Proceedings of the First PhD Symposium on Sustainable Ultrascale Computing Systems (NESUS PhD 2016)
Timisoara, Romania

Jesus Carretero, Javier Garcia Blas
Dana Petcu
(Editors)

February 8-11, 2016

Volume Editors

Jesus Carretero
University Carlos III
Computer Architecture and Technology Area
Computer Science Department
Avda Universidad 30, 28911, Leganes, Spain
E-mail: jesus.carretero@uc3m.es

Javier Garcia Blas
University Carlos III
Computer Architecture and Technology Area
Computer Science Department
Avda Universidad 30, 28911, Leganes, Spain
E-mail: fjblas@arcos.inf.uc3m.es

Dana Petcu
West University of Timisoara
Department of Computer Science
Faculty of Mathematics & Informatics
B-dul V.Parvan 4, 300223 Timisoara, Romania
E-mail: petcu@info.uvt.ro

Published by:
Computer Architecture, Communications, and Systems Group (ARCOS)
University Carlos III
Madrid, Spain
http://www.nesus.eu

ISBN: 978-84-608-6309-0

This document also is supported by:

Printed in Madrid — February 2016

Preface

Network for Sustainable Ultrascale Computing (NESUS)

We are happy to present the Proceedings of the First PhD Symposium on Sustainable Ultrascale Computing Systems (NESUS PhD 2016), an output of the PhD Symposium held in the First Winter School of the COST Action (IC1035) (www.nesus.eu) <http://www.nesus.eu)>.

The First PhD Symposium of the COST Action IC1035 was held on February 10, 2016, in Timisoara. Twenty PhD students belonging to NESUS action made a presentation of their PhD Thesis research work and contributed with a short paper reflecting the main ideas of their PhD Thesis.

The PhD Symposium was a very good opportunity for the young researchers to share information and knowledge, to present their current research, and to discuss topics with other students in order to look for synergies and common research topics. The idea was very successful and the assessment made by the PhD Student was very good. It also helped to achieve one of the major goals of the NESUS Action: to establish an open European research network targeting sustainable solutions for ultrascale computing aiming at cross fertilization among HPC, large scale distributed systems, and big data management, training, contributing to glue disparate researchers working across different areas and provide a meeting ground for researchers in these separate areas to exchange ideas, to identify synergies, and to pursue common activities in research topics such as sustainable software solutions (applications and system software stack), data management, energy efficiency, and resilience.

Prof. Jesus Carretero
University Carlos III of Madrid
NESUS Chair

February 2016

TABLE OF CONTENTS

First NESUS PhD Symposium (PhD-NESUS 2016)

85 List of Authors

Computational Intelligence Modeling of Pharmaceutical Properties

Hossam M. Zawbaa

Faculty of Mathematics and Computer Science, Babes-Bolyai University, Romania
Faculty of Computers and Information, Beni-Suef University, Egypt
hossam.zawbaa@gmail.com

Abstract

In the pharmaceutical industry, a good understanding of the casual relationship between product quality and attributes of formulations is very useful in developing new products, and optimizing manufacturing processes. Feature selection is mandatory due to the abundance of noisy, irrelevant, or misleading features. The selected features will improve the performance of the prediction model and will provide a faster and more cost effective prediction than using all the features. With the big data captured in the pharmaceutical product development practice, computational intelligence (CI) models and machine learning algorithms could potentially be used to identify the process parameters of formulations and manufacturing processes. That needs a deep investigation of roller compaction process parameters of pharmaceutical formulations that affect the ribbons production. In this work, we are using the bio-inspired optimization algorithms for feature selection such as (grey wolf, Bat, flower pollination, social spider, antlion, moth-flame, genetic algorithms, and particle swarm) to predict the different pharmaceutical properties.

Keywords Computational Intelligence, Pharmaceutical Roll Compaction, Bio-inspired Optimization, Feature Selection

I. Introduction

A feature is an measurable property of the problem under observation, over the past years the domain of features in machine learning and pattern recognition applications have expanded from tens to hundreds of variables or features used in such applications. Hence the use of reduction or selection techniques is essential to reduce the large number of feature in the problem. Feature selection is a process of selecting a subset of features from a larger set of features, which leads to the reduction of the dimensionality of features space for a successful classification task. Feature selection provides a way for identifying the important features and removing irrelevant or redundant features from a dataset [1]. Feature Selection helps in understanding data, reducing computation requirement, reducing the effect of curse of dimensionality and improving the predictor performance [2].

Formerly, an exhaustive search for the optimal or near to optimal solution in a enormous search space may be impracticable, many researches seek to model the feature selection as a optimization problem [3]. One of the most used methods to solve the feature selection problems are evolutionary and swarm intelligence methods. Swarm intelligence is a computational intelligence-based approach which is made up of a population of artificial agents and inspired by the social behavior of animals (fish, birds, fireflies, etc.) from the real world. Example of such methods are ant colony optimization [4], bat algorithm [5], and particle swarm optimization (PSO) [6].

Roller compaction is method of preparing drug granules for capsules or for tablet formulations used in the pharmaceutical industry with suitable densification. The most common filler binder excipient

used in roller compaction are microcrystalline cellulose (MCC), dibasic calcium phosphate (DCP), and lactose. Roller compaction is a particle size enlargement technique that granulated the powder materials to obtain materials of intermediate sizes in tablets production. The use of latest technology facilitates to efficient production of high quality granules. The selection of the critical roll compaction parameters such as (constant compacting pressure and constant roller gap) is very important.

Being a part of the development of in-silico process models for roll compaction (IPROCOM) project, Marie Curie. IPROCOM project employs a multidisciplinary approach to understand the fundamental mechanisms of particulate manufacturing processes involving roll compaction, and to develop predictive in-silico tools that can be used by various industrial sectors in Europe. In addition, we in need to establish a computational intelligence framework that identifies the critical material and process parameters and defines the design spaces for robust formulations and efficient production.

The aggregate aim of this work is to propose the bio-inspired optimization algorithms for feature selection that maximize feature reduction and obtaining comparable or even better prediction results of roll compaction parameters from using full features and conventional feature selection techniques.

II. Related work

Evolutionary computational (EC) algorithms have been used in feature selection issues such as genetic algorithm (GA), genetic programming (GP), ant colony optimization (ACO), and particle swarm optimization (PSO). GA was the first evolutionary based algorithm introduced in the literature and developed based on the natural process of evolution through reproduction [7]. Particle swarm optimization (PSO) is one of the well-known swarm algorithms. In PSO, each solution is considered as a particle with specific characteristics (position, fitness, and speed vector) that defines the moving direction of each particle [8]. A hybrid methods can also be applied in which two evolutionary algorithms are used to solve the problem, for example [9] proposed a new feature selection approach that is based on the integration of GA and PSO. Artificial bee colony (ABC) is a numerical optimization algorithm based on foraging behavior of honeybees. In ABC, the employer bees try to find food source and advertise the other bees. The onlooker bees follow their interesting employer and the scout bee fly spontaneously to find the best food source [10]. Social spider optimization (SSO) algorithm is a population based algorithm and one of the comparatively recent swarm algorithms [11].

A virtual bee algorithm (VBA) is applied to optimize the numerical function in 2-D using a swarm of virtual bees, which move randomly in the search space and interact to find food sources. From the interactions between these bees results the possible solution for the optimization problem [12]. A proposed approach based on natural behavior of honeybees, which randomly generated worker bees are moved in the direction of the elite bee. The elite bee represents the optimal (near to optimal) solution [13]. Ant colony optimization (ACO) wrapper-based feature selection algorithm was applied in network intrusion detection with rough set theory [14]. Artificial fish swarm (AFS) algorithm mimics the stimulant reaction by controlling the tail and fin. AFS is a robust stochastic technique based on the fish movement and its intelligence during the food finding process [15].

III. Thesis idea

The main goal of this thesis study was to investigate the roller compaction and granulation characteristics of pharmaceutical formulations. During the roller compaction operation, uniformly mixed powder blends are passed continuously through the gap between a pair of counter rotating compression rolls to form solid ribbons or sheets which are then passed through a mill or granulator with a suitable sized screen to form dry granules. Compared to wet granulation processes, dry granulation by roller compaction has various advantages such as simpler manufacturing procedure, easier scale up and higher production throughput. Dry granulation is also energy efficient and suitable for processing pharmaceutical agents that are sensitive to moisture and heat. The complex-

ity of formulation design is a highly specialised task, requiring specific knowledge and often years of experience. In this work, we have applied bio-inspired optimization algorithms such as (grey wolf optimization, Bat optimization, cuckoo search, flower pollination algorithm, social spider optimization, etc) for feature selection and prediction of different pharmaceutical properties. After that, we use machine learning techniques like (artificial neural network, k-nearest neighbour, extreme learning machine, etc) to predict the different pharmaceutical properties such as (true density, porosity, tensile strength, fines, etc).

Each optimization algorithm is run for 20 times to test the algorithm convergence capability. The used evaluation indicators to compare different optimization algorithms are:

1. **Average reduction** represents the average size of selected features to the total number of features.

2. **Mean square error (MSE)** measures the average of squared errors that means the difference between actual output and predicted ones.

The two evaluation criteria or objective function in the wrapper feature selection is commonly reflecting the regression performance as well as the feature reduction. A generic representation of the fitness function representing for both regression performance and feature reduction as described in equation (1):

$$f_\theta = \alpha * E + (1 - \alpha)\frac{\sum_i \theta_i}{N}, \tag{1}$$

where f_θ is the fitness function given a vector θ sized N with $0/1$ elements representing unselected / selected features, N is the total number of features in the dataset, E is the prediction error, and α is a constant controlling the importance of regression performance to the number of features selected.

A random controlling term (α) is used to balance the trade-off between exploration and exploitation and hence should be carefully adapted. Therefore, at the beginning of optimization (α) has its maximum value to allow for maximum exploration and at the end of optimization it has minimum value for more exploitation of search space. Each bio-inspired algorithm is initialized with n random agents, each agent (solution) representing a given selected feature combination. After that, each algorithm is iteratively applied for a number of iterations hoping to converge to a good solution. Individual solution is represented as a continuous valued vector with same dimension as number of attributes in the given dataset. The solution vector continuous values are limited to the range [0, 1]. At the solution fitness function evaluation the continuous valued solution is threshold to its binary representation using equation (2).

$$y_{ij} = \begin{array}{l} 0 \text{ If}(x_{ij} < 0.5) \\ 1 \text{ Otherwise} \end{array} \tag{2}$$

where x_{ij} is the continuous value of the solution number i in dimension j, and y_{ij} is a discrete representation of solution vector x.

IV. Conclusion and future work

In this work, bio-inspired optimization algorithms were proposed and applied for feature selection in wrapper mode. The most recent bio-inspired optimization algorithms such as (GWO, ALO, BAT, SSO, and FPA) are hired in the feature selection domain for evaluation and results are compared against well-known feature selection methods particle swarm optimization (PSO) and genetic algorithm (GA). The evaluation is performed using a set of evaluation criteria to assess different aspects of the proposed system.

Acknowledgment

This work was supported by the IPROCOM Marie Curie initial training network, funded through the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme FP7/2007-2013/ under REA grant agreement No. 316555. In addition, this work was partially supported by NESUS.

References

[1] Chizi, Barak and Rokach, Lior and Maimon, Oded and Wang, J, "A Survey of Feature Selection Techniques", 2009.

[2] Chandrashekar, Girish, Sahin, Ferat, "A survey on feature selection methods", Computers & Electrical Engineering, pp. 16-28, Vol. 40, No. 1, 2014.

[3] Duda, Richard O and Hart, Peter E and Stork, David G, "Pattern classification", John Wiley & Sons, 2012.

[4] Forsati, Rana and Moayedikia, Alireza and Jensen, Richard and Shamsfard, Mehrnoush and Meybodi, Mohammad Reza, "Enriched ant colony optimization and its application in feature selection", Neurocomputing, pp. 354-371, Vol. 142, 2014.

[5] Rodrigues, Douglas and Pereira, Luís AM and Nakamura, Rodrigo YM and Costa, Kelton AP and Yang, Xin-She and Souza, André N and Papa, João Paulo, "A wrapper approach for feature selection based on Bat Algorithm and Optimum-Path Forest", Expert Systems with Applications, pp. 2250-2258, Vol. 41, No. 2, 2014.

[6] Inbarani, H Hannah, Azar, Ahmad Taher, Jothi, G, "Supervised hybrid feature selection based on PSO and rough sets for medical diagnosis", Computer methods and programs in biomedicine, pp. 175-185, Vol. 113, No. 1, 2014.

[7] Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, John Henry Holland, MIT press, 1992.

[8] R. C. Eberhart, and J. Kennedy, "A New Optimizer Using Particle Swarm Theory", Proceeding of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39-43, 1995.

[9] Ghamisi, Pedram and Benediktsson, Jon Atli, "Feature selection based on hybridization of genetic algorithm and particle swarm optimization", Geoscience and Remote Sensing Letters, IEEE, pp. 309-313, Vol. 12, No. 2, 2015.

[10] Dervis Karaboga, Bahriye Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm", Journal of Global Optimization, Vol. 39, No. 3, pp. 459-471, 2007.

[11] Cuevas, E., Cienfuegos, M., Zaldivar, D., Perez-Cisneros, M., "A swarm optimization algorithm inspired in the behavior of the social-spider", Expert Systems with Applications, Vol. 40, No. 16, pp. 6374-6384, 2013.

[12] Yang XS, "Engineering optimizations via nature-inspired virtual bee algorithms", In: Lecture notes in computer science, Springer (GmbH), pp. 317-323, 2005.

[13] Sundareswaran K, Sreedevi VT, "Development of novel optimization procedure based on honey bee foraging behavior", IEEE International conference on systems, man and cybernetics, pp. 1220-1225, 2008.

[14] H. Ming, "A rough set based hybrid method to feature selection", in Proc. Int. Symp. KAM, pp. 585-588, 2008.

[15] X. L. Li, Z. J. Shao, J. X. Qian, "An Optimizing Method Based on Autonomous Animates: Fish-swarm Algorithm", Methods and Practices of System Engineering, pp. 32-38, 2002.

Towards a Smart Selection of Hybrid Platforms for Multimedia Processing

Sidi Ahmed Mahmoudi and Pierre Manneback

University of Mons, Belgium
sidi.mahmoudi@umons.ac.be

Abstract

Nowadays, images and videos have been present everywhere, they can come directly from camera, mobile devices or from other peoples that share their images and videos. The latter are used to illustrate different objects in a large number of situations. This makes from image and video processing algorithms a very important tool used for various domains related to computer vision such as video surveillance, medical imaging and database (images and videos) indexation methods. The performance of these algorithms have been so reduced due the the high intensive computation required when using new image and video standards. In this paper, we propose a new framework that allows users to select in a smart and efficient way the processing units (GPU or/and CPU) within heterogeneous systems, when treating different kinds of multimedia objects : single image, multiple images, multiple videos and video in real time. The framework disposes of different image and video primitive functions that are implemented on GPU, such as shape (silhouette) detection, motion tracking using optical flow estimation, edges and corners detection. We have exploited these functions for several situations such as indexing videos, segmenting vertebrae in in X-ray and MR images, detecting and localizing event in multi-user scenarios. Experimentation showed interesting accelerations ranging from 6 to 118, by comparison with sequential implementations. Moreover, the parallel and heterogeneous implementations offered lower power consumption as a result for the fast treatment.

Keywords GPU, Heterogeneous architectures, Image and video processing, Medical imaging, Motion tracking

I. Introduction

Recently, the architecture of CPUs has so changed and evolved that the number of integrated computing units has been multiplied. This evolution is reflected in both general (CPU) and graphic (GPU) processors which present a large number of computing units, their power has far exceeded the CPUs ones. In this context, image and video processing algorithms are well adapted for acceleration on the GPU by exploiting its processing units in parallel, since they are mainly based on applying the same computation over many points or pixels. Many GPU and parallel computing approaches have been developed recently. Although they present a great power of GPU architecture, any is able to process high definition image and video efficiently and accordingly to the type of Medias (single image, multiple image, multiple videos and video in real time). Thus, there was a need to develop a framework capable of addressing the outlined problem. In literature, we can categorize two types of related works based on the exploitation of parallel and heterogeneous platforms for multimedia processing: one related to image processing on GPU such as presented in [1], [2] which proposed GPU implementations that use CUDA [1] for basic image processing and medical imaging algorithms. A performance evaluation of GPU-based image processing algorithms is presented in [3]. These implementations offered high improvement of performance thanks to the exploitation of the GPU's computing units in parallel. However, these accelerations are so reduced when processing image databases with different resolutions. Thus, an efficient exploitation of CPU, GPU and

[1] CUDA. https://developer.nvidia.com/cuda-zone

hybrid (Multi-CPU/Multi-GPU) platforms is needed with an effective management of the related memories. Notice also that the processing of images with low resolutions cannot benefit from the high power of GPUs since few computations will be launched. This implies an analysis of algorithms complexities before their parallelization. On the other hand, video processing algorithms require generally a real-time treatment. We may find several methods in this category, such as understanding human behavior, event detection, camera motion estimation, etc. These methods apply mainly motion tracking algorithms that can exploit several techniques such as optical flow estimation [4], block matching technique [5], and scale-invariant feature transform (SIFT) [6] descriptors. In this case also, several GPU implementations have been proposed for sparse [7] and dense [8] optical flow computation.

II. Research idea

Despite the high speedups presented in the previous section, none of the above-mentioned implementations can provide real-time processing of high definition videos. Therefore, we propose a new framework that allows a smart, effective and adapted processing of different type of Medias exploiting parallel and heterogeneous platforms. This framework enables to select the units (GPU or/and CPU) for processing, and also the related implementations to be applied. The latter are selected after checking the type of media to treat and the algorithm complexity. The framework offers several scheduling strategies that allow an equivalent distribution of tasks over the available processors. The data transfer times are also reduced as a result of the efficient management of GPU memories and to the overlapping (CUDA streaming) of data copies by kernels executions. Otherwise, the framework disposes of several GPU-based image and video primitive functions, such as shape detection, motion tracking using optical flow estimation, edges and corners extraction. We have exploited these functions for several situations such as indexing videos, segmenting vertebrae in X-ray and MR images, detecting and localizing event in multi-user scenarios. The primitive functions are presented in detail in our previous publication [9]. Figure 1 illustrate the proposed framework, presenting different applications that can exploit in an adapted way the heterogeneous systems, which offers a low energy consumption as a result for the fast and accelerated treatment. The main contributions of our framework can be summarized within five points :

1. Smart selection of resources (CPU or/and GPU) based on the estimated complexity and the type of media. Additional computing units are exploited only in case of intensive and tasks;

2. Several image and video GPU primitive functions;

3. Efficient scheduling of tasks and management of memories in case of heterogeneous computation;

4. Acceleration of real-time image and video processing applications;

5. low energy consumption.

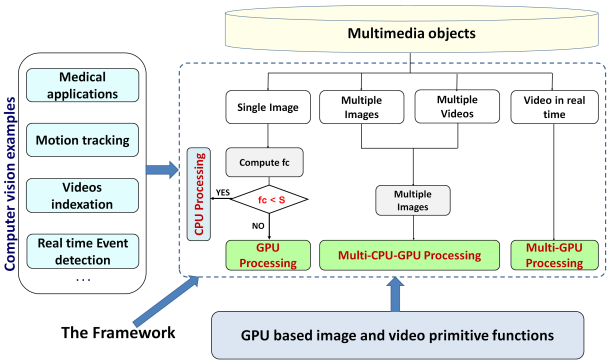


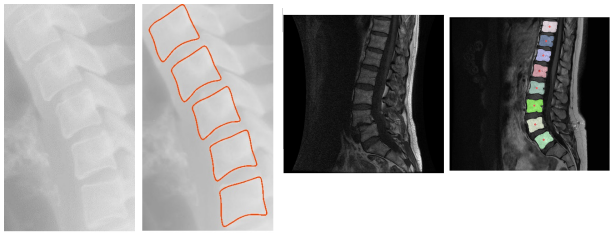
Figure 1: Multi-CPU/Multi-GPU based Framework for Multimedia Processing

III. Experimental results

The proposed framework has been exploited in several high intensive applications related to image and video processing such as vertebra segmentation, videos indexation, event detection and localization, etc.

III.1 Heterogeneous vertebra segmentation

The main objective of this method is the cervical vertebra mobility analysis on X-Ray or MR images. The aim is to detect vertebra automatically. The computation time presents one of the most important requirements for this application. Based on our framework, we propose a hybrid implementation of the most intensive steps, which have been defined within our estimation complexity equation [9]. Our solution for vertebra detection on Multi-CPU/Multi-GPU platforms is detailed in [10] for X-Ray images, and in [11] for MR images. Fig. 2(a) presents the results of vertebra detection in X-ray images, while Fig. 2(b) is related to present the detected vertebra in MR images. Notice that the use of heterogeneous platforms allowed to improve performance with a speedup of 30 $\times$ for vertebra detection within 200 high resolution (1472$\times$1760) X-ray images, and a speedup of 118 $\times$ when detecting vertebra in a set of 200 MR images (1024$\times$1024).



(a) X-ray images (b) MR images

Figure 2: Vertebra detection in X-ray and MR images

III.2 Multi-CPU/Multi-GPU based videos indexation

The context of this application is to develop a new browsing environment for images and videos databases. This method consists on calculating similarities between videos sequences (composed of consecutive images), based on detecting the feature of images (frames) that compose videos [12]. The main drawback of this application is the high computing time that increases considerably when enlarging videos databases and definitions. Using our framework, we developed a hybrid version of the most intensive step of the features extraction process. This step, detected within our complexity estimation equation defined in [9], consists of contours extraction algorithm that provides relevant information for localizing motion's areas. This implementation is detailed in [13] showing a total gain of 80%, compared to the total time of the sequential version, when treating 800 frames of a video sequence (1080x720).

III.3 Multi-GPU based Event detection and localization in real time

The aim of this method is to detect and localize events in video sequences in real time. The method is based on modeling normal behaviors, and then estimating the difference between the normal behavior model and the observed events of behaviors. The detected variations are labeled as emergency events, and the deviations from examples of normal behavior are used to characterize abnormality. After the detection of each event, we localize the related areas in video frames where motion behavior is surprising compared to the rest of motion. Using our framework, we developed a Multi-GPU version of the most intensive steps of the application. The latter are detected within our complexity estimation equation defined in [9]. This implementation is detailed in [14]. Notice that performed tests show that our application can turn in multi-user scenarios, and in real time even when processing high definition videos such as Full HD or 4K standards. The scalability of our results is also achieved thanks to the effective exploitation of multiple GPUs. A demonstration of GPU based features detection, features tracking, and event detection in crowd video is shown in this video sequence: `https://www.youtube.com/watch?v=PwJRUTdQWg8`.

IV. Conclusion and future work

We proposed in this paper a new framework that allows a smart and efficient exploitation of Multi-CPU/Multi-GPU platforms accordingly to the type of multimedia (single image, multiple images, multiple videos, video in real time) objects. This framework

enables to select the units (GPU or/and CPU) for processing, and also the related implementations to be applied. The latter are selected after checking the type of media to treat and the algorithm complexity. Experimental results showed different use case applications that have been improved thanks to our framework. Each application has been integrated in an adapted way for exploiting resources in order to reduce both computing time and energy consumption. As future work, we plan to port our algorithms on GPU Tegra Mobile Processors [2] that allow to reduce significantly the power consumption, with maintaining high performance of computation.

Acknowledgment

Authors would like to thank the support of European COST NESUS action IC1305 " Network for Sustainable Ultra-scale Computing"

References

[1] Yang. Z and Zhu. Y and pu. Y, " Parallel Image Processing Based on CUDA " *HPCCE Workshop, IEEE International Conference on Cluster Computing*, pp. 198-201, 2008.

[2] Mahmoudi. Sidi Ahmed and Lecron. F and Manneback. P and Benjelloun. M and Mahmoudi. S, " GPU-Based Segmentation of Cervical Vertebra in X-Ray Images " *HPCCE Workshop, IEEE International Conference on Cluster Computing*, pp. 1-8, 2010.

[3] Park. Kyu and Nitin. Singhal and Man. Hee Lee, " Design and Performance Evaluation of Image Processing Algorithms on GPUs " *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, pp. 1-14, 2011.

[4] Horn. B. K and Schunk. B. G, " Determining Optical Flow " *Artificial Intelligence*, vol. 2, pp. 185-203, 1981.

[5] Shan Zhu and Kai-Kuang Ma, "A new diamond search algorithm for fast block-matching motion estimation " *IEEE Transactions on Image Processing*, vol. 9, pp. 287-290, 2000.

[6] Lowe. D. G, " Distinctive image features from scale-invariant keypoints " *International Journal of Computer Vision (IJCV)*, vol. 60(2), pp. 91-110, 2004.

[7] Mahmoudi. Sidi Ahmed and Kierzynka. Michal and Manneback. Pierre and Kurowski. K, " Real-time motion tracking using optical flow on multiple GPUs " *Bulletin of the Polish Academy of Sciences: Technical Sciences*, vol. 62, pp. 139-150, 2014.

[8] Marzat. J and Dumortier. Y and Ducrot. A, " Real-time dense and accurate parallel optical flow using CUDA " *In Proceedings of WSCG*, pp. 105-111, 2009.

[9] Mahmoudi. Sidi Ahmed and Manneback. Pierre, " Multi-CPU/Multi-GPU Based Framework for Multimedia Processing " *Computer Science and Its Applications*, vol. 456, pp. 54-65, 2015.

[10] Lecron. Fabian et al., " Heterogeneous Computing for Vertebra Detection and Segmentation in X-Ray Images " *International Journal of Biomedical Imaging: Parallel Computation in Medical Imaging Applications*, vol. 2011, pp. 1-12, 2011.

[11] Larhmam. Mohammed Amine et al., " A Portable Multi-CPU/Multi-GPU Based Vertebra Localization in Sagittal MR Images ", *International Conference on Image Analysis and Recognition, ICIAR 2014*, pp. 209-218, 2014.

[12] Damien Tardieu et al., " Video Navigation Tool: Application to browsing a database of dancers' performances " , *QPSR of the numediart research program*, vol. 2, number. 3, pp. 85-90, 2009.

[13] Mahmoudi Sidi Ahmed and Manneback Pierre, " Efficient exploitation of heterogeneous platforms for images features extraction " *3rd International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pp. 91-96, 2012.

[14] Mahmoudi Sidi Ahmed and Manneback Pierre, " Multi-GPU based event detection and localization using high definition videos " *International Conference on Multimedia Computing and Systems (ICMCS)*, pp. 81-86, 2014.

[2] Tegra Mobile Processors :http://www.nvidia.com/object/tegra.html

Energy aware execution environments and algorithms on low power multi-core architectures

Sandra Catalán, Rafael Rodríguez-Sánchez, Enrique S. Quintana-Ortí

Universitat Jaume I, Spain
catalans@uji.es, rarodrig@uji.es, quintana@uji.es

Abstract

Energy consumption is a key aspect that conditions the proper functioning of nowadays data centers and high performance computing just like the launch of new services, due to its environmental negative impact and the increasing economic costs of energy.

The energy efficiency of the applications used in these data centers could be improved, especially when systems' utilization rate is low or moderate, or when targeting memory bounded applications. In this sense, energy proportionality stands for systems which power consumption is in line with the amount of work performed in each moment. As a response to these needs, the main objective of this project is to study, design, develop and analyze experimental solutions (models, programs, tools and techniques) aware of energy proportionality for scientific and engineering applications on low-power architectures. With the aim of showing the benefits of this contribution, two applications, coming from the image processing and dynamic molecular simulation fields, have been chosen.

Keywords Energy, low-power architectures, linear algebra, NESUS

I. Motivation

Nowadays there is a vast variety of scientific, industrial and engineering applications that have great computing power and storage requirements, and their demand is still growing. In order to obtain more precise solutions in these applications, scientists need to build and work with sophisticated physical and mathematical models. Scientific computation (seen as the elaboration of mathematical models and the use of computers to analyze and solve scientific problems) is an efficient tool to make scientific discoveries that are complementary to the most traditional methods based on theory and experimentation [1] As a consequence, new data processing systems and high performance computing centers collapse just a few weeks later from their commissioning [2].

To face the mathematical formulation at the bottom of the physical laws advanced numerical algorithms are required: linear algebra, spectral methods (e.g, FFT), N-body methods, mesh methods to solve partial differential equations, as well as searching, classification and optimization algorithms, among others [2] are required. In particular, the main part of the computations demanded to solve these scientific, industrial and engineering applications can be decomposed into a reduced number of well known matrix computation problems, e.g., simple operations of linear algebra, linear equation systems, minimum least square problems, eigenvalue and eigenvector problems. In this way, the efficiency of these computation problems determine as a last resort the effectiveness of the software application.

Large scale HPC (high performance computing) systems are great energy consumers, using computing resources and auxiliary systems to work [1]. This energy consumption has a direct impact on the operation costs and maintenance of the computing centers, threatening their existence and complicating the acquisition of new facilities. However, electricity cost is not the only problem; energy consumption results in carbon dioxide emissions dangerous to the environment and public health, and the heat reduces the reliability of the hardware components [3].

HPC centers' pressure forced hardware manufacturers to improve their designs to get better energy efficiency: CPU, memory and disks (the main energy consumers in a system, followed by the network and

the power supply unit) provide some energy saving strategies, based on the system transition to a low power consumption state or the dynamic adaptation of frequency and voltage (DVFS or Dynamic Voltage Frequency Scaling) [4]. On the other hand, software systems, communication libraries and, specially, computational libraries and application codes used in HPC centers have been, traditionally, unaware of power consumption. In fact, the Top500 [5] list is a good example. Computers listed in this ranking are classified depending on their sustained performance (in FLOPS) when running the Linpack test (basically, solving a dense linear system of scalable dimension). However, the numerical method behind this test, LU factorization, is far from being representative for most real scientific codes [6].

Despite the great benefits [7] that HPC energy aware solutions can provide in terms of run time optimization and energy conservation, this topic is still at an early research stage if compared with energy study in other segments. Recently, HPC community has presented energy aware metrics, e.g., Energy Delay Product (EDP), Energy To Solution (ETS), FLOPS/Watt or FTTSE [8], that are becoming more significant when evaluating algorithms and computers performance. In fact, the Green500 [9] ranking, which uses these metrics to compare and classify supercomputers all around the world regarding their energy efficiency, is becoming more considered every day.

II. Related work

Nowadays HPC linear algebra libraries make use of hardware concurrency in multi-core processors using multi-threaded implementations highly optimized for a small set of linear algebra kernels (particularly, BLAS matrix-vector product and matrix-matrix product). For years, this approach was successfully followed by the scientific community, since it provides an interface that has allowed the development of complex and architecture independent packages of numerical methods with portable performance. However, with the increasing number of cores (e.g, Intel Xeon Phi), this solution became suboptimal due to the fact that concurrency at BLAS level implies a high number of thread synchronizations, causing a high overhead.

Recently, many projects demonstrated the benefits of applying parallelism at a higher level, in both dense and sparse linear algebra, through applications that decompose operations in fine grain tasks, with out-of-order execution by means of an scheduling aware of the tasks' dependencies. Examples of this successful solution are libflame (SuperMatrix [10]), PLASMA (Quark) [11], SMPSs [12], StarPU [13], etc., based on the ideas/techniques firstly proposed by the project Cilk [14] of MIT. These execution frameworks aim at the gross performance as final value for the user. However, they are completely energy unaware. Initial research efforts showed the possibility of keeping isoefficiency/isoscalability in a parallel solver while getting low power consumption, and the benefits derived from this approach. This can be done, for instance, scheduling non-critical tasks to less powerful and low power consumption cores (on heterogeneous environments) or through processor frequency adjustment, and promoting idle cores to low power states [15, 16, 17].

Previously mentioned solutions try to efficiently identify and make use of task parallelism in software applications. To this end, they provide the user with an explicit or implicit mechanism to identify tasks and dependencies among them. There is a part in this framework that builds a Directed Acyclic Graph (DAG) that gathers all the dependencies, and this information is used by the scheduler, which in turn issues tasks to execution when their dependencies are solved and there are enough free computational resources. Some of these frameworks also tackle the existence of multiple address spaces, providing the programmer with an explicit transfer mechanism or, alternatively, a memory control mechanism built in the scheduler that performs transparent transfers for the programmer. Scheduling algorithms at the bottom of these execution frameworks aim at optimizing performance, but generally, they do not consider energy as a variable to make decisions. However, for some operations, it is possible to improve energy efficiency during the dynamic execution of a DAG if some non-critical tasks are executed at a lower speed (via, e.g., the frequency reduction of cores applying DVFS).

On the way towards the construction of exaflop supercomputers, some research lines stand for the utilization of highly heterogeneous systems, composed of some nodes, with a huge amount of simple and low-power multi-core processors, combined with some other nodes, featuring hardware accelerators [18]. In the same vein, some recent works reveal energy advantages when using low-power processors, such as Intel Atom, ARM A-15, or more specialized systems, like ARM+NVIDIA Carma, composed of an ARM A-9 processor and a small Quadro 1000M GPU, or the Digital

Signal Processors (DSP) of Texas Instruments [19, 20].

III. Thesis idea

The main objective of the research proposal is to study, design, develop and experimentally analyze solutions that are aware of the energy proportionality (models, programs, tools and techniques) of scientific and engineering applications running on low-power architectures. This objective is composed of two specific targets:

- Studying, characterizing and modeling low-power architectures' performance and energy efficiency, which include, Intel Atom, ARM Cortex-A15, Texas Instruments DSP C66x, among others.
- Designing, developing and evaluating energy proportional solutions for scientific applications in the field of hyperspectral image processing and macromolecular simulations.

So far the improvement of these kind of applications was focused on increasing their performance, through traditional parallel systems that were to a large extent energy proportionality oblivious. The novelty of this proposal is founded on the study of specific HPC techniques for low-power architectures, capable of making the best of the greater energy proportionality of these systems.

To achieve the proposed goal, the first stage of the work will consist of analyzing, modeling and optimizing basic kernels on low-power architectures. To this end, a representative number of low-power architectures will be selected in order to build experimental energy models with an appropriate collection of parameters and to determine computing and memory access costs in terms of energy. In addition, the same basic kernels will be used to characterize the energy consumption of the different components in a given architecture. After this initial study, the improvement of hyperspectral image processing problems and macromolecular simulations will be tackled. In both cases, the exploitation of parallelism at different levels (fine grain, gross grain and task parallelism) and the use of the MPI paradigm will be key to get the best of these applications on low-power architectures.

IV. Conclusion and future work

Apart from the computational implications explained along this text, from the economical and digital society point of view, this proposal is also part of the greenhouse gas reduction challenge and the energy efficiency goal. Moreover, this project is strongly connected with the climate change action and the use of raw materials and natural resources. On the other hand, the macromolecular simulations, and to a large extent also the hyperspectral image processing, make use of and produce huge amounts of data/results. Consequently, these two kind of applications belong to the "big data" category, being also characterized as a priority topic by the economical and digital society challenges.

As future work, the improvement of dense linear algebra operations (focused on the BLIS library [21]) on low-power architectures has to be completed and the improvement of hyperspectral image processing problems and macromolecular simulations need to be performed.

Acknowledgment

This work is partially supported by EU under the COST Program Action IC1305: Network for Sustainable Ultrascale Computing (NESUS) and the FPU program of MECD.

References

[1] J. Dongarra, *et al*, The international ExaScale software project roadmap, Int. J. of High Performance Computing & Applications 25 (1) (2011) 3–60.

[2] International technology roadmap for semiconductors, `http://www.itrs.net/` (2013).

[3] W.-c. Feng, X. Feng, R. Ge, Green supercomputing comes of age, IT Professional 10 (1) (2008) 17 –23. `doi:10.1109/MITP.2008.8`.

[4] W. Y. Lee, Energy-saving DVFS scheduling of multiple periodic real-time tasks on multi-core processors, in: Distributed Simulation and Real Time Applications, 2009. DS-RT '09. 13th IEEE/ACM International Symposium on, 2009, pp. 216 –223. `doi:10.1109/DS-RT.2009.12`.

[5] The Top 500 list, `http://www.top500.org/` (2014).

[6] P. Kogge, K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, J. Hiller, S. Karp, S. Keckler, D. Klein,

R. Lucas, M. Richards, A. Scarpelli, S. Scott, A. Snavely, T. Sterling, R. S. Williams, K. Yelick, ExaScale computing study: Technology challenges in achieving ExaScale systems (2008).

[7] S. Albers, Energy-efficient algorithms, Commun. ACM 53 (2010) 86–96.

[8] C. Bekas, A. Curioni, A new energy aware performance metric, Computer Science - Research and Development 25 (2010) 187–195.

[9] The Green 500 list, http://www.green500.org/ (2014).

[10] E. Chan, F. G. Van Zee, P. Bientinesi, E. S. Quintana-Ortí, G. Quintana-Ortí, R. van de Geijn, SuperMatrix: A multithreaded runtime scheduling system for algorithms-by-blocks, in: ACM SIGPLAN 2008 symposium on Principles and practices of parallel programming (PPoPP'08), 2008, to appear.

[11] F. Song, S. Tomov, J. Dongarra, Enabling and scaling matrix computations on heterogeneous multi-core and multi-gpu systems, in: Proceedings of the 26th ACM International Conference on Supercomputing, ICS '12, ACM, New York, NY, USA, 2012, pp. 365–376, http://doi.acm.org/10.1145/2304576.2304625. doi:10.1145/2304576.2304625.

[12] R. M. Badia, J. R. Herrero, J. Labarta, J. M. Pérez, E. S. Quintana-Ortí, G. Quintana-Ortí, Parallelizing dense and banded linear algebra libraries using SMPSs, Conc. and Comp.: Pract. and Exper. 21 (2009) 2438–2456.

[13] R. M. Badia, J. R. Herrero, J. Labarta, J. M. Pérez, E. S. Quintana-Ortí, G. Quintana-Ortí, Parallelizing dense and banded linear algebra libraries using smpss, Concurrency and Computation: Practice and Experience 21 (18) (2009) 2438–2456.

[14] R. D. Blumofe, C. F. Joerg, B. C. Kuszmaul, C. E. Leiserson, K. H. Randall, Y. Zhou, Cilk: An efficient multithreaded runtime system, Vol. 30, ACM, 1995.

[15] P. Alonso, M. F. Dolz, F. D. Igual, R. Mayo, E. S. Quintana-Ortí, DVFS-control techniques for dense linear algebra operations on multi-core processors, Computer Science - Research and Development 1–10http://dx.doi.org/10.1007/s00450-011-0188-7.

[16] P. Alonso, M. F. Dolz, R. Mayo, E. S. Quintana-Ortí, Saving energy in the LU factorization with partial pivoting on multi-core processors, 2012, to appear.

[17] P. Alonso, M. Dolz, R. Mayo, E. Quintana-Ortí, Improving power efficiency of dense linear algebra algorithms on multi-core processors via slack control, Proc. Int. Conf. High Performance Computing & Simulation–HPCS (2011) 463–470.

[18] Mont-blanc project, http://www.montblanc-project.eu/ (2013).

[19] J. I. Aliaga, H. Anzt, M. Castillo, J. C. Fernández, G. León, J. Pérez, E. S. Quintana-Ortí, Performance and energy analysis of the iterative solution of sparse linear systems on multicore and manycore architectures, Springer, 2014.

[20] M. Castillo, J. Fernández, F. Igual, A. Plaza, E. Quintana-Ortí, A. Remón, Hyperspectral unmixing on multicore DSPs: Trading off performance for energy, Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal ofDOI:10.1109/JSTARS.2013.2266927.

[21] F. G. Van Zee, R. A. van de Geijn, BLIS: A framework for generating BLAS-like libraries, ACM Trans. Math. Soft.To appear. http://www.cs.utexas.edu/.

CuDB: a Relational Database Engine Boosted by Graphics Processing Units

Samuel Cremer, Michel Bagein, Saïd Mahmoudi, Pierre Manneback

University of Mons, Belgium
samuel.cremer@heh.be,michel.bagein@umons.ac.be,
said.mahmoudi@umons.ac.be,pierre.manneback@umons.ac.be

Abstract

GPUs benefit from much more computation power with the same order of energy consumption than CPUs. Thanks to their massive data parallel architecture, GPUs can outperform CPUs, especially on Single Program Multiple Data (SPMD) programming paradigm on a large amount of data. Database engines are now everywhere, from different sizes and complexities, for multiple usages, embedded or distributed; in 2012, 500 million of SQLite active instances were estimated over the world. Our goal is to exploit the computation power of GPUs to improve performance of SQLite, which is a key software component of many applications and systems. In this paper, we introduce CuDB, a GPU-boosted in-memory database engine (IMDB) based on SQLite. The SQLite API remains unchanged, allowing developers to easily upgrade database engine from SQlite to CuDB even on already existing applications. Preliminary results show significant speedups of 70x with join queries on datasets of 1 million records. We also demonstrate the "memory bounded" character of GPU-databases and show the energy efficiency of our approach.

Keywords Relational Database, In-Memory, SQLite, GPU

I. Instruction

One of the most common components in many applications is related to database management. Compared to explicit data management (like C/C++ container), the main advantage of a relational database engine is its flexibility in data storage and manipulation. Relational databases are used in enterprise systems (ERP, CRM), in e-business applications (Apache, MySQL, PHP), in many personal applications (FireFox, Skype, GoogleGears, etc.), in embedded systems (iPhone and low cost cellular phones), and also as a native component in OS (e.g. Android and Symbian). With currently more than a billion copies of implementation, SQLite is probably currently the most widely deployed SQL database engine.

In 2004, a first attempt was made to process some database operations with a GPU [1]. At that time, the GPU architectures were not sufficient mature for general-purpose processing. GPGPU frameworks appeared much later. Since the first releases in 2007 of the CUDA framework and in 2009 for the OpenCL framework, it has become common to use GPUs in HPC environments for boosting scientific simulations. Nevertheless, GPUs are not commonly used for boosting database engines. Our goal is to show that a GPU-boosted relational database engine can provide drastic speedups while improving energy efficiency. In this paper we briefly introduce CuDB, a GPU boosted version of SQLite.

II. Related Works

In 2007 appeared GPUQP [2], one of the first experimental relational query processing engine working on a Graphics Processing Unit. With GPUQP, each operator of generated query plans could be processed either on CPU or GPU. The source code did not offi-

cially evolve since 2009 but it contributes to provide a reference database engine for many other contributions. In 2010, two researchers proposed Sphyraena [3], a GPU boosted version of SQLite. Unlike other solutions, Sphyraena does not split the query plans into sequences of parallel primitives which require multiple kernel calls. With Sphyraena, the whole query plan is processed on GPU with a single kernel call. Those previous researches have motivated us to start our own GPU-sided relational database engine. We described some specificities of our GPU-sided database, named CuDB, in a previous paper [4].
Meanwhile other teams started different types of researches, with GPU-database engines as central thematic. Sphyraena was used as base for Virginian [5], with as aim the development of a GPU-adapted table-structure. A group of researchers decide to study the impact of transaction mechanisms within GPU-databases and published the experimental GPUTx engine [6]. The main drawback of GPUTx is that it executes only pre-compiled procedures. Another experimental project is GPUDB [7] which was mainly build to run the Star Schema Benchmark. GPUDB has contributed to prove potential performances of GPU-databases with a reference benchmark.
Another group of researchers wanted to create a database engine which is able to run on different hardware architectures. They used GPUQP as reference engine, and developed the OmniDB [8] engine. The experimental CoGaDB [9] database engine allow the generation of query-plans which are dynamically adapted to the target hardware. Unlike most of previous cited solutions, the online available source code is currently still updated.Note also that two commercial solutions of GPU-sided database engines currently exist [10, 11] and a third database engine just started beta phases [12]. Those commercial solutions are more designed for Geographic Information Systems and the Big Data market. They do not encounter all the issues of a full relational DBMS.

III. THESIS IDEA

Before explaining the internel architecture of CuDB, it is necessary to understand how our reference engine, SQLite, works. SQLite is subdivided into 4 modules: (1) the interface which receive SQL queries, (2) SQL Command Processor which parses the queries and generates query plans, (3) Virtual Database Engine which executes the query plans, and (4) the database. Current version of CuDB engine preserves SQLite API and Command Processor. With CuDB, the Virtual Database Engine and the Database are replaced by our GPU versions. The CPU unit is in charge of parsing queries and translating it into query plans in the first two modules. A query plan is formed by a sequence of opcodes to be processed by a Virtual Machine. Our Virtual Machine is natively designed for GPU parallel architecture as well as our In-Memory Database Engine. This hybrid design was motived by several points: parsing and processing could not expect high speedup although process and storage operation on data can largely benefit of SIMD GPU architectures (several hundreds of synchronized cores). Figure 1 shows the internal architecture of CuDB.

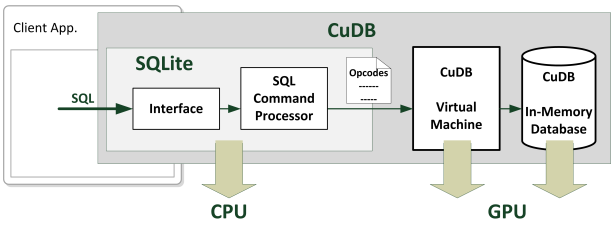


Figure 1: Internal architecture of CuDB.

CuDB engine preserves the original SQLite API, enabling fast, easy and efficient update of existing applications with minor source code updates.To take benefit of the high computation power of GPUs, with GPU-sided virtual machine, each GPU-thread processes the same query plan on its own records, allowing significant speedups with large datasets.
In 2013, a paper specific to the implementation of *SELECT WHERE* and *SELECT JOIN* queries with a GPU-database engine was published [13]. The chosen approach, for the implementation of join operations, was a trivial Cartesian product of tables, which procures a quadratic time complexity. With our engine, we preferred to use a temporary indexation structure for the processing of join-queries, which procure a quasi-linear time complexity. We made performance tests with JOIN queries on two non-indexed tables that are

composed by multiple numerical columns. The selectivity of the queries starts at 10% for small datasets and decreases to 0.1% for the one million row tables. Tables count both the same amount of records. We compared the execution time of CuDB, with a standard SQLite CPU implementation in which tables are stored in RAM memory. The specificities of the hardware we used for this performance evaluation are shown on table 1. Figure 2 shows the average execution time of the multiple join queries.

	CPU	GPU1	GPU2
Reference	Core i7 2600K	GT740	GTX770
Units	4 + HT	384	1536
Frequency	3.8GHz	~1GHz	~1GHz
Bandwidth	21GB/s	80GB/s	224GB/s

Table 1: Hardware specificities

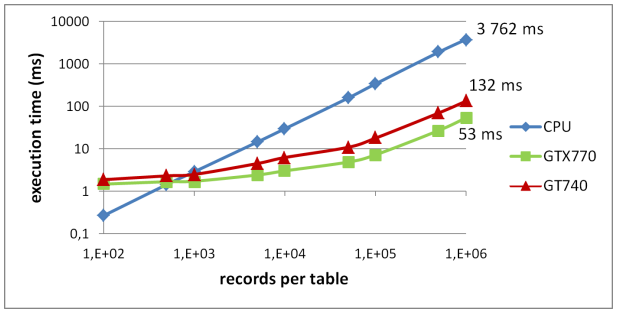


Figure 2: Average execution times with JOIN queries.

Our GPU database becomes as fast as the CPU version when the tables count a minimum of 800 records with GPU1 and 600 records with GPU2. We obtain relevant speedups on large datasets, and even modest GPUs like our GPU1 are able to procure substantial speedups. Our measures also show that performances of our system are clearly memory bounded and depending of query types, the processing time can be more impacted by the memory bandwidth than by the computation power of GPUs.
These results are encouraging but they are produced on non-indexed tables. When the record number of one table increases, performance of a indexed search in $O(log(n))$, running on a single thread CPU, overtakes a trivial parallel brute-force implementation $O(n/p)$, where p is the number of cores. Therefore, we are also currently working on indexation mechanisms for CuDB with better complexity.
During the performance evaluations, we also measured the total power consumption of our platforms. From the measured values we subtracted the idle power consumption to only show the part of energy consumption involved by the computation of the database system. Figure 3 shows the resulting total consumed energy.

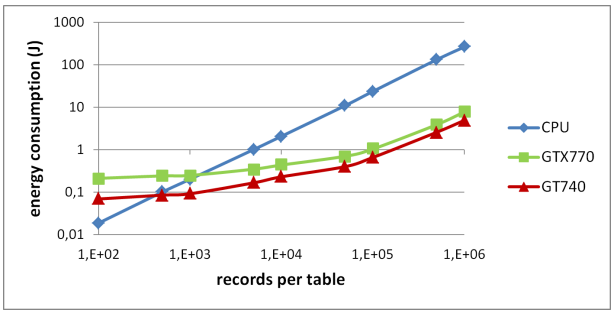


Figure 3: Average energy consumption with JOIN queries.

With our energy consumption tests, we show that the small GPU1 (manufactured in 28 nm) is more efficient than GPU2 (also 28 nm) because of its better "memory bandwidth" over "number of computation units" ratio, what confirms that our GPU database is memory bounded. With CuDB, we are currently working on different types of storage engines with different levels of data compactness and data types. We are also working with SoC architectures to provide a CuDB(m) version which will be dedicated to mobile and embedded applications. Instead of large systems, where the major manufacturers challenge was mainly focused on the processing speed over energy efficiency, small systems dedicated to embedded applications have major energy constraints, particularly due to the portable nature of devices (smartphone, auricular devices). In this field, SoC now offer higher energy efficiency than large systems, mainly due to better integration between components on the same chip (shared memory between CPU and GPU units). So, these small systems using less energy and boosted by environmental constraints, could offer a valuable alternative to existing HPC facilities.

IV. Conclusion and Future Works

In this paper, we have introduced CuDB, a GPU boosted relational database engine. CuDB is based on SQLite and preserves its user interface. We measured relevant speedups while the energy efficiency was increased up to 54 times with large datasets. With join queries, our GPU database always outperforms SQLite when tables counted more than one thousand records. Some significant SQL clauses like ORDER BY are still not being supported by our engine. The SQL support of CuDB needs to be improved, as aiming to run full database benchmarks. We need to deal with the GPU memory limitations and we plan to make a hybrid version of our engine where the CPU cores will process queries on small datasets, while the GPU still manages the greediest processing. We also showed that a GPU-boosted database engine is a memory bounded application. The future GPU architectures with stacked memory will drastically improve the available memory bandwidths. NVidia speaks about 1 TB/s with its next Pascal GPU architecture which will still increase the performances of GPU-database engines.

Acknowledgment

The authors would like to acknowledge the contribution of the Nesus COST Action IC1305.

References

[1] N.K. Govindaraju, B. Lloyd, W. Wang, M. Lin, and D. Manochad, "Fast computation of database operations using graphics processors," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, Paris, France, June 2004, pp. 215-216.

[2] R. Fang, B. He, M. Lu, K. Yang, N.K. Govindaraju, Q. Luo, and P.V. Sander, "GPUQP: query co-processing using graphics processors," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, Beijing, China, June 2007, pp. 1061-1063.

[3] P. Bakkum and K. Skadronr, "Accelerating SQL database operations on a GPU with CUDA," in *Proceedings of the 3rd Workshop on General-Purpose Computation on Graphics Processing*, Pittsburgh, Pennsylvania, March 2010, pp. 94-103.

[4] N. Dechamps, M. Bagein, M. Benjelloun, and S. Mahmoudi, "Boosting Open-Source Database Engines with Graphics Processors," in *Proceedings of the 2012 Seventh International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, Victoria, Canada, November 2012, pp. 262-266.

[5] P. Bakkum and S. Chakradhar, "Efficient Data Management for GPU Databases," 2012. *http://pbbakkum.com/virginian/paper.pdf* Accessed : 2015-08-11.

[6] B. He, and J. Xu Yu, "High-throughput transaction executions on graphics processors," *VLDB Endowment*, vol. 4, no. 5, pp. 314-325, 2011.

[7] S. Zhang, J. He, B. He, and M. Lu, "OmniDB: towards portable and efficient query processing on parallel CPU/GPU architectures," *VLDB Endowment*, vol. 6, no. 12, pp. 1374-1377, 2013.

[8] Y. Yuan, R. Lee, and X. Zhang, "The Yin and Yang of processing data warehousing queries on GPU devices," *VLDB Endowment*, vol. 6, no. 10, pp. 817-828, 2013.

[9] S. Breß, N. Siegmund, L. Bellatreche, and G. Saake, "An operator-stream-based scheduling engine for effective GPU coprocessing," *Advances in Databases and Information Systems*, vol. 8133, pp. 288-301, 2013.

[10] Parstream, "Parstream - turning data into knowledge," White Paper, November 2010.

[11] GPUdb, *www.gpudb.com*, Accessed : 2015-07-23.

[12] T. Mostak, "An overview of MapD (massively parallel database)," White Paper, Massachusetts Institute of Technology, 2013.

[13] M. Pietron, P. Russek, and K. Wiatr, "Accelerating select where and select join queries on a GPU," *Computer Science (AGH)*, vol. 14, no. 2, pp. 243-252, 2013.

The analysis of parallel OpenFOAM solver for the heat transfer in electrical power cables

Andrej Bugajev, Raimondas Čiegis

Vilnius Gediminas Technical University, Sauletekio ave. 11, Vilnius
andrej.bugajev@vgtu.lt

Abstract

Here we present the part of results obtained in PhD thesis "The investigation of efficiency of physical phenomena modelling using differential equations on distributed systems" by Andrej Bugajev. This work is dedicated to development of mathematical modelling software. While applying a numerical method it is important to take into account the limited computer resources, the architecture of these resources and how do methods affect software robustness. Three main aspects of this investigation are that software implementation must be efficient, robust and be able to utilize specific hardware resources. The hardware specificity in this work is related to distributed computations. The investigation is done for FVM method usage to implement efficient calculations of a very specific heat transferring problem. That lets to create technological components that make a software implementation robust and efficient. OpenFOAM open source software is selected as a basis for implementation of calculations and a few algorithms to solve efficiency issues are proposed. The FVM parallel solver is implemented and analyzed, it is adapted to heterogeneous cluster Vilkas.

Keywords Finite Volume Method, OpenFOAM, parallel algorithms, domain decomposition, distributed computing, parallel computing

I. Motivation

This work is dedicated to proposal of technological solutions for developing design rules for power transmission lines and cables (1, [1]), which have to meet the latest power transmission network technical and economical requirements.

In order to do that it is necessary to develop specific software solutions. At present, sizes of the power lines are up to 60% bigger than is necessary in terms of transmitted power. However, as the new distributed generating capacities are installed e.g. large wind farms, bio-gas plants or waist-to-energy plants, the infrastructure of power grid must be re-designed or new optimization strategies for the available grid must be developed. Power cables for power distribution applications are still rated according to IEC 287 and IEC 853 standards, which use the Neher and McGrath meth-

Figure 1: Typical high-voltage (110 kV) cables [1]

ods proposed in 1957 [2]. Obviously, these formulas cannot accurately account for the various conditions under which the cables are actually installed and used. They estimate the cable's current-carrying capacity (so-called *ampacity*) with significant margins to stay on the safe side [3]. The safety margins can be quite large and result in 50–70% usage of actual resources. A more accurate mathematical modelling is needed to meet the latest technical and economical requirements and to elaborate new, improved, cost-effective design rules and standards. Today there are many applications where analytical and heuristic formulas cannot describe precisely enough the conditions under which the cables are installed. The present standards require that the cable's current-carrying capacity must be reduced according to the worst-case scenario. To be on the safe side this rule is acceptable, but today the cost effective designing of cable installations comes first as the copper price level has reached its maximum value.

When we need to deal with mathematical models for the heat transfer in various media (metals, insulators, soil, water, air) and non-trivial geometries, only the means of parallel computing technologies can allow us to get results in an adequate time. To solve numerically selected models, we develop our numerical solvers using the OpenFOAM package.

II. Related work

The knowledge of dynamics (in time) of heat distribution in/around electrical cables is necessary to optimize the usage of electricity transferring infrastructure. It is important to determine: maximal electric current for the cable, optimal cable parameters in certain circumstances, cable life expectancy, other engineering factors. To solve the optimization problem it is necessary to implement an efficient modelling software for heat distribution in cables. Fundamentals of the heat distribution in cables are given in [4], but for further readings refer [5, 6, 7]. [8] and [9] presented efficient parallel numerical algorithms for simulation of temperature distribution in electrical cables for mobile devices and cars and solved inverse problem for fitting the diffusion coefficient of the air-isolation material mixture to the experimental data. Numerical algorithms for parabolic and elliptic problems with discontinuous coefficients have been widely investigated in many papers. The use of standard finite element method (FEM) to solve interface problems is equivalent to arithmetic averaging of discontinuous coefficients. The mixed FEM leads to the harmonic averaging if special quadrature formula are used – see, e.g. works by [10] and [5]. Conservative finite-difference schemes for approximation of parabolic and elliptic problems were derived by [11] and [12]. These schemes are robust and use only general assumptions on the position of the interface. Also such finite difference schemes were proposed, which approximate with the second order of accuracy both – the solution and the normal flux through the interface – see [13, 14] for details.

In recent years, scalability and performance of parallel OpenFOAM solvers are actively studied for various applications and HPC platforms. In [15] it is noted that the scalability of parallel OpenFOAM solvers is not very well understood for many applications when executed on massively parallel systems.

We note that an extensive experimental scalability analysis of selected OpenFOAM applications is one of the tasks solved in PRACE (Partnership for Advanced Computing in Europe) project, see [16], [17]. In [16] are presented results on IBM BlueGene/Q (Fermi) and Hewlett Packard C7000 (Lagrange) parallel supercomputers for a few CFD applications with different multiphysics models. The presented experimental results are showing a good scaling and efficiency with up to 2048–4096 cores. It is noted that such results are expected when balancing between computation, message passing and I/O work is good. Obviously, the next generation of ultrascale computing systems will cause additional challenges due to their complexity and heterogeneity.

The most important challenges for parallel solvers implemented in OpenFOAM are the following: a) efficiency of solvers on hybrid heterogeneous parallel systems, b) sensitivity of the parallel preconditioners to data distribution algorithms, c) workload balancing on heterogeneous parallel systems. For mathematical models describing coupled multi-physics problems, it is important to investigate two different approaches to design robust and efficient solvers for such problems [18]. Monolithic solvers operate directly on the

system of nonlinear algebraic equations, obtained after the discretization ofthe system of PDEs. In the partitioning approache the discrete system is solved by using the single-physics solvers in decoupled fixed-point iterations. The latter aproach is implemented in OpenFOAM. A good review for a comparison of some popular fixed-point methods is given in [19].

III. Thesis idea

In this work, we study the performance of parallel OpenFOAM-based solver for heat conduction in electrical power cables. For computational experiments, we use the following 2D benchmark problem:

$$\begin{cases} c\rho\dfrac{\partial T}{\partial t} = \nabla\cdot(\lambda\nabla T) + q, & t\in[0,t_{max}], x\in\Omega, \\ T(x,0) = T_b, & \text{when } x\in\Omega, \\ T(x,t) = T_b, & \text{when } x\in\partial\Omega, \\ [T] = 0, \quad [\lambda\nabla T] = 0 & \text{when } x\in\partial\Omega_D, \end{cases} \tag{1}$$

here $x = (x_1, x_2)$, $T(x,t)$ is temperature, $\lambda(x) > 0$ is heat conductivity coefficient, $q(x,t,T)$ is the source function, $\partial\Omega$ is the contour of domain Ω, $\rho(x) > 0$ defines mass density, $c(x) > 0$ is specific heat capacity, T_b, t_{max} are given constants. Operator $\nabla\cdot(\lambda\nabla T) = \sum\limits_{j=1}^{2}\frac{\partial}{\partial x_j}\left(\lambda\frac{\partial T}{\partial x_j}\right)$ is the diffusion operator. The solution and flux continuity conditions are satisfied on boundaries of domains with different diffusion coefficients $\partial\Omega_D$.

When we need to deal with 2D and 3D mathematical models for the heat transfer in various media (metals, insulators, soil, water, air) and non-trivial geometries, only parallel computing technologies can allow us to get results in an adequate time. To solve numerically selected models, we develop our numerical solvers using the OpenFOAM package. OpenFOAM is a free, open source CFD software package. It has an extensive set of standard solvers for popular CFD applications. It also allows us to implement new models, numerical schemes and algorithms, utilizing the rich set of OpenFOAM capabilities. The important consequence of this software development approach is that numerical solvers can automatically exploit the basic parallel computing capabilities already available in the OpenFOAM package.

In this work, we study and analyze the parallel performance of OpenFOAM-based solver for heat conduction in electrical power cables. The main goal is to consider the scalability and efficiency of the developed parallel solver in the case when the parallel system is not big, but it consists of non homogeneous multicore nodes. The mesh is adaptive and it is partitioned by using Scotch method. Then load balancing techniques must be used in order to optimize the parallel efficiency of the solver. The second aim is to investigate the sensitivity of parallel preconditioners with respect to the number of processes.

IV. Conclusions and future work

1. Smaller problems enable a better caching and give a hardware-based speed-up for computations.
2. The uniform distribution of problems sizes is enough to solve the problem on homogeneous set of nodes, however this strategy is inefficient on heterogeneous set of nodes.
3. The load balancing lets to use different nodes efficiently in a heterogeneous cluser.
4. The future investigation of parallel efficiency dependence on preconditioners may lead to additional optimization of parallel solvers. This is especially important for large parallel systems.
5. One of the main challenges in future work is modelling the problem with multi-physics on parallel systems. In this case some parts of the whole domain have effects, described by Navier-Stokes equations and the rest part has diffusion only.

Acknowledgment

The paper was supported by NESUS project “Winter School & PhD Symposium 2016”.

References

[1] Z. Dongping. “Optimierung zwangsgekühlter Energiekabel durch dreidimensionale FEM-

Simulationen," *Doctoral thesis, Universität Duisburg-Essen*, 2009.

[2] J. H. Neher, M. H. McGrath. "The Calculation of the temperature rise and load capability of cable systems," *AIEE Transactions*, Vol. 76, Part III, pp. 752–772, 1957.

[3] I. Makhkamova. "Numerical Investigations of the Thermal State of Overhead Lines and Underground Cables in Distribution Networks," *Doctoral thesis, Durham University*, 2011.

[4] F. Incropera, P. DeWitt, P. David. *Introduction to heat transfer,* John Willey & Sons, New Yourk, 1985.

[5] A. Ilgevicius. "Analytical and numerical analysis and simulation of heat transfer in electrical conductors and fuses," *Doctoral thesis, Universität der Bundeswehr München*, 2004.

[6] A. Ilgevicius, H.D. Liess. "Calculation of the heat transfer in cylindrical wires and electrical fuses by implicit finite volume method," *Mathematical Modelling and Analysis*, Vol. 8, No. 3, pp. 217–228, 2003.

[7] J. Taler, P. Duda. *Solving Direct and Inverse Heat Conduction Problems,* Springer, Berlin, 2006.

[8] R. Čiegis, A. Ilgevičius, H. Liess, M. Meilūnas, O. Suboč. "Numerical simulation of the heat conduction in electrical cables," *Mathematical modelling and analysis*, Vol. 12, No. 4, pp. 425–439, 2007.

[9] Raim. Čiegis, Rem. Čiegis, M. Meilūnas, G. Jankevičiūtė, V. Starikovičius "Parallel numerical algorithm for optimization of electrical cables," *Mathematical modelling and analysis*, Vol. 13, No.4, pp. 471–482, 2008.

[10] R. Falk, J. Osborn, "Remarks on mixed finite element methods for problems with rough coefficients," *Math. Comp.*, Vol. 62, No. 205, pp. 1–19, 1994.

[11] A.A. Samarskii, *The Theory of Difference Schemes*. Marcel Dekker, Inc., New York–Basel, 2001.

[12] A.N. Tichonov, A.A. Samarskii, "Homogeneous finite difference schemes," *Zh. Vychisl. Mat. Mat. Fiziki*, Vol. 1, No. 1, pp. 5–63, 1961.

[13] V.P. Il'in, "High order accurate finite volumes discretization for Poisson equation," *Siberian Math. J.*, Vol. 37, No.1, pp. 151–169, 1996.

[14] R. LeVeque, Z. Li. Erratum, "The immersed interface method for elliptic equations with discontinuous coefficients and singular sources," *SIAM J. Numer. Anal.*, Vol. 32, No 5, pp. 1704–1704, 1995.

[15] O. Rivera, K. Furlinger, D. Kranzimuller, "Investigating the scalability of OpenFOAM for the solution of transport equations and large eddy simulations," *Lecture Notes in Computer Science*, Vol. 7017, pp. 121–130, 2011

[16] P. Dagna. "OpenFOAM on BG/Q porting and performance," *Prace report*, CINECA, Bologna, Italy 2012.

[17] M. Culpo. "Current bottlenecks in the scalability of OpenFOAM on massively parallel clusters," *Prace white papers*, CINECA, Bologna, Italy 2012.

[18] R. Muddle, M. Milhajlovic, M. Heil. "An efficient preconditioner for monolithically-coupled large-displacement fluid-structure interaction problems with pseudo-solid mesh updates," *Journal of Computational Physics*, Vol. 231, No. 21, pp. 7315–7334, 2012.

[19] U. Kuettler, W. Wall. "Fixed-point fluid-structure interaction solvers with dynamic relaxation," *Computational Mechanics*, Vol. 43, No. 1, pp. 61–72, 2008.

Cloud Resource Management

Tychalas Dimitrios
PhD Student
Aristotle University of Thessaloniki, Greece
dtychala@csd.auth.gr

Karatza Helen
Supervisor
Aristotle University of Thessaloniki, Greece
karatza@csd.auth.gr

Abstract

Nowadays computational needs increase exponentially every year. We analyze, calculate and process large data sets every day and the "traditional" servers do not meet these computational criteria. As a result cloud computing was "invented" offering multiple resources at an affordable cost. Besides that, Cloud Computing supports scalability, fault tolerance and high availability [2] [16]. Our goal is to delve deeper into Cloud Computing to be able to carry out independent research to study and improve the state of the art load balancing techniques.

Keywords Ultrascale systems, NESUS, Cloud computing, Load balancing, Fault tolerance, High availability, Scalability

I. Introduction

Cloud computing is one of the most fast-growing fields in computer science [2]. Almost everyone has access to Internet via his smart-phone/tablet/PC [18] and access his data from anywhere. In the near future everything would be on the "cloud" making the network needs to grow exponentially. As a result the next-generation of cloud computing will thrive on how effectively the infrastructure is used and if the available resources can be utilized dynamically [1]. Load balancing distributes the load across multiple virtual machines to ensure that the service is always accessible and the resources are utilized in the best effort. Moreover a "good" load balancer should adapt its decisions to the changing environment [17] [19].

The main goal of this thesis is to examine the known load balancing techniques and algorithms and improve them in the cost and energy saving aspects [19].

II. Related work

The most used load balancing techniques [15] are:

1. **Round Robin:** Incoming requests are distributed sequentially across the available virtual machines. All virtual machines should be homogeneous.
2. **Weighted Round Robin:** Incoming requests are distributed across the virtual machines in a sequential manner, while taking account of a static "weight" that can be pre-assigned per VM. This method is preferred on heterogeneous VMs.
3. **Least Connection:** Incoming requests are distributed on the basis of the connections that every VM is currently maintaining. The VM with the least number of active connections automatically is selected.
4. **Weighted Least Connection:** Incoming requests are distributed across the virtual machines with the fewer active connections, while taking account of predefined "weight" for each VM.

There are a number of works that are employing load balancing algorithms that take in account current requirements for CPU performance like [4] [8] [9] [20]. However despite the high performance achieved by the aforementioned algorithms, they lead to high energy consumption. This resulted in the development of many routing algorithms for power awareness as [11] [21] [24].

III. Thesis idea

Cloud computing is so involved in our every day lives and spread among many different aspects of research. It is the ideal area for aspiring computer scientists to keep themselves up to date with the latest technologies. In our research we will study the load balancing technologies and we will address open issues.

In order to examine the state of the art algorithms and techniques in this field, we first developed a Web Framework that uses more than one Virtual Machine in order to address the problems of the "classic" servers. The main problems are faults, as power failure, errors on system or on hardware, expensive hardware when scalability is needed and of course the overloading on the server when multiple users are connected simultaneously. The system is intended to deal with all the aforementioned problems using:

1. Virtual Machines, by ∼okeanos [12]
2. MySQL Cluster [3] [13]
3. Apache as Load Balancer [6] [14]
4. GlusterFS [23]

The system employs load balancing to handle the multiple requests. There are many ways to balance traffic between systems [15], but the most effective one is using weights. The weight is determined by counting the requests that each server has and how much time is needed to serve all of them. The output of this study was published in [22].

Secondly we utilized the package JPPF (Java Parallel Processing Framework) which enables applications with large processing power requirements to be run on any number of computers. This is done by splitting an application into smaller parts and executes them simultaneously on different machines [7]. We used the above package in order to write our own load balancing rules and use it in co-operation between a Desktop PC and a Raspberry. Our load balancing algorithm works with meta-tags in every task. If the meta-tags of a task meet the minimum needs, then the Raspberry is used in order to process the task, alternatively the task is processed by the Desktop.

Finally we are developing our own simulation program in C in order to test the above systems with more virtual machines or with more Servers - Raspberries.

IV. Future work

As future work we are going to use KVM [5] as virtualization solution because we can increase or decrease the number of CPUs and the amount of RAM on-the-fly, without the need of restarting the virtual machine [10]. As a result we can increase the resources when it is needed and decrease them in order to save energy and money.

V. Acknowledgment

We would like to acknowledge the contribution of the academic cloud service ∼okeanos [12] for giving us the ability to create the necessary virtual machines for the above case study. We would also like to acknowledge the contribution of the COST Action IC1305 NESUS (Network for Sustainable Ultrascale Computing).

References

[1] Omer F. Rana Antonio Corradi. “The management of cloud systems”. In: *Future Generation Computer Systems* 32 (2014), pp. 24–26.

[2] Michael Armbrust et al. “A view of cloud computing”. In: *Communications of the ACM* 53.4 (2010), pp. 50–58.

[3] Charles Bell, Mats Kindahl, and Lars Thalmann. *MySQL high availability: tools for building robust data centers*. " O’Reilly Media, Inc.", 2010.

[4] Anton Beloglazov and Rajkumar Buyya. “Energy Efficient Resource Management in Virtualized Cloud Data Centers”. In: *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. CCGRID ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 826–831. ISBN: 978-0-7695-4039-9. DOI: 10.1109/CCGRID.2010.46. URL: http://dx.doi.org/10.1109/CCGRID.2010.46.

[5] Anton Beloglazov et al. "Deploying OpenStack on CentOS using the KVM Hypervisor and GlusterFS distributed file system". In: *Cloud Computing and Distributed Systems (CLOUDS) Laboratory Department of Computing and Information Systems, The University of Melbourne, Australia* (2012).

[6] Trieu C Chieu et al. "Dynamic scaling of web applications in a virtualized cloud computing environment". In: *e-Business Engineering, 2009. ICEBE'09. IEEE International Conference on*. IEEE. 2009, pp. 281–286.

[7] L. Cohen. *Java Parallel Programing Framework*. 2005. URL: http://www.jppf.org (visited on 12/26/2015).

[8] Shridhar G Domanal and G Ram Mohana Reddy. "Optimal load balancing in cloud computing by efficient utilization of virtual machines". In: *Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on*. IEEE. 2014, pp. 1–4.

[9] James Michael Ferris. *Load balancing in cloud-based networks*. US Patent 8,849,971. Sept. 2014.

[10] *Hotplug (qemu disk,nic,cpu,memory)*. 2015. URL: https://pve.proxmox.com/wiki/Hotplug_(qemu_disk,nic,cpu,memory) (visited on 12/26/2015).

[11] Myungsun Kim et al. "Utilization-aware load balancing for the energy efficient operation of the big. LITTLE processor". In: *Proceedings of the conference on Design, Automation & Test in Europe*. European Design and Automation Association. 2014, p. 223.

[12] Vangelis Koukis, Constantinos Venetsanopoulos, and Nectarios Koziris. "~ okeanos: Building a Cloud, Cluster by Cluster". In: *IEEE Internet Computing* 3 (2013), pp. 67–71.

[13] Arjen Lentz. "MySQL Cluster Introduction". In: *White Paper* (2006).

[14] Quanzhong Li and Bongki Moon. "Distributed cooperative Apache web server". In: *Proceedings of the 10th international conference on World Wide Web*. ACM. 2001, pp. 555–564.

[15] *Load Balancing Scheduling Methods Explained | LoadBalancerBlog.com*. 2013. URL: http://loadbalancerblog.com/blog/2013/06/load-balancing-scheduling-methods-explained (visited on 12/26/2015).

[16] Ioannis A Moschakis and Helen D Karatza. "Enterprise HPC on the Clouds". In: *Cloud Computing for Enterprise Architectures*. Springer, 2011, pp. 227–246.

[17] Ioannis A Moschakis and Helen D Karatza. "Evaluation of gang scheduling performance and cost in a cloud computing system". In: *The Journal of Supercomputing* 59.2 (2012), pp. 975–992.

[18] Ioannis A Moschakis and Helen D Karatza. "Towards scheduling for Internet-of-Things applications on clouds: a simulated annealing approach". In: *Concurrency and Computation: Practice and Experience* (2013).

[19] Ioannis Moschakis, Helen D Karatza, et al. "Performance and cost evaluation of Gang Scheduling in a Cloud Computing system with job migrations and starvation handling". In: *Computers and Communications (ISCC), 2011 IEEE Symposium on*. IEEE. 2011, pp. 418–423.

[20] Kumar Nishant et al. "Load balancing of nodes in cloud using ant colony optimization". In: *Computer Modelling and Simulation (UKSim), 2012 UKSim 14th International Conference on*. IEEE. 2012, pp. 3–8.

[21] George Terzopoulos and Helen Karatza. "Power-aware load balancing in heterogeneous clusters". In: *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2013 International Symposium on*. IEEE. 2013, pp. 148–154.

[22] Dimitris Tychalas and Helen Karatza. "A cloud system for health care". In: *Proceedings of the 19th Panhellenic Conference on Informatics*. ACM. 2015, pp. 169–170.

[23] YANG Yong. "Distribution Redundancy Storage Based on GlusterFS". In: *Journal of Xifffdfffdfffdan University of Arts & Science (Natural Science Edition)* 4 (2010), pp. 67–70.

[24] Andrew J Younge et al. “Efficient resource management for cloud computing environments”. In: *Green Computing Conference, 2010 International*. IEEE. 2010, pp. 357–364.

Techniques for Autotuning Algorithms on Heterogenous Platforms

Adrián P. Diéguez, Margarita Amor, Ramón Doallo

University of A Coruña, Spain
{adrian.perez.dieguez,margarita.amor,ramon.doallo}@udc.es

Abstract

*Current GPUs (*Graphic Processing Units*) can obtain high computational performance in scientific applications. Nevertheless, programmers have to use suitable parallel algorithms for these architectures and have to consider optimization techniques in the implementation in order to achieve that performance. This thesis is focused on designing and implementing parallel prefix algorithms into GPU architectures with little effort. For that, we have developed a very optimized library called BPLG (Tuning Butterfly Processing Library for GPUs) and based on a set of building blocks that enable to easily design well-known algorithms such as FFT, tridiagonal systems solvers, scan operator, sorting or signal processing. This library is designed under a tuning methodology based on two-stages indentified as GPU resource analysis and operator string manipulation. Specifically, this strategy is focused on a set of parallel prefix algorithms that can be represented according to a set of common permutations of the digits of each of its element indices [4], denoted as Index-Digit (ID) algorithms. So far, the proposed methodology has obtained very good results with respect to state-of-art libraries, as CUFFT, CUSPARSE, CUDPP or ModernGPU.*

Keywords CUDA, parallel prefix algorithms, GPU, ID-algorithms, tuning

I. Motivation

In recent years, GPUs (Graphics Processing Units) have experienced a noticeable increase in its relevance and usage in high performance computing. Nevertheless, programmers have to use suitable parallel algorithms for these architectures that also require special languages such as NVIDIA CUDA or OpenCL; and finally, have to consider optimization techniques in the implementation in order to achieve high performance.

The algorithms examined in this thesis are described using a parallel prefix approach [17], one of the most popular parallel paradigms. Some parallel prefix algorithms may be also represented according to a set of common permutations of the digits of each element index [4], denoted as Index-Digit (ID) algorithms. In this thesis, we have focused on the following ID-algorithms: FFT, Tridiagonal Systems Solvers, Scan Operator and Sorting algorithms.

The *FFT* is a highly important operation for many applications, such as image and digital signal processing, filtering, compression or partial differential equation resolution. Tridiagonal linear systems arise in many scientific and engineering problems such as fluid dynamics, heat conduction, numerical analysis, ocean models or cubic spline approximations. The scan operator is widely used in areas such as the construction of summed area tables, stream compaction, image filtering, or cryptography, among many others. Sorting is a computational building block of high importance, being one of the most studied algorithms due to its impact. Many algorithms rely on the efficiency of sorting routines. For example, computer graphics, and geographic information systems or MapReduce patterns.

Thus, it is relevant the importance of efficently solving these algorithms. For that, GPUs provides an excellent hardware desing where executing these parallel algorithms. For achieving this goal, there are several proposals in order to facilitate the programmability of these architectures: automatic parallelization, directive-based compiler approaches and auto-tuning frameworks or libraries.

Automatic parallelization and performance optimization of affine loop nests on GPU is developed using a polyhedral compiler model of data dependence abstraction and program transformation. In [2], a compiler algorithm revises data placement across different types of GPU resources using input optimized programs. Shared memory multiplexing [22] allows a higher number of thread blocks to be executed concurrently. GPU caches suffer contention due to massive multithreading, an adaptive cache bypass is presented in [20] in order to reduce contention and preserve space for reused cache lines.

Frameworks using directive-based compiler approaches [19, 18] have been developed to automatically optimize GPU programs. Most of this kind of libraries require to have GPU expertise, specifying the number of threads to be used, which loops are parallelised or when to synchronize. Furthermore, the code is not easily readable, complicating the tuning process, and there are some limitations as programmer cannot use CUDA intrinsic functions within the accelerator region.

Autotuning is a very interesting option for applications whose execution time, memory usage or energy consumption can vary depending on a set of parameters and their execution environment. These parameters can take a small number of values and the autotuner determines the best combination to maximise an user-defined metric. On GPUs, there are various tunable parameters, such as the number of warps per block or the *workload* per thread. Nevertheless, this technique requires writing code in a parametrized way to accommodate various performance tuning parameters. Taking into account previous proposals disadvantages, we have decided to focus my thesis on this approach.

II. Related Work

There are several implementations on GPU for each cited algorithm. Furthermore, there are also some GPU methodologies based on an autotuning approach. All of them are studied in this section.

There are some auto-tuning proposals for FFTs on *GPUs*, achieving high performance, such as [21]. Specifically, approaches focused on large 1D FFT on a single coprocessor is [21]. However, the most used and well-known *GPU* implementation is *NVIDIA's CUFFT* [12]. There are some *GPU* tridiagonal solvers implementations based on different algorithms, such as [23, 10]. There are also *GPU* proposals based on auto-tuning design for tridiagonal solvers in [1]. Most scan implementation on GPU are based on either the Kogge-Stone or the Brent-Kung parallel prefix patterns, being important [8] and [9]. Finally, there are several parallel sorting algorithms which have been developed for GPUs. Radix sort for GPUs was efficiently implemented in [11] and Quicksort algorithm in GPU was implemented in [3].

Most of previous approaches provide a solution focused in just one algorithm; however, there is a growing trend of using acceletared libraries that solve this and other parallel algorithm being devoted to a set of algorithms. Our proposal gives a solution based on the development of a small number of efficient parametrizable skeleton building blocks carefully designed to achieve high level of efficiency in CUDA architecture and thought to be used by a set of parallel prefix algorithms instead of focusing on just one. Other examples are CUSPARSE [16] and CUDPP [14], accelerated libraries developed by *NVIDIA*; Merrill's CUB [15] and ModernGPU [13].

III. Thesis idea

The thesis is focused on developping a 2-stage methodology for implementing efficient parallel prefix algorithms on GPU architectures. In the first stage,

performance parameters are obtained from a GPU performance analysis in order to achieve a set of premises such as the maximum parallelism to keep all elements of the GPU occupied. In the second stage, CUDA kernels are obtained from a combination of two techniques called *index-digit permutations* and *tuned mapping vector*, which are used to adjust the data distribution in the GPU according to the resource analysis made at the first stage and the digits of the element's index. Furthermore, our code is designed as building blocks. That means, the functions used are very abstract and they can be reused for the different algorithms. These functions, or building blocks, are parameterized (data types and performing variables are unspecified) and then, the corresponding tuned parameters for each architecture are selected at compile-time and sent them to these functions. So, in the end, thanks to this parametrization of the code, we are designing GPU algorithms with little effort and obtaining very competitive performance with respect to other approaches.

Depending on the size of the problem, we have divided the development of our methodology in three phases:

- The problem data fits in shared memory. Each problem is assigned to a single CUDA block, using the shared memory to perform communications.
- The problem size is bigger than shared memory but can be allocated in the GPU memory of a single GPU. The work is distributed among several blocks, using several kernels for coordinating them.
- The problem size is bigger than the GPU memory of a single GPU, using streams and MPI for dealing with that in a *MultiGPU* approach.

So far, we have implemented FFT, Hartley transform, Discrete cosine transform, different tridiagonal systems solvers, different scan operator algorithms and an algorithmic variant of Bitonic Sort for sorting; obtaining very good results [5, 6, 7] with respect to other state-of-art libraries such as *CUDPP*, *CUSPARSE*, *CUFFT* and *ModernGPU*.

IV. Conclusions

This thesis presents a two-stages methodology for efficiently implementing parallel prefix algorithms into GPU architectures with little effort. Specifically, the strategy is focused on a set of algorithms known as ID-algorithms. In the first stage a *GPU resource analysis* is performed, where performance parameters are obtained from a GPU performance analysis. In the second stage, *operators string manipulation*, kernels are obtained after adjusting the data distribution in the GPU according to the first stage. These kernels are developed with a set of *building blocks* that enable to easily design flexible code, and are integrated in our BPLG library (*Tuning Butterfly Processing Library for GPUs*).

Depending on the problem size, three different strategies have been considered. So far, we have tested this methodology for small and medium problem sizes, outperforming well-known libraries as *CUFFT*, *CUSPARSE*, *CUDPP* and *ModernGPU*.

Acknowledgment

This work is supported by EU under the COST Program Action IC1305: Network for Sustainable Ultrascale Computing (NESUS).

References

[1] A. Davison and J. D. Owens. Register Packing for Cyclic Reduction: A Case Study. In *Proc. of the Fourth Workshop on General Purpose Processing on Graphics Processing Units (GPGPU-4)*, pages 4:1–4:6, 2011.

[2] C. Li, Y. Yang, Z. Lin and H. Zhou. Automatic Data Placement into GPU On-Chip Memory Resources, booktitle = Proceedings of the 13th Annual IEEE/ACM International Symposium on Code Generation and Optimization, CGO'15, year = 2015, pages = 23–33.

[3] Daniel Cederman and Philippas Tsigas. GPU-Quicksort: A Practical Quicksort Algorithm for

Graphics Processors. *J. Exp. Algorithmics*, 14:4:1.4–4:1.24, January 2010.

[4] D. Fraser. Array Permutation by Index-Digit Permutation. *Journal of ACM*, 23(2):298–309, 1976.

[5] Adrián P. Diéguez, Margarita Amor, and Ramón Doallo. Efficient Scan Operator Methods on a GPU. In *Proceedings of the 2014 IEEE 26th International Symposium on Computer Architecture and High Performance Computing*, SBAC-PAD '14, pages 190–197, 2014.

[6] Adrián P. Diéguez, Margarita Amor, and Ramón Doallo. BPLG-BMCS: GPU-sorting algorithm using a tuning skeleton library. *The Journal of Supercomputing*, pages 1–13, 2015.

[7] Adrian P. Diéguez, Margarita Amor, and Ramon Doallo. New Tridiagonal Systems Solvers on GPU architectures. In *Proceedings of IEEE International Conference on High Performance Computing (2015), HiPC'15 (accepted)*, 2015.

[8] D.Merrill and A. Grimshaw. Parallel scan for stream architectures. In *Technical report*. Dept. of Computer Science, Univ. of Virginia, December 2009.

[9] Yuri Dotsenko, Naga K. Govindaraju, Peter-Pike Sloan, Charles Boyd, and John Manferdelli. Fast scan algorithms on graphics processors. In *Proceedings of the 22Nd Annual International Conference on Supercomputing (2008)*, pages 205–213, 2008.

[10] H.-S. Kim, S. Wu, L.-W. Chang, W.W. Hwu. A Scalable Tridiagonal Solver for GPU. In *Int. Conf. on Parallel Processing*, pages 444–453, 2011.

[11] Mark Harris, Shubhabrata Sengupta, and John D Owens. Parallel prefix sum (scan) with CUDA. *GPU Gems*, 3(39):851–876, 2007.

[12] NVIDIA. *CUDA CUFFT Library*, 2012. v5.0.

[13] Nvidia Comp. Modern gpu library, 2013.

[14] Nvidia Comp. CUDPP: CUDA Data Parallel Primitives Library, 2014.

[15] Nvidia Comp. Cub library, 2015.

[16] NVIDIA-Corporation. CUDA CUSPARSE Library. 2012.

[17] R. E. Ladner and M. J. Fischer. Parallel Prefix Computation. *Journal of the ACM*, 27(4):831–838, 1980.

[18] S. Wienke, P. Springer, C. Terboven, D. an Mey. OpenACC: First Experiences with Real-world Applications. In *Proceedings of the 18th International Conference on Parallel Processing, EuroPar12*, pages 859–870, 2012.

[19] T. Han and T. Abdelrahman. hiCUDA: A High-level Directive-based Language for GPU Programming. In *GPGPU-2: Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units*, pages 52–61, 2009.

[20] X. Chen, S. Wu, L.-W. Chang, W.-S. Huang, C. Pearson, Z. Wang and W.-M. W. Hwu. Adaptive Cache Bypass and Insertion for Many-core Accelerators. In *Proceedings of International Workshop on Manycore Embedded Systems, MES'14*, pages 1:1–1:8, 2014.

[21] Y. Dotsenko, S.S. Baghsorkhi, B. Lloyd and N.K. Govindaraju. Auto-Tuning of Fast Fourier Transform on Graphics Processors. In *Proceedings of Principles and Practice of Parallel Programming (PPoPP '11)*, pages 257–266, 2011.

[22] Y. Yang, P. Xiang, M. Mantor, N. Rubin and H. Zhou. Shared memory multiplexing: A novel way to improve gpgpu throughput. In *Proceedings of the 21st International Conference on Parallel Architectures and Compilation Techniques*, PACT '12, pages 283–292. ACM, 2012.

[23] Y. Zhang, J. Cohen, J.D. Owens. Fast Tridiagonal Solvers on the GPU. In *Proceedings of the 15th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP 2010)*, pages 127–136, 2010.

Resilience of Parallel Applications

Nuria Losada, María J. Martín, Patricia González

Universidade da Coruña, Spain
{nuria.losada, mariam, pglez}@udc.es

Abstract

Future exascale systems are predicted to be formed by millions of cores. This is a great opportunity for HPC applications, however, it is also a hazard for the completion of their execution. Even if one computation node presents a failure every one century, a machine with 100.000 nodes will encounter a failure every 9 hours. Thus, HPC applications need to make use of fault tolerance techniques to ensure they successfully finish their execution. This PhD thesis is focused on fault tolerance solutions for generic parallel applications, more specifically in checkpointing solutions. We have extended CPPC, an MPI application-level portable checkpointing tool developed in our research group, to work with OpenMP applications, and hybrid MPI-OpenMP applications. Currently, we are working on transparently obtaining resilient MPI applications, that is, applications that are able to recover themselves from failures without stopping their execution.

Keywords Fault Tolerance, Checkpointing, Resilience, MPI, OpenMP

I. Motivation

Current petascale systems are formed by hundreds of thousands of cores. Schroeder and Gibson [16] have analysed failure data collected at two large high-performance computing sites, showing failure rates from 20 to more than 1,000 failures per year, depending mostly on system size. That can be translated in a failure every 8.7 hours. Future exascale systems will be formed by several millions of cores, and they will be hit by error/faults much more frequently than petascale systems due to their scale and complexity [5]. Therefore, long-running HPC applications in these systems will need to use fault tolerance techniques to ensure the successful execution completion.

The MPI (Message Passing Interface) standard is the most popular parallel programming model in petascale systems. Moreover, current HPC systems are clusters of multicore nodes that can benefit from the use of a hybrid programming model, in which MPI is used for the inter-node communications while a shared memory programming model, such as OpenMP, is used intra-node [20, 8]. However, these programming models lack fault tolerance support. In this scenario, checkpointing is a widely used fault tolerance technique, in which the computation state is saved periodically to disk into checkpoint files, allowing the recovery of the application when a failure occurs.

This PhD. thesis is focused on the study of efficient fault tolerance solutions for those parallel programming models that will likely be the most used in the exascale era. For this purpose, new strategies and protocols will be implemented in CPPC (ComPiler for Portable Checkpointing) [14], a portable and transparent checkpointing infrastructure for MPI parallel applications, to adequate it for the exascale era.

II. CPPC Overview

CPPC is an open-source checkpointing tool for MPI applications available at `http://cppc.des.udc.es` under GNU general public license (GPL). CPPC is made up of a compiler tool and a runtime library, and its main characteristics are:

- It constitutes a transparent solution for the final user, since at compile time the CPPC source-to-source compiler automatically transforms a paral-

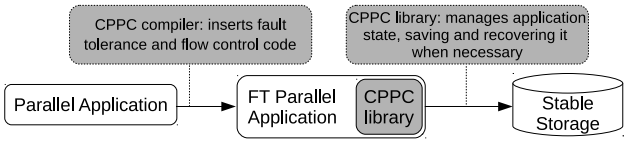


Figure 1: CPPC global flow

lel code into an equivalent fault-tolerant version instrumented with calls to the CPPC library, as exemplified in Figure 1.

- It applies a spatially coordinaded checkpointing. The CPPC compiler identifies safe points, that is, code locations in which it is guaranteed that no inconsistencies due to messages may occur. The usage of safe points guarantees data consistency and no inter-process communications or runtime synchronization are necessary when checkpointing. Thus, reducing the checkpointing protocol overhead.
- It uses an application-level checkpointing, including in the checkpoint files only those application variables indispensable for the successful recovery. The CPPC compiler automatically performs a liveness analysis to identify the relevant variables, minimizing the checkpoint file size and, thus, reducing the checkpointing overhead.
- It results in a portable solution, thanks to the use of portable storage formats and the exclusion of architecture-dependent state from checkpoint files, allowing the recovery on machines with different architectures and/or operating systems than those in which the checkpoint files were generated.

III. Thesis Work

In the literature, there exists some works focused on fault tolerance for shared memory systems, in which OpenMP is the de-facto standard for parallel programming on this systems. Some of these proposals are based on redundancy [7, 18], however, they can not tolerate multiple failures. On the other hand, the available checkpointing proposals for shared memory applications lack portability, whether code portability [13, 17] (allowing its use on different architectures) or checkpoint files portability [2, 4] (allowing to restart on different machines). In this context, we have extended CPPC to cope with OpenMP applications using a coordinated checkpointing protocol for data consistency [12], and applied different optimization techniques to minimize the overhead introduced during its operation [11]. Afterwards, we have extended that solution to cope with hybrid MPI-OpenMP applications using a hybrid protocol: coordinated checkpointing across OpenMP threads and uncoordinated across MPI processes (thanks to the use of safe points). We have evaluated the performance of this hybrid MPI-OpenMP solution on applications from the ASC Sequoia Benchmark Codes and the NERSC-8/Trinity benchmarks on over 6144 cores, obtaining overheads below 1.1% when checkpointing 50 GB of data. Additionally, the choice of an application-level approach and the portability of the checkpoint files allow building adaptable applications, that is, applications that are able to be restarted in a different resource architecture and/or number of cores, varying the number of OpenMP threads used by the application. This feature will be specially useful on heterogeneous clusters, allowing the adaptation of the application to the available resources.

Whether using the MPI or the hybrid MPI-OpenMP model, upon a single process/thread failure the entire application is aborted. This is the default behaviour because the state of MPI is undefined upon failure and, thus, there are no guarantees that the program can successfully continue its execution. Therefore, traditional fault tolerant solutions for these applications rely on stop&restart checkpointing: the application state is periodically saved into checkpoint files, so that, upon failure, a new job can be relaunched for restarting the application using the state files. However, a complete restart is unnecessary since, after a failure, most of the computation nodes used by a job will still be alive. Moreover, a complete restart introduces overheads both for re-queuing the job and for moving the checkpointed data across the cluster to the new granted resources. Thus, in the last years, new methods have emerged to provide fault tolerance support to MPI applications, such as failure avoidance approaches [6, 21] that preemptively migrate processes

from processors that are about to fail. Unfortunately, these solutions are not able to cope with already happened failures.

Recently, the Fault Tolerance Working Group within the MPI forum proposed the ULFM (User Level Failure Mitigation) interface [3] to integrate resilience capabilities in the future MPI 4.0. It includes new semantics for process failure detection, and communicator revocation and reconfiguration. Thus, it enables the implementation of resilient MPI and hybrid MPI-OpenMP applications, that is, applications that are able to recover themselves from failures. Nevertheless, incorporating the ULFM capabilities in already existing codes is not a simple task. Different approaches for resilience using the new ULFM functionalities have emerged. Some of these solutions are Algorithm-Based Fault Tolerance (ABFT) techniques, which means that they are specific to one or a set of applications and they can not be generally applied [9, 1]. Other proposals, such as [15, 19] present a more general scope, however they rely on the developers to instrument their MPI applications in order to obtain fault tolerance support, which is, in general, a complex and time-consuming task.

In this scenario, we have exploit the ULFM new functionalities using CPPC to transparently obtain resilient MPI applications from generic MPI SPMD (Single Program Multiple Data) programs [10]. By means of the CPPC instrumentation of the original application code, failures in one or several MPI processes are tolerated using a non-shrinking backwards recovery based on checkpointing. In this solution, after a failure, the failed processes are re-spawned and all the processes rolled back to the last checkpoint available, so that the application can continue its execution with the same number of MPI processes.

IV. Future Work

Our MPI resilience proposal combining CPPC and ULFM avois the overheads both for requeuing the job and for moving all the checkpointed data across the cluster. However, upon a failure, all the MPI processes roll back to a previous saved state to recover the application. In this situation, not only some computation done by the failed processes is lost, but also some computation performed by the survivor processes, as all of them roll back to the last checkpoint available and continue the execution from that point. Therefore, to adequate this proposal to the exascale era, we plan on designing and implementing a local recovery strategy, so that, only the failed processes have to roll back to a previous state, while the survivors can continue their computation. Apart from improving the scalability of the proposal, this strategy can reduced the energy consumption, as survivor processes do not repeat any part of their computation.

Acknowledgment

This research was supported by the Ministry of Economy and Competitiveness of Spain and FEDER funds of the EU (Project TIN2013-42148-P, and the predoctoral grant of Nuria Losada ref. BES-2014-068066) and by EU under the COST Program Action IC1305: Network for Sustainable Ultrascale Computing (NESUS).

References

[1] M. M. Ali, J. Southern, P. Strazdins, and B. Harding. Application Level Fault Recovery: Using Fault-Tolerant Open MPI in a PDE Solver. In *IEEE International Parallel Distributed Processing Symposium Workshops*, pages 1169–1178, 2014.

[2] J. Ansel, K. Arya, and G. Cooperman. DMTCP: Transparent Checkpointing for Cluster Computations and the Desktop. In *Proceedings of the 23rd IEEE International Parallel and Distributed Processing Symposium*. IEEE, 2009.

[3] W. Bland, A. Bouteiller, T. Herault, J. Hursey, G. Bosilca, and J. J. Dongarra. An evaluation of User-Level Failure Mitigation support in MPI. *Computing*, 95(12):1171–1184, 2013.

[4] G. Bronevetsky, K. Pingali, and P. Stodghill. Experimental evaluation of application-level checkpointing for OpenMP programs. In *Proceedings of the 20th Annual International Conference on Supercomputing*, pages 2–13, 2006.

[5] F. Cappello. Fault tolerance in petascale/exascale systems: Current knowledge, challenges and re-

search opportunities. *International Journal of High Performance Computing Applications*, 23(3):212–226, 2009.

[6] I. Cores, G. Rodríguez, P. González, and M. J. Martín. Failure avoidance in MPI applications using an application-level approach. *The Computer Journal*, 57(1):100–114, 2014.

[7] H. Fu and Y. Ding. Using Redundant Threads for Fault Tolerance of OpenMP Programs. In *Proceedings of the 2010 International Conference on Information Science and Applications*, pages 1–8, 2010.

[8] H. Jin, D. Jespersen, P. Mehrotra, R. Biswas, L. Huang, and B. Chapman. High Performance Computing using MPI and OpenMP on Multi-core Parallel Systems. *Parallel Computing*, 37(9):562 – 575, 2011.

[9] I. Laguna, D.F. Richards, T. Gamblin, M. Schulz, and B.R. de Supinski. Evaluating User-Level Fault Tolerance for MPI Applications. In *European MPI Users' Group Meeting*, pages 57–62, 2014.

[10] N. Losada, I. Cores, M. J. Martín, and P. González. Resilient MPI applications using an application-level checkpointing framework and ULFM. In *Journal of Supercomputing. [In Press]*, 2016.

[11] N. Losada, M. J. Martín, G. Rodríguez, and P. González. I/O Optimization in the Checkpointing of OpenMP Parallel Applications. In *Proceedings of the 23rd Euromicro International Conference on Parallel, Distributed and Network-Based Processing*, pages 222–229, 2015.

[12] N. Losada, M.J. Martín, G. Rodríguez, and P. González. Extending an Application-Level Checkpointing Tool to Provide Fault Tolerance Support to OpenMP Applications. *Journal of Universal Computer Science*, 20(9):1352–1372, 2014.

[13] M. Prvulovic, Z. Zhang, and J. Torrellas. ReVive: cost-effective architectural support for rollback recovery in shared-memory multiprocessors. In *Proceedings of the 29th Annual International Symposium of Computer Architecture*, pages 111–122, 2002.

[14] G. Rodríguez, M.J. Martín, P. González, J. Touriño, and R. Doallo. CPPC: a compiler-assisted tool for portable checkpointing of message-passing applications. *Concurrency and Computation: Practice and Experience*, 22(6):749–766, 2010.

[15] K. Sato, A. Moody, K. Mohror, T. Gamblin, B.R. De Supinski, N. Maruyama, and S. Matsuoka. FMI: Fault Tolerant Messaging Interface for Fast and Transparent Recovery. In *IEEE International Parallel and Distributed Processing Symposium*, pages 1225–1234, 2014.

[16] B. Schroeder and G. A. Gibson. A large-scale study of failures in high-performance computing systems. *IEEE Transactions on Dependable and Secure Computing*, 7(4):337–350, 2010.

[17] D.J. Sorin, M.M.K. Martin, M.D. Hill, and D.A. Wood. SafetyNet: improving the availability of shared memory multiprocessors with global checkpoint/recovery. In *Proceedings of the 29th Annual International Symposium on Computer Architecture*, pages 123–134, 2002.

[18] O. Tahan and M. Shawky. Using dynamic task level redundancy for OpenMP fault tolerance. In *Proceedings of the 25th International Conference on Architecture of Computing Systems*, pages 25–36, 2012.

[19] K. Teranishi and M.A. Heroux. Toward Local Failure Local Recovery Resilience Model Using MPI-ULFM. In *European MPI Users' Group Meeting*, pages 51–56, 2014.

[20] R. Thakur, P. Balaji, D. Buntinas, D. Goodell, W. Gropp, T. Hoefler, S. Kumar, E. Lusk, and J. L. Träff. MPI at Exascale. *Proceedings of Scientific Discovery through Advanced Computing*, 2, 2010.

[21] C. Wang, F. Mueller, C. Engelmann, and S. L. Scott. Proactive process-level live migration in HPC environments. In *Proceedings of the 2008 ACM/IEEE conference on Supercomputing*, page 43, 2008.

Beamforming filtering with real-time constraints on mobile embedded devices

Fran J Alventosa[1], Pedro Alonso[1], Gemma Piñero[2] and Antonio M Vidal[1]

[1]Dpto. de Sistemas Informáticos y Computación (DSIC)
[2]Instituto de Telecomunicaciones y Aplicaciones Multimedia (iTEAM)
Universitat Politècnica de València, Spain
{[1]fraalrue,[1]palonso,[1]avidal}@dsic.upv.es
[2]gpinyero@iteam.upv.es

Abstract

Nowadays Tables and Smart phones are equipped with low power processor. Some of them, like the NVIDIA Tegra SoC, also come with a GPU integrated so that both, the CPU and the GPU have access directly to the same RAM memory. In another vein, one the main limitations of microphone array algorithms for audio processing is the high computational cost required to reproduce real acoustics environments when real-time signal processing is absolutely required. One of these algorithms is the Beamforming Algorithm*, which is used to recover acoustic signals from their observations when they are corrupted by noise, reverberation and other interfering signals. In order to achieve real-time processing executing this algorithm we have employed high performance libraries such as OPENBLAS, LAPACK, CUBLAS, PLASMA and MAGMA, and a particular tune programming for these mobile devices.*

Keywords Heterogeneous Computing, Low Power Processors, ARMv7 and ARM Cortex-A15, Beamforming Filter

I. Motivation

The field of High-Performance Computing (HPC) has always been oriented to achieve good performance in terms of execution time. For this reason research in HPC has traditionally focused on applications of large computational cost on computers equipped with high-performance processors capable of performing large amounts of floating-point operations. Also on software tools and hardware resources addressed to large clusters of computers capable of working with large amounts of data. However, also in the field of high performance computing has always existed another type of needs represented by applications that, while not requiring the processing of a large amount of data (such as simulations), they do need immediacy in obtaining the result (real-time), as for example, a large set of applications of digital signal processing. It is also important to emphasize that we are experiencing a fundamental change in the conception of the Information and Communication Technologies ICT, moving from an oriented approach to the optimization of computational power and speed processes and applications to another approach more oriented to achieve maximum performance benefits at a low energy efficiency cost. This model change requires a new orientation in which efforts should be focused on the sustainability of the developments to ensure the optimum use of resources. The processor manufacturers are aware of this fact and design new devices that offer not only high computational performance but also a low consumption. For instance, the NVIDIA company delivers their graphics cards as devices of a high ratio Gflops per watt [1]. The ARM [2] is another example of processor that needs low energy to operate since it has been designed to be the core of mobile devices and, therefore, should be aware of the consumption to get the maximum availability.

II. Related work

There are many problems in engineering that can benefit from the good ratio of computational power by energy consumption offered by current processor architectures. The research group in which this doctoral thesis is integrated has a large experience in the design of high performance algorithms that address problems like 3D audio [3, 4], design of passive components based on microwave and electromagnetic devices applied to telecommunications [5, 6], systems analysis of detection of Multiple-Input Multiple-Output (MIMO systems) [7, 8, 9, 10], etc.

Typical paradigms of signal processing (detection, location, source tracking, feature extraction, etc.) have taken an extensive development in recent years in the form of distributed processed signals partly because of the increase of applications that have emerged around wireless sensor networks or, to be more specific, "Smart Sensors Networks" (*SSN*) obtained when the nodes of the network have processing and "decision making" capacities.

III. Thesis idea

The main target of this thesis is the design and implementation of algorithms for digital signal processing of sound signals in mobile devices. In an early step, we have tested the behaviour of high performance libraries of such HPC like BLAS [11], LAPACK [12], CUBLAS [13], PLASMA [14], and MAGMA [15], on an embedded system to evaluate their usability to solve our problem since many of the operations on which the algorithms are based can be cast in terms of linear algebra functions. We also have used parallel programming standards like OpenMP [16] and MPI [17].

The applications that can benefit from the work of this thesis are, e.g. applications of spatial sound (3D audio), filtering multichannel, echo cancellers of crosstalk, tracking and tracing of sources, classification and signal enhancement, etc. Among the applications, we will focus on processing distributed and collaborative signals around SSN's. Due to the high computational requirements to achive real-time processing we will try to get the best of the promissing NVIDIA solution SDK Jetson DevKit [18].

IV. The Beamforming Algorithm

In this section we make a brief introduction to the work being carried out in the framework of the thesis. This work consists in the effcient implementation of the Beamformer algorithm for the Jetson TK1.

Let $s_m(k)$, $m = 1, \ldots, M$, be signals emitted by M loudspeakers, the goal is to develop N filters g_n, $n = 1, \ldots, N$, where N is the number of microphones in the system, that allow to rebuild the original signals once cleaned from noise and room reverberation. To this end, we use channel responses of the room, represented as h_{nm}, for values of n and m stated before.

The output of the n-th microphone is given by:

$$x_n(k) = \sum_{m=1}^{M} \sum_{j=1}^{L_h} h_{nm}(j) s_m(k-j) + v_n(k) \ .$$

where L_h is the length of longest room impulse response of all the acoustic channels h_{nm}, and $v_n(k)$ is the noise signal. (For the sake of clarity, we will not consider the noise term hereafter.) Also for clarity and computation efficiency, we rewrite the form of the output signal of each microphone as

$$x_n(k) = \sum_{m=1}^{M} \mathbf{h}_{nm}^T \mathbf{s}_m(k) \ ,$$

where $\mathbf{s}_m(k)$ is the column vector defined as

$$\mathbf{s}_m(k) = \left[\ s_m(k) \quad s_m(k-1) \quad \cdots \quad s_m(k-L_h+1) \ \right]^T \ ,$$

and $\mathbf{h}_{nm}$ is the $\mathbb{R}^{L_h \times 1}$ acoustic channel vector from loudspeaker m to microphone n.

Considering now the problem of recovering source signals $s_m(k)$ from the recorded observations $x_n(k)$, beamforming filters g_n have to be designed so that the output signal $y(k)$ is a good estimate of $s_m(k)$, that is, $y(k) = \hat{s}_m(k-\tau)$ with minimum error. Given a maximum length of L_g taps for each of the N filters g_n, the broadband beamforming output signal is expressed in a similar form as

$$y(k) = \sum_{n=1}^{N} \mathbf{g}_n^T \mathbf{x}_n(k) \ ,$$

where $\mathbf{g}_n$ is the $\mathbb{R}^{L_g \times 1}$ vector containing the ordered taps of beamforming filters g_n, and $\mathbf{x}_n(k) = [x_n(k) x_(k-1) \cdots x_n(k-L_g+1)]^T$.

The algorithm of Beamformer filter called LCMV (Linearly Constrained Minimum Variance) [19] calculates beamforming filters as:

$$\mathbf{g}^{\mathrm{LCMV}} = \hat{\mathbf{R}}_x^{-1}\mathbf{H}_{:m}[\mathbf{H}_{:m}^T\hat{\mathbf{R}}_x^{-1}\mathbf{H}_{:m}]^{-1}\mathbf{u}_m\ , \qquad (1)$$

where $\mathbf{g}^{\mathrm{LCMV}}$ is formed by the concatenation of filters $\mathbf{g}_n$, i.e. $\mathbf{g}^{\mathrm{LCMV}} = [\mathbf{g}_1^T,\ldots,\mathbf{g}_N^T]^T$, and matrix $\mathbf{H}_{:m}^{(NL_g)\times(L_g+L_h-1)}$ is a partition of the channel impulse matrix that only includes the impulse responses from the m-th source to the N microphones used in *Sylvester* matrix form. Matrix $\hat{\mathbf{R}}_x$ is the correlation matrix of the recorded signals and $\mathbf{u}_m$ is the vector of zeros except for a one at the proper vector component in order to compensate the room impulse response delay.

The implementation of the LCMV proposed seeks for efficiency and accuracy, and its mainly based on the QR decomposition. Firstly, we form the following matrix $\mathbf{X} \in \mathbb{R}^{NL_g\times K}$,

$$\mathbf{X} = \frac{1}{\sqrt{K}}\begin{pmatrix} \mathbf{x}_1(k) & \mathbf{x}_1(k+1) & \ldots & \mathbf{x}_1(k+K-1) \\ \mathbf{x}_2(k) & \mathbf{x}_2(k+1) & \ldots & \mathbf{x}_2(k+K-1) \\ \vdots & \vdots & & \vdots \\ \mathbf{x}_N(k) & \mathbf{x}_N(k+1) & \ldots & \mathbf{x}_N(k+K-1) \end{pmatrix}, \qquad (2)$$

where $K\ (> NL_g)$ is the number of samples used. The algorithm computes the **qr** decomposition of $\mathbf{X}^T$, i.e. $\mathbf{X}^T = \mathbf{QR}$, where $\mathbf{Q}$ is orthogonal and $\mathbf{R}$ is upper triangular. Thus, in order to use LAPACK routines we build directly matrix X^T in column major order representation. Using matrix $\mathbf{X}$, matrix $\hat{\mathbf{R}}_x$ can be defined as

$$\hat{\mathbf{R}}_x = \mathbf{X}\mathbf{X}^T = \mathbf{R}^T\mathbf{Q}^T\mathbf{Q}\mathbf{R} = \mathbf{R}^T\mathbf{R}\,.$$

Now, we define for convenience matrix $\mathbf{W} = \hat{\mathbf{R}}_x^{-1}\mathbf{H}_{:m}$ so that the LCMV beamformer filter $\mathbf{g}^{\mathrm{LCMV}}$ (1) can be expressed as

$$\mathbf{g}^{\mathrm{LCMV}} = \mathbf{W}[\mathbf{H}_{:m}^T\mathbf{W}]^{-1}\mathbf{u}_m\ . \qquad (3)$$

We define matrix $\mathbf{Z}$ as the solution of the linear system

$$\mathbf{R}^T\mathbf{Z} = \mathbf{H}_{:m}\ ,$$

then, using the **qr** decomposition of matrix $\mathbf{X}$ we have

$$\mathbf{W} = \hat{\mathbf{R}}_x^{-1}\mathbf{H}_{:m} = (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{H}_{:m} = \mathbf{R}^{-1}\mathbf{R}^{-T}\mathbf{H}_{:m} = \mathbf{R}^{-1}\mathbf{Z}\ ,$$

where clearly matrix $\mathbf{W}$ is the solution of the linear system $\mathbf{RW} = \mathbf{Z}$.

The solution to get the beamforming filters proceeds by solving the linear system

$$\mathbf{A}\mathbf{b}_m = \mathbf{u}_m\ , \qquad (4)$$

where $\mathbf{A} = \mathbf{H}_{:m}^T\mathbf{W} = \mathbf{H}_{:m}^T\mathbf{R}^{-1}\mathbf{Z} = \mathbf{Z}^T\mathbf{Z}$. Also here, the solution of the linear system (4) is obtained through a **qr** factorization, in this case, of matrix $\mathbf{Z}$. Let $\mathbf{Z} = \mathbf{Q}'\mathbf{R}'$ be the **qr** decomposition of matrix $\mathbf{Z}$, then vector $\mathbf{b}_m$ can be computed by solving the following two triangular linear systems:

$$\begin{aligned} \mathbf{R}'^T\mathbf{y} &= \mathbf{u}_m\ , \\ \mathbf{R}'\mathbf{b}_m &= \mathbf{y}\ . \end{aligned}$$

Finally, it is easy to see that the computation of the beamformer filter (1) can be computed using the last obtained objects, i.e. $\mathbf{R}$, $\mathbf{Z}$, and $\mathbf{b}_m$, this way:

$$\mathbf{g}^{\mathrm{LCMV}} = \mathbf{R}^{-1}\mathbf{Z}\mathbf{b}_m\ ,$$

which involves a matrix vector product and a triangular linear system solution.

The results have been carried out on the NVIDIA Jetson TK1, which consists of an ARM cortex A-15 with four cores and an NVIDIA GPU Kepler with 192 cores integrated all together in a single chip. The cost of the QR decomposition of matrix X (2) is $\approx 70\%$ the total cost of the algorithm, thus we focused our efforts on optimizing this operation. For the reduction in time of the QR decomposition we wrote different implementations based on libraries BLAS and LAPACK. After some testing we selected the optimized BLAS implementation OPENBLAS for the architecture ARMV7 as the best. We also used CUBLAS, PLASMA and MAGMA libraries to involve the GPU in the computations and, thus, to reduce the execution time.

In a first assessment we realize that MAGMA library is not (yet) optimized for devices with the characteristics of the Jetson (CPU and GPU ensambled on a single chip), since the cost of the QR decomposition by MAGMA is higher than the cost of our own implementation of the QR decomposition. Our implementation uses the same scheme as function GEQRF of LAPACK, but some operations are deliverd to the ARM processor cores using OPENBLAS and other operations are driven to the GPU using the CUBLAS library.

V. Conclusion and future work

Probably, the main conclusion of our incipient work is that yet exists a large room for improvement, both in the hardware devices as in the implementations that can exploit these devices. One of the solutions in which we are working on now consists of the QR updating. With this idea, many operations involved in the original algorithm that computes the QR factorization from scratch at each iteration can be avoided, allowing thus to reduce significantly the execution time.

References

[1] NVIDIA JETSON TK1, http://blogs.nvidia.com/blog/2013/11/20/10-greenest-powered-by-nvidia-gpus/, (accessed 2016 January 13).

[2] ARM Processors, http://www.arm.com/products/processors/, (accessed 2016 January 13).

[3] J. A. Belloch, M. Ferrer, A. González, F. J. Martínez and A. M. Vidal, "Headphone-Based Virtual Spatialization of Sound with a GPU Accelerator" in *J. Audio Eng. Soc.*, vol. 61, no. 7/8, pp. 546-561, 2013.

[4] J. A. Belloch, A. González, F. J. Martínez and A. M. Vidal, "Multichannel Massive Audio Processing using GPU" in *Integrated Computer-Aided Engineering (ICAE)*, vol. 20, no. 2, pp. 169-182, 2013.

[5] A. M. Vidal, A. Vidal, V. E. Boria and V. M. García, "Parallel computation of arbitrarily shaped waveguide modes using BI-RME and Lanczos Methods" in *Communications in Numerical Methods in Engineering*, vol. 23, no. 4, pp. 273-284, 2007.

[6] V. M. García, A. Vidal, V. E. Boria and A. M. Vidal, "Efficient and accurate waveguide mode computation using BI-RME and Lanczos methods" in *International Journal for Numerical Methods in Engineering*, vol. 65, no. 11, pp. 1773-1788, 2006.

[7] C. Ramiro, A. M. Vidal, A. González and S. Roger, "MIMOPack: a high-performance computing library for MIMO communication systems" in *Journal of Supercomputing*, vol. 71, no. 2, pp. 751-760, 2014.

[8] C. Ramiro, M. A. Simarro, F. J. Martínez, A. M. Vidal and A. González, "A GPU implementation of an iterative receiver for energy saving MIMO ID-BICM systems" in *Journal of Supercomputing*, vol. 70, no. 2, pp. 541-551, 2014.

[9] V. M. García, A. M. Vidal, A. González and S. Roger, "Improved Maximum Likelihood detection through sphere decoding combined with box optimization" in *Signal Processing*, vol. 98, no. 1, pp. 284-294, 2014.

[10] S. Roger, C. Ramiro, A. González, V. Almenar and A. M. Vidal, "An Efficient GPU Implementation of Fixed-Complexity Sphere Decoders for MIMO Wireless Systems" in *Integrated Computer-Aided Engineering (ICAE)*, vol. 19, no. 4, pp. 341-350, 2012.

[11] BLAS Library, http://www.netlib.org/blas/, (accessed 2016 January 13).

[12] LAPACK Library, http://www.netlib.org/lapack/, (accessed 2016 January 13)..

[13] CUBLAS Library, http://docs.nvidia.com/cuda/cublas/, (accessed 2016 January 13)..

[14] PLASMA Library, http://icl.cs.utk.edu/plasma/, (accessed 2016 January 13).

[15] MAGMA Library, http://icl.cs.utk.edu/magma/, (accessed 2016 January 13).

[16] OpenMP, http://openmp.org/wp/, (accessed 2016 January 13).

[17] MPI, http://www.mpi-forum.org/, (accessed 2016 January 13).

[18] NVIDIA JETSON TK1, https://developer.nvidia.com/embedded/develop/hardware, (accessed 2016 January 13).

[19] Jorge Lorente, Gemma Piñero, Antonio M. Vidal, Jose Antonio Belloch, Alberto GonzÃąlez, "Parallel implementations of Beamforming desgign and filtering for microphone array applications," in *European Signal Processing Conference (EUSIPCO)*, Barcelona, Spain, August 2011, pp. 501-505.

Data mining for autonomous wearable sensors used for elderly healthcare monitoring

Aileni Raluca Maria[1, 2], Strungaru Rodica[1], Valderrama Carlos[2]

[1]Politehnica University of Bucharest, Faculty of Electronics, Telecommunication and Information Technology

[2]Mons University, Faculty of Engineering, Department Electronics and Microelectronics

Abstract

The paper presents some aspects regarding data mining used modeling and prediction of the patients' health state parameters.

The proposed wearable device integrated by using wireless personal networks (WPNs) can sense, process and communicate vital signs through internet for healthcare monitoring. These WPNs are fitted for medical applications and offer continuous ambulatory health monitoring by using non-invasive methods. Generally, the body sensor network (BSN) for medical applications are based on big data fusion and cloud computing technologies (PaaS, SaaS - for data storage and sharing solutions).

The big data fusion includes preprocessing (filter the noise), feature extraction (data abstraction), data fusion computation (modeling different information type and fusion), and data compression (reducing the information stored in memory and transmitted by the transceiver).

The fusion between wearable wireless body sensor network (WWBSN), IoT and Cloud Computing will allow doctors, emergency stations or caregivers to track and receive data from BSNs about patients in different places. By using biomedical sensors can be studied the human behavior and physiology, the body's response physiologically and emotionally to various physical and mental diseases. The WWBSN can cover monitoring for cardiovascular, diabetic problems or mental disorders (Alzheimer).

***Keywords**: data mining, elderly healthcare, sensors*

Motivation

The motivation source for doctoral thesis study was the case of elderly patients monitoring (fig. 1). The elderly patients are dealing with comorbidity phenomena characterized by association of diseases like cardiovascular problems (hypertension, hypotension), cardiovascular problems (hypertension, hypotension), nonphysical activities (obesity) and Alzheimer. Comorbidity is associated with worse health outcomes, complex clinical management and increased health care costs.

The monitoring of the elderly patients in their living environment by using wireless sensors network (optical sensors, gyroscopes and accelerometers) presents a high interest for scientists in order for failure detection [1].

For diabetic elderly and for person with cardiovascular diseases the posture of the body and rapidity on changing the body posture coordinates can indicate critical situations like failure, tremors or heart attack.

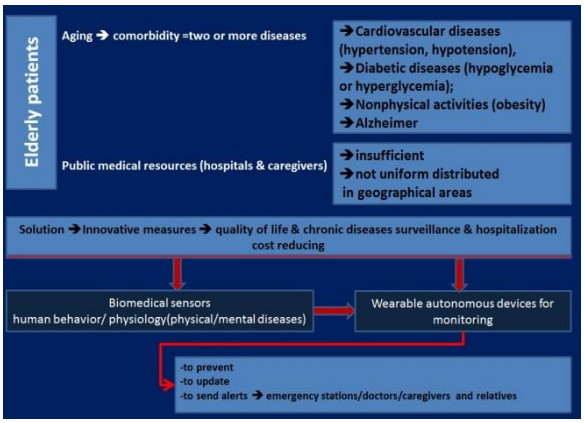


Fig. 1 Wearable monitoring system-motivation

Thesis idea

The doctoral thesis "Theoretical and experimental contributions to the monitoring of vital parameters using intelligent control systems based on sensors integrated into textile structures and Cloud Computing services" idea is to track vital parameters data from wearable sensors integrated in textile structures.
The purpose of this thesis is to create a wearable monitoring system for elderly patients.
The textile technology allow the weaving, sewing and knitting of conductive yarns into the flexible structures, but in case of integration of the electronic components (sensors, actuators and computational devices) on the textile surface (e-textile), may occurs constraints related to system design which require high computational performance, low power consumption and fault tolerance.
The nature of the textile (discrete model) and the faults which occur due to the open and short circuits can disconnect/drain the battery and can affect both battery life and the performance of the textile with conductive yarns, which finally affect the accuracy signals from the textile structure made with conductive yarns [2].
Usage of the semiconductors in textiles structures for the connections sensors/actuators – motherboard affect signals data accuracy because of the yarns resistivity modifications with temperature and skin humidity variations, body thermal flow and due to the textile property to be good thermal conductor [2].
Big data in medical, physical sciences and financial area generate a huge volume of data collected, which required new technologies and complex algorithms and software for collecting, storage and managing the big data.
For big data from biomedical sensors analysis, data mining methods allow predictive modeling of data in order to obtain the disease risk assessment and disease model in correlation with patient behavior.

Conclusion

By defining fault like a physical defect or imperfection that occurs in some hardware (sensors, actuators) or software component (a short circuit between two adjacent interconnects, a broken pin, or a software bug) and knowing the cause-effect model for fault-error-failure (faults cause errors and errors causes for failures effects) can conclude that usage of conductive textile yarns for data transmission can cause system monitoring failure and false data.
Wearable sensors system for health monitoring should allow [2]:
-fault tolerance control implementation;
-big data fusion for extract the values and establish optimal decisions based on predictive modeling;
-sensor data processing algorithm for reducing the noise and data discretization;
Wearable electronics integrated in textile structure experience a data losses and low accuracy signals due to the textile structure properties. In design of textile structures with electronics integrated must consider the noise that could occur due to the conductive yarns length and resistivity in correlation with temperature and skin humidity.
In case of diabetic patient study case the critical values for biomedical signal (pulse, temperature, humidity and breath rhythm) are sent to fault tolerance control unit and after comparison is selected the optimal decision and are sent the message alerts.

In case of diabetic elderly patient for establish the critical situation we analyze the correlation between breath rhythm, humidity, pulse and temperature values obtained from wearable sensors:

Hypoglycemia=f (temperature, pulse, breath rhythm, pulse)

Hyperglycemia=f (temperature, pulse, breath rhythm, pulse)

In many cases the sensors output may generate the errors which can be considered like fault events [2]:

-partial or total output loss;

-abrupt/continuous switching between modes of functioning;

- Nonlinear aberrations;

Future work

For developing the monitoring system will be required to analyze, collect and storage the big data.

For analyzing the parameters from patients will be developed a support decision system (fig. 2). The system architecture will consist in 5 levels:

➔Level 1 - data transmission (biomedical sensors aggregators);

➔Level 2 - big data (data collecting, discretization and storage);

➔Level 3 - medical information (data mining)

➔Level 4 - diseases knowledge (data synthesis)

➔Level 5 - decision support system

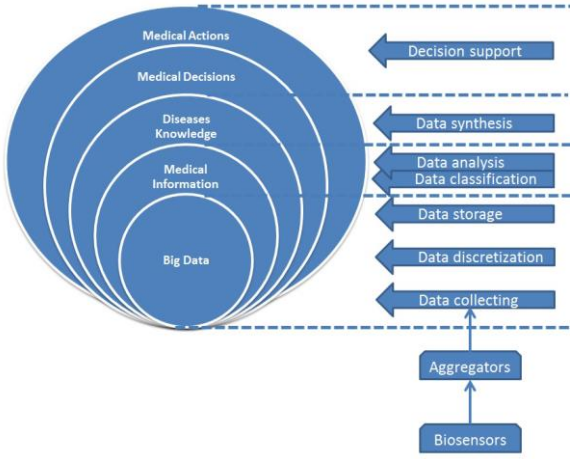


Fig. 2 Decision system architecture- big data monitoring [3]

The software will be available in two versions – for smartphone (fig. 3) and pc and will offer:

➔Usability
➔Autonomy
➔Portability

Fig. 3 Patient data management software-mobile app

Acknowledgment

This paper was presented at NESUS Winter School & PhD Symposium (8-11 February 2016) with financial support from NESUS COST Action.

References

1. P. Augustyniak, M. Smolen, Z. Nikrut, E. Kantoch, "Seamless Tracing of Human Behavior Using Complementary Wearable and House-Embedded Sensors", *Sensors Journal* 2014, 14(5):7831-7856
2. R.M. Aileni, S. Pasca, C. Valderrama, "Biomedical sensors data fusion algorithm for enhancing the efficiency of fault-tolerant systems in case of wearable electronics device", *ROLCG*, 2015, IEEE.
3. R.M. Aileni, S. Pasca, C. Valderrama,"Cloud computing for big data from biomedical sensors monitoring, storage and analyze", *ROLCG*, 2015, IEEE.
4. R.M. Aileni, Wearable Wireless Body Sensor Network (WWBSN) for Health Monitoring, *Grascomp*, UNamur, Belgium, 2015

Processor Model for the Instruction Mapping Tool

Roman Mego

Brno University of Technology, Czech Republic
roman.mego@phd.feec.vutbr.cz

Abstract

This paper describes the model designed for the instruction mapping tool, which can be used for generating the low level assembly code for the digital signal processing algorithms. The model is based on the Very Long Instruction Word architecture. The Texas Instrument TMS320C6678 was the pattern and finally was described with the created model. The paper is showing the parameters of the hardware resources and also the instruction set.

Keywords Processor model, Instruction mapping, VLIW

I. Introduction

Several years ago, in applications for digital signal processing applications, the critical code was not written using high level languages, but it was hand optimized in the assembly language. This approach was chosen because of the non-effective results generated by the compilers. This procedure resulted in the long development time and high cost. The other complication is that the final code cannot be used on the different processor architecture. In the case of the migration on the different processor, the code must be rewritten into the different form.

Nowadays, the modern compilers are capable of generating effective code. This statement applies mainly for the scalar processor architectures. It is given by the wide use of the scalar processors in different sectors, from the industrial and medical equipment, to the customer electronics, which leaded to the development of the effective compilers. There are also frameworks, where the architecture can be defined for various architectures such as [1] or [2].

But there are also different architectures, not widely used, where the use of high level languages leads to the ineffective code. These processors are usually the ones that use instruction level parallelism, such as superscalar or Very Long Instruction Word (VLIW). To avoid the problems related with the software creating using assembly language, the new tool for DSP algorithm mapping under development [3].

This paper is dealing with the processor model used in the tool. The next chapters will show the model structure based on the VLIW architecture.

II. Model Description

To cover the majority of possible cases of the processor internal structure, the more complex processor was chosen as the reference. It was the TMS320C6678 [4] which is 8-core digital signal processor based on the C66x CorePac [5] made by Texas Instruments.

Single C66x DSP core contains 8 functional units and 64 general purpose registers. Its simplified structure is shown in figure 1. At first sight, it may seem that the core has quite large amount of the resources for parallel operation, but it has its limitation.

The first is that the functional units are not equal. They are not capable to execute the same instructions. Functional units are marked .L1, .L2, .S1, .S2, .D1, .D2 and .M1, .M2. The .D units are primary used for the loading and storing data into the memory. The .L and .S units are designed for the general arithmetic, logic and branch operations as well. The last, .M units, are able to perform multiply operations with single and double precision floating point values. All of the units

are also able to execute other types of instructions, but not with all data types.

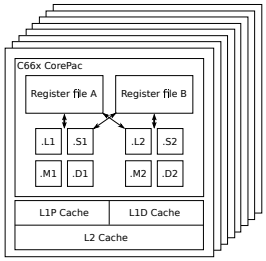


Figure 1: Simplified structure of the TMS320C6678.

The second limitation is caused by the division of the previously mentioned hardware resources into 2 identical data paths. These data paths are marked as Data Path A and Data Path B. Because of this it is not possible to directly access registers from Data Path A with functional unit from Data Path B. It can be done only through the Register File Cross Paths marked 1x and 2x. The single cross path in the C66x is capable to transfer 64-bit operand in the instruction. In addition, this operand can be used in multiple instructions in the same execute packed, which was not allowed in the older C64x core.

The model itself is aimed only on the description of the processor core, not the processor as the entire unit. The main parts of the model are:

- hardware resources of the core;
- instruction set.

II.1 Hardware Resources

The topology of the model is based on the VLIW architecture with the multiple data path.

II.1.1 Data Paths

From the outside view, the data path is the top level element, which contains all basic hardware resources. For this reason, the part of the model with the hardware resources is set of structures describing the data path.

The selected TMS320C6678 has 2 practically identical data paths, so the model in this case can contain only the template of one data path and information about the number of the data paths in the given architecture. But in general, the processor may consist of several different data paths, so every element in the model has its own definition.

Each data path contains the physical and virtual (or logical) resources, what will be explained later in the paper.

II.1.2 Cross Paths

As it was mentioned in the TMS320C6678 description, the data paths work as the separated units. The data cannot be directly moved between the register files and the functional units cannot read the register value. For this purpose, the model is able to define cross paths.

Each cross path is defined by the following parameters:

- source data path with register file;
- maximum width of the transferred data;
- maximum number of operands where the value can be used.

The meaning of the source data path is clean. The target data path is not defined at this point, because the functional units in the TMS32C6678 are not handling the operands in the same way. The .D, .M and .S units can read only the second operand through the cross path and the .L units can access to the different register file for both operands (figure 2). For this reason, the destination of the cross paths is defined individually on the functional units.

The maximum width of transferred data is given by the bus width, which is 64-bit in the selected processor despite the fact, that the register size is 32-bit. There is no need to define this parameter to different value than the multiply of register width, so the model keeps only the number of possible transferred registers.

The requirement of parameter which can tell if it is possible to use the operand transferred by the cross path in the multiple operations is given by the difference between the C66x and C64x cores. In the C64x, it is possible to use the data from the cross path only

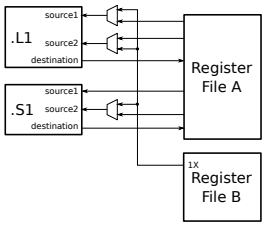


Figure 2: Example of the cross path connection to the functional units [5].

in the one functional unit at once in compare with the C66x where this limitation does not exists.

II.1.3 Functional Units

Each data path includes the set of functional units. The only one parameter, except the name, of the functional unit is the identification of the operand input connection to the cross path. The referenced C66x and also the older C64x are composed of the 2 data paths, so in this case the parameter could be only with the meaning connected or disconnected. But in general, the processor could have more than 2 data paths and therefore it is needed to identify which cross path is connected into the functional unit input.

II.1.4 Registers

The last physical hardware resources in the presented model are the general purpose register files. Each data path has one register file defined by the set of the registers. The registers are identified only by their names. Even the width of the registers is not mentioned in the model. To determine how many and which registers to represent data type, virtual resources are used. They will be described in the next chapter parts.

II.1.5 Register Groups and Data Types

Register groups are only logical definitions for the tool, to determine which registers can be used together as the single value (figure 3). As it was mentioned, the model is not working with the physical width with the registers. Also the registers can handle different number of bits on different architectures, so the decision which group to use as given data type cannot be made. For this reason, the data types supported by the tool are assigned to the created register groups.

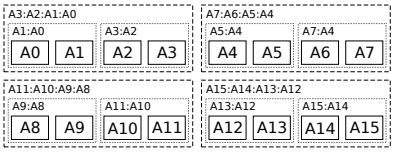


Figure 3: Creating register groups from the physical registers.

III. Instruction Set

The instruction set is next big part in the model describing the processor. It is not divided into other segments as the hardware resources. It is only the list of the instructions that can fit into the operation abstraction of the tool. It includes the arithmetical and logical operations and the data loading and storing instructions.

Each instruction is represented by the following attributes:

- name of the instruction;
- instruction format;
- instruction operation;
- data type of the operands;
- functional units capable to execute instruction;
- number of cycles needed to read the instruction and operands;
- number of cycles needed to write result to registers;
- total number of cycles needed to execute the instruction.

The meaning of the instruction name is clear. Its purpose is only the identification by the user.

The instruction format gives the position of the parameters in the final notation of the generated code.

Some of the instructions are able to process data with different number representation. For example the ABS instruction in the C66x is able to process 32-bit integers and 64-bit integers as well. That is why this parameter is list of the data types.

Functional units are another list acting as the instruction parameter. This list contains the functional units from all data paths. They are not divided into smaller groups.

The last group of parameters defines the timing of the instruction. The full instruction cycle was reduced into 3 stages. During the read stage, the functional unit is fetching instruction and the input value must be prepared in the registers. After this stage, the functional unit can be used for other purpose and the input register can be overwritten. The write stage moves the result of the operation into the destination registers. At this stage, the register must be prepared to receive new data to prevent overwrite the valid values for other operations. The instruction is executed between these stages and the resources can be freely used without limitations. Figure 4 shows the timing of the MPYDP instruction as the example.

Pipeline stage	1	2	3	4	5	6	7	8	9	10
Read	src1_l src2_l	src1_l src2_h	src1_h src2_l	src1_h src2_h						
Write									dst_l	dst_h
Unit in use	.M	.M	.M	.M						

Figure 4: MPYDP instruction pipelining.

IV. Implementation

The processor model is implemented as part of the instruction mapping tool. This tool is written in the C++ language and the model is not specified directly. It is in form of classes and the tool is reading user specified JSON file [6], which contains the structure of the specific architecture.

The simple command line tool to editing the architecture was also created. This editor is helpful during the defining the new architecture, because it keeps the valid format of the files, which could be corrupted by the mistype and also watches over the right connection between the parameters.

V. Conclusion

This paper presented the processor model designed for the instruction mapping tool, which was primary intended for VLIW architectures. The model was implemented as the part of the instruction mapping tool. Its functionality was verified with the mentioned tool on the TMS320C6678 processor. The model is primary aimed on the VLIW architectures, but it should be able to define other architectures such as the scalar or superscalar processors. This was not verified and it will be the part of the future work.

Acknowledgment

Publication of this paper was supported by the COST action IC1305, Network for Sustainable Ultrascale Computing (NESUS).

References

[1] I. Povazan et al., "A Retargetable C Compiler for Embedded Systems," in *Engineering of Computer Based Systems (ECBS-EERC) 2013 3rd Eastern European Regional Conference*, August 2013.

[2] S. Rajagopalan et al., A retargetable VLIW compiler framework for DSPs with instruction-level parallelism, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, issue 11.

[3] R. Mego and T. Fryza, "Tool for algorithms mapping with help of signal-flow graph approach", in *Radioelektronika 2014 24th International Conference*, April 2014.

[4] Texas Instruments, *Multicore fixed and floating-point digital signal processor* [online], Available: http://www.ti.com/lit/ds/symlink/tms320c6678.pdf.

[5] Texas Instruments, *TMS320C66x CorePac user guide* [online], Available: http://www.ti.com/lit/ug/sprugw0c/sprugw0c.pdf.

[6] ECMA International, *ECMA-404 The JSON Data Interchange Format*, 1st Edition, Available: http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf.

Distributed Processing in Cloud Computing

Ilias Mavridis
Aristotle University of Thessaloniki, Greece
imavridis@csd.auth.gr

Eleni Karatza
Supervisor
Aristotle University of Thessaloniki, Greece
karatza@csd.auth.gr

Abstract

Cloud computing offers a wide range of resources and services through the Internet that can been used for various purposes. The rapid growth of cloud computing has exempted many companies and institutions from the burden of maintaining expensive hardware and software infrastructure. With characteristics like high scalability, availability and fault tolerance, cloud computing meet the new era needs for massive data processing at an affordable cost. In our doctoral research we intend to study, analyze, evaluate and make proposals in order to further improve the performance of cloud computing.

Keywords Cloud computing

I. Introduction

Cloud computing has evolved into a major computing platform that is used by many companies. By using cloud computing, companies offer their services or process their data without the need of in-house IT infrastructure [1]. The term of cloud computing usually refers to providing computational services as utilities via the Internet [2]. These services may include infrastructure, platform and software. The increasing use of cloud computing can be explained by the fact that cloud offers "on-demand" scalability, high availability, flexible cost policy, ease of customization and other elements that positions it ahead of classic distributed technologies such as the Grid [1].

The aim of this thesis is to address open issues and limitations in cloud computing and propose techniques in order to overcome the potential obstacles and improve the performance of cloud computing. Through the doctoral research we will study the current bibliography and we will conduct several experiments to analyze and evaluate the current cloud computing technologies.

II. Ongoing Study

At the first phase of our research we investigated the use of main memory in cloud computing and we studied how it affects the computation performance. We analyzed and compared the widespread cloud computing framework Hadoop[3] with the relatively new general engine for large-scale data processing Spark[4]. Spark (unlike Hadoop's MapReduce) uses effectively the main memory and claims that can achieve up to one hundred times higher performance for certain applications compared to Hadoop's MapReduce [4].

In order to experimental evaluate the two frameworks we developed and executed log file analysis application in both frameworks. Log file analysis in cloud was proposed and investigated by many papers [5] - [15] for various reasons. Also many big companies like Facebook, Amazon, ebay, etc. use cloud computing solutions to analyze the enormous amount of log data that they produce. However to the best of our knowledge this is the first work that investigates and compares the performance of real log analysis applications in Hadoop and Spark.

In bibliography there are many papers that investigate the performance of cloud computing from different perspectives and explore how various factors affect

it [16] - [25]. To evaluate the performance of the two frameworks we focus on three performance indicators. The execution time, resource utilization and scalability. The experimental results showed that Spark presents almost the same scalability as Hadoop but Spark is significantly faster and makes better resource utilization than Hadoop.

The output of this study is published in the proceedings of the Second International Workshop on Sustainable Ultrascale Computing Systems (NESUS 2015) in Krakow, Poland; paper entitled "Log File Analysis in Cloud with Apache Hadoop and Apache Spark" [26] and an extended version of this work is submitted to an international journal.

III. Related Work

As we mentioned before in order to evaluate the performance of Hadoop and Spark we developed log file analysis applications in both frameworks. After an extensive search in bibliography we found that cloud computing for log analysis has been investigated and proposed by many papers, however the majority of them studied and proposed Hadoop-based algorithms and systems.

In papers [5] - [9] the authors recognized that logs are produced in higher rate than traditional systems can serve. To overcome the bottleneck of massive data processing of traditional relational databases they proposed and implemented log file analysis using Hadoop cluster.

The paper [10] presents a Hadoop-based log analysis system for intrusion detection and in [11] a MapReduce log analysis algorithm was used to identify security threats and problems. In both works they used Hadoop MapReduce in order to improve the response time of large log files analysis applications and as a result to achieve a faster reaction by the system's administrator.

In [12] the authors implemented a MapReduce-based framework for anomaly detection that follows a specific methodology to analyze log files. First, it collects logs from each node of the monitored cluster to the analysis cluster. Then, it applies K-means clustering algorithm to integrate the collected logs. Finally executes a MapReduce-based algorithm to parse these clustered log files.

A Hadoop-based flow logs analyzing system was proposed in paper [13]. This system uses for log analysis a new script language called Log-QL, which is a SQL-like language that was translated and submitted to the MapReduce framework. After experiments the authors concluded that their distributed system is faster and can handle much bigger datasets compared to a centralized system.

Paper [14] presents a scalable platform named Analysis Farm, for network log analysis with fast aggregation and agile query. To achieve storage scale-out, computation scale-out and agile query, OpenStack was used for resource provisioning, and MongoDB for log storage and analysis.

A cloud platform for log data analysis with the combination of Hadoop and Spark was presented in paper [15]. The authors proposed a cloud platform with batch processing and in-memory computing capabilities by using at the same time Hadoop, Spark and Hive/Shark. They claim that the proposed platform managed to analyze logs with higher stability, availability and efficiency than standalone Hadoop-based log analysis tools.

IV. Thesis idea

Cloud computing has been a focused area of research in the last years and there is still a great research interest in cloud computing. In our research we will study the state of the art cloud technologies and we will deal with open issues. As we continue our research we will study current trends in cloud computing and we will identify and try to propose solutions to problems.

V. Conclusion and future work

At the beginning of our research we dealt with the effective use of main memory in cloud computing and we studied how it can significantly improve its performance. We will continue our research in different areas of cloud computing with the goal of further improve the cloud performance.

Acknowledgment

We would like to acknowledge the contribution of the academic cloud service okeanos [27] for giving us the ability to create the necessary virtual machines for the above case study. We would also like to acknowledge the contribution of the COST Action IC1305 NESUS (Network for Sustainable Ultrascale Computing).

References

[1] I.A. Moschakis and H.D. Karatza, "A meta-heuristic optimization approach to the scheduling of Bag-of-Tasks applications on heterogeneous Clouds with multi-level arrivals and critical jobs," *Simulation Modelling Practice and Theory*, Elsevier, vol. 57, pp. 1-25, 2015.

[2] G.L. Stavrinides and H.D. Karatza, "A cost-effective and QoS-aware approach to scheduling real-time workflow applications in PaaS and SaaS clouds," in *3rd International Conference on Future Internet of Things and Cloud (FiCloud'15)*, Rome, Italy, August 2015, pp. 231-239.

[3] `http://hadoop.apache.org/`

[4] `http://spark.apache.org/`

[5] B. Kotiyal, A. Kumar, B. Pant and R. Goudar, "Big Data: Mining of Log File through Hadoop," in *IEEE International Conference on Human Computer Interactions (ICHCI'13)*, Chennai, India, August 2013, pp. 1-7.

[6] C. Wang, C. Tsai, C. Fan and Sh. Yuan, "A Hadoop based Weblog Analysis System," in *7th International Conference on Ubi-Media Computing and Workshops (U-MEDIA 2014)*, Ulaanbaatar, Mongolia, July 2014, pp. 72-77.

[7] S. Narkhede and T. Baraskar, "HMR log analyzer: Analyze web application logs over Hadoop MapReduce," *International Journal of UbiComp (IJU)*, vol.4, no.3, pp. 41-51, 2013.

[8] H. Yu and D.i Wang, "Mass Log Data Processing and Mining Based on Hadoop and Cloud Computing," in *7th International Conference on Computer Science and Education (ICCSE 2012)*,Melbourne, Australia, July 2012, pp. 197.

[9] H. Kathleen and R. Abdelmounaam, "SAFAL: A MapReduce Spatio-temporal Analyzer for UNAVCO FTP Logs," in *IEEE 16th International Conference on Computational Science and Engineering (CSE)*, Sydney, Australia, December 2013, pp. 1083-1090.

[10] M. Kumar and Dr. M. Hanumanthappa, "Scalable Intrusion Detection Systems Log Analysis using Cloud Computing Infrastructure," in *2013 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Tamilnadu, India, December 2013, pp.1-4.

[11] S. Vernekar and A. Buchade, "MapReduce based Log File Analysis for System Threats and Problem Identification," in *Advance Computing Conference (IACC), 2013 IEEE 3rd Internationa*, Patiala, India, February 2013, pp. 831-835.

[12] Y. Liu, W. Pan, N. Cao and G. Qiao, "System Anomaly Detection in Distributed Systems through MapReduce-Based Log Analysis," in *3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, Chengdu, China, August 2010, pp. V6-410 - V6-413 .

[13] J. Yang, Y. Zhang, S. Zhang and Dazhong He, "Mass flow logs analysis system based on Hadoop," in *5th IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT)*, Guilin, China, November 2013, pp. 115-118.

[14] J. Wei, Y. Zhao, K. Jiang, R. Xie and Y. Jin, "Analysis farm: A cloud-based scalable aggregation and query platform for network log analysis," in *International Conference on Cloud and Service Computing (CSC)*, Hong Kong, China, December 2011, pp. 354-359.

[15] X. LIN, P. WANG and B. WU, "Log analysis in cloud computing environment with Hadoop and Spark," in *5th IEEE International Conference on Broadband Network and Multimedia Technology (IC-BNMT 2013)*, Guilin, China, November 2013, pp. 273-276.

[16] J.Conejero, B. Caminero and C. Carron, "Analysing Hadoop Performance in a Multi-user IaaS Cloud," in *High Performance Computing and Simulation (HPCS)*, Bologna, Italy, 21-25 July 2014, pp. 399 - 406.

[17] G. Velkoski, M. Simjanoska, S. Ristov and M. Gusev," CPU Utilization in a Multitenant Cloud," in *IEEE EUROCON 2013*, Zagreb, Croatia, 1-4 July 2013, pp. 242-249.

[18] L. Gu and H. Li, "Memory or Time: Performance Evaluation for Iterative Operation on Hadoop and Spark," in *IEEE 10th International Conference on High Performance Computing and Communications and 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC EUC)*, Zhangjiajie, China, 13-15 Nov. 2013, pp. 721-727.

[19] P.R. Magalhaes Vasconcelos and G. Azevedo de Araujo Freitas, "Performance analysis of Hadoop MapReduce on an OpenNebula cloud with KVM and OpenVZ virtualizations," in *9th International Conference for Internet Technology and Secured Transactions (ICITST)*, London, 8-10 Dec. 2014, pp. 471-476.

[20] Eug. Feller, Lav. Ramakrishnan and Chr. Morin, "Performance and energy efficiency of big data applications in cloud environments: A Hadoop case study," *Journal of Parallel and Distributed Computing Special Issue on Scalable Systems for Big Data Management and Analytics*, vol. 79âĂŞ80, pp. 80âĂŞ89, May 2015.

[21] B.G. Batista, J.C. Estrella, M.J. Santana, R.H.C. Santana and S. Reiff-Marganiec, "Performance Evaluation in a Cloud with the Provisioning of Different Resources Configurations," in *2014 IEEE World Congress on Services (SERVICES)*, Anchorage, Alaska, June 27-July 2 2014, pp. 309-316.

[22] B. El Zant and M. Gagnaire, "Performance evaluation of Cloud Service Providers," in *2015 International Conference on Information and Communication Technology Research (ICTRC2015)*, Paris, France, May 17-19 2015, pp. 302-305.

[23] J. Gao, P. Pattabhiraman, B. Xiaoying and W.T. Tsai, "SaaS Performance and Scalability Evaluation in Clouds," in *2011 IEEE 6th International Symposium on Service Oriented System Engineering (SOSE)*, Irvine, USA, 12-14 Dec. 2011, pp. 61-71.

[24] T. Jiang, Q. Zhang, R. Hou, L. Chai, S.A. Mckee, Z. Jia and N. Sun, "Understanding the behavior of in-memory computing workloads," in *2014 IEEE International Symposium on Workload Characterization (IISWC)*, Raleigh, USA, 26-28 Oct. 2014, pp. 22-30.

[25] T.C. Chieu, A. Mohindra and A.A. Karve, "Scalability and Performance of Web Applications in a Compute Cloud," in *2011 IEEE 8th International Conference on e-Business Engineering (ICEBE)*, Beijing, China, 19-21 Oct. 2011, pp. 317-323.

[26] I. Mavridis and E. Karatza, "Log File Analysis in Cloud with Apache Hadoop and Apache Spark," in *Second International Workshop on Sustainable Ultrascale Computing Systems (NESUS 2015)*, Krakow, Poland, 10-11 Sept. 2015, pp. 51-62.

[27] `https://okeanos.grnet.gr`

The Analysis of Diachronic Variation in Romanian Print Press

Daniela Gîfu

Alexandru Ioan Cuza University, Faculty of Computer Science, 16, General Berthelot St., 700483, Iaşi
daniela.gifu@info.uaic.ro

Abstract

The paper describes a study based on diachronic exploration of Romanian texts in order to implement a technology for detecting automatically the morpho-lexical from 1840 to nowadays. The chosen timings put in evidence the language changes, describing, also, the phenomena related to the evolution of the Romanian language, especially, in print press. We define a complex methodology for recovering of old Romanian texts in two different spaces: Romania (until 1918, representing 3 countries, Moldova, Wallachia and Transylvania) and Republic of Moldavia, the last being a territory lost of Romania after the historic events. The aim of this survey it to analyse the morphology and lexical-semantics of Romanian language, based on important corpus starting with the middle of the 19th century until today, in order to compare them, emphasizing the language differences and similarities. This work could be of interest to lexicographers and computational linguistics specialists, who want to clarify the linguistic identity.

Keywords diachronic study, lexicon, morphosyntax, print press, WEKA.

I. Motivation

This research is anchored in diachrony (over the centuries, Romanian language has crystallized some structures which continue to be preserved as we show later) at the expense of synchrony, since today, despite language innovations (Coşeriu, 1997) appeared, things seems to be more stable (Ciompec, 1985). It is about how can we investigate the linguistic deviations that affect the multilingual Republic of Moldova in parallel with the Romanian language, using natural language processing (NLP) methodology for tracking diachronic changes from the middle of the 19th century?

II. Related work

Up to the 16th century almost all scientific writing in Europe was conducted in Latin. The construction and annotation of historical corpora is challenging in many ways (Lüdeling et al. 2005; Chiarcos et al., 2008; Claridge, 2008; Rissanen, 2008; Kytö, 2011; Kytö and Pahta, 2012, among many others).

In general, the creation of a parallel corpus of diachronic language is constituted by biblical texts, because the Bible is one of the earliest sizable coherent texts documented for many languages (especially European). The reason is obvious, the digital text is freely available in an unparalleled variety of languages and it has been repeatedly updated in different periods of time (Resnik et al., 1999) becoming very useful for comparative and diachronic studies. For instance, for older Germanic languages (Sukhareva and Chiarcos, 2014).

The diachronically and synchronically comparative studies of the Romance languages expose the presence of many similarities, especially in diachronic studies (Densuianu, 1902). Latin was the starting point, but issues about substratum, superstratum and adstratum which contributed to differentiate language were not set aside.

Contributions assigned to this section are closely related to the previous ones, as many of the ideas in Romance linguistics are also found in diachronic or diatopic study of the Romanian language. Linguists are known to call for language facts from the Romanesque

in order to explain some form and vice versa. We should mention contributions of Al. Rosetti (Rosetti, 1968; Rosetti et al., 1971), Iorgu Iordan (Iordan, 1975), Al. Graur (Graur, 1968), Valeria Guţu-Romalo (Guţu Romalo, 1972; 2005), Florica Dimitrescu (Dimitrescu, 1978, 1982), Marius Sala (Sala, 1998), Victor Iancu (Iancu, 2000), Narcisa Forăscu (Forăscu, 2001), Angela Bidu-Vrănceanu (Bidu-Vrănceanu, 1986), Theodor Hristea, (Hristea, 1984) followed by those of Adriana Soichiţoiu-Ichim (Stoichiţoiu-Ichim, 2001), Rodica Zafiu (Zafiu, 2001), Grigore Brâncuş (Brâncuş, 2004) or Adrian Chricu (Chircu, 2012).

Reading the studies published by our predecessors helped us to better perceive the differences occurring in the Romanian language, in the diachronicy and diatopic. Taking over the way how to interpret the language facts from them, our system is developed based on morphological and syntactical analysis of the words found in analyzed ancient texts as highlighted by the methodology proposed in this paper.

The rich literature tells its own story regarding the usefulness of technology and information services (Carstensen et al., 2009; Jurafsky & Martin, 2009; Manning & Schütze, 1999; Cole et al., 1998; Tufiş & Filip, 2002; Cristea & Butnariu, 2004; Trandabă
ct et al., 2012, Gîfu, 2015). The development and use of software for natural language processing (NLP) highlight the defining aspects of the text (morphological and syntactic analysis, semantic analysis and, more recently, pragmatic analysis).

The similarities between languages are interesting for historical and comparative linguistics, as well as for machine translation and language acquisition as well. Scannell (2006) and Hajič et al. (2000) argue for the possibility of obtaining a better quality in translation using simple methods for very closely related languages. Koppel and Ordan (2011) studied the impact of the distance between languages on the translation product and conclude that it is directly correlated with the ability to distinguish translations from a given source language from non-translated text. It has been established that some genetically related languages have a high degree of similarity to each other, and its speakers are able to communicate without prior instructions (Gooskens, 2006; Gooskens et al., 2008).

The approach for the study of the evolution of Romanian language is focusing only on the orthographic similarity. The basis for this approach consists of the idea that phonetic alterations have an orthographic correspondent, thus an alphabetic character correspondences (Delmestri and Cristianini, 2010).

Different approaches have been used in previous case studies in order to assess the orthographic distance similarity between related words. Their accuracy has been investigated and compared (Frunza et al., 2005; Rama and Borin, 2014), but a clear conclusion could not be drawn with respect to which method is the most appropriate for a given task. Metrics will be used to determine the orthographic similarity between related words. For the moment, we have the syllabic similarities of the Romanian language in different geographic areas and periods of time, starting by the Ciobanu and Dinu works (Ciobanu and Dinu, 2014). They used orthographic metrics like: the edit distance, the longest common subsequence ratio, and the rank distance.

III. Thesis idea

This survey describes the work methodology, starting with two collections of publications (Romanian and Moldavian), written at the middle of the 19th century, in order to compare them, emphasizing the language differences. In this sense, a modular structure is presented, including text processing, extracting quotes, WEKA statistics, and language similarity computation. As an illustration of the possible synergies between diachronic textual resources and linguistic research, a diachronic architecture is described using statistical machine learning techniques to infer probabilistic context-sensitive rules for the automatic delimiting in time and space of unknown words.

This amount of parallel data is of crucial interest to philologists and comparative linguists. Out of this context, it is also important for aligned journalistic corpora with the most important Romanian language resources as DEX-online and eDTLR, the last being developed by the Romanian Academy and âĂIJAlexandru Ioan CuzaâĂİ University of Iaşi.

IV. Authors and Affiliations

Formatting the authors' names and their affiliation depends on the number of authors and the number of different affiliations. Both names and affiliations spread over both columns.

V. Conclusion and future work

Language was not and is not static but the feature that characterizes language is the dynamism, whether it focuses on internal processes of word formation or loanwords. We were able to successfully create a search system for unknown words, acting especially on old text fields, these facts representing a premiere for Romanian language. For elaboration, symbolic method was used, combining efficiently rules created manually and a carefully organized external collection of files. It has been used two instruments of the Faculty of Computer UAIC, thus proving their usefulness: morphological and syntactic Tagger (WebPosRo) and Graphical Grammar Studio, and also improving existing findings. This resource can be useful in other projects on the same topic, where you only need to import.

By collecting all the information from an important resource we generate a large corpus that can be easily used in this application, but also this may be a way to extend the variation of programs that will use it. In this case, all this work of collecting content in order to get a large database will influence the final output of the main application.

Using the Naïve Bayse classifier available in WEKA, we managed to implement a mechanism which can find the words region and the period of time with 91% of correctly classified instances.

In the future we want to apply a few metrics in order to determine the orthographic similarity between related texts from the same period of time, but different areas. Moreover, we plan to extend this analysis for other kind of texts (literature, for instance), and to combine the orthographic approach with semantic evidence for a wider perspective on Romanian language similarity.

Acknowledgment

I would like to thank NESUS for supporting this article.

References

[1] Bidu-Vrănceanu, A. Structura vocabularului limbii române contemporane, Bucureşti, 1986.

[2] Carstensen, K-U., Ebert, C., Ebert, C., Jekat, S., Langer, H. and Klabunde, R. (eds.). ComputerlinguistikundSprachtechnologie: EineEinführung. Spektrum Akademischer Verlag, 2009.

[3] Chiarcos, C., Dipper, S., Götze, M., Leser, U., Lüdeling, A., Ritz, J. & Stede, M. A Flexible Framework for Integrating Annotations from Different Tools and Tag Sets. Traitment automatique des langues, 49, 2008, pp. 271-293.

[4] Chircu, A. Influenţa slavă asupra limbii române pe baza ALRM I. Terminologia corpului omenesc. Harta 1 (Corp), în Katalin Balazs, Ioan Herbil (eds.), Lucrările simpozionului internaţional âĂđDialogul slaviştilor la începutul secolului al XXI-leaâĂİ (Cluj-Napoca, 8-9 aprilie 2011), Cluj-Napoca, Casa Cărţii de ştiinţă, 2012, pp. 92-98.

[5] Ciobanu, A. and Dinu, L. An Etymological Approach to Cross-Language Orthographic Similarity. Appilcation on Romanian in Proceedings of EMNLP-2014, Oct. 25-29, 2014, Doha, Quatar, pp. 1047-1058

[6] Ciompec, G. Morfosintaxa adverbului românesc. Sincronie şi diacronie, Bucureşti, Editura Ştiinţifică şi Enciclopedică, 1985, p. 283.

[7] Claridge, C. Historical Corpora. In A. Lüdeling, & M. Kytö (Eds.), Corpus Linguistics. An International Handbook, Volume 1. Berlin: De Gruyter, 2008, pp. 242âĂŞ259.

[8] Cole, R., Mariani, J., Uszkoreit, H., Battista V., Giovanni, Zaenen, Annie and Zampolli, Antonio (eds.). Survey of the State of the Art in Human Language Technology. Cambridge University Press, 1998.

[9] Coşeriu, E. Sincronie, diacronie şi istorie. Problema schimbării lingvistice, versiune în limba română de Nicolae Saramandu, BucureŞti, Editura Enciclopedică, 1997.

[10] Cristea, D., Butnariu C. Hierarchical XML representation for heavily annotated corpora. In: Proceedings of the LREC 2004 Workshop on XML-Based Richly Annotated Corpora, Lisbon, Portugal, 2004.

[11] Delmestri, A. and Cristianini, N. String Similarity Measures and PAM-like Matrices for Cognate Identification. Bucharest Working Papers in Linguistics, 12(2), 2010, pp. 71âĂŞ82.

[12] Densusianu, O. Filologia Romanică în universitatea noastră, Bucureşti, J. V. Socecu Editeur, 1902, p. 23.

[13] Dimitrescu, Florica (coord.). Istoria limbii române, BucureŞti, Editura Didactică Şi Pedagogică, 1978.

[14] Dimitrescu, Florica. Dicţionar de cuvinte recente, Bucureşti, Editura Albatros, 1982.

[15] Forăscu, N. Dificultăţi gramaticale ale limbii române, Ed. Univ., Bucureşti, 2001.

[16] Frunza, O., Inkpen, D., and Nadeau, D. A text processing tool for the Romanian language. Proceedings of the EuroLAN 2005 Workshop on Cross-Language Knowledge Induction, 2005.

[17] Gîfu, D. Contrastive Diachronic Study on Romanian Language. In: Proceedings FOI-2015, S. Cojocaru, C.Gaindric (eds.), Institute of Mathematics and Computer Science, Academy of Sciences of Moldova, 2015, pp. 296-310.

[18] Gooskens, C. Linguistic and extra-linguistic predictors of Inter-Scandinavian intelligibility. In: Van de Weijer, J. & Los, B. (eds.). Linguistics in the Netherlands, 23, 101-113. Amsterdam: John Benjamins, 2006.

[19] Gooskens, C., Beijering, K. & Heeringa, W. Phonetic and lexical predictors of intelligibility. International Journal of Humanities and Arts Computing 2 (1-2), 2008, pp. 63-81.

[20] Graur, Al. TendinÅčele actuale ale limbii române, Ed. Ştiinţifică, BucureŞti, 1968.

[21] Guţu Romalo, V. Corectitudine Şi greŞeală. (Limba română de azi), Bucureşti, 1972.

[22] Guţu-Romalo, V. Aspecte ale evoluţiei limbii române, col. "Repere", Bucureşti, Editura Humanitas Educaţional, 2005.

[23] Hajič, J., Hric, J., and Kuboň, V. Machine translation of very close languages. In Proceedings of the 6th Applied Natural Language Processing Conference, pages 7âĂŞ12. Association for Computational Linguistics, 2000.

[24] Hristea, Th. Sinteze de limba română, Editura Albatros, 1984.

[25] Iancu, V. Istoria limbii române, col. "Argumente", BucureŞti, Editura Fundaţiei Culturale Române, 2000.

[26] Iordan, I. Stilistica limbii române, Ed. Ştiinţifică, BucureŞti, 1975.

[27] Kytö, M. Corpora and historical linguistics. Revista Brasileira de Linguistica Aplicada, 11(2), 2011, pp. 417-457.

[28] Kytö, M., & Pahta, P. Evidence from historical corpora up to the twentieth century. In T. Nevalainen, & E. C. Traugott (Eds.), The Oxford Handbook of the History of English. Oxford o.a.: Oxford University Press, 2012, pp. 123-133.

[29] Lüdeling, A., Poschenrieder, T., Faulstich, L. C. et al. DeutschDiachronDigital - Ein diachrones Korpus des Deutschen. Jahrbuch für Computerphilologie 2004, 2005, pp. 119-136.

[30] Manning, C. D. and Schütze, H. Foundations of Statistical Natural Language Processing. MIT Press, 1999.

[31] Rama, T and Borin, L. Comparative Evaluation of String Similarity Measures for Automatic Language Classification. In George K. Mikros and Jan Macutek, editors, Sequences in Language and Text. De Gruyter Mouton, 2014.

[32] Resnik, P., Broman Olsen, M. and Diab, M. The Bible as a Parallel Corpus: Annotating the 'Book of 2000 Tongues'. Computers and the Humanities 33, 1999, pp. 129-153.

[33] Rissanen, M. Corpus linguistics and historical linguistics. In: Corpus Linguistics: an International Handbook. Vol. 1, ed. by Anke Lüdeling and Merja Kytö. Berlin and New York: Walter de Gruyter. 2008, pp. 53-68.

[34] Rosetti, Al. Istoria limbii române, de la origini până în secolul al XVII-lea, cu 6 hărţi afară din text, BucureŞti, Editura pentru literatură, 1968.

[35] Rosetti, Al., Cazacu, B., Onu, L. Istoria limbii române literare, Bucureşti, Editura Minerva, 1971.

[36] Sala, M. De la latină la română, col. "Limba română", nr. 1, Bucureşti, Editura Univers Enciclopedic & Academia Română, 1998.

[37] Scannel, K. Statistical models for text normalization and machine translation. In Proceedings of the First Celtic Language Technology Workshop, pages 33âĂŞ40, Dublin, Ireland, August 23 2014.

[38] Stoichitoiu-Ichim, A. Vocabularul limbii romane actuale. Dinamica, influente, creativitate, Bucureşti, Editura All, 2001.

[39] Sukhareva, M. And Chiarcos, C. Diachronic proximity vs. data sparsity in cross-lingual parser projection. A case study on Germanic in Proceedings of the First Workshop on Applying NLP Tools to Similar Languages, Varieties and Dialects, Dublin, Ireland, August 23, 2014, pp. 11âĂŞ20.

[40] Trandabăţ, D., Irimia, E., Barbu Mititelu, V., Cristea, D., Tufiş, D. The Romanian Language in the Digital Age. In: White Paper Series, Georg Rehm and Hans Uszkoreit (eds.), Berlin, Springer, 2012

[41] Tufiş, D., Filip, F. Gh. (coord.). Limba română în Societatea informaţională âĂŞ Societatea Cunoaşterii, Ed. Expert, Bucureşti, 2002.

[42] Zafiu, R. Diversitate stilistică în româna actuală, BucureŞti, 2001.

Dynamic Management of Resource Allocation for OmpSs Jobs

Sergio Iserte* Antonio J. Peña† Rafael Mayo* Enrique S. Quintana-Ortí*
Vicenç Beltran†

*Universitat Jaume I (UJI), Spain
†Barcelona Supercomputing Center (BSC-CNS), Spain

Abstract

The main purpose of this thesis is to research in the relation between task-based programming models and resource management systems in order to provide a smart autonomous load-balancing and fault-tolerant system. Thus, taking advantage of MPI malleable applications and execution models such as SMPD and MPMD we will dig in the principle of the dynamical reconfiguration. Apart from providing an overview of the thesis idea, this paper explains our initial motivation and reviews briefly the most remarkable work done in this field.

Keywords Exascale, heterogeneous systems, dynamic reconfiguration, OmpSs, resource management

I. Introduction and motivation

It is consensually believed that Exascale performance will only be achieved by adopting specialized hardware, what inevitably will turn systems into heterogeneous facilities. Dealing with heterogeneous hardware not only involves a tougher management of the cluster, but also a rise in the complexity of the applications which wanted to use all the resources available.

The vast majority of scientific applications have been developed using the Message Passing Interface (MPI) [7], in order to distribute the work among the nodes of a cluster. Two execution flows can be followed in this programming model:

- **Single Program Multiple Data (SPMD)** is the traditional and most extended approach. In this mode, all the processes will execute the same code working on different parts of the data.
- **Multiple Program Multiple Data (MPMD)**. This more recent mode does not restrict all processes to execute the same code. Usually, MPI applications are composed of several computational stages. If these stages can be executed independently and can be accelerated in specific hardware, we could refer to that as an offloading of the code in a device. This model fits better in heterogeneous environments.

The vast majority of MPI applications are moldable; they can be launched with different numbers of resources, which remain constant during all the application execution time. On the contrary, malleable applications can vary the amount of resources used in their execution, what means that applications are able to adapt themselves to changes in the environment.

Dynamic reconfiguration of MPI applications has been an important issue for many years. Its importance resides in the necessity of maximizing the utilization rate of the resources in an HPC cluster. Furthermore, it can reduce waiting times in queues by sizing jobs to the available resources or distributing sets of nodes among jobs. Hence, considerable effort made in the field of reconfiguration has been focused on the ability of malleability. This reconfiguration can be triggered by

the application itself or by the Resource Management System (RMS)—in the literature we can find defined this last set as *evolving applications*.

Nevertheless, dynamic reconfiguration is still a hot topic due to the blooming of new programming models which try to exploit heterogeneous HPC systems. One of the most extended modes is OmpSs [8] (developed by The Barcelona Supercomputing Center) which extends OpenMP with new directives to support asynchronous parallelism and heterogeneity. OmpSs enables asynchronous parallelism by using data dependencies among the tasks of the program. Offloading the MPI kernels dynamically using the OmpSs programming model could foster the adoption of the recenly emerged MPMD execution model [5] .

Moreover, the execution of these applications are generally handled by a RMS conscious of the status of all the hardware available in the facilities. If an application decided to change its allocated resources for different others, the RMS should be noticed in order to grant the operation at a given time.

II. Related Work

On the one hand, we find many contributions in the field of process malleability, having as a result excellent reconfiguration techniques or tools. For instance, authors in [3] explored the integration of malleability extensions in the process checkpointing and migration library (PCM) [4]. They take advantage of moldability to make the applications malleable by finishing and restarting them again. Also, there are contributions that make easier the adoption of malleability in applications with mechanisms of dynamic load-balancing [10], as well as reconfiguration techniques that are able to redistribute the workload and change the number of processes of a running application to obtain a certain performance [6].

On the other hand, projects that go further than just malleability techniques have been paving the road to exascale performance. One of the most remarkable is the DEEP Project [2]. DEEP is an innovative response to the exascale challenge, where a new organization is proposed: instead of providing the nodes with accelerators, the devices are put aside in an acceleration cluster, called "booster". In this scheme, both sides are interconnected by a high performance network. Applications offload their tasks to the "boosters" by using the OmpSs programming model.

[5] presents an extension of OmpSs to support dynamic offload of tasks among MPI processes. This provides flexibility, performance and scalability. However, the integration of that extension in a RMS is not addressed.

[1] presents a study of how to interact with an OmpSs job and the RMS that manages the facility. This work addresses the following limitations:

- The resources have to be requested on submission time, and the request is invariable. Hence, regardless of whether the application is using the "booster" or not, the resources are allocated.
- Queue and resource management. DEEP does not know the status of the nodes and its resources, making scheduling virtually impossible.

The work is concluded with a series of scripts to communicate the job and the RMS in order to perform the reconfiguration. However, an intelligent system with capacity of decision is left for future work.

III. Thesis Idea

The main objective of this thesis is to provide a user-friendly methodology to manage the resources assigned to a running job. Following partly the work in [1] (see Section II), our idea is still based on the fact that heterogeneous systems are paving the road to the exascale era, and that taking advantage of a programming model that supports asynchronous parallelism is crucial. Hence, combining the OmpSs multi-task (internally handled by threads) support with the capabilities of MPI to make the most of the distributed programming, the two most common programming models will be explored:

- SPMD: MPI + OmpSs (OpenMP). The user code should be adapted to provide a malleable MPI application (similar to application-based checkpoint/restart). Here, the application actively asks for a change of its assigned resources on response to a resource change request from the RMS.

- MPMD: MPI + OmpSs offload + OmpSs (OpenMP). In this scenario, the offloaded stage could be assigned with more or less resources depending on the decisions automatically taken by both the OmpSs runtime and the logic of the resource manager. However, having a malleable kernel like the one described in the previous point could boost the benefits.

Technically speaking, the OmpSs application should count with synchronization points where a reassignation of resources could be performed (whether a variation in the quantity or only a replacement). The synchronization points will be managed by a series of directives. Thus, the OmpSs runtime will be the responsible for moving data among tasks in different machines.

In addition, another interesting study case is that related to the *states*. Occasionally, servers save their own states as a guarantee of recovery in the case of a physical failure. This state could be loaded in another server and the execution of its jobs could be resumed. In this scenario the runtime of OmpSs should take additional care and provide more information about the states of its jobs in the different servers in order to let the RMS decide an appropriate strategy to reschedule the jobs and the resources.

In order to take reallocation decisions, four situations may happen:

1. An OmpSs job requests more resources: if the RMS has available resources, the job will be provided with them; otherwise, the request will be ignored or postponed.
2. An OmpSs job finishes a computational stage giving as a result a release of part of the allocated resources: the OmpSs runtime will notify the RMS about which resources are made available.
3. The RMS decides to assign more resources to an OmpSs job: at a given time, the RMS realizes that there are unused resources. Hence, if a job that previously had requested an expansion is still running, Slurm will assign more resources to it.
4. The RMS notices a stress situation (the queue is growing dramatically and the wait times have increased sharply) or the priority of other jobs is higher than that of the running job. If any running OmpSs job in the queue has been provided with the capability of reducing its allocation, the RMS could remove resources from the job. Of course, the OmpSs runtime will be aware of the location of the job data in order to redistribute it appropriately.

On the side of the RMS, we have decided to make use of Slurm [9]. Having an open source tool which provides a complete API and has proven that can reassign resources during the execution of a job [1] will increase the adoption of this project. Slurm is aware of the status of all the hardware under its control and ultimately the responsible for granting any reallocation operation.

To summarize, the main contributions that we expect from this work are:

- Integration of process malleability features in the OmpSs programming model, with the following actions:
  - We will propose extensions to the current application programming interface which will be considered for the OpenMP programming model.
  - We will develop the required functionality into the current OmpSs runtime and compiler.
  - We will define two APIs to face the triggered actions from both the RMS and the OmpSs application:
    * The first API will allocate/release resources.
    * While the second will check if there is a need for changing the resources currently assigned. In this case, once the RMS informs the application about a resource change, the application should use the first API to reallocate new resources.
- Novel dynamic reallocation scheduling policies with the enough intelligence to perform smart reallocation actions.

- Extensive performance evaluations in order to demonstrate the viability of using this new approach.

IV. Conclusion and Future Work

So far, the project is in an embryonic stage where we are still pursing an MPI malleable application. The application at issue will be used to measure the performance among versions.

Apart from the immediate appealing of having a process-malleable user-friendly environment, we strongly believe that this work can be directly applied on the resilience field, due to the capacity of adaptation to the environment that it presents. Exascale performance will involve a massive number of nodes working together. Such quantity of hardware increases the likelihood of experiencing a malfunction. Working at that scale a failure that entailed the re-execution of a job would represent a large waste of money and time. Having a system capable of reallocating efficiently resources in execution time, would transparently be highly beneficial.

Acknowledgment

This work is partially supported by EU under the COST Program Action IC1305: Network for Sustainable Ultrascale Computing (NESUS); and the Project TIN2014-53495-R from MINECO and FEDER.

References

[1] Marco D'Amico. Extending deep offload programming model. Master's thesis, 2015.

[2] DEEP Project. http://www.deep-project.eu.

[3] Kaoutar El Maghraoui, Travis J. Desell, Boleslaw K. Szymanski, and Carlos A. Varela. Dynamic malleability in iterative MPI applications. In *Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid '07)*, pages 591–598. IEEE, May 2007.

[4] Kaoutar El Maghraoui, Boleslaw K. Szymanski, and Carlos Varela. An architecture for reconfigurable iterative MPI applications in dynamic environments. In *Parallel Processing and Applied Mathematics*, pages 258–27. 2006.

[5] V. Beltran F. Sainz and J. Labarta. Collective offload for heterogeneous cluster. *2nd IEEE International Conference on High Performance Computing (HiPC)*, Dec 2015.

[6] Gonzalo Martín, David E. Singh, Maria-Cristina Marinescu, and Jesús Carretero. Enhancing the performance of malleable MPI applications by using performance-aware dynamic reconfiguration. *Parallel Computing*, 46:60–77, Jul 2015.

[7] MPI Standard 3.1. http://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf.

[8] OmpSs. https://pm.bsc.es/ompss.

[9] SLURM Workload Manager. http://slurm.schedmd.com.

[10] Masha Sosonkina, Layne T. Watson, Nicholas R. Radcliffe, Rafael T. Haftka, and Michael W. Trosset. Adjusting process count on demand for petascale global optimization. *Parallel Computing*, 39(1):21–35, Jan 2013.

Spatial and Temporal Cache Sharing Analysis in Tasks

Germán Ceballos, David Black-Schaffer

Uppsala University, Sweden
firstname.lastname@it.uu.se

Abstract

Understanding performance of large scale multicore systems is crucial for getting faster execution times and optimize workload efficiency, but it is becoming harder due to the increased complexity of hardware architectures. Cache sharing is a key component for performance in modern architectures, and it has been the focus of performance analysis tools and techniques in recent years. At the same time, new programming models have been introduced to aid the programmer dealing with the complexity of large scale systems, simplifying the coding process and making applications more scalable regardless of resource sharing. Task-based runtime systems are one example of this that became popular recently. In this work we develop models to tackle performance analysis of shared resources in the task-based context, and for that we study cache sharing both in temporal and spatial ways. In temporal cache sharing, the effect of data reused over time by the tasks executed is modeled to predict different scenarios resulting in a tool called StatTask. In spatial cache sharing, the effect of tasks fighting for the cache at a given point in time through their execution is quantified and used to model their behavior on arbitrary cache sizes. Finally, we explain how these tools set up a unique and solid platform to improve runtime systems schedulers, maximizing performance of execution of large-scale task-based applications.

Keywords Task-based runtime systems, cache sharing, performance analysis, NESUS

I. Introduction

Maximizing applications performance on the multi-cores era is hard due to sharing resources, such as the caches, as it can have a negative or positive impact on the total execution time. To deal with this, newest programming models simplify the coding process of large scale parallel applications. Task based programming is one example of this, where the code is disaggregated in small units of code (independent functions) called tasks, and a runtime system determines their execution order and placement. The task based approach is simpler to reason for the programmer while it is also a good approach for performance as it can adapt the scheduling to the effective resource sharing. However, it is a very different dynamic of execution, making harder to understand performance of these systems due to the lack of models and tools.

In this paper we look at two key types of cache sharing (both *temporal* and *spatial*, in a task based context. An application might reuse data brought to the cache in the past, meaning that the cache is being shared in a *temporal* way. On the other hand, two applications might contend for the cache at the same moment in time, fighting to install and keep data in it, meaning that the cache is being shared in a *spatial* notion.

To do so, we develop efficient modeling techniques to predict performance with the goal of improving runtime scheduling decisions based on task sensitivity to hardware resource sharing, maximizing performance of large scale parallel applications.

To achieve this we first developed StatTask, a fast and efficient method to predict cache miss ratios for any arbitrary schedule from information sampled from a single execution. This method addresses temporal cache sharing between tasks: how sensitive tasks are to inter-task data reuse over time. An example can be seen in Figure 1 for tasks A, B and C. Tasks A

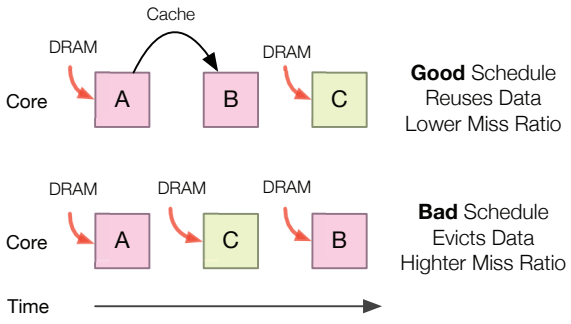


Figure 1: Temporal Locality in Task Based Systems.

and B share data, and B might reuse it from the cache However, executing tasks C in between could evict this shared data, causing data to be fetched from memory and increasing the execution time.

Second, we developed a method for predicting performance of co-running applications combining both statistical cache models and performance models for regular applications. Previous works did not take into account parallelism in the memory hierarchy in combination with statistical cache models, which is a key factor for performance.

Later, we extended this method to address tasks spatial resource sharing: how the memory hierarchy is shared at a given moment in time during execution. An example is display in Figure 2, where tasks A1 and B1 executing in parallel will bring data at the same time to the caches with different ratios. Since they fight for the cache, both tasks will end up with smaller cache portions impacting on their performance. However, if tasks would have been co-executed with tasks sharing data (respectively A2 and B2) sharing could have reduced their misses.

The method we present is able to predict quickly, accurately and with low-overhead, how multiple tasks running in parallel will compete for the caches.

Third, we explain how our models for temporal and spatial cache sharing can be combined improve schedulers of task-based runtime systems by giving them feedback.

II. Related work

There are three categories of related work: existing profiling tools that identify bottlenecks of task-based applications, task-scheduling optimization techniques,

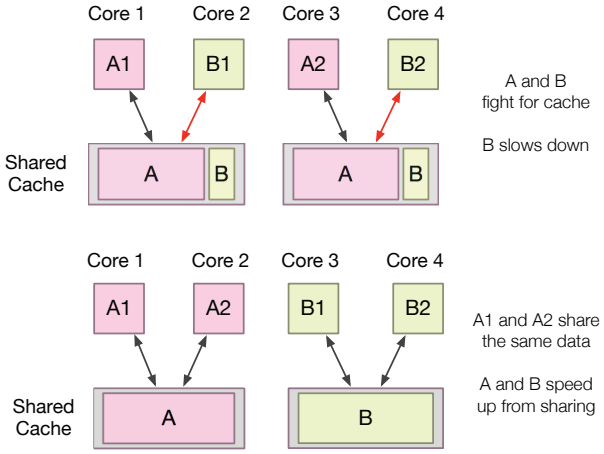


Figure 2: Spatial Locality in Task Based Systems.

and finally techniques to analyze and understand data locality properties of applications.

Many tools exist to profile scheduling and load-balancing of tasks. Ding et. al. [8] presented a generic and accessible tool for task monitoring, independent of any program or library and able to acquire rich information with very low overhead, targeting load balancing and scheduling problems unrelated to data reuse. Lorenz et. al developed [16], a library for identifying performance problems inherent to tasking with OpenMP through direct instrumentation. Schmidl et. al. [17] surveyed different techniques to analyze data delivered by instrumentation of task-based programs in order to integrate parallel performance modeling to the automation of load-balancing. Ghosh et. al. [14] have proposed OpenMP extensions to support dependence-based synchronization, Brinkmann et al. presented a graphical debugging tool for task parallel programming that works with most of the production frameworks. Weng and Chapman [19] looked at the task graph for OpenMP applications to optimize load balance.

In the second category, work has been done on improving scheduling strategies. The standard work-stealing approach was carefully analyzed by Blumofe and Leiserson in [5] and [1]. Strategies accounting for the tasks types were presented by Wimmer et. al. [20]. Adaptive cut-off scheduling to take advantage of data locality and reduce the runtime overhead were considered in [9]. Recently, important work on cache-aware task stealing was carried out in [7] by Chen et. al. Qian Cao et. al.

[6] proposed a hybrid scheduling policy for heterogeneous multicores using breadth-first over the available task-pool.

None of these approaches for task-based profiling have incorporated a general method for understanding the data reuse implications of the tasks and schedules. In this category, characterization of data reuse has been done theoretically in [12] by Frigo. Practically, this can be done through instrumentation based techniques as presented by Aamer et. al. in [15] and Weidendorfer in [18].

Statistical cache modeling, first introduced in [2], is another widely used way to characterize data locality. This work has been extended to other cache replacement policies by Eklov in [11], and to support thread-based or multicore shared caches in [4, 3, 10].

III. Thesis Idea

Our main contribution is the development models that address the prediction of temporal and spatial cache sharing for arbitrary cache sizes for task-based runtime systems. These model preserve fundamental properties to be used in conjuntion with runtime schedulers for better scheduling: both models are fast and low-overhead, portable (easy to implement across different runtimes) and architectural-independent (working seamlessly with different architectures).

III.1 Temporal resource sharing

For task-based programs, data is initially brought into the cache by a task, and if it is reused, this reuse can come from either the same task (private reuse) or by a subsequent one (shared reuse). Other tasks that execute between tasks with shared data also bring new data into the cache that may evict the shared data, turning reuses from the cache into a cache misses, and hurting performance.

Thus, we classify memory accesses in three types, depending on where they come in the memory hierarhcy: First accessses to a particular memory location must be brought from DRAM, for example cold cache misses, and therefore we will call them *DRAM Accesses*. Second, memory accesses to addresses previously lodaded by another task, and which we will call *shared reuses*. This reuses will be able to bring data from the cache if it is large enough to hold the data sets of the sharing tasks and the data is not evicted by other tasks before the shared reuse. Finally, memory accesses to addresses previously loaded by the same task, called *private reuses*. This type of accesses will bring data from the cache if it is large enough to hold the entire task's data set while it is executing. With this classification, we are able to improve statistical cache models to support memory access information per-task.

A key property of statistical cache models is that are able to sample a memory access stream from an application during execution, build a profile depending on a *distance notion* that determines how close/far the data reuses happened, and use statistical inference to predict cache miss ratios for different cache sizes very quickly. However, if these methods are used on task-based applications, the profile would be built based on information collected from the execution of a particular schedule. Since changing the tasks' schedule can affect observed data reuses, predictions for cache misses given by these models would be wrong.

StatTask extends existing statistical cache models collecting extra information during the memory profiling stage. Memory access samples are taken for a particular task schedule and then classified on a task basis. Later, multiple profiles are built for different schedules, adapting what would happen to the distances in the reuses on each of those cases. With these new profiles, statistical inference is used to get cache miss ratios for the new schedules, predicting the correct scenarios. This enables accurate prediction of cache behavior for arbitrary schedules of tasks and cache sizes.

III.2 Spatial resource sharing

When analyzing cache behavior in multi-program workloads, previous statistical cache models did not treat memory level parallelism, which now became crucial in latest architectures. In modern multicore processors, a last level cache miss might queue a new request in the memory controller's queue, which might be handled in parallel with a previous miss. Thus two consecutive misses are likely to overlap, hiding the latency for the second miss compared to the case of treating them sequentially having a drastic improvement in performance.

The number of parallel misses treated on average throughout execution can be measured and is often known as memory level parallelism (*MLP*). Our second contribution is a technique that combines statistical cache modeling with a modern performance model, adding support for memory level parallelism, that is able to predict a breakdown of performance (measured

in CPI) of co-running applications.

To do so, applications memory accesses are sampled with binary instrumentation, running on isolation. Later, a statistical cache model called StatCC is used to predict cache miss ratios of co-running application for arbitrary cache sizes, assuming an initial performance. Later, a realistic and advanced performance model called Interval model [13] is used to calculate the number of cycles spent on memory per-application when co-running. The interval model is based on the abstraction that the execution time is driven by long-latency events, such as long latency loads and branch misses. However, the number of cycles calculated can change the ratio in which each application miss in the cache. Thus, StatCC is used iteratively, predicting new miss-ratios and recomputing the number of cycles spent on memory with the interval model as a fixed-point iterative solver.

This method needs to be adapted for the task-based context. To do that, it is necessary to add the same support for identifying tasks as in Section III.2 generating a per-task profile. In addition, the MLP modeling has to be done on a per-task basis as well. Our method runs on a pair profiles tasks sequences profiles and applies the technique described above to estimate the CPI of both sequences of co-running tasks.

IV. Conclusion and future work

Multicore architectures have the potential for high performance on parallel applications, but they are hard to optimize for due to the complexities of resource sharing. In this work we have presented two contributions to understand cache sharing in a task-based context based on the analysis of memory access samples. First, we presented StatTask, an efficient statistical cache model that predicts cache miss ratios for arbitrary task schedules, addressing the temporal cache sharing problem. Second, we introduced a new method that quickly predicts the effect of simultaneous cache sharing on the tasks performance, addressing the spatial cache sharing issue. Both of our methods use the same low-overhead, sampled input information, and can be easily combined to enable performance modeling of arbitrary task schedules. With these new capabilities we will be able to to develop more intelligent task scheduling policies that take into account the effects of temporal and spatial cache sharing, and thereby enable task-based programs to automatically adapt to the complexities of modern multicore resource sharing.

Acknowledgments

The work presented in this paper has been partially supported by EU under the COST programme Action IC1305,'Network for Sustainable Ultrascale Computing (NESUS)', and by the Swedish Research Council, carried out within the Linnaeus centre of excellence UPMARC, Uppsala Programming for Multicore Architectures Research Center.

References

[1] U. Acar, G. Blelloch, and R. Blumofe. The data locality of work stealing. *Theory of Computing Systems*, 35(3):321–347, 2002.

[2] E. Berg and E. Hagersten. Statcache: A probabilistic approach to efficient and accurate data locality analysis. *Proceedings of the 2004 IEEE International Symposium on Performance Analysis of Systems and Software*, 2004.

[3] E. Berg and E. Hagersten. Fast data-locality profiling of native execution. *SIGMETRICS Perform. Eval. Rev.*, 33(1):169–180, June 2005.

[4] E. Berg, H. Zeffer, and E. Hagersten. A statistical multiprocessor cache model. In *Performance Analysis of Systems and Software, 2006 IEEE International Symposium on*, pages 89–99, March 2006.

[5] R. D. Blumofe and C. E. Leiserson. Scheduling multithreaded computations by work stealing. *J. ACM*, 46(5):720–748, Sept. 1999.

[6] Q. Cao and M. Zuo. A scheduling strategy supporting OpenMP task on heterogeneous multicore. In *26th IEEE International Parallel and Distributed Processing Symposium Workshops & PhD Forum, IPDPS 2012, Shanghai, China, May 21-25, 2012*, pages 2077–2084, 2012.

[7] Q. Chen, M. Guo, and Z. Huang. Cats: Cache aware task-stealing based on online profiling in multi-socket multi-core architectures. In *Proceedings of the 26th ACM International Conference on Supercomputing*, ICS '12, pages 163–172, New York, NY, USA, 2012. ACM.

[8] Y. Ding, K. Hu, and Z. Zhao. Performance monitoring and analysis of task-based OpenMP. 2013.

[9] A. Duran, J. Corbalan, and E. Ayguade. An adaptive cut-off for task parallelism. In *High Performance Computing, Networking, Storage and Analysis, 2008. SC 2008. International Conference for*, pages 1–11, Nov 2008.

[10] D. Eklov, D. Black-Schaffer, and E. Hagersten.

Statcc: A statistical cache contention model. In *Proceedings of the 19th International Conference on Parallel Architectures and Compilation Techniques*, PACT '10, pages 551–552, New York, NY, USA, 2010. ACM.

[11] D. Eklöv and E. Hagersten. Statstack : Efficient modeling of LRU caches. In *Proc. International Symposium on Performance Analysis of Systems and Software : ISPASS 2010*, pages 55–65. IEEE, 2010.

[12] M. Frigo and V. Strumpen. The cache complexity of multithreaded cache oblivious algorithms. *Theory of Computing Systems*, 45(2):203–233, 2009.

[13] D. Genbrugge, S. Eyerman, and L. Eeckhout. Interval simulation: Raising the level of abstraction in architectural simulation. In *In High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*, pages 1–12. IEEE, 2010.

[14] P. Ghosh, Y. Yan, D. Eachempati, and B. M. Chapman. A prototype implementation of OpenMP task dependency support. In *OpenMP in the Era of Low Power Devices and Accelerators - 9th International Workshop on OpenMP, IWOMP 2013, Canberra, ACT, Australia, September 16-18, 2013. Proceedings*, pages 128–140, 2013.

[15] A. Jaleel, R. S. Cohn, C. keung Luk, and B. Jacob. Cmp$im: A pin-based on-the-fly multi-core cache simulator.

[16] D. Lorenz, P. Philippen, D. Schmidl, and F. Wolf. Profiling of OpenMP tasks with Score-P. In *41st International Conference on Parallel Processing Workshops, ICPPW 2012, Pittsburgh, PA, USA, September 10-13, 2012*, pages 444–453, 2012.

[17] D. Schmidl, P. Philippen, D. Lorenz, C. Rössel, M. Geimer, D. an Mey, B. Mohr, and F. Wolf. Performance analysis techniques for task-based OpenMP applications. In *OpenMP in a Heterogeneous World - 8th International Workshop on OpenMP, IWOMP 2012, Rome, Italy, June 11-13, 2012. Proceedings*, pages 196–209, 2012.

[18] J. Weidendorfer, M. Kowarschik, and C. Trinitis. A tool suite for simulation based analysis of memory access behavior. In *In Proceedings of International Conference on Computational Science*, pages 440–447. Springer, 2004.

[19] T. Weng and B. Chapman. Towards optimisation of openmp codes for synchronisation and data reuse. *Int. J. High Perform. Comput. Netw.*, 1(1-3):43–54, Aug. 2004.

[20] M. Wimmer, D. Cederman, J. L. Träff, and P. Tsigas. Work-stealing with configurable scheduling strategies. In *Proceedings of the 18th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '13, pages 315–316, New York, NY, USA, 2013. ACM.

Application Partitioning and Mapping Techniques for Heterogeneous Parallel Platforms

Rafael Sotomayor, J. Daniel Garcia

University Carlos III, Spain
rsotomay@inf.uc3m.es, jdgarcia@inf.uc3m.es

Abstract

Parallelism has become one of the most extended paradigms used to improve performance. Legacy source code needs to be re-written so that it can take advantage of multi-core and many-core computing devices, such as GPGPU, FPGA, DSP or specific accelerators. However, it forces software developers to adapt applications and coding mechanisms in order to exploit the available computing devices. It is a time consuming and error prone task that usually results in expensive and sub-optimal parallel software.

In this work, we describe a parallel programming model, a set of annotating techniques and a static scheduling algorithm for parallel applications. Their purpose is to simplify the task of transforming sequential legacy code into parallel code capable of making full use of several different computing devices with the objetive of increasing performance, lowering energy consumption and increase the productivity of the developer.

Keywords Parallel computing, heterogeneous computing, programming models, kernel partitioning

I. Introduction

In recent years, traditional approaches to improving CPU performance have reached a limit due to the limitations of sequential programming models as well as the physical constraints related to clock speed (such as heat dissipation or power consuption). As a result, efforts have turned to developing heterogeneous hardware architectures, combining several computing devices other than CPUs (such as GPUs, FPGAs or DSPs), programmed in a highly parallel fashion.

This approach, however, has limitations. Firstly, each kind of device has a different architecture, and it is usually necessary to follow a very specific programming model. This makes it very difficult to write code that makes full use of these heterogeneous architectures. Secondly, a very intimate knowledge of both architectures and programming models is necessary to make an efficient use of these devices with regards to high performance and low energy consupmtion.

The purpose of this work is to develop a unified programming model that can be used in this kind of heterogeneous parallel platforms in order to: (1) reduce power consumption, (2) improve performance and (3) increase productivity realizing designs.

The rest of the paper is organized as follows. In section II, the related work is summarized. Section III presents the proposed model. Finally, section IV shows the results and conclusions drawn so far, and outlines some future work.

II. Related work

In the literature we can find pragma-based frameworks that allow executing code in multi-devices. Some works take advantage of open standards in order to execute legacy code block in GPUs. Some examples are Wienke et al. [2], based on OpenACC, and Bertolli et al. [3] who use the newest version of OpenMP 4.0.

From a semantic viewpoint, C++11 attributes provide some advantages over pragma-based frameworks

[4]. They do not need support from the preprocessor, they can be applied to every syntactic element in the code, and they provide a portable way of annotating code.

Automatic kernel selection techniques is an important research field for automatic serial code parallelization [5], including GPU source code transformation. Multi-core as target devices is also considered for automatic source code transformation. For example, polyhedral tools are used in order to create source code that improves cache accesses with tiling optimizations [6]. However, all of these tools focus on one particular kind of optimization , such as CPU-only, accelerators-only)

The Open Computing Language (OpenCL) [1] is a C-based programming model, used for different computing devices (e.g. CPUs, GPGPUs, DSP, FPGA, accelerators) that has become widely accepted and supported by major vendors. OpenCL is based on parallel code regions, called kernels, that can be executed on a device. OpenCL allows the development of heterogeneous parallel applications that could use more than one computing device, improving application efficiency. It's use with CPU for HPC systems has been studied in recent years [9], concluding that the performance is close to OpenMP and other library solutions.

III. Thesis project

As explained before, the goal of this work is to develop a unified programming model targeting several programming devices under a single annotation system based on C++11 attributes. This model draws heavily from OpenCL. The different sections of the code susceptible to parallelization are refered to as kernels. Also, the memory model is host-centric, and the CPU, acting as said host, is in charge of orchestrating memory transfers to and from device memory space. Also, a set of code annotation techniques are developed to allow the transformation and optimization of sequential code into parallel heterogeneous code.

To this end, the following milestones have been set:

III.1 Hardware description tool

In order to split the parallel kernels between the different devices in an efficient manner in line with the goals of performance and energy consumption, it is necessary to know the capabilities and limitations of the different devices that make up an heterogeneous parallel platform. From now on, we will refer to an heterogeneous parallel platform as one made of multicore, GPGPU,with CPU for HPC systems has been studied in recent years [9], concluding that the performance is close to OpenMP and library solutions FPGA, DSP or combinations of all the previous.

The Heterogeneous Parallel Platform Description Language (HPP-DL) is a specification of a description language that provides all the relevant details of an heterogeneous parallel platform. It is designed to be human readable, so that automated and non-automated descriptions of platforms can be made. JSON (JavaScrip Object Notation) format has been adopted to represent the HPP-DL information.

HPP-DL allows to express the characteristics of a hardware system via a hierarchical model. Its intended use is making sure that platform-specific information is made available to (1) expert programmers and (2) tools such as auto-tuners, compilers or run-time systems. The HPP-DL format is independent of the programming model used. This means that it can be used as a virtual platform for other offline simulations. HPP-DL makes use of existing tools, mainly Hardware Lister (lshw) and Hardware Locality (hwloc) tool.

With the HPP-DL, the hardware parallel platform can be described in terms of:

- **Components**: each of the parts that make up the whole HPP, such as processors, memory banks, cache or different devices, and they are interconnected in various ways. Different devices contain different information.
- **Links**: this entity represents the relationships between two different components of the HPP. It describes a one-way connection in terms of throughput and latency. An example of link is the PCIe between the board and a GPGPU. It currently does not cover connections between different computers.
- **Resources**: refer to IS-specific information about resources used by/allocated to a component, such as I/O ports, IRQs or address ranges. Their main

use is to develop code for FPGA boards, where low-level memory operations are necessary.

III.2 Software annotation mechanisms

This mechanisms are based on an ad-hoc set of C++11 attributes [10]. Their purpose is to include semantic information about the kernels, so that the sequential code may be automatically optimized and, ultimately, transformed for a specific device.

These attributes can be used to define kernels in a code base, their behaviour (e.g. `rpr::map`) and their parameters (e.g. `rpr::in`). We refer to these parallel regions as *kernels*. An attribute is attached to a syntactic entity (e.g. a statement, loop, or definition), as defined by the standard C++ grammar. In general, an annotation precedes the syntactic element it is annotating to and does not require any preprocessing (a key difference with *pragma* based solutions).

Listing 1: Block-based matrix multiplication with `REPARA` attributes.

```
[[rpr::kernel, rpr::map,
  rpr::in(A,B,C,AN,BN,CN,b),
  rpr::out(C)]]
for(long i=0;i<mblocks;++i)
 for(long j=0;j<nblocks;++j)
  for(long k=0;k<pblocks;++k) {
   double *Aik = &A[b*(i*AN + k)];
   double *Bkj = &B[b*(k*BN + j)];
   double *Cij = &C[b*(i*CN+j)];
   MMul(b,Aik,AN,Bkj,BN,Cij,CN);
  }
```

Listing 1 shows a basic map computations using our attributes. The attribute `rpr::kernel` annotates the subsequent single or compound statement expressing the programmers intent of marking it as a kernel region. Kernel nesting is not considered in our model, therefore when two or more kernels are nested, inner annotations are ignored.

Additional attributes may be applied to a kernel region to refine intentions and to provide additional information. For example, `rpr::map` or `rpr::farm` can be used to express the expected parallel pattern transformation.

Additionally, the `rpr::in` and `rpr::out` attributes are used to identify input and output parameters of the kernel, respectively. Input/output sets do not need to be disjoint, allowing a parameter to be both input and output when needed.

A tool has been created to automatically detect kernels and transform the sequential code to OpenCL code [11]. We propose a workflow containing four different stages:

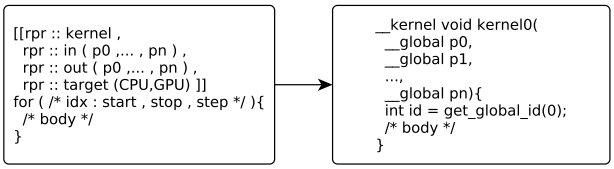


Figure 1: Basic source-to-source transformation process.

Kernel detection. Applies a set of rules to find potential kernels in a legacy C++ source code.

IR multi-version generation. This stage takes the output from the previous stage and generates a set of possible versions for each kernel.

Multi-version selection. For each set of versions, we apply Multiple Attribute Decision Making (MADM) techniques to filter the most promising versions.

OpenCL code generation. For each version provided in the previous stage, we generate the *OpenCL* equivalent code that performs the configuration of the kernel parameters, executes the kernel, and performs the cleanup process. An example of code generation from an independent for-loop to an equivalent kernel is shown in Figure 1.

III.3 Static software partitioning and scheduling techniques

After profiling both hardware and software, it is necessary to schedule the kernels marked in an application to be run into specific devices. Currently, a static, offline scheduling algorithm has been implemented [8].

This algorithm is based on four key aspects: kernels, input/output size, devices and transfer rates. Each pair of kernel and input/output size takes a certain time to run. Also related to the data size is the transfer rate. Lastly, each device has its own strengths and limitations, and as such their performance will vary from kernel to kernel.

With this, we represent the different possible schedules as nodes in a tree, where the root node is an empty schedule, and the leaf nodes are full schedules where all kernels have been assigned to a device. With this, we take the schedule that takes the least time. It is possible to add feedback on energy consuption and, by introducing weights for each measure, prepare the model so that the user can configure it as needed.

IV. Conclusions and future work

In this work, we have developed a unified heterogeneous model that allows to describe heterogeneous parallel platforms composed of different computational devices. It also allows to orchestrate different kernels into said devices to be executed in parallel.

We also have developed several tools that automatize the hardware description, code annotation and scheduling optimization. The last two have been tested in several existing benchmarks. In the first case, we compared our automatic kernel detection against an existing OpenMP version of the tested benchmarks. We had a 95% success, with 3% of the misses being false negatives due to manually introduced constraints. As for the static scheduling algorithm, our predicted schedules are usually in the top 5%.

We will expand the work done on the static scheduling by introducing dynamic scheduling techniques. In order to test this techniques, we will integrate our work with a parallel programming framework, such as FastFlow [7].

Acknowledgments

The work presented in this paper has been partially supported by EU under the COST programme Action IC1305, 'Network for Sustainable Ultrascale Computing (NESUS)'

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007- 2013) under grant agreement n. 609666 and by the Spanish Ministry of Eco- nomics and Competitiveness under the grant TIN2013-41350-P.

References

[1] The Khronos Group, The OpenCL Specification, http://www.khronos.org/ (Sep. 2014).

[2] Wienke, Sandra et al., OpenACC: First Experiences with Real-world Applications, Proceedings of the 18th International Conference on Parallel Processing, Euro-Par'12.

[3] Bertolli, Carlo et al., Coordinating GPU Threads for OpenMP 4.0 in LLVM, Proceedings of the 2014 LLVM Compiler Infrastructure in HPC, LLVM-HPC '14.

[4] B. Kolpackov, C++11 generalized attributes, apr, 2012.

[5] Nugteren, Cedric and Corporaal, Henk, Bones: An Automatic Skeleton-Based C-to-CUDA Compiler for GPUs, ACM Trans. Archit. Code Optim., January 2015.

[6] Johannes Doerfert and Clemens Hammacher and Kevin Streit and Sebastian Hack, SPolly: Speculative Optimizations in the Polyhedral Model, jan, 2013.

[7] Danelutto, Marco and Torquati, Massimo, Structured Parallel Programming with "core" FastFlow, 2015.

[8] J. Daniel Garcia et al., Static partitioning and mapping of kernel-based applications over modern heterogeneous architectures, Simulation Modelling Practice and Theory, 2015.

[9] Sanchez, Luis Miguel et al., A Comparative Study and Evaluation of Parallel Programming Models for Shared-Memory Parallel Architectures, New Generation Computing, 2013.

[10] Marco Danelutto et al., Introducing Parallelism by using REPARA C++11 Attributes, 2016, ACCEPTED, PENDING PUBLICATION.

[11] Rafael Sotomayor et al., Automatic CPU/GPU Generation of Multi-versioned OpenCL Kernels for C++ Scientific Applications, High-level Parallel Programming and Applications, 2015, ACCEPTED, PENDING PUBLICATION.

A Framework for Knowledge Management using Complex Networks Methods

Alex Becheru

University of Craiova, Romania
becheru@gmail.com

Abstract

In a world where complexity is constantly increasing due to the technological advancement, large scale of data available and increased interaction between various phenomena there was a need for a field of study to model and understand such complex systems. One such field of research is called Complex Networks Analysis *(CNA) or Network Science. The heart of this research field leverages on Graph Theory and Computer Science. In this paper we shall briefly present a common framework for knowledge management using CNA methods. The power of the framework shall be proven by extracting knowledge from various heterogeneous domains like: Tourism, E-learning, Freight Transportation , and Organisational Analysis.*

Keywords Complex Networks, Knowledge Management, Graph Theory, Tourism, E-Learning, Organisational Analysis

I. Introduction

Our current understanding of the surrounding world shows us that nature is formed out of complex interconnecting systems. Networks created by these systems support phenomena that are far from being deterministic trough traditional methods. Each element influences the network, while the network puts its mark on every element. Now we can say with certainty that the *butterfly effect* imagined by Edward Lorenz is truly possible.

The complexity of real world networks comes from the modelling and evaluation of overlapping and interdependent phenomena, that are neither purely regular nor purely random. Also complexity may come with the sheer size of the network itself.

In order to understand complex interconnected systems a new field of research emerged – *Network Science* (NS) or *Complex Networks Analysis* (CNA). The heart of this new research field leverages on *Graph Theory* and *Computer Science*. NS investigates non-trivial features of graph problems that usually are not addressed by lattice theory or random graphs.The understanding of such non-trivial features is of high interest, as they frequently occur in real world problems.

Our aim is to develop a common framework for knowledge management using CNA methods. Thus we can extract information from various heterogeneous domains. The development of the frameworks implies determining and adding scientific contributions to the following research fields:

1. data acquisition
2. data preprocessing
3. data storage
4. complex network creation
5. methods of analysis
6. proof of concept in various domains

In order develop and test the common framework we chose to try and resolve real world problems from the following domains: Tourism, E-learning, Freight Transportation , and Organisational Analysis. The domains just enumerated are diverse and should give

a sufficient generality to the framework to be called a common framework.

The paper is structured as follows. The next sections focuses on background information and related work. The third section briefly describes the framework. The last section presents the current status of our research and future work.

II. Background and Related Work

Two important papers stand as the building blocks of *Complex Networks Analysis*. Paul Erdös and Alfréd Rényi wrote about random graphs in 1959 [1]. In 1973, Mark Granovetter discovered the "strength of weak ties" [2]. A graph usually consists of a number of subgraphs, nodes inside these subgraphs are tightly connected among them and loosely (weak ties) connected with other subgraphs. One may think that those weak ties are not relevant, but without their presence the graph of subgraphs would not exist. CNA emerged at the beginning of the 1990's as a result of the progress in applied computational sciences. But the most important factor was the access to data describing real world networks. The emergence of the World Wide Web, as well as the explosion of the interest in detailed mapping across many sciences, especially in biology and economics, opened a multitude of research paths.

Stanley Milgram [3] and Watts et al. [4] discovered and defined the *small world phenomenon*. Otherwise called *six degrees of separation*, this phenomenon is found in many real world large networks, where contrary to the size of the network the average path length between two nodes has a very low value (6 or less). Barabasi et al. [5] showed that real world networks have a *scale free degree distribution*, also called Pareto or Zipf distribution. This means that very few nodes have high *Degree* while the majority has almost the same very low *Degree*. An explanation for the appearance of the *scale free distribution of degree* is the *preferential attachment* [6] of nodes, a node has a greater probability to be linked with nodes that have high *Degree* than with nodes with low *Degree*. Another phenomenon that is of great interest for NS is *Homophily*, described as the tendency of individuals (nodes in our case) to associate and bond with similar others [7].

CNA can be used in many application domains. For example, internet companies like *Google* and *Facebook* are practically built on complex networks. In medicine, the spread of diseases is now studied with the help of CNA [8]. Security forces map the networks of acquaintances of wanted individuals, maps which could lead to alternative ways to reach them. The famous Saddam Hussein was captured using methods from NS [9]. Large oil companies use a branch of CNA known as *Organisational Network Analysis* to enhance the flow of information exchange within the companies [10]. CNA was even used to determine the best tennis players respective to different scenarios [11], e.g. best tennis player on the grass surface.

III. Framework

The first aspect in the design of the framework should be it's universality. We are looking to develop the framework such that in can be used and easily adapted for diverse use cases no matter the domain of the problem. But we want also to put some restrictions in order to ensure the quality of the results. Therefor some of the guidelines shall be mandatory but the majority are optional. The guidelines are extracted from our experience in the already mentioned domains.

The main restriction in using the framework is modelling the domain of interest into a graph. Although this might seem a considerable restriction keep in mind that it is very easy to abstract the real work into objects and relations among the objects. By object we understand phenomenon/ living thing /material object that can be described as a sum of states at a certain point in time.

The main feature of framework is the power to analyse the resulted graph/graphs from various granularity levels:

1. from the perspective of the entire graph/network
   (a) evolution in time, with possibility to predict further evolution.
   (b) the level of resilience of the graph, with indications on how to increase or reduce the resilience.
   (c) the ability of the graph to support information/knowledge exchange between the ob-

jects, with indications on how to improve information/knowledge exchange.

(d) detection of graph particularities, with possibility of detecting similar graphs based on those particularities.

(e) social phenomenon detection, e.g. small world.

(f) knowledge extraction based on visualisation.

2. from the perspective of communities inside the graph

   (a) community detection using traditional artificial intelligence algorithms, complex networks algorithms or hybrid algorithms

   (b) the ability of the graph to support information/knowledge exchange between communities, with indications on how to improve information/knowledge exchange.

3. from the objects's perspective

   (a) determining the objects with high centrality, with the option of developing/optimising centrality measures for particular domains.

   (b) identification of particular objects.

   (c) hybrid object recommendation system based on CNA metrics and other scientific methods, e.g. natural language processing.

Before each particular use of the framework the user needs to determine the objects and their defining states. Objects can be represented strictly conforming to a pattern, where the domain is well defined, or in a schema-less mode, especially useful when the domain of research is entirely regulated. E.g the tourism domain is not entirely regulated, a king size bed may also be known as a sultan size bed due to cultural differences. Relations need to be thoroughly defined.

Regarding data acquisition a multitude of tools can be used or developed depending on the on the source, e.g. web crawlers. But before a source of data is selected it is mandatory to check for its quality, garbage in garbage out. If the data is extracted from multiple sources it is mandatory to understand and consider similarities and dissimilarities between the sources in the data acquisition process, e.g. multiple definitions of the same thing need to be avoided. As much as possible include also temporal data, thus evolutionary analysis can be conducted.

Data prepossessing is not mandatory if the source of data is clean, e.g. data from U.S. patent bureau, otherwise we need to clean the data. The amount of preprocessing is research but at least duplicate, unreadable data and data that gives no added value should be eliminated. Detecting outliers and eliminating them could have a significant improvement in the end results. Natural language processing of texts can be usefull in eliminating parts of speech or stop words that represent no valuable data. Twitter tag expansion can also be valuable, as it brings relevant keywords in the analysis, e.g. from "#thebestcity" becomes "the best city". By using RDF resources like DBpedia[1] we can enrich the knowledge base.

Data can be stored in many forms and in many systems. We recommend using a database system. The choice depends on how much "joggle" with the data is needed. For very ambitious "joggle" we recommend NoSQL graph data bases, like Ne04j[2], as jumping and combining relations is very easy. If the objects that shall be analysed are schema-less and the aggregation structure needs no change then NoSQL aggregate-oriented databases are the best choice, e.g. MongoDB[3]. Otherwise traditional SQL should be used.

The creation of the complex network/networks is possibly the most important step as the way the objects are put together has significant on knowledge extraction. A "mud-ball" graph consisting all objects and all relations might give some information but usually that is not true. Thus a series of trial and-error construction of complex networks have to be attempted. A good knowledge of the research domain is needed. Usually a graph is created for each relations defined at the beginning, an only after these are analysed multigraphs[4] are created and analysed. Based on the definitions of the relations between objects the decision to create directed graphs or undirected graphs is made. We recommend

[1] http://wiki.dbpedia.org/

[2] http://neo4j.com/

[3] https://www.mongodb.org/

[4] a multigraph is a graph which is permitted to have parallel edges

using both types, as the directed graphs can better pin point objects with high centrality, while undirected graphs reveal structural objects (those objects that keep the graph together but don't have high centrality). We also recommend using weighted graphs as they are more accurate in the abstraction of a research domain.

The methods of analysis are also research domain dependent. A major part of our research focuses on developing and optimising methods / techniques / ontologies both at a general level for specific domains. Among the algorithms used by us we mention: centrality algorithms, graph topological detection algorithms (e.g. clique detection), community detection algorithms, textual complexity algorithms. Besides algorithms we also use ontologies to define states and complex networks types. We also employed statistical methods calculating correlations.

IV. Current Status and Future Work

Regarding Freight Transportation we were able to develop a system for freight brokering using ICNET negotiation algorithm and based on an ontology developed by us for an exhaustive list of freight types. Next we plan to conceive a recommender system to recommend transport companies based on their previous contracts with freight owners.

Based on touristic reviews extracted from the Internet site *AmFostAcolo.ro* we were able to analyse the graph of information exchange and extract knowledge on information exchange and network expansion. Another recommeder system is in development to suggests tourist locations based on community preferences.

Based on messages exchange by students in an e-learning environment we were able to tie the textual complexity of students to their grades. In the future we plan to conceive a grade prediction system based on students textual complexity.

On the Organisational Analysis we've proven that the SCRUM agile development method support better information exchange and innovation than the classical hierarchical scheme. Also we analysed the information exchange in a small academic organisation and we were able to identify bottlenecks and suggest improvements. For the future we plan to analyse other agile development methods.

Acknowledgment

We acknowledge support from COST Action IC1305 NETWORK FOR SUSTAINABLE ULTRASCALE COMPUTING (NESUS).

References

[1] Erdős, P., Rényi, A.: On random graphs. Publicationes Mathematicae Debrecen **6** (1959) 290–297.

[2] Granovetter, M.: The strength of weak ties. American journal of sociology **78** (1973) 1

[3] Milgram, S.: The small world problem. Psychology today **2** (1967) 60–67

[4] Watts, D.J., Strogatz, S.H.: Collective dynamics of 'small-world' networks. nature **393** (1998) 440–442

[5] Barabási, A.L., et al.: Scale-free networks: a decade and beyond. science **325** (2009) 412

[6] Newman, M.E.: Clustering and preferential attachment in growing networks. Physical Review E **64** (2001) 025102

[7] McPherson, M., Smith-Lovin, L., Cook, J.M.: Birds of a feather: Homophily in social networks. Annual review of sociology (2001) 415–444

[8] Barabási, A.L., Gulbahce, N., Loscalzo, J.: Network medicine: a network-based approach to human disease. Nature Reviews Genetics **12** (2011) 56–68

[9] Wilson, C.: Searching for saddam: Why social network analysis hasn't led us to osama bin laden. Slate, February **26** (2010)

[10] Cross, R.L., Singer, J., Colella, S., Thomas, R.J., Silverstone, Y.: The organizational network fieldbook: Best practices, techniques and exercises to drive organizational innovation and performance. John Wiley & Sons

[11] Radicchi, F.: Who is the best player ever? a complex network analysis of the history of professional tennis. PloS one **6** (2011) e17249

A generic I/O architecture for data-intensive applications based on in-memory distributed cache

Francisco Rodrigo Duro, Javier Garcia Blas, Jesus Carretero

University Carlos III, Spain

frodrigo@arcos.inf.uc3m.es, fjblas@arcos.inf.uc3m.es, jesus.carretero@uc3m.es

Abstract

The evolution in scientific computing towards data-intensive applications and the increase of heterogeneity in the computing resources, are exposing new challenges in the I/O layer requirements. We propose a generic I/O architecture for data-intensive applications based on in-memory distributed caching. This solution leverages the evolution of network capacities and the price drop in memory to improve I/O performance for I/O-bounded applications adaptable to existing high-performance scenarios. We have showed the potential improvements of our proposed solution applied on three scenarios: clusters, cloud, and mobile cloud computing environments.

Keywords Ultrascale systems, NESUS, generic I/O architecture, distributed I/O, data-intensive applications, workflow, cloud computing, in-memory storage

I. Introduction

In the last decade, the scientific computing scenario is greatly evolving in two main areas. First, the focus in scientific computation is changing from CPU-intensive jobs like large scale simulations or complex mathematical applications towards a data-intensive approach. This new paradigm greatly affects the underlying architecture requirements, slowly vanishing the classical CPU bottleneck and exposing bottlenecks in current I/O systems.

Second, the evolution in computing technologies and science funding restrictions are changing the computing resources available in the scientific community. Cloud computing offers a virtually limit-less pool of computing resources in a pay-per-use approach, but most of the research institutions still have access to clusters or supercomputing resources. This heterogeneity in the nature of the available resources leads to new demands in the flexibility of the I/O layer, requiring a more generic approach.

Current trends in bandwidth and latency improvements in high-speed networks in conjunction with the RAM price drop and the near advent of non-volatile memory, present a bright opportunity for improving I/O performance through the use of in-memory I/O solutions. The possibility of using spare memory in compute nodes, and the performance offered by state-of-the-art network technologies, can lead to distributed in-memory solutions where the number of I/O nodes deployed can be flexibly adjusted depending on the performance required by each application, or even by each different experiment. This flexibility in the number of I/O nodes can tackle the I/O bottleneck present in current parallel file systems using fixed configurations.

We propose a new generic I/O architecture for data intensive applications based on in-memory distributed cache targeting both the I/O bottlenecks and the heterogeneity of computing resources. The architecture design is guided by four main objective: flexibility, scalability, performance, and ease of deployment. In an effort to demonstrate the flexibility and capabilities of our solution, we present three different successful scenarios where our proposed solution has been applied: a workflow engine running on a cluster infrastructure,

a data mining framework running on a cloud infrastructure, and a mobile cloud computing scenario.

II. Thesis idea

The main goal of this thesis is to propose a novel generic I/O architecture design for an in-memory storage system based on distributed caching [2]. As shown in Figure 1, the front-end of the architecture is a user-level library and the back-end consists of Memcached servers enhanced with persistence and other performance tweaks. The memory distributed among the server nodes is offered to the user as a unified storage space that can be accessed through the use of easy-to-use APIs: POSIX-like, MPI-IO, and put/get.

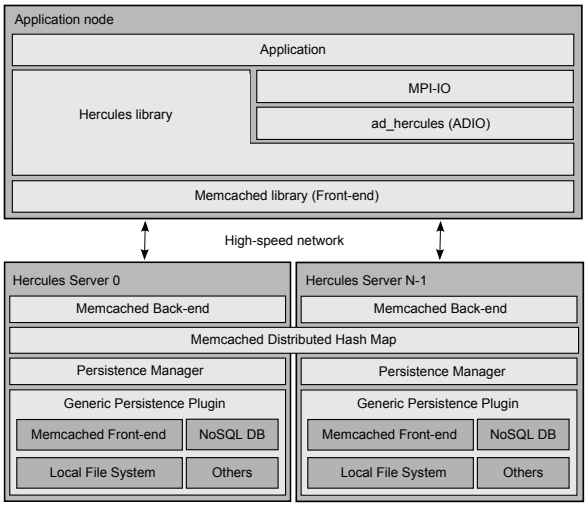


Figure 1: Current version of our proposed generic I/O architecture, namely Hercules [3]

Internally, the I/O nodes behave as stateless servers composing a distributed key-value store where data and metadata are completely distributed. The unified memory space is used as a virtual device. In every key-value pair stored, the key acts as the block ID, and the value represents the block contents. Thanks to this approach, every block ID can be calculated instead of being stored, simplifying the algorithms for data placement and retrieval.

The architecture design targets four objectives: scalability, flexibility, easy deployment, and performance. **Scalability** is achieved by fully distributing data and metadata among all the available I/O nodes, avoiding any possible bottleneck derived from centralized services. Data placement is fully calculated client-side by a hashing algorithm, minimizing storage and communications for data retrieval.

Flexibility is tackled in both client and server sides. On the front-end, the APIs offered to the user are widely used in existing applications, facilitating the use of existing applications with minimum changes. The layered design simplifies the addition of new APIs and persistence plugins. On the back-end, the servers are completely state-less, permitting the deployment of any number of I/O nodes depending on the characteristics of the infrastructure, even on different levels of the I/O hierarchy if necessary. The only information needed by the clients are the IP addresses of the I/O nodes. The servers, on the other end, do not need any information about other servers running on the same hierarchy level.

Ease of deployment is especially important in order to design an architecture as generic as possible. Both the user-level library and the I/O nodes can be deployed on any Linux system in user mode, without requiring any special privileges.

Performance-wise, our solution supports parallel I/O accesses to enhance applications throughput. Each I/O node available can be accessed independently, multiplying the maximum throughput peak performance. Furthermore, the multi-threading implementation increases the level of parallelism for serving requests.

Scalability, flexibility, and easy deployment work together to adjust the system for the best possible performance required by each situation. The user can deploy as many I/O nodes as necessary depending on the throughput requirements of each application, or even for different runs of the same application.

III. Application scenarios

This work presents an I/O architecture design aiming to be generic. In order to demonstrate the capabilities of our I/O solution for adapting to different infrastructures, we present three different scenarios where our proposed architecture has been successfully applied.

III.1 Workflow engine over cluster infrastructure

The first scenario consists of deploying our in-memory architecture as an I/O accelerator for the Swift/T workflow engine [3] in collaboration with Argonne National Laboratory (USA), developer of the Swift/T workflow engine and runtime.

This scenario is motivated by the I/O contention suffered by classic parallel file systems available in HPC infrastructures, in applications with a high number of worker nodes accessing concurrently to the shared file system. Classic parallel file systems are deployed in a static configuration, thus number of I/O nodes available for the applications can not be dynamically configured. The aggregated bandwidth of the I/O nodes is shared among all the workers accessing concurrently, which is translated in high I/O contention during peak I/O loads.

As shown in Figure 2 our solution (labeled as Hercules) is deployed as an alternative storage space for temporary files in the workflow life-cycle. Most of the files generated by each task of the workflow are consumed by other task. Deploying one Hercules I/O node sharing resources with each worker node, we target two main objectives.

First, the number of I/O nodes scales with the number of worker nodes available. This is translated into a better scalability in the maximum available bandwidth available for I/O operations, especially when compared with the default shared file system.

Second, the possibility of exposing and exploiting data locality. Our storage space is allocated using spare memory of the worker nodes. Offering information about data placement to the scheduler can expose data locality. Co-locating tasks and data in the same node, data locality can be exploited. Additionally, the data placement policy is also optimized for data locality purposes. Another advantage offered by this approach is the isolation from the shared file system noise obtained through the deployment of I/O nodes dedicated to one specific application.

Evaluated against GPFS, our solution scales better when the number of available worker nodes is increased. In the most extreme cases, our proposed solution was able of converting an I/O bounded problem (where the total execution time increased when scaling the worker nodes as a result of I/O contention) into a CPU-bounded application (where the execution time always decreased while increasing the number of worker nodes available).

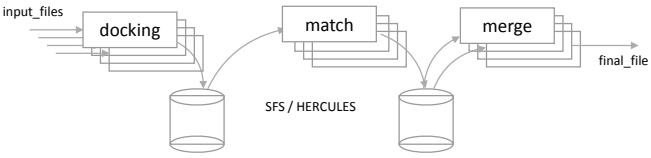


Figure 2: Example of workflow. Temporary files can be stored in the default shared file system or in Hercules for improving maximum throughput and data locality [3]

III.2 Data mining framework over cloud infrastructure

The objective targeted by this second scenario is shared with the previous one, aiming to accelerate the I/O accesses over temporary files in a data mining workflow through the use of in-memory storage. The main difference is the infrastructure where the workers are deployed, using cloud resources instead of a cluster. The idea behind this scenario is a collaboration with the DIMES group at University of Calabria (Italy), developers of the Data Mining Cloud Framework (DMCF) [5].

This collaboration shows the potential performance of our proposed solution deployed over the Microsoft Azure infrastructure and evaluated against the Azure Storage, the default storage provided by Microsoft. The collaboration has followed with the full integration of DMCF and Hercules, and it is still active for exposing and exploiting data locality.

In order to show the flexibility of our solution, additionally, it has been deployed over another cloud provider, Amazon AWS in this case. Hercules was deployed on Amazon EC2 instances and evaluated against S3 using S3FS and I/O performance was evaluated through specifically designed micro-benchmarks, with successful results [4].

III.3 Mobile cloud computing scenario

In 2013 we developed CoSMiC, a version of our proposed architecture especially adapted for the emerging Mobile Cloud Computing field. Leveraging the ease of deployment and flexibility of our architecture, the objective of this work was improving the storage capabilities of mobile devices, especially on public places and limited connectivity scenarios.

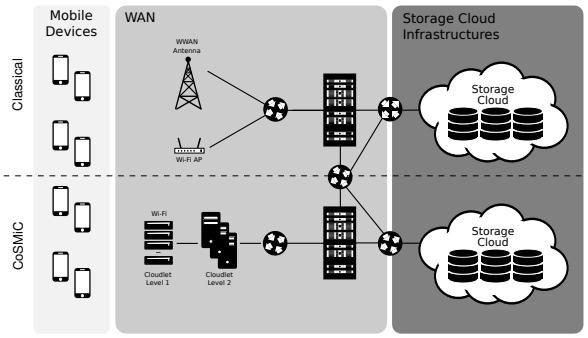


Figure 3: Application of our generic architecture into a Mobile Cloud Computing scenario, based on the cloudlet concept [1]

As shown in Figure 3 our solution presents an alternative data path for mobile device users based on the cloudlet concept. The advantage of this approach is a result of the proximity of the storage in contrast with the classic cloud approach. Due to this proximity, mobile device storage is expanded, latency is significantly reduced, and energy-efficiency is improved through the use of Wi-Fi instead of 3G/HSDPA/4G. MNOs are also benefited, relieving the pressure in their WAN infrastructures by caching popular contents in public places, especially on highly crowded scenarios, leading to a win-win situation for every participant.

IV. Conclusions and future work

This Thesis presents a new generic I/O architecture for data intensive applications based on in-memory distributed cache. Our solution tackles the I/O system bottlenecks exposed by new trends of scientific computing while tends to be generic in order to be usable in legacy HPC infrastructures and other resources gaining popularity such as public clouds.

The flexibility and performance capabilities of our proposed solution are presented as four heterogeneous scenarios where our solution has been successfully applied, supported by publications on prestigious international journals, conferences, and workshops.

Acknowledgment

This work is partially supported by EU under the COST Program Action IC1305: Network for Sustainable Ultrascale Computing (NESUS). This work is partially supported by the grant TIN2013-41350-P, *Scalable Data Management Techniques for High-End Computing Systems* from the Spanish Ministry of Economy and Competitiveness.

References

[1] Francisco Rodrigo Duro, Francisco Javier García Blas, Daniel Higuero, Oscar Pérez, and Jesús Carretero. CoSMiC: A hierarchical cloudlet-based storage architecture for mobile clouds. *Simulation Modelling Practice and Theory*, 50:3–19, 2015.

[2] Francisco Rodrigo Duro, Javier Garcia Blas, and Jesus Carretero. A Hierarchical parallel storage system based on distributed memory for large scale systems. EuroMPI '13, pages 139–140, New York, NY, USA, 2013. ACM.

[3] Francisco Rodrigo Duro, Javier Garcia Blas, Florin Isaila, Justin Wozniak, Jesus Carretero, and Rob Ross. Exploiting data locality in Swift/T workflows using Hercules. NESUS 2014, pages 71–76, Porto, Portugal, 2014. UC3M.

[4] Francisco Rodrigo Duro, Javier Garcia-Blas, Florin Isaila, and Jesus Carretero. Experimental evaluation of a flexible I/O architecture for accelerating Workflow engines in cloud environments. DISCS '15, pages 6:1–6:8, New York, NY, USA, 2015. ACM.

[5] Francisco Rodrigo Duro, Fabrizio Marozzo, Javier Garcia Blas, Jesus Carretero, Domenico Talia, and Paolo Trunfio. Evaluating data caching techniques in DMCF workflows using Hercules. NESUS 2015, pages 95–106, Krakow, Poland, 2015.

Machine Learning Methods Applied to Biometrics

Cristina M. Noaica

University of Bucharest, Bucharest, Romania
noaica@irisbiometrics.org

Abstract

Biometrics is a challenging field which uses physiological and behavioral characteristics of persons in order to establish their identities. Biometrics research requires the fusion of several other fields, fields that are in a continuous development. Among these fields we find image processing, pattern recognition and machine learning. There are many research oportunities in this field, some of the most recent ones being cross-sensor comparisons, liveness detection (in iris recognition), behavioral biometrics and mobile biometrics. My PhD thesis will contribute by applying Machine Learning methods at least for some of the enumerated research oportunities.

Keywords Biometrics, Pattern Recognition, Image Processing, Machine Learning

I. Introduction

Biometrics is a field with continuously increasing areas of application, such as financial services, mobile device access or border control. Shortly, biometrics is represented by automated methods of identifying persons, based on their physiological and behavioral traits. Some of the physiological traits are iris, face, fingerprint, and vein. Examples of behavioral traits are signature, gait and keystroke dinamics. In the past years there have been made many advancements in this field, especially when it comes to biometrics in controlled environment, where factors such as lighting are held under control. Lately, the researchers attention started to switch to unrestricted environments, where there is no human agent present to supervise the proper usage of biometric systems.

II. Related Work

In my opinion, the most impressive research results have been reported in the past two years. For instance, in 2014 Yaniv Taigman et. al. published a paper [3] on face recognition in which they presented a method of verifying identities with an accuracy up to 97.35%, really close to the human performance, which is 97.5%. As far as I know, these are the best results published in biometrics scientific literature on face recognition, up to the moment. Other important results have been presented by Marios Savvides, from Carnegie Mellon University's CyLab Biometrics Center. Savvides and his colleagues have developed an iris recognition system [2] that is able to establish the identity of individuals from approximately 12 meters. The biometric system is designed especially for police cars, helping to establish the identity of the car drivers that are pulled over, by acquiring images of their eyes from the side-view mirrors.

Getting closer to the area of my recent work, in [1] the authors proposed an iris segmentation algorithm for CASIA-Iris V4 Lamp database, algorithm that acquires a 95.63% accuracy in detecting the pupillary boundary and a 90.52 overall segmentation accuracy (i.e. determining both the pupillary and limbus boundaries).

III. Thesis Idea

The need to correctly establish the identity of individuals is constatly increasing nowadays, mainly due to technological progress. This is why biometrics is a still

flourishing domain, enjoying the attention of many researchers. The following directions are some of the ones that captured my attention as well:

- Developing new image segmentation procedures;
- Signal processing;
- Machine Learning algorithms;
- Solving problems that characterize the biometric systems which have an increasing number of users. When the number of users is increased, the chance of occurring false accept or false reject cases is high.
- Also, one of the interests in biometrics research is evaluating and analyzing the weaknesses of biometric systems. In other words, it is important to correctly identify any forgery attempts.

My work is primarily focused on applying Machine Learning methods in biometrics. For instance, one of my previous work consisted in applying a modified unsupervised neural network, ART 1 (Adaptive Resonance Theory of type 1), in iris recognition. The neural network is modified in order to classify input patterns (iris codes) given in a random order. This new characteristic of the network allows an easy identification of any person who attempts to pass a biometric verification by using a fake identity. False Acceptance Rate and False Rejection Rate, two performance indicators in biometrics, have obtained null values in all of the tests performed with the modified version of ART.

IV. Conclusions and Future Work

My research was focused so far on iris recognition, wehther it was about testing concepts such as biometric menagerie, testing several neural networks, such as PNN (Probabilistic Neural Network) or ART 1, or performing cross-sensor comparison. During the remaining time of my PhD studies I intend to extend my research to other biometric traits as well, but, for the moment, I work on developing a segmentation algorithm for iris images, algorithm that will allow me to approach other current research topics in iris recognition, such as liveness detection.

Acknowledgment

I would like to thank NESUS for supporting this article.

References

[1] Cheng, Guojun, Wenming Yang, Dongping Zhang, and Qingmin Liao. *A Fast and Accurate Iris Segmentation Approach.* In Image and Graphics, pp. 53-63. Springer International Publishing, 2015.

[2] Iris recognition of driver 40 feet away through side view mirror `http://www.sciencedirect.com/science/article/pii/S0969476515300679`

[3] Taigman, Yaniv, Ming Yang, Marc'Aurelio Ranzato, and Lars Wolf. *Deepface: Closing the gap to human-level performance in face verification.* In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, pp. 1701-1708. IEEE, 2014.

Work in progress about enhancing the programmability and energy efficiency of storage in HPC and cloud environments

PhD Student
Pablo Llopis
University Carlos III, Spain
pllopis@arcos.inf.uc3m.es

PhD Advisor
Javier Garcia Blas
University Carlos III, Spain
fjblas@arcos.inf.uc3m.es

PhD Advisor
Florin Isaila
University Carlos III, Spain
florin@arcos.inf.uc3m.es

Abstract

We present the work in progress for the PhD thesis titled "Enhancing the programmability and energy efficiency of storage in HPC and cloud environments". In this thesis, we focus on studying and optimizing data movement across different layers of the operating system's I/O stack. We study the power consumption during I/O-intensive workloads using sophisticated software and hardware instrumentation, collecting time series data from internal ATX power lines that feed every system component, and several run-time operating system metrics. Data exploration and data analysis reveal for each I/O access pattern various power and performance regimes. These regimes show how power is used by the system as data moved through the I/O stack. We use this knowledge to build I/O power models that are able to predict power consumption for different I/O workloads, and optimize the CPU device driver that manage performance states to obtain great power savings (over 30%). Finally, we develop new mechanisms and abstractions that allow co-located virtual machines to share data with each other more efficiently. Our virtualized data sharing solution reduces data movement among virtual domains, leading to energy savings I/O performance improvements.

Keywords NESUS, PhD Symposium, Energy Efficiency, I/O, Storage, Data movement, HPC, Cloud

I. Introduction

Modern scientific discoveries have been driven by an insatiable demand for high performance computing. However, as we progress on the road to Exascale systems, energy consumption becomes a primary obstacle in the design and maintenance of HPC facilities. A simple extrapolation shows that an Exascale platform based on the most energy efficient hardware currently available in the Green500 would consume 120 MW. However, the desirable goal has been set by the DOE to 20 MW [2]Actually, hardware vendors are already trying to provide more energy-efficient parts and software developers are gradually increasing power-awareness in the current software stack, from applications to operating systems.

Data movement has been identified as an extremely important challenge among many others on the way towards the Exascale computing [2]. As the power cost of computation decreases, the cost of data movement increasingly becomes a more relevant issue [1]. The low performance of the I/O operations continues to present a formidable obstacle to reaching Exascale computing in the future large-scale systems especially in I/O-intensive scientific domains and simulations. This issue triggers a special interest in optimizing storage systems in data centers, and motivates the need for more research to improve the energy efficiency of storage technologies. Therefore, a first step to develop I/O optimizations is to further understand how energy is consumed in the complete I/O stack.

We focus on gaining a clear understanding of how

power is used during I/O operations across the software stack, and using this knowledge to provide solutions that optimize energy utilization and I/O performance.

II. Thesis overview

The purpose of this section is to present an overview that provides an holistic description of the work introduced in this thesis. The contributions constitute work that studies and optimizes data movement across different levels of the operating system's I/O stack. More precisely, we propose contributions to the understanding and optimization of I/O power consumption that span from virtualized environments, through the operating system's I/O stack, and including low-level CPU device drivers, as depicted in Figure 1. Our contributions show that through the understanding of the different operating system layers and their interaction, it is possible to achieve coordinations that optimize the energy consumption and increase performance of I/O workloads.

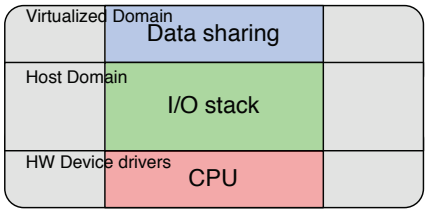


Figure 1: The contributions of this thesis span multiple levels of the software I/O stack.

The thesis starts with the goal of better understanding how power is used in the operating system's I/O stack. We perform a detailed study of power and energy usage across all across all system components during various I/O-intensive workloads [5]. To achieve an exhaustive examination, our work combines software and hardware-based instrumentation in order to study I/O data movement through exploratory data analysis. This data-driven process reveals detailed knowledge about how the system shifts between different power and performance regimes (depicted for a sequential file write in Figure 2), and which layers and algorithms of the I/O stack are responsible. As a result of our analysis and characterization, we provide I/O power models that are able to predict power consumption of I/O workloads that perform various access patterns. Figure 3 shows three workloads that do different combinations of random read/write, sequential read/write, strided reads combined re-reads (resulting in various page cache hit ratios). Our models are able to predict energy consumption with a normalized standard error under 5%.

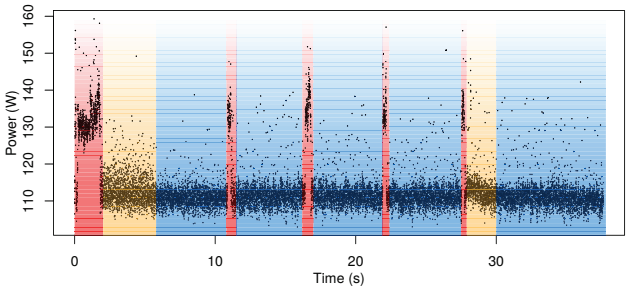


Figure 2: Power regimes during a sequential write of a 4 GiB file. Colors correspond to different regimes. Regimes correlate with speeds at which data is moved through the I/O stack, either put into the page cache or written to disk.

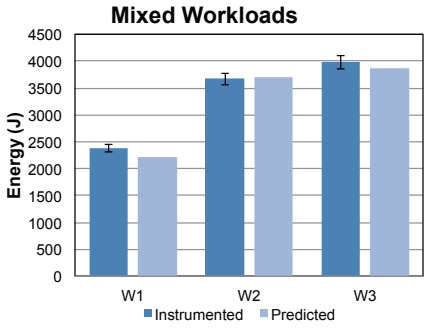


Figure 3: Comparison of measured energy with model predicted values for three workloads that mix reads and writes using different I/O patterns.

Our work continues into the hypervisor-based virtualization layer. We focus on optimizing data sharing between co-hosted virtual machines. In our work we refer to this as intra-domain data sharing, which mainly differs from existing solutions in the way the data moves across the software I/O stack. We develop virtualized data sharing (VIDAS) in order to

reduce data movement across virtual environments [6, 4]. VIDAS proposes new abstractions and mechanisms to more efficiently coordinate storage I/O across virtual domains, reduce data movement by creating intra-domain shared access spaces, relax POSIX consistency to allow flexible data write and update policies, and expose data locality. We argue that these abstractions and mechanisms can be used to build an efficient para-virtualized file system, and demonstrate reduced energy consumption and increased performance for various collective I/O access patterns. Figure 4 depicts the results for collectively writing and reading data to/from a 512MB object/file. The domains are accessing non-overlappingly interleaved strided vectors of 2MB blocks. Our solution uses a shared buffer space between domains/virtual machines, which reduces data movement. On the other hand, ROMIO collective operations copy the data into collective buffers before sending them to disks, which makes performance drop dramatically when increasing the number of virtual machines.

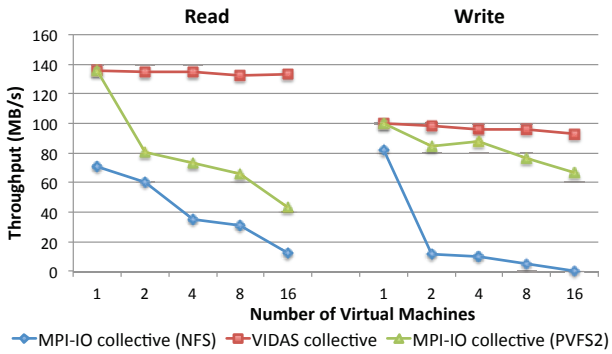


Figure 4: Comparison of VIDAS collective I/O and ROMIO collective I/O

Finally, we focus on the CPU, motivated by the fact that it is one of the most power-hungry components in a system. We examine the behavior of the CPU under I/O intensive workloads, and make two observations. First, we learn that in spite of being the most power-proportional component, the CPU does not shift performance states based on the I/O power and performance regimes revealed during our analysis of the operating system's I/O stack. Second, we note that there is a thermal imbalance that causes the CPU behave like a heterogeneous system. We develop kernel modules that use internal CPU mechanisms for thermal sensing and performance state selection, and demonstrate that we are able decrease energy consumption for I/O workloads for each of these two cases. Motivated by our first observation, we develop I/O-aware performance state selection. We are able to detecting I/O regimes and shift power states accordingly in order to lower CPU power usage without reducing performance. By adaptively setting performance states based on I/O performance regimes, we are able to reduce CPU energy consumption during write I/O by an average of 33%. Figure 5 depicts the difference between our solution and the Linux default CPU p-state driver in average CPU consumption, temperature (3.5°*C* improvement), and runtime (9% improvement).

Our second observation motivates us to develop thermal and I/O-aware thread placement, where computationally intensive and I/O intensive workload threads are placed in a thermal-aware fashion to optimize CPU power consumption. We are able to obtain up to 2.9% less energy consumption just by placing computation threads on the coldest CPU cores.

In conclusion, work shows that data movement within the host can be optimized to obtain performance and power consumption improvements. We not only analyze I/O power consumption in detail, but also demonstrate that data movement and I/O optimizations can be achieved on multiple layers of the system, spanning from the CPU device drivers, to virtual environments.

Acknowledgments

We would like to thank the community participating in this NESUS Action for making this PhD Symposium possible.

III. Related work

Our work is related to large body of research, but this Section will only highlight a few works. VIDAS builds upon and extends the paravirtualization concepts introduced first introduced Xen [8] to improve I/O performance in virtualized environments. Manousakis

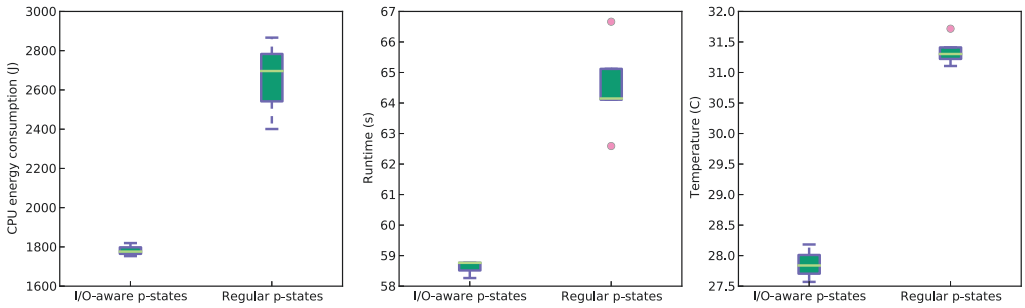


Figure 5: I/O-regime aware p-state selection driver consumes 33% less energy (left) than Intel's driver during write operations, takes 10% less time (middle), and decreases average CPU core temperature by 3.5° (right).

et al. [7] present a feedback-driven controller that improves DVFS for I/O intensive applications. They detect I/O phases and periodically switch the CPU frequency to all possible states, selecting the optimum setting power/performance ratio based on power readings from an internal power meter. Our solution does not rely on instrumented power readers, and detects power/performance regimes within I/O phases to shift p-states automatically. Our power meter instrument is based on the work provided in Powerpack [3]. Our CPU optimizations are also related to the work by [9], that addresses thermal variation and does thermal and workload-aware application placement.

References

[1] S. Borkar and A. A. Chien. The future of microprocessors. *Communications of the ACM*, 54(5):67–77, 2011.

[2] U. Department of Energy. Top Ten Exascale Research Challenges. Technical report, Department of Computer Science, Michigan State University, February 2014.

[3] R. Ge, X. Feng, S. Song, H.-C. Chang, D. Li, and K. W. Cameron. Powerpack: Energy profiling and analysis of high-performance systems and applications. *Parallel and Distributed Systems, IEEE Transactions on*, 21(5):658–671, 2010.

[4] P. Llopis, J. Blas, F. Isaila, and J. Carretero. Vidas: object-based virtualized data sharing for high performance storage i/o. In *Proceedings of the 4th ACM workshop on Scientific cloud computing*, pages 37–44. ACM, 2013.

[5] P. Llopis, M. F. Dolz, J. García-Blas, F. Isaila, J. Carretero, M. R. Heidari, and M. Kuhn. Analyzing power consumption of i/o operations in hpc applications. *Ultrascale Computing Systems (NESUS 2015) Krakow, Poland*, page 107, 2015.

[6] P. Llopis, G. Martin, B. Bergua, and J. Carretero. Virtual i/o forwarding for cloud-based hpc applications. In *Proceedings of the 2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications*, pages 869–870. IEEE Computer Society, 2012.

[7] I. Manousakis, M. Marazakis, and A. Bilas. Fdio: A feedback driven controller for minimizing energy in i/o-intensive applications. In *Presented as part of the 5th USENIX Workshop on Hot Topics in Storage and File Systems, Berkeley, CA*, 2013.

[8] I. Pratt, K. Fraser, S. Hand, C. Limpach, A. Warfield, D. Magenheimer, J. Nakajima, and A. Mallick. Xen 3.0 and the art of virtualization. In *Linux Symposium*, page 65. Ottawa, Ontario, Canada, 2005.

[9] K. Zhang, S. Ogrenci-Memik, G. Memik, K. Yoshii, R. Sankaran, and P. Beckman. Minimizing thermal variation across system components. In *Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International*, pages 1139–1148. IEEE, 2015.

List of Authors