6th International Workshop
on
Grid Computing
for Complex Problems

GCCP 2010
BOOK OF ABSTRACTS

November 8 – 10, 2010
Bratislava, Slovakia

The workshop is organized by

Institute of Informatics, Slovak Academy of Sciences
Faculty of Electrical Engineering and Informatics, Technical University of Košice
Masaryk University, Brno, Czech Republic

The workshop is supported by

EGI-InSPIRE - EU FP7 RI project: Integrated Sustainable Pan-European Infrastructure for Researchers in Europe (2010-2014) FP7-261323

Program Committee

Ladislav Hluchý	IISAS - Institute of Informatics, Slovak Academy of Sciences
Ján Astaloš	IISAS - Institute of Informatics, Slovak Academy of Sciences
Jozef Černák	Faculty of Science, P. J. Šafárik University, Košice
Miroslav Dobrucký	IISAS - Institute of Informatics, Slovak Academy of Sciences
Ladislav Hudec	FIIT Slovak University of Technology in Bratislava
Jacek Kitowski	Cyfronet Cracow Poland
Jan Kmuníček	CESNET Praha, Masaryk University, Brno, Czech Republic
Ján Kollár	Faculty of Electrical Engineering and Informatics, TU Košice
Tibor Kožár	Institute of Experimental Physics SAS, Košice
Robert Lovas	MTA SZTAKI Budapest Hungary
Karol Matiaško	Management Science & Informatics, University of Žilina
Norbert Meyer	PSNC Poznan Poland
Ľudovít Molnár	FIIT Slovak University of Technology in Bratislava
Luboš Neslušan	Astronomical Institute SAS, Tatranská Lomnica
Ján Paralič	Faculty of Electrical Engineering and Informatics, TU Košice
Monique Petitdidier	CNRS - Centre National de la Recherche Scient., Paris, France
Ján Sarnovský	Faculty of Electrical Engineering and Informatics, TU Košice
Branislav Sitár	FMFI UK Bratislava
Jarmila Škrinárová	UMB Banská Bystrica
Milan Šujanský	Faculty of Electrical Engineering and Informatics, TU Košice
Viet Tran	IISAS - Institute of Informatics, Slovak Academy of Sciences
Claudio Vuerli	INAF Trieste Italy
Ivan Zahradník	Institute of Molecular Physiology and Genetics SAS, Bratislava
Peter Závodný	University of Economics in Bratislava

Organizing Committee

Ladislav Hluchý
Miroslav Dobrucký
Peter Kurdel
Jolana Sebestyénová
Oľga Schusterová
Institute of Informatics, Slovak Academy of Sciences
Dúbravská cesta 9, 845 07 Bratislava, Slovakia
E-mail: {ladislav.hluchy, miroslav.dobrucky, peter.kurdel, sebestyenova, sekr.ui}@savba.sk

Preface

Welcome to the 6th International Workshop on Grid Computing for Complex Problems GCCP 2010. The workshop is a three-day combined event for grid users: workshop with invited lectures, plenary discussions, accompanied by course for users of EGEE Grid sites, which is in the scope of EGI-InSPIRE - EU FP7 RI project: Integrated Sustainable Pan-European Infrastructure for Researchers in Europe (2010-2014) FP7-261323.

The topics of the workshop are:

- Distributed Computing and Large Scale Applications
- Environmental Applications and Distributed Computing
- Use of Knowledge and Semantics in Distributed Computing
- Astronomy & Astrophysics and High Energy Physics
- Grid and Service-oriented Computing
- Computational Chemistry & Material Science
- Distributed Computing and Large Scale Simulations
- Course on Development of Grid Applications

The next goal of the workshop is an associate action to create national Grid initiative "Sprístupnenie Gridu pre elektronickú vedu na Slovensku" (Making the Grid accessible for electronic science in Slovakia) which will help to improve the e-Science in Slovakia through the creation of virtual organizations for individual science branches. The associate action aims to join Grid specialists with complex application users, to provide a medium for the exchange of ideas between theoreticians and practitioners to address the important issues in computational performance and computational intelligence towards Grid computing.

The workshop on Grid Computing for Complex Problems GCCP 2010 has attracted 32 paper contributions and active participations from Czech Republic, Germany, Hungary, Poland, Sweden, Ukraine and Slovakia. This book is a collection of abstracts of papers from International Workshop on Grid Computing for Complex Problems – GCCP 2010. Workshop's papers will be published after the workshop as edited proceeding.

Many people have assisted in the success of this workshop. I would like to thank all the members of the Program and Organizing Committees, the workshop Secretariat for their work and assistance of the workshop. I would like to express my gratitude to all authors for contributing their research papers as well as for their participation in the workshop that made our cooperation more fruitful and successful.

Ladislav Hluchý
November 2010
Bratislava, Slovakia

Table of Contents

Invited lectures

Vendor session

Section 1
Distributed Computing and Large Scale Applications

Section 2
Environmental Applications and Distributed Computing

Section 3
Use of Knowledge and Semantics in Distributed Computing

Section 4
Astronomy & Astrophysics and High Energy Physics

Section 5
Grid and Service-oriented Computing

Section 6
Computational Chemistry & Material Science

Section 7
Distributed Computing and Large Scale Simulations

Course

Invited lectures

Structure and Recent Advancements of National Grid Infrastructure in Poland

Jacek Kitowski [1, 2]

[1] Institute of Computer Science, AGH-UST, al. Mickiewicza 30, 30-059, Kraków, Poland
[2] Academic Computer Center CYFRONET, ul. Nawojki 11, 30-950 Kraków, Poland

kito@agh.edu.pl, {z.mosurska, r.pajak}@cyfronet.pl

Abstract. The Polish Grid Initiative commenced in 2009 in the context of the PL-Grid Project funded under the framework of the Innovative Economy Operational Programme. The main purpose of this Project is to provide the Polish scientific community with an IT platform based on Grid computer clusters, enabling e-science research in various fields. The Project is establishing a country-wide Polish Grid infrastructure, which supports scientific research through integration of experimental data and results of advanced computer simulations carried out by geographically-dispersed teams. PL-Grid aims at significantly extending the amount of computing resources provided to the Polish scientific community and constructing a Grid system facilitating effective and innovative use of the available resources. In the paper some basic facts concerning the PL-Grid Project goals are outlined together with achieved results represented by several examples of innovative grid services and software developed within PL-Grid as well as user support procedures. Polish Grid Initiative has been considered to be the first working NGI in Europe.

Extending Service Grids with Desktop Grids

Peter Kacsuk

MTA SZTAKI, Budapest, Hungary

kacsuk@sztaki.hu

Current Grid systems can be divided into two main categories: service grids (SG) and desktop grids (DG). Service grids are typically organized from managed clusters and provide a 24/7 service for a large number of users who can submit their applications into the grid. The service grid middleware is quite complex and hence relatively few managed clusters take the responsibility of providing grid services. As a result the number of processors in SGs is moderate typically in the range of 1.000-50.000. Even the largest SG system, EGEE has collected less than 200.000 computers.
Desktop grids are collecting large number of volunteer desktop machines to exploit their spare cycles. These desktops have no SLA requirement, their client middleware code is extremely simple and hence typical number of volunteer desktops in desktop grids is in the range of 10.000-1.000.000. However, their drawback is that they can execute only some very limited number of pre-registered applications, typically compute-intensive bag-of-task applications. The most well-known volunteer desktop grid is SETI@home that collected over 2 Million CPUs.

Comparing the price/performance ratio of SGs and DGs the creation and maintenance of DGs is much cheaper than the one of SGs. Therefore it would be most economical if the compute-intensive bag-of-task applications could be transferred from the expensive SG systems into the cheap DG systems and executed there. The recognition of these advantages of extending SGs with DGs led to the initiation of the EDGeS (Enabling Desktop Grids for e-Science) EU project that was launched in January 2008 with the objective of integrating these two kinds of grid systems into a joint infrastructure in order to merge their advantages into one system. The EDGeS project extended gLite-based service grids with BOINC and XtremWeb DG systems.

To make these systems interoperate EDGeS has developed the 3G Bridge (Generic Grid-Grid Bridge) technology that enables the interconnection of any service and desktop grids. This bridge was used in EDGeS to create the gLite<->BOINC and gLite->XtremWeb bridges. The concept of 3G Bridge is so generic that it was successfully applied in the EELA-2 project, too in order to interconnect the OurGrid P2P desktop grid with gLite service grids. Based on the 3G Bridge technology EDGeS has created a production infrastructure where any gLite VO can be extended with volunteer and institutional DG systems. EDGeS has also ported 12 gLite applications from various scientific areas to the EDGeS infrastructure and created an application repository from where gLite users can access these applications.

Built on the success of EDGeS a follow-up project, called as EDGI (European Desktop Grid Initiative), was launched in June 2010. The objectives of EDGI include the extension of the EDGeS infrastructure with ARC and Unicore service grid support and to enable the execution of even data-intensive applications. Further on to provide QoS for the integrated infrastructure DG systems of the EDGI infrastructure will be extended with Cloud resources when required.

All these experiences of EDGeS will be explained in detail in the talk. At the end, some future plans of the EDGI project will be shown giving details how to support QoS requirements in the DG part of the integrated SG-DG infrastructure by supporting DG systems with some dedicated local academic clouds.

The Swedish Computing Infrastructure

Erwin Laure

High Performance Computing Center at the Royal Institute of Technology (KTH) in Stockholm, Sweden

erwinl@pdc.kth.se

Sweden aims at providing a full range of compute and storage services, ranging from high-throughput Grid systems to leading edge massively parallel HPC systems to Swedish scientists. This infrastructure is coordinated by a metacentrum, SNIC, the Swedish National Infrastructure for Computing, responsible for the strategic and scientific development and funding of computing and storage resources in Sweden.

This infrastructure is being used by a wide variety of Swedish scientists and builds the backbone of the recently launched national e-Science special research programs. It also builds the basis for Sweden's participation in major European infrastructures, particularly EGI and DEISA/PRACE.

In this talk we give an overview on the Swedish Computing Infrastructure with a particular focus on its usage and relations to international efforts.

D-Grid Infrastructure

Stefan Freitag

Robotics Research Institute in Germany

stefan.freitag@udo.edu

Like similar endeavors in other countries, the D-Grid initiative sponsored by the Federal Ministry of Education and Research has been started in 2005 to establish a national e-infrastructure that is particularly targeted towards public research and private-public partnerships involving small and medium enterprises. During the last five years, more than thirty projects of D-Grid have produced new insights and technological advances. Amongst others these projects helped to identify future challenges by extensively utilizing the D-Grid infrastructure with altogether more than 145 Mio CPUh in 2009.

This talk gives an overview on the organizational and structural aspects of D-Grid and describes its characteristics on the technical level. Furthermore, it addresses Cloud computing and quality- of-service provisioning as two of the identified technical challenges.

GPU for Bio-inspired Computing

Václav Snášel

VŠB - Technical University of Ostrava, Ostrava – Poruba, Czech Republic

vaclav.snasel@vsb.cz

Slovak participation in the World LHC Computing Grid

Branislav Sitár

Faculty of Mathematics, Physics and Informatics, Comenius University, Bratislava, Slovakia

sitar@fmph.uniba.sk

Designing Efficient Programs for Parallel Heterogeneous Platforms Using Rewriting Rules Technique

Anatoliy Doroshenko, Kostiantyn Zhereb , Mykola Kotyuk

Institute of Software Systems of National Academy of Sciences of Ukraine,
Glushkov prosp. 40, 03187 Kyiv, Ukraine
doroshenkoanatoliy2@gmail.com, zhereb@gmail.com, km.mysha@gmail.com

Recent hardware developments have lead to appearance of new platforms for parallel computations. Such platforms include multicore processors and graphical processing units (GPUs). These hardware devices are widely available and can be encountered in many computing environments, e.g. desktop and mobile computers, clusters and GRIDs. They also offer possibilities of significant improvement in program performance. However to realize such possibilities a programmer has to write optimized programs specifically targeted at given parallel platform. Creating efficient parallel programs for new parallel platforms remains a complex task because of novel computational paradigms, insufficient tool support and inherent complexity of parallel programs. Therefore there is a need of facilities that help developer create correct and efficient parallel programs utilizing all capabilities of hardware platform.

In this paper we describe our approach of developing efficient parallel programs by applying formalized transformations to models of sequential or parallel programs. We use formal facilities to automate designing of efficient parallel programs for heterogeneous platforms. This paper concentrates on two such facilities: algebraic program models are used to concisely describe high-level algorithm structure and rewriting rules technique allows applying program transformations in automated way.

We use the following approach to design efficient parallel programs. We start with a serial program (in source code form) implementing given algorithm. This program is represented as high-level model using Glushkov's algebra of algorithms [1]. Then we develop program transformations represented as sets of rewriting rules. We consider two varieties of such transformations: *parallelizing* transformations convert a serial program into a parallel program for a given platform implementing the same algorithm; and *optimizing* transformations improve performance of parallel programs. Developed rulesets are then automatically applied to program model, resulting in a new model of parallelized or optimized program. Finally this model is transformed back into source code form.

We have automated many steps of described process. In particular, conversion between source code and high-level algebraic model of a program is completely automated for a given problem domain and programming language. Applying parallelizing and optimizing transformations is also automated: user only has to specify which transformation should be applied and what fragment of code (i.e. loop) it should affect. The most important manual part of described process is creation of transformation rules. However, many of such rules are generic enough to be applied

to multiple programs. Therefore such rules can be created once and then reused in different programs.

To automate program transformations we use Termware rewriting rules system [2]. We have developed Termware parser and code generator for C# language, which allows working with program in source code form. Developed parser converts C# code into a term representing syntax structure of a program. This term essentially represents low-level model of program. Code generator can be used to transform such low-level model back into source code form.

While low-level (syntax) model already allows automating code transformations, it is verbose and requires specifying minor details of code structure. Therefore, we transform low-level syntax model into high-level algebraic model. Transition between low-level and high-level models of a program is performed using rewriting rules, as both kinds of models can be represented as terms. These rules take form of Termware *patterns* that consist of a pair of terms t_p – a designation of pattern (an element of high-level model) and t_g – an implementation of pattern (an element of low-level model). Rulesets $R_p = \{t_g \to t_p\}$ are used to convert low-level model into its high-level form, and rulesets $R_g = \{t_p \to t_g\}$ for backward conversion.

We have developed program transformations aimed at both multithreaded and GPU parallel platforms. In the paper we present parallelizing transformations that apply to sequential *for* loop and turn it into parallel loop that use threading constructs from Microsoft .Net framework (multithreaded platform) or facilities provided by NVidia CUDA platform (GPU platform).

As an example of use of proposed approach we have applied developed transformations to C# program implementing matrix multiplication. From initial sequential version we created both multithreaded and GPU parallel programs. For GPU program we also applied optimizing transformation to use shared memory instead of global memory. Performance measurements suggest high efficiency of developed transformations: in multithreaded case, we were able to obtain speedup of 3x-4x on 4 core processor (compared to sequential program), and in GPU case, speedup of 40x-300x depending on matrix size.

Our approach based on applying formal program transformations can be used to design and develop correct and efficient parallel programs for heterogeneous platforms and automates significant steps of development process. Using high-level algebraic models allows concise representation of programs and transformations, while rewriting rules technique enables automated application of developed transformations and transition between high-level models and source code. Further research directions include development of additional parallelizing and optimizing transformation and using them to improve performance of various programs. We plan adding support for emerging GPU programming technologies, such as OpenCL and exploring possibilities of combining multithreaded and GPU capabilities in single program in order to use heterogeneous platforms more efficiently.

References

1. Andon, P.I., Doroshenko, A.Yu., Tseitlin, G.O., Yatsenko, O.A.: Algebra-algorithmic models and methods of parallel programming (in Russian). Academperiodika, Kiev (2007)
2. Doroshenko, A., Shevchenko, R.: A Rewriting Framework for Rule-Based Programming Dynamic Applications. Fundamenta Informaticae, Vol. 72, N 1–3 (2006) 95–108

Vendor session

IBM a oblasť vysoko výkonného počítania Partner so skúsenosťami, na ktorých sa oplatí stavať

Marián Kováčik

IBM Slovakia, s.r.o., Bratislava, Slovakia
marian.kovacik@sk.ibm.com

Vysoko výkonné počítanie získava na Slovensku a v celej strednej Európe čoraz viac priaznivcov v kruhoch odbornej verejnosti zaujímajúcich sa o rôznorodú paletu technológií a riešení, tak v oblasti široko rozprestrenej výpočtovej sily prepojenej do gridových infraštruktúr, ako aj vo forme samostatných klástrov s koncentrovaným výpočtovým výkonom na jednom mieste. V dôsledku technického a technologického pokroku, v porovnaní s minulosťou, sú tieto riešenia čoraz výkonnejšie a na druhej strane aj dostupnejšie pre nasadenie. Týmto sa ešte viac zvyšuje ich popularita a záujem o ich využívanie v oblasti vedy, výskumu, vývoja a v konečnom dôsledku vo využití ich výsledkov v hospodárskej praxi.

Čo sa však skrýva za pojmom vysoko výkonné počítanie? Je to len hrubá výpočtová sila založená na určitom počte výpočtových jadier alebo niečo viac?

IBM vníma klástre pre vysoko výkonné počítanie vždy ako unikátne riešenie založené na výpočtových požiadavkách aplikácií, ktoré sú definované koncovými užívateľmi týchto aplikácií. Na podporu dosiahnutia požadovanej výkonnosti a výsledkov je potom možné použiť rôzne druhy hardvérových a softvérových komponentov. Z tohto pohľadu by bolo možné za primárnu devízu IBM považovať široké a rôznorodé portfólio hardvérových a softvérových produktov pokrývajúce potreby výpočtov. Dôležité je však aj niečo iné. A to sú skúsenosti spojené s inštalovanou bázou. Tieto skúsenosti nesiahajú len do oblasti implementácie riešenia, tá je len jednou fázou. Začínajú už zrodom myšlienky, ktorá súvisí s potrebou využitia výpočtov pre dosiahnutie vytýčeného cieľa, cez návrh riešenia, implementáciu až po starostlivosť o riešenie v poimplementačnej fáze. V konečnom dôsledku práve skúsenosti vedú k optimálnemu nasadeniu riešenia so správnymi komponentami, ktoré sú nevyhnutne potrebné a užitočné pre dosiahnutie stanoveného účelu a cieľov.

V oblasti vysoko výkonného počítania IBM ponúka riešenia kombinujúce širokú škálu hardvérových a softvérových komponentov. Z hardvérovej strany pokrýva IBM portfólio klástrové a gridové požiadavky od základného všeobecného použitia a to prostredníctvom serverov IBM System x, cez silné monolitické riešenia postavené na serveroch IBM POWER Systems, až po špeciálne riešenia postavené na IBM BlueGene. Tieto riešenia môžu byť účelne doplnené zariadeniami ako úložiská dát, či zálohovacie knižnice, ktoré sú častokrát nevyhnutnou súčasťou nasadenia klástrov. Na záver, nad vybranou hardvérovou platformou a operačným systémom IBM umožňuje stavať aj softvérovú vrstvu optimalizujúci nasadenie riešenia, jeho manažovanie, monitorovanie, či paralelný prístup k ukladaným dátam (xCAT, IBM Systems Direc-

tor, IBM Tivoli, IBM General Paralel File System). To všetko je ešte doplnené rozsiahlymi skúsenosťami implementátorov.

Vysoko výkonné počítanie predstavuje komplexnú oblasť, ktorá si vyžaduje dobre premyslené riešenia, častokrát založené na nadčasových požiadavkách a trendoch. IBM technológie tieto požiadavky a trendy približujú bližšie k reálnemu nasadeniu.

Section 1
Distributed Computing and Large Scale Applications

GPU performance within numerical precision of OpenCL operations

Martin Jurečko[1], Jana Kočišová[1,3], Ján Buša Jr.[1,4], Tomáš Kasanický[1,6], Marek Domiter[2,4], and Marián Zvada[1,2,5]

{jurecko, kocisova, jbusa, kasanicky, domiter, zvada}@sors.com

[1] FURT Solutions, s.r.o., Strojárenská 3, Košice, Slovakia,
[2] SORS Holding, Strojárenska 3, Košice, Slovakia
[3] Pavol Jozef Šafárik University in Košice, Šrobárová 3, Košice, Slovakia
[4]Technical University of Košice, Letná 9, Košice, Slovakia
[5]Karlsruhe Institute of Technology, Karlsruhe, Germany
[6]Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

Abstract. There are many projects focused on performance measurements of GPUs but there is no unifying test framework that could be used for evaluating generic floating point intensive applications. This work describes the testing suite for evaluating GPUs that measures raw performance and numerical precision of a subset of OpenCL operations, and analyzes results obtained from several commonly available high-end GPUs.

Extended Abstract

There are many interesting projects focused on performance measurements of GPUs but there are very few frameworks utilizing OpenCL, and there is no unifying test framework that could be used for evaluating generic, floating point intensive applications. The majority of testing suites [4–5] aim to measure the GPU performance and accuracy only on the basis of typical algorithms such as matrix multiplication, matrix transpose, binary search or by evaluating some custom, application specific algorithm. However, the use of such algorithms often leads to specific results that are bound to the domain of the tested algorithm. We aim to find a more generic approach, or the metric, that could be used to evaluate GPU cards on some basis relative to the most of the GPU algorithms. There are several programming frameworks written directly for native GPUs [1] hardware (NVIDIA, ATI), but as of today, the OpenCL [2] is the only framework that is supported by the majority of the commonly available GPUs. Therefore we decided to test the accuracy and performance of the most common OpenCL arithmetic operations, namely addition, subtraction, division, multiplication, sin, cbrt, exp, log, abs (fabs), floor, fmod, pow, fmax and mad. Our work is supported by FACADE (Financial Analysis Computing Architecture in Distributed Environment) project, in which we are creating cluster environment based on

GPUs[3]. Different GPU vendors and the generation gaps between the GPU units we currently have was the main reason to use the OpenCL. For the purpose of choosing the best GPU cards to buy, we needed to create a testing suite for easy and automated evaluation of graphics processing units available on the market. We tested NVIDIA Tesla C1060, NVIDIA GTX285, ATI HD5870 and we are presenting our findings in this paper.

Test results were produced by our in-house written OpenCL testing suite. Tests aimed to exercise subset of typical arithmetic operations used, or expected to be used in our applications. Tests operated on randomly generated input data with maximum possible length, limited only by the amount of allocable GPU global memory. At the time of testing, the amount of allocable GPU memory of early OpenCL implementations was limited; also, some data types and operations were not fully supported. To eliminate data corruption caused by faulty hardware or by possibly incorrect algorithm design, tests ran multiple times on the same inputs and their outputs were compared to each other.

The testing suite presented in this paper shows to be capable of evaluating the performance and numerical precision of GPU cards by measuring the execution times and by validating the results of testing kernels. The kernels cover representative combinations of arithmetic operations, input element types and local sizes. To account for kernel setup time, the combinations also cover repetitive arithmetic operations. Tested element types include most of scalar and vector integers and floating point types with the exception of half types. Results were validated by comparing the number of differences in test results between the GPU results and the golden results computed on CPU, with regards to OpenCL specification. The tests were run multiple times on different host platforms and on multiple GPU cards. The results provided an evaluation of completeness and robustness of current OpenCL driver implementations and they also provided an insight into architecture of tested GPUs. Analysis of the results presented in this work proved to be essential for planning the hardware strategy of the FACADE project.

References

1. Kamran Karimi, Neil G. Dickson, Firas Hamze: A Performance Comparison of CUDA and OpenCL. [Online]. Available: http://arxiv.org/ftp/arxiv/papers/1005/1005.2581.pdf
2. KHRONOS Group: OpenCL Overview [Online]. Available:http://www.khronos.org/opencl/
3. Gregg Ch., Brantley J., Hazelwood K.: Contention-Aware Scheduling of Parallel Code for Heterogeneous Systems. 2nd USENIX Workshop on Hot Topics in parallelism (HotPar). Berkeley, CA. (June 2010).
4. Krawezik G. and Poole G.: Accelerating the ANSYS Direct Sparse Solver with GPUs. SAAHPC workshop 2009 Symposium on Application Accelerators in High Performance Computing (SAAHPC'09) NCSA at UIUC, Urbana, Illinois
5. Kurpad Anupama Shankar: A comparative performance analysis of the phase recovery algorithm for microstructure reconstruction. Drexel Theses and Dissertations (Aug 2009).

On the Load Balancing of Parallel Maximum Clique Algorithm

Karol Grondzak[1] and Penka Martincová[1]
[1]Department of Informatics, Faculty of Management Science and Informatics, University of Zilina

To define a maximum clique problem, let us consider undirected graph $G = (V, E)$, where V is the set of vertices and E is a set of edges. Complete (or full) graph is a graph for which there is an edge between every pair of vertices. Subgraph $G' = (V', E')$ of graph G is such a graph, that $V' \subseteq V$ and $E' \subseteq E$.

If subgraph of some graph is complete, then we will call it *clique*. It is obvious, that there can be many complete subgraphs of a given graph. For some applications, cliques of some special properties are of interest.

Maximal clique of graph G is a clique, which can not be extended using any other vertex of the graph G. *Maximum* clique is a clique, which has maximal number of vertices among any cliques of the graph G.

In the coding theory maximum clique problem is solved to find largest binary code able to correct prescribed number of errors. In computational biology problem of maximum or maximal cliques is applied for solving the problems of genome mapping or 3-D protein structure alignment. Interesting area of application of maximal clique problem is biology, where it is used to study food chains.

Some tasks in computer vision and pattern recognition areas include the problem of finding either maximum or maximal cliques.

Problem of finding maximum clique has been proven to be NP-complete ([1]) and requires exponential time to be solved. During the time, many exact and heuristic algorithms were proposed to solve this problem.

First exact algorithm for a general graph was published in late 1950's ([2]). Later researchers oriented towards the minimization of memory requirements and reducing the computational time. In 1970's backtracking approach was proposed and implemented. Some researchers also proposed to apply branch and bound method ([3]).

Because of the computational complexity, for some problems of finding maximal clique, heuristic methods are applied ([3]). Greedy heuristic algorithm is based on the idea of repeated adding nodes into existing clique, i.e. expanding it. Another approach is to remove repeatedly nodes from a set of vertices, which are not a clique until clique is obtained.

Other optimization algorithms based on simulated annealing, tabu-search, genetic algorithms and neural networks are also are also applied to solve the problem of maximum clique.

Exact maximum clique algorithms are based on the idea to construct all possible subgraphs of a given graph G and test, if they are cliques ([4]). Evaluating all constructed cliques, the clique with maximum number of vertices is obtained. This brute-force algorithm is very inefficient for even medium size graphs (with thousands of vertices and edges) and it has been improved by many authors. Improvements are based on some information, obtained during the process of construction and evaluation of subgraphs. This information is then used to make a decision, whether some subgraph can yield a better solution than the actual one. Only promising subgraphs are considered for further evaluation.

The basic idea of all exact algorithms is to construct search-tree and perform depth-first search of that tree. Recently with the emerge of parallel and distributed computing the idea of dividing search-tree into parts and performing depth-first search of these subtrees in parallel became very promising to improve the performance of maximum clique search algorithm ([5],[6]). Simple algorithms to evaluate search tree in parallel are not efficient, because of the fact, that the search tree is usually unbalanced. Thus some of the processors finish their task very quickly and are sitting idle, while other processors are still busy performing their part of the work. Published papers indicate that linear speedup is difficult to obtain ([7]).

To understand the nature of the search-tree of maximum clique finding problem, we have studied it on the problem of PIN generation. It is a problem of finding a set of PIN codes, such that a distance (defined on a set of PIN codes) between any two codes is larger than some given constrain. It can be used to generate a set of PIN codes for users of electronic devices which require some kind of access authorization, e.g. copy machines, or PBX to get access to public land lines from private telephone network.

Obtained results indicate that the search-tree of the PIN generation problem is highly unbalanced as it was expected.

In this paper we present a simple load balancing technique applied to the exact maximum clique algorithm. The modified algorithm is evaluated in the terms of load balancing quality and overall performance.

References

[1] Lawler, E.L., Lenstra, J.K., Kan, A.H.G.R.: Generating all maximal independent sets: NP-hardness and polynomial-time algorithms. SIAM Journal on Computing **9** (3), 1980, pp. 558-565

[2] Harary, F., Ross, I.C.: A procedure for clique detection using the group matrix, Sociometry **20** (3), 1957, pp. 205-215

[3] Du, Z. Z., Pardalos, P.M. (Eds.): Handbook of Combinatorial Optimization. Supplement Volume A. Kluwer Academic Publishers, 1999

[4] Carraghan, R., Pardalos P.M. An exact algorithm for the maximum clique problem. Operations Research Letters 9 (1990), pp. 375-382

[5] Pardalos, P. M., Rappe, J., Resende, M. G. C. An exact parallel algorithm for the maximum clique problem. In High Performance and Software in Nonlinear Optimization, Kluwer Academic Publishers, 1997, pp. 279-300

[6] Schmidt, M.C., Samatova, N., F., Thomas, K., Park, B.: A scalable, parallel algorithm for maximal clique enumeration. J. Parallel Distrib. Comput. **69**, 2009, pp. 417-428

[7] Du, N., Bin, W., Liutong, B., Bai, W., Xin, P.: A parallel algorithm for enumerating all maximal cliques in complex networks. Proc. of ICDM Workshops, 2006, pp. 320-324

Remarks on the GRID computation of the characteristics of Boolean matrices

(Extended abstract)

Matúš Jókay *

KAIVT FEI STU, Ilkovičova 3, Bratislava, Slovakia

Grosek et. al. [1] proposed a new matrix test of randomness based on characteristics of Boolean matrices, namely: index, and period statistics of the set of Boolean matrices. Recently, we were able to compute the statistics for the matrix dimension up to $n = 7$, using parallel computation, and SIMD within register techniques [2]. Our original goal was to compute the statistics for $n = 8$, since this dimension is favorable for the optimal bit packing, and implementation of the test, respectively. In [2] this goal seemed to be too computationally difficult for the resources available. However, we were finally able to reach this goal using a novel algorithm [3], and by employing a large scale computation on the GRID facility at the University of Bergen. In this contribution, we provide some remarks on the technical details of the GRID computation.

In the first part of the contribution we describe the GRID center at the University of Bergen (where the computation took place). The contribution will provide more details on:

- the available computational power,
- provided development tools and applications,
- control framework,
- administrativnu cast spravy centra.

The second part of the contribution details the design and architecture of the control program for our distributed computation, and the gathering of statistics, respectively. We use the MPI framework, and the specific scripts for the following tasks:

- scheduling and task stack management,
- backup tools,
- recovery tools.

The statistics of the computational effort will be mentioned, such as:

- the volume of processed data,
- average computational times,

* This material is based upon work supported under the grant NIL-I-004 from Iceland, Liechtenstein and Norway through the EEA Financial Mechanism and the Norwegian Financial Mechanism, and under the grant VEGA 1/0244/09. Distributed computing power was provided by the NorGrid at Bergen University under grant NIL-I-004.

– GRID utilization statistics,
– comparison of the computational effort on the large UiB GRID with the smaller computations in the GRID laboratory at FIIT STU BA,
– comparison of the real timings and the expected computational effort.

We conclude with future possibilities, and predictions for the computation of statistics of even larger matrices.

References

1. Grošek, O., Vojvoda, M., Krchnavý, R.: A new matrix test for randomness. Computing 85, (2009) 21–36
2. Jókay, M., Zajac, P.: Parallelization Techniques for the Matrix Test Precomputation. In: 5th International Workshop on Grid Computing for Complex Problems. GCCP 2009 : Bratislava, Slovak Republic, 26.-28.10.2009. (2009) 103–109.
3. Zajac, P., Jókay, M.: Computing indexes and periods of all Boolean matrices up to dimension $n = 8$. preprint. (2010)

GPU Computing on CUDA Cluster

Tibor Kožár

Department of Biophysics, Institute of Experimental Physics, Slovak Academy of Sciences,
04001 Košice, Slovak Republic

tibor@saske.sk

Section 2
Environmental Applications & Distributed Computing

Experimental and computational study of automobile fires

P.Weisenpacher[1], J.Glasa[1], L.Halada[1], P.Poledňák[2], G.Okša[3]
[1]Ústav informatiky SAV, Bratislava,
[2]Fakulta špeciálneho inžinierstva ŽU v Žiline,
[3]Matematický ústav SAV, Bratislava

1.Introduction

In spite of the fact, that the number of fires due to technical or electrical malfunction and self ignition has decreased somewhat, the number of automobile fires is increasing in many countries. There are several reasons of this occurrence:

- great number of automobiles of different types and their concentration on roads, in car park and tunnels,
- amount of combustible material in standard automobiles is in the range 150-250 kg and this material is stored relatively in a small compartment,
- number of automobile fires due to arson has also unfortunately increasing tendency especially in big cities.

Therefore, human survivability during automobile fires, as well as prevention and protection of materials in tunnels or car parks are studied and detail investigated in detail. One approach, how to analyze above mention problems, is by full-scale experiment. This approach is usually costly and non effective.

2.Full-scale automobile-fire experiments

However, some full-scale automobile fire experiments are necessary to accomplish, because every ten years new generation of cars are supplied to market and for each category of such cars it is important to have basic parameters such as the average car mass, mass of combustible materials, the energy release or heat release rate and so on. Generally, materials used to the construction of new car category can be different than materials used for earlier cars. As a result, the graph of heat release rate vs. time for old and new cars can be significantly different. Therefore, based on energy released in automobile fire, European cars can be classified in four categories. For each category average car mass and energy released can be found in Table 1 [1].

Category	Car mass (kg)	Mass of combustible materials (kg)	Related energy (MJ)
1	850	200	6000
2	1000	250	7500
3	1250	320	9500
4	1400	400	12000

Table 1.

Full-scale experiments in an open car park with a floor surface 15 x 32 m and height 3 m, with cars parked together side-by-side, were documented in [1]. During the test, fire always started by ignition under the middle car at the level of the gearbox and the fire continued until the full burn-out of adjacent cars. Fire spread from one car to another always occurred, but the fire propagation time was different depending upon wind condition and the orientation with respect to the cars. This experiment was used also as reference example for comparison with numerical analysis result to check the validity and the accuracy of numerical models.

A similar experiment was analyzed in paper [2]. The rectangular garage, with inner dimensions 15.8 x 15.8 x 3.2 m and the door for cars 3.2 x 2.5 m, had three windows 1.0 x 0.6 m on one side. The next door for entrance and exit people stands (was placed) on the opposite side of the door for cars. The garage capacity was 12 vehicles in two rows. In the garage two technical solutions included the mechanical design of ventilation system for the case of fire were analyzed and appropriate solution was suggested.

Another experiment, realized by General Motors, was analyzed in [3]. The authors concluded from the fire test that in front-end collisions, where fire originates in the engine, compartment flames penetrate the vehicle interior within 10-20 minutes. In rear-end collisions with a gasoline pool fire, flames penetrated the vehicle interior through body openings within 2 minutes. Consequently, once flames penetrate the passenger cabin from either the front or the rear, dead of occupants can occur during a few minutes. Thus, the human survivability in motor vehicle fires is a serious problem.

2. Computer simulation of automobile fires

The second possibility, how to analyze a fire development, is to use CFD method with suitable computer program. Unlike the conventional approach, CFD approach enable in different geometrical area, with specific boundary and time conditions, relatively precisely to predict the fire development and to compute many useful parameters such as fluid velocity, temperature, density, turbulence, etc. Thus, the numerical modelling makes it possible to analyse and to visualise car fire or multiple car fire in different car parks or tunnels under various fire scenarios and conditions. In fact, this advanced approach has been largely used for fire analysis in many research projects.

3. Paper results

In our paper, we present a full-scale experiment of automobile fire of Kia ceed made in open experimental space in Považský Chlmec. The development of this fire is documented by graph of temperature vs. time measured in different places of this automobile during the experiment. The data obtained during this full-scale experiment are consequently used for computer simulation of automobile fire (Fig. 1). In the next part simulation of fire in 180 m long tunnel will be analyzed. Suitable design of mechanical ventilation systems in tunnels is very important and has to be tested before the construction of tunnel. The optimal configuration of a supply and exhaust system applied in tunnel is complicate a problem, in general (Fig. 2).

References

[1] Bin Zhao, Joel Kruppa: Structural behaviour of an open car park under real fire scenarios. Fire and materials, vol. 28, pp. 269-280, 2004.

[2] M.J.Banjac, B.M.Nikolič: Computational study of smoke flow control in garage fires and optimization of the ventilation system. Thermal science, vol. 13, n. 1, pp.69-78, 2009.

[3] K.H.Digges and all.: Human survivability in motor vehicle fires, Fire and materials, vol. 32, pp. 249-258, 2008.

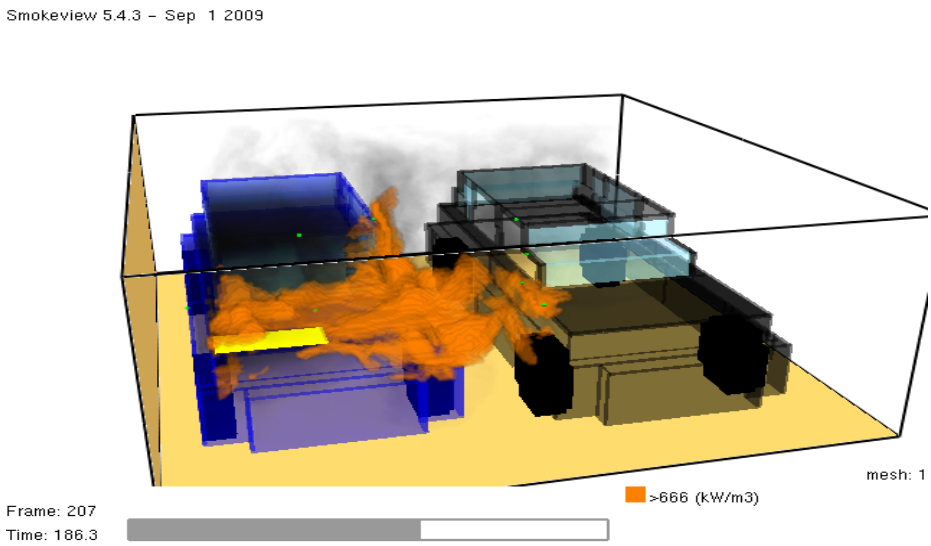

Fig. 1 Computer simulation of automobile fire

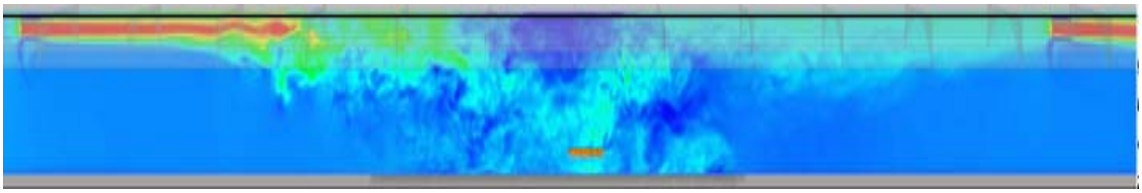

Fig.2. Automobile fire initialised in the middle of the tunnel with supply and exhaust ventilation system is shown in the left and the right border of the tunnel, respectively.

Data Mining for Prediction in Meteorology and Hydrology

Peter Krammer[1], Ondrej Habala[1], Martin Šeleng[1], Viet Tran[1], Ladislav Hluchý[1]

[1] Institute of Informatics oft he Slovak Academy of Sciences, Dúbravská cesta 9
84507 Bratislava, Slovakia
{peter.krammer, ondrej.habala, martin.seleng, viet.tran, hluchy.ui}@savba.sk

Abstract. The use of data mining techniques in environmental applications of IT, and specifically in hydro-meteorological predictions, is not a new topic. However, the range of phenomena and environmental parameters to which data mining has been applied is not nearly exhausted yet, and in this paper we present the use of data mining on two innovative scenarios. In one scenario we try to predict the change of water level and water temperature in the Orava river below the Orava reservoir, and how they react to discharge from the reservoir. In the second scenario we use radar imagery from weather radars for accurate, short-term prediction of precipitation.

Keywords: data mining, meteorology, hydrology, weather radar

1 Introduction

The project ADMIRE[2] (Advanced Data Mining and Integration Research for Europe [1]) is a 7th FP EU ICT project aims to deliver a consistent and easy-to-use technology for extracting information and knowledge. The project is motivated by the difficulty of extracting meaningful information by data mining combinations of data from multiple heterogeneous and distributed resources. It will also provide an abstract view of data mining and integration, which will give users and developers the power to cope with complexity and heterogeneity of services, data and processes.

There are three scenarios, which are in different stage of complete in the ADMIRE project. These scenarios have been selected from more than a dozen of candidates provided by hydro-meteorological, water management, and pedological experts in Slovakia. The main criterion for their selection was their suitability for data mining application. The scenarios are named ORAVA, RADAR and SVP, and they are in different stages of completion, with ORAVA being complete, and SVP only in the beginning stages of its implementation. In this paper we will mainly describe the ORAVA scenario, its implementation, and experimental results.

[2] This work is supported by projects ADMIRE FP7-215024, APVV DO7RP-0006-08, SMART ITMS: 26240120005, SMART II ITMS: 26240120029, VEGA No. 2/0211/09.

2 ORAVA Scenario

The primary goal of Orava-scenario is to predict variables at Orava river, which are water level and water temperature. Our objective is to precede water floods caused by high level of water or blocking water flow by icebergs. Get information can be used for water power-plant , or in ecology.

3 Integration of spatio-temporal data

In environmental management use-cases in ADMIRE, we deal with the spatio-temporal environmental data stored in gridded form. We use following methodology for integration of spatio-temporal data in ADMIRE:

- Data retrieval: in this step, raw data are retrieved from available (possibly heterogeneous) data sources and preprocessed to common data representation (set of tuples).
- Spatial representation transformation: in case that data sets being integrated use different coordinate system on spatial dimensions, coordinate representations must by unified prior to the data integration.
- Missing data handling: it is often the case, that there are missing data from measurement instruments (e.g. due to their failure). To avoid unexpected problems with the data integration in following steps, we introduce the step for dealing with missing records. Concrete procedure depends on the data integration scenario.
- Temporal synchronization: when integrating the data sets with different record frequencies (e.g. hourly records vs. daily records), the record frequencies must be synchronized (e.g. records with higher frequencies are aggregated to match the frequency of other data sets).
- Spatial synchronization: when integrating spatial data in grid form with different grid cell sizes and different grid cells centers, the grid representation must be unified; this involves transformation of a data sets grid and values associated with grid cells.
- Data integration: after having the data sets synchronized on spatial and temporal dimensions and having taken care of missing measurement values, we can finally integrate the data into homogeneous set of tuples.

References

1. Advanced Data Mining and Integration Research for Europe (ADMIRE), EU FP7 ICT project no. 215024. http://www.admire-project.eu (accessed Sept. 2010).

Experiments with a Hybrid-Parallel Model of Weather Research and Forecasting (WRF) System

Viera Šipková,[1] Andrej Lúčny[2] and Martin Gažák[2]

[1] Ústav informatiky, Slovenská akadémia vied
845 07 Bratislava, Dúbravská cesta 9
http://www.ui.sav.sk/
[2] Microstep-MIS, Monitorovacie a informačné systémy
804104 Bratislava, Čavojského 1
http://www.microstep-mis.com/

Abstract. The Weather Research and Forecasting (WRF) system represents a numerical weather prediction model suitable for research and operational user communities. WRF is designed to run on a variety of platforms, either serially or in parallel, with or without multi-threading. In this paper we describe the performance investigation of the real WRF simulation process which is configured as a workflow of three consecutive jobs (pre-processing, modeling, and post-processing) running on a multi-core cluster locally, and remotely on the EGEE Grid. The workflow was compiled and validated for two parallel scenarios: the pure MPI, and the hybrid MPI+OpenMP.

Keywords: WRF model · distributed & shared memory architecture · parallel programming model · MPI · OpenMP · cluster computing · grid computing

1 Introduction

Numerical modeling plays a significant role in the earth sciences filling in the gap between experimental and theoretical approach. Because of the vast amounts of observed data, and the need to store, process and refine them, the employment of high performance parallel computing is the only effective way to ensure the real usability of computation- and/or data-intensive numerical applications. Cluster and grid computing belong to the key technologies in the field of computational sciences. Grids, which may encompass a huge number of geographically-spread autonomous high performance resources (clusters, and more), enable their coordinated sharing, selection, and aggregation, dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements.

In the process of the software development, the selection of an appropriate programming model which is capable to exploit effectively each of the available computing resources (e.g. multiple cores in a CPU, GPU accelerators, processors and clusters available across a network or in a grid), is fundamental and crucial for increasing the performance. For the last several years, most computing clusters have been built of a collection of multi-core nodes, with a shared memory between cores of a node, and a distributed memory between nodes. So, applying a hybrid-parallel programming paradigm, where shared memory, message-passing, and grid programming techniques are combined within a single application, seems to fit well.

References

1. The Weather Research and Forecasting (WRF) Model. http://wrf-model.org/
2. The users home page for the Weather Research and Forecasting (WRF) modeling system. http://www.mmm.ucar.edu/wrf/users/
3. A Description of the Advanced Research WRF Version 3. http://www.mmm.ucar.edu/wrf/users/docs/arw_v3.pdf
4. PBS - Portable Batch System. http://www.mcs.anl.gov/research/projects/openpbs/
5. EGEE - Enabling Grids for E-sciencE, Grid project. http://www.eu-egee.org/
6. gLite - Lightweight Middleware for Grid Computing. http://www.glite.org
7. S. Burke, S. Campana, E. Lanciotti, P.M. Lorenzo, V. Miccio, C. Nater, R. Santinelli, A. Sciaba: gLite 3.1 Users Guide. https://edms.cern.ch/file/722398/1.3/gLite-3-UserGuide.pdf
8. Open MPI - A High Performance Message Passing Library. http://www.open-mpi.org/
9. OpenMP - Open Multi-Processing, API Specification for Parallel Programming. http://openmp.org/
10. F. Pacini: Job Description Language HowTo. http://www.infn.it/workload-grid/docs/DataGrid-01-TEN-0102-0 2-Document.pdf
11. F. Pacini: Job Description Language Attributes Specification for the gLite middleware (submission through WMProxy service). https://edms.cern.ch/document/590869/1/
12. EGI - European Grid Initiative. http://www.egi.eu/
13. EMI - European Middleware Initiative. http://web.eu-emi.eu
14. S. Masoud Sadjadi et al.: Transparent Grid Enablement of Weather Research and Forecasting. In Procs. of the 15th ACM Mardi Gras Conference: From lightweight mash-ups to lambda grids: Understanding the spectrum of distributed computing requirements, applications, tools, infrastructures, interoperability, and the incremental adoption of key capabilities, Baton Rouge, Louisiana, 2008. MG'08; Vol. 320, ISBN 978-1-59593-835-0
15. Juraj Bartok, Peter Bednár, Martin Gažák, Ondrej Habala, Ladislav Hluchý, Andrej Lúčny, Ján Paralič: Prediction of Significant Meteorological Phenomena Using Data Mining. In Proc. of the 5th International Workshop on Grid Computing for Complex problems (GCCP'2009), October 2009, Bratislava, Slovakia. ISBN 978-80-970145-1-3, pp. 56-62.

The grid scheduling

Jarmila Skrinarova

UMB, Banská Bystrica, Slovakia

skrinar@fpv.umb.sk

Section 3
Use of Knowledge and Semantics in Distributed Computing

New aspects on sentiment analysis of slovak texts.

Ivana Budinská[1], Andrej Kadora[2]

[1] Ústav informatiky Slovenskej akadémie vied, Dúbravská cesta, 9,
84507 Bratislava, Slovakia.
budinska@savba.sk
[2]andrej.kadora@gmail.com

1 Abstract

Research in the area of sentiment analysis of written texts is very actual. The application of sentiment analysis methods are in many areas from marketing to political situation monitoring. Broader employment of these techniques in everyday life requires carrying out and evaluation of many experiments. This is the way how to gain reference data to be used for given application domains.
The paper discusses some problems connected to the text classification on the basis of sentiment analysis. An implementation of Naive Bayes classificator is used to perform a set of experiments in the domain of movie reviews. A reference set of movie reviews by Pang and Lee was used. The set consists of 700 negative and 700 positive reviews. In this set a number of 5000 subjective sentences were determined. Experiments were realized on a set of movie review from the portal www.csfd.cz. The reviews were processed prior classification. Some different methods of pre-processing were used and the influences of the preprocessing methods on the classification results were evaluated. The removing of diacritics and some letters that are not important for the text sentiment leads to achievement of considerably better results. A machine translation as another method of text processing prior to classification was used. Recently there exist some very good online translators that can be used to automatic translation of texts. The results of classification were surprisingly good after translation of texts from the original language (Slovak) to English. The classification was then done on the translated English texts. However the results were little bit worse comparing to the classification of original English texts. The reason for this may be in:

- The reviews from the www.csfd.cz are not as homogenous in language expressions as original English texts, because there are more different authors of the reviews. Also there are many spelling errors.
- The automatic translation of original texts into English is not accurate. The better translator will be used the better results we will reach

The velocity of processing of continuously rapidly increasing number of information resources (online discussions, blogs, etc.) becomes a very important factor. From experiences, the velocity is even more important for users than precision of the classification. The following research is focused on rapid and still enough precise methods for sentiment analysis.

Evaluating Grid Infrastructure for Natural Language Processing

Radovan Garabík[1], Jan Jona Javoršek[2], and Tomaž Erjavec[2]

[1] Ľ. Štúr Institute of Linguistics, Slovak Academy of Sciences, Bratislava, Slovakia
[2] Jožef Stefan Institute, Ljubljana, Slovenia

garabik@kassiopeia.juls.savba.sk, jan.javorsek@ijs.si, tomaz.erjavec@ijs.si

Increasing computing requirements for acquiring and processing large data-sets and working with big corpora in Natural Language Processing (NLP) and related disciplines allow us to work on NLP algorithms and tasks that were impractical just a few years ago. The core of the problem is shifting from obtaining access to enough computation power and from optimizing algorithms into developing efficient ways of allocating the computing resources to various tasks and into finding efficient ways of dealing with huge amounts of data. The Grid network provides a good distributed environment for heavy-duty computational tasks, oriented towards the use by scientific community, and the possibility of using it for NLP-related tasks is obvious. Typical NLP tasks that could benefit from the Grid environment include language modelling (including algorithm and language model training, rather computationally demanding tasks), data conversion and indexing (especially when dealing with huge text corpora), and various kinds of language analysis. Some of these jobs can be easily parallelized or need to be run simultaneously many times with different parameters. Conversely, querying corpora is an interactive task that depends on low latency response and is best carried out by dedicated servers outside of the Grid environment.

Contemporary NLP tasks are rather varied; some of them require a lot of "pure" computing power, but many tasks, especially in the area of corpus linguistics, merely process large data files. From the software point of view, the tools used are very diverse -- they are often programmed in typical computer languages, like C or C++, but a lot of data processing is done in scripting languages, such as Perl or Python, and Java is increasingly popular, and more often than not, one specific task uses several different tools bound by short programs written in a shell script. From this follows than the tools are often fragile and require a specific environment, which sometimes means that even using a different GNU/Linux distribution that the one the software has been developed on can be a major problem.

Moreover, the actual deployment of Grid computing in the natural language processing area faces specific legal issues -- the data being processed are in majority of cases copyrighted, and the research institutions either have very strict legal agreements governing the use of the data, or are operating entirely on copyright law sections allowing scientific and research use of the data. The situation is somewhat simi-

lar to the problems the users of Grid computing in health care systems -- though in that case, metadata are the most sensitive and protected part of the data-set, while in corpus linguistics the data (i.e. texts) are sensitive, but the metadata is usually freely accessible.

In the extreme (but very frequent) case, the research institution using the data for research does not have the right to distribute the data at all. However, it might be still advantageous to use the Grid infrastructure for computing clusters of the institution itself, and use middleware functions to restrict data-replication to those processing nodes and data storage elements physically located in the organization. This way, the whole Grid can still be used for less sensitive tasks, or for post-processing the results of operations on sensitive data. It is nevertheless desirable to protect the data leaving the organization premises from casual snooping.

Sensitive data has to be suitably protected where it is permanently stored. Therefore we propose to store the corpus data in encrypted form in a dedicated storage element and set up the access authorization in such a way that access is restricted to VO (Virtual Organization) members who belong in a specific VO user group, where membership is restricted to those users who signed the necessary legal agreements to access the data. Furthermore, we propose that the data is transferred to the untrusted environment of Grid worker nodes, where jobs perform their computations, in the encrypted form and that the decryption keys are issued to the jobs protected with asymmetric encryption decryptable only by the job's Grid proxy keys so that only the jobs can access the keys and decrypt the data.

In this manner, access and decryption is regulated with the authorization of embedded VOMS attributes in the proxy certificate without any additional authorization steps, while the data is never shipped or stored in unencrypted form. If the tools used by the job have to store temporary files on disk, these are protected from other processes (with the exception of system administrators, who are already bound by strong agreements pertaining to data security on the Grid) and are in addition of short-lived nature. The simplest implementation of the system described involves the use of a decryption filter in the job script and is rather simple to deploy. A more flexible solution, based on CryptoSRM (cryptographic storage resource manager) and Hydra Key Storage (a distributed fragmented encryption key storage system) is possible.

From Grid point of view, the best way to use the specific software is to install it inside a runtime environment which is made available to the jobs when submitted to the Grid. This is directly supported by the Grid infrastructure and requires no additional steps or privileges. However, at this time this requires a significant effort, since all the tools and their dependencies have to be compiled (or installed in a non-standard location inside the runtime environment) on the standard SLC distribution, which can be problematic if the software has many external dependencies.

There are two possible solutions: to run under a chroot environment or to use virtualization, and several different approaches that lie somewhere in between those two extremes, ranging from paravirtualization, which requires cooperation from the guest operating system kernel, used e.g. by the XEN virtualization solution; to compartmentalization (e.g. Linux virtual servers and OpenVZ), which divides the host operating system into different compartments with completely separated processes, network

access and file systems but sharing the same kernel; to vanilla kernel namespace support, which only separates user and process management (slightly extending chroot separation).

Many of the commonly used distributions already have support for (at least partial) installation inside a chroot environment built in. But in the context of Grid infrastructure this solution has a significant disadvantage, since it requires support from the cluster administrator since chroot environments are not a standard feature of the Grid environment. Use of the chroot environment neither supports transparent encryption of the filesystem in such a way that the files are not directly accessible from the administrator of the host environment.

However, installing and using virtual machines requires not just administrator cooperation, but often also nonstandard host operating system extensions (such as special kernel modules). One of the more interesting virtualization systems in this context is User Mode Linux, which does not require any special host support, runs as an ordinary user process and provides a complete guest Linux kernel environment. Unfortunately, guest environment in this case suffers from a big I/O performance degradation, which can be a noticeable problem when dealing with very large corpus data.

While there is significant research in the use of different kinds of virtualization in the context of Grid technologies, this is not a wide spread feature at this time.

However, in order to truly exploit the Grid potential, we envisage a scheme where the linguistic data (especially text corpora) are stored on the Grid infrastructure as well and the existing Grid access control infrastructure is extended in order to be provide secure access to the data to third parties interested in accessing the data in such a way that all the limitations and conditions arising from the copyright law and other binding agreements are met. For this, it might be necessary to devise a standardised system of efficient storing and accessing of sensitive data.

Our vision for ergonomic use of the Grid infrastructure is to be able to create independent virtual machines with arbitrary GNU/Linux distribution inside (or even with other operating systems), optionally with encrypted base file system image, with transparent access to encrypted data residing at the Grid storage (in some limited and hackish manner, this is possible to achieve with existing tools -- using Usermode linux as a virtual machine not requiring any elevated privileges on the host system, and combination of hostfs and encryptfs to access the encrypted data on the host file system). Taking the idea one step further, the virtual machines can present to the guest operating systems a massively parallel multiprocessing environment that the guest applications can take advantage of and that are distributed across many Grid nodes in the real hardware.

Objectives for Migration and Operation Support of Legacy Applications in Cloud

Zoltán Balogh, Emil Gatial, Ladislav Hluchý

Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

{zoltan.balogh, emil.gatial, ladislav.hluchy}@savba.sk

Abstract. A new model for operating Information Technologies (IT) is heading towards a model of infrastructure, platform and software rental for required time and in required capacity. Cloud computing is capable to provide such services while providing higher reliability and lower costs as the traditional IT operation model. Because of the mentioned reasons there is a trend in migration of applications to the Cloud. The aim of this article is to define requirements for a platform for migration and operation support of applications into the Cloud. The article aims to focus on applications for legacy platforms, e.g. platforms which were not primarily intended for deployment in Cloud. The core requirement is a semantically consistent description of resources in Cloud - such description would enable to define specification of platforms, application requirements, dynamic discovery and generation of platforms as well as automatic migration of applications to alternative platforms. Such concept implementation would ease migration and operation of applications in Cloud thus provide more potential users access to Cloud as well as simplify entry of new Cloud services providers to the market.

Section 4
Astronomy & Astrophysics and High energy Physics

The formation of ice giants in the solar nebula with Jupiter and Saturn

Marian Jakubik[1], Lubos Neslusan[1], Jan Astalos[2]

1 Astronomical Institute of the Slovak Academy of Sciences, 05960 Tatransk´a Lomnica, Slovakia
2 Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

mjakubik@ta3.sk, ne@ta3.sk, jan.astalos@savba.sk

ABSTRACT. The formation of ice-giant planets in the Solar System, Uranus and Neptune, is still unsolved problem of the Solar-System cosmogony. The observed chemical composition of these planets, consisting mainly of the chemical elements heavier than helium, indicates that they had to accrete after the dissipation of once existing gaseous, hydrogen and helium dominated, solar nebula. It means that the Uranus and Neptune formed in an era, when the gaseous giants, Jupiter and Saturn, were already formed.

In the presented contribution, we describe our work intended to determine the formation sites of the Uranus and Neptune in the solar nebula at the end of its dissipation taking into account a still acting gas drag as well as the gravitational perturbations of Jupiter and Saturn.

We especially describe the usage of the GRID computational system of a number of independently working CPUs, which is suitable to be used and was used to compute this task. The usage of the GRID in this task is forced by a large extent of algebraic operations over a large bulk of data. The GRID helps us to perform the calculations in a reasonable time also with respect of our requirement to make not only one, but 8 runs. In each run, three massive objects, the Sun, Jupiter, and Saturn, acts each on other and on the 4946 TPs considered. The dynamical behaviour of each TP is not influenced any other TP, therefore the motion of a subset of TPs can be numerically integrated independently, on a separate CPU. In this context, we can classify our task as a "separable extensive task".

We design each run to be computed on 100 CPUs. Thus, 4946 TPs can be divided into 100 subsets, each consisting of 49 or 50 TPs. Within a given subset, 6 interactions among the massive objects and 3 times 49 (or 3 times 50) interactions between the massive objects and TPs are to be computed in every integration step. In addition, the gas drag effect is also calculated at each TP. The massive bodies influence each other, therefore the integration of their motion cannot be separated. We solve this problem repeating this particular integration, comprising 6 interactions, on every CPU. Another 147 or 150 integrations can, however, be computed independently. Although the repetition of 6 interactions enlarges the total computational by cca 6/148.5, i.e. about 4%, the possibility of the usage 100 CPUs simultaneously allows

to complete the computation in a much shorter real time than would be the time necessary for a single-CPU computation without the repetition.

Two of the runs are alternatives, with varying input parameters. The part of computations covering these three runs can be classified as a "parametric task".

According our result, there existed two regions, in 9-13 and 15-18 AU of the accumulation of matter, if the macroscopic counterpart of the solar nebula, i.e. protoplanetary disc, was dominated by relatively small-sized objects.

Dynamics of streams of chosen comets

Dušan Tomko, Luboš Neslušan, Marián Jakubík

Astronomical Institute, Slovak Academy of Sciences, 05960 Tatranská Lomnica, Slovakia

dtomko@ta3.sk, ne@ta3.sk, mjakubik@ta3.sk

ABSTRACT. When a comet travels near the Sun, it develops its coma and dust tail and can lose several hundred million tons of dust and vapor. The closer it gets to the Sun, the more solids and gases are released. This material remains in orbit around the Sun, and the solid pieces are called meteoroids. A meteoroid becomes a meteor when it falls through the atmosphere, and we see it shooting across the sky.

The motion of meteoroids can be perturbed by the gravitational fields of major planets. We know, the Jupiter's gravitational influence is capable to reshape an asteroid's orbit from the main belt so that it dives into the inner solar system and crosses the orbit of Earth. A similar large change of orbit can also happen at meteoroids.

The majority of short-period comets have associated a meteoroid stream. However, we can usually observe only the meteors in the Earth's atmosphere originating from the bodies in an orbit passing in a vicinity of the orbit of our planet. The distant streams cannot be usually detected on the Earth. Nevertheless, the orbits of some meteoroid streams can still be modified in such a way that the meteoroid can collide with the Earth.

In this contribution, we deal with several such the distant theoretical streams.

We model a theoretical stream associated with chosen comets, which we assume to be the parent bodies of meteoroid streams, and study the dynamics of the modeled meteoroid particles. Particulars of tested particles were monitored with numerical integration. The perturbations from 8 planets are considered, but non-gravitational forces are not considered in the modeling.

The simulation was performed in the following steps:

1. The integration of the parent body into the past over the period equal to 1000 orbital revolutions of the theoretical parent body.
2. The modeling of theoretical meteoroid stream in the time of perihelion passage.
3. The numerical integration of all theoretical particles up to the present.
4. Identification of the modeled particles orbits with the orbits of actual, photographically observed meteors.

The integration of theoretical particles is performed using the GRID computational system. The results of the modeling procedure are components of the position and velocity vectors. The data are divided into 100 parts and every part includes 100 test particles.

To integrate each part one processor in the GRID is utilized. Every input file includes: input files with parameters for the integration and file containing the input

data on test particles, parent body, and perturbing planets. Specifically, it includes the components of position and velocity vectors of 100 particles, parent body, and 8 big planets.

The computational time of a single run is about 7 hours. So, the total computational time using a single CPU would be 700 hours (29 days) of the stream of one comet. The GRID helps us to perform the calculations in a reasonable time.

We have been studying 4 comets: 122P/de Vico, 126P, 161P and 149P. In our paper, we describe the dynamics of modeled theoretical meteor stream associated with comet 122P/de Vico. From 10000 test particles only 897 approach the Earth's orbit to 0.05 AU or less. The radiant of this stream is characterized with: RA= 317.871°, DEC= 46.030°, Vg = 47.182 km/s. The maximum of the activity of predicted meteor shower is on June 28.49924. The main step is the identification of the orbits of modeled particles with the orbits of actually photographed meteors in the IAU Meteor Data Center (IAU MDC) or with the orbits determined from radar observations. The number of identified particles is low. The meteoroid stream asociated with comet 122P/de Vico is not proved with the IAU MDC data.

Section 5
Grid and Service-oriented Computing

Performance analysis of Cloud middleware

Binh Minh Nguyen, Viet Tran
Institute of Informatics, Slovak Academy of Sciences
Email: minh.ui@savba.sk

In the recent years, cloud computing became more and more topical for research also for industry. Although the idea of providing computational resources via Internet dynamically on demand and paid per use was mentioned in also in grid computing, it is fully realized in cloud computing. One of the reasons is the maturity of virtualization technologies including hardware support for virtualization like Intel VT-x and AMD-V. Today, many companies already provide services (infrastructure, software) via cloud computing, most notably Amazon (EC2, S3), Google Apps, Microsoft Azure.

Cloud computing can be classified into three levels: infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS). In IaaS, the computer infrastructure (hardware, network) is often virtualized and delivered via internet as a service to users. The users can specify in their request the hardware configuration, receive full access to the required hardware, use it and pay only the duration the hardware is used. PaaS, as is in its name, delivers a platform (e.g. Java, Python, .NET) where the users can write their applications and deploy them on the servers provided by service providers. In SaaS, applications are delivered via internet to users, so the users can use them without installation or servers on their side.

Beside the public clouds, provided by external service providers like Amazon, Google, there are also private clouds, i.e. the clouds provided by the same organizations/companies which use them internally. There are several open-source cloud middleware that every organization can use for creating a private cloud. Private clouds can be useful for companies, which by some reasons cannot use public clouds (e.g. problems with data security or privacy) but want to exploit the advantages of cloud computing.

In this paper, we will focus on performance analysis of two existing middleware for creating private or hybrid cloud: Eucalyptus and OpenNebula. We will test how the virtualization technologies affect the performance of hardware, as well as the effect of cloud middleware.

Eucalyptus is an open source cloud middleware that most resemble Amazon EC2 interface. The middleware implements many features of EC2 like creation of virtual machines, images management, credential management, storage management, elastic IP address and so on.

OpenNebula, on the other hand, was implemented as software for management of virtual infrastructure. While the middleware provide rich features for managing virtual machines as well as interfaces programming and adding new features, it still lacks of features for full user management and storage management.

Our tests consist of several parts: CPUs, disk performance and network performance. For performance test of disk operations, we use Bonnie++ and for network benchmarking, we use netperf tool. All tools are open sources and available for Ubuntu 10.4, which we are used as operating system for both host and virtual machines.

Our tests shows the virtualization technologies perform well in term of CPU and disk performance. The network performance is one of the most critical point from the view of high performance computing: while the network throughput of virtual machines is still comparable with the physical hardware, the long latencies caused by virtualization can degrade performance of parallel applications.

The technology of information security in grid computing systems

Aleksandr Palagin[1], Nadir Alishov[1], Aleksandr Gromovski[1], Sergei Zinchenko[1], Vitaliy Marchenko[1], Aleksandr Mischenko[1]

[1] V.M.Glushkov Institute of cybernetics of NAS of Ukraine, Prospekt akad. Glushkova 40, 03680 Kiev, Ukraine;

palagin_a@ukr.net, anio_n@mail.ru, _zot_@mail.ru, zinchenkosv@gmail.com, marchenkov@ukr.net

Abstract. The paper presents an approach to solving problems of security in grid computing systems. Considered problems of creating a complex security based on cryptographic hardware and software security solutions, data store, data transfer over the network and streaming media over the network.

Keywords: grid computing system, security, cryptography, network, protocol.

1 Introduction

Grid - a geographically distributed infrastructure that unites many different types of resources (processors, and long-term memory, storage and databases, networks), access to which you can get from anywhere, regardless of their location. Grid requires a collective shared mode access to resources and related services in the globally distributed environments (virtual organizations), consisting of companies and individuals that share common resources. In each virtual organization has its own policy conduct of its members who must abide by the rules. Virtual organization can be formed dynamically and have a limited lifetime. This kind of organization of distributed computing systems gives rise to a complex problem of security as the functioning of the system and data security in it. In addition, a significant factor is the "integrity" of data to be online analytical processing. Addressing security issues in distributed computing systems require new technical approaches. In this paper propose the technology of security in distributed computing systems based on hardware with advanced security subsystem information.

References

1. Dolev D., Yao A.C. On the security of public key protocols // Proc. of the IEEE 22nd Annual Symposium on Foundations of Computer Science. – 1981. – P. 350–357.
2. Kimberly Getgen. Encryption and Key Management Industry Benchmark Report. – www.trustcatalyst.com 2009. – 33 p.

3. Alishov N.I., Marchenko V.A., Orudjeva S.G. Indirect steganography as a new mode of transmission of classified information (Косвенная стеганография как новый способ передачи секретной информации) // Komp'yuterni zasobi merezhi ta systemi – Kiev.: NAS, Institute of cybernetics, 2009. – № 8. – P. 105–112.
4. Cryptographic Service Providers. – http://msdn.microsoft.com/en-us/library/aa380245%28VS.85%29.aspx
5. Intercepting API functions on Windows (Перехват API функций в Windows). – http://www.wasm.ru/article.php?article=apihook_1
6. ISO/IEC 9594. The Directory: Overview of concepts, models and services. – 2008. – 23 p.
7. John T. Kohl, B. Clifford Neuman, Theodore Y. Ts'o. The Evolution of the Kerberos Authentication Service // Distributed Open Systems. – IEEE Computer Society Press. – 1994. – 78–94p.
8. Goto A. Safe and Secure Ubiquitous Communication. – international workshop on network security and wireless communications 27 Jan 2005.– http://www.it.ecei.tohoku.ac.jp/~kato/workshop2005/NTT-goto-slides.pdf.
9.Network Security: Know It All / J. Joshi, S. Bagchi, B.S. Davie et al. – Morgan Kaufmann. – 2008. – 368p.
10.Network Security: Know It All / J. Joshi, S. Bagchi, B.S. Davie et al. – Morgan Kaufmann. – 2008. – 368p.
11. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", July 2003. http://tools.ietf.org/pdf/rfc3550.pdf

Advanced configuration with DIANE

Viet Tran
Institute of Informatics, Slovak Academy of Sciences
Email: viet.ui@savba.sk

Over the last years, the development and acceptance of Grid technologies have been forwarded incrementally. Grid technologies connect distributed computational resources of dynamic multi-institutional virtual organization together and provide aggregate computational powers for solving very complex and computation demanding problems. The technologies make the infrastructures for researchers to share resources and knowledge, allows them to collaborate on solving common problems.

One of largest classes of applications running in the Grid is parametric studies. These applications have a (very) large number of independent tasks with the same code and different input/output data. As these tasks do not require communications among each others, they can be easily distributed among available computation nodes in Grid and are executed in parallel. The speedup of such execution is practically equal the number of processors in the infrastructure located for the applications. In other word, the speedup is virtually unlimited if the number of tasks is large enough, the more processors are available for the application, the faster the application will run.

DIANE is a framework for running parametric study developed by CERN. In comparison with built-in gLite support for parametric study, DIANE improves the reliability and efficiency of job execution by providing automatic load balancing, fine-grained scheduling and failure recovery. The model is based on master-worker architecture. This approach is also known as agent-based computing or pilot jobs in which a set of worker agents controls the resources. The resource allocation is independent from the application execution control and therefore may be easily adapted to various use cases. DIANE uses the Ganga interface to allocate resources by sending worker agent jobs, hence the system supports a large of computing backends: LSF, PBS, SGE, Condor, LCG and gLite in EGEE.

In this paper, we will introduce some advanced configuration of DIANE for parametric study. These cases include separating DIANE master from gLite job submission, connection with Windows machines and using DIANE with some workflow managers.

Running Parallel MATLAB on EGEE Grid

Peter Kurdel, Jolana Sebestyénová

Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia
peter.kurdel@savba.sk, sebestyenova@savba.sk

Abstract. The ability to solve very large problems by scaling computer programs to run on multi-core workstations, clusters, grids, and clouds can help engineers gain significant research and competitive advantages. The MathWorks parallel computing tools let users develop Matlab and Simulink applications that can work in a variety of environments, from a multi-core desktop to computer clusters and grids. MATLAB Distributed Computing Server enables users to execute the same MATLAB codes and Simulink models in the grid that were executed on the desktop. MATLAB supports implicit as well as explicit multi-processing. Scheduler manages interaction between user desktop and cluster computers.

1 Introduction

Parallel computing technologies offer engineers the means to accelerate solutions of their computational problems by using multiple hardware resources [3]. The ability to solve very large problems by scaling computer programs to run on multi-core workstations, clusters, grids, and clouds can help engineers gain significant research and competitive advantages.

The Enabling Grids for E-sciencE project is no longer active. The distributed computing infrastructure built by the projects DataGrid (2002-2004), EGEE-I, -II and -III (2004-2010) is now supported by the European Grid Infrastructure [5]. This long-term organisation coordinates National Grid Initiatives, which form the country-wide building blocks of the pan-European Grid.

The Slovak National Grid Initiative SlovakGrid on a regular basis organises conferences, workshops, training courses, and information days for researchers, scientists, students, and practitioners interested in high performance computing, especially in using EGEE grid. Recently, one of the main aims of such meetings was to find out which software packages are the most required and useful for future users of Slovak national Grid infrastructure. Huge interest was displayed to usage of MATLAB and ANSYS systems.

MATLAB (matrix laboratory) [8] is a numerical computing environment and fourth-generation programming language. It is halfway between a programming language (a user must do everything) and a menu-driven application [1] (the user only makes high level decisions). The MathWorks parallel computing tools let users develop Matlab and Simulink applications that can work in a variety of environments,

from a multi-core desktop to computer clusters and grids. MATLAB Distributed Computing Server enables users to execute the same MATLAB codes and Simulink models in the grid that were executed on the desktop.

References

1. Burkardt J.: MATLAB Parallel Computing, May 2009, http://people.sc.fsu.edu/~jburkardt/presentations/fdi_2009_matlab.pdf
2. Chakravarti Arjav J., Grad-Freilich Silvina, Laure Erwin, Jouvin Michel, Philippon Guillaume, Loomis Charles, Floros Evangelos: *Enhancing e-Infrastructures with Advanced Technical Computing*: Parallel MATLAB® on the Grid, http://www.mathworks.com
3. Chakravarti A., Ivanov P.S. and Rorison J.: Scaling Beyond the Desktop with MATLAB - Parallel MATLAB apps make programming simple and portable, *Technology for Design Engineering*, http://www.deskeng.com/articles/aaarpz.htm, June 30, 2009
4. Cikovskis L., Znots E.: Running Licensed Software on Grid, *2nd BalticGrid-II AHM*, 13.05.2009, Riga, http://2ahm.sigmanet.lv/uploads/workshop_licenced_soft-1.ppt
5. European Grid Infrastructure http://www.egi.eu/
6. Floros V.: gLite and MATLAB® Integration, Parallel Computing with MATLAB R2008B on EGEE cluster for Windows Client, 09-Jun-2009, https://twiki.cern.ch/twiki/bin/view/EGEE/GLiteAndMATLABToolkitIntegration
7. Jejkal T., Stotzka R., Sutter M., Hartmann V.: GridMate – The Grid Matlab Extension, http://www.ipe.fzk.de/~stotzka/publications/publications/jejkal2009.1.pdf
8. MATLAB http://www.mathworks.com/products/matlab/
9. Samsi Siddharth: Parallel computing using MATLAB, *SC08* Engineering Track, http://sc08.sc-education.org/conference/engineering/

Section 6
Computational Chemistry and Material Science

Distributed Way of Biomolecular Conformational Space Exploration - Experience of grid - CICADA Tool Utilization

Petr Kulhánek

Faculty of Natural Science at Masaryk University in Brno, Czech Republic

kulhanek@chemi.muni.cz

The use of NAMD cluster computing in molecular dynamics study of inhibition mechanism for aldose reductase

Magdaléna Májeková[a], Miroslav Dobrucký, Peter Slížik

[a] Institute of Experimental Pharmacology and Toxicology, Slovak Academy of Sciences, Institute of Informatics, Slovak Academy of Sciences, Bratislava
exfamaje@savba.sk

The enzyme aldose reductase (ALR2) plays an important role in the polyol pathway of glocose mechanism and is related to long-term diabetic complications. The inhibition of ALR2 is expected to have large biological impact and is under detailed study.
We started the simulation of one step in the inhibition mechanism, which could explain the role of so called “safety-belt” in the ALR2 structure. For this purpose we used the program package NAMD for scalable parallel molecular dynamics calculations. The program was implemented in computer cluster of the Institute of Informatics, later intended to be used in Grid infrastructure currently developed.
The pdb input 2J8T was taken from BPDB and parametrized by VMD package in charmm27 force field. Several solvation shells were added to ensure the mobility of the system in expected state. The pure protein + water system was tested in several time periods.

Section 7
Distributed Computing and Large Scale Simulations

Mathematical Model of Multicriteria Scheduling in Grid Environment as a Tool to Determine Efficiency of Scheduling Algorithms: Improved Strong Pareto Evolutionary Algorithm (SPEA2)

Michal ULBRICHT, Michal MURÍN

University of Žilina, Faculty of Management Science and Informatics, Žilina, Slovak Republic
{michal.ulbricht, michal.murin}@fri.uniza.sk

Extended Abstract

Since the first publication of scheduling appeared in 1954 it was formulated many problems on this subject but the difference was only in the environment of machines, restrictive conditions and objective function. Until the later eighties there was in scheduling known only one criterion. This approach was in many areas insufficient and it leads to the development of the multicriteria scheduling – also in grids. Grid can be defined as hardware and software infrastructure providing reliable, continuous, consistent, pervasive and inexpensive access to computational resources.

There are many options how the grid environment can be designed. A Typical method is when the global scheduler is the highest in the hierarchy and makes decisions based on information from the grid information service, which resource will be allocated to the task. Global scheduler then sends this task to the local scheduler, which decides the way the task will be scheduled in the resource in its autonomous domain. For simplicity of model presentation, this environment will be simplified to only one scheduler (with queue containing tasks that are waiting to be planned) and a basic set of entities - a set of users who assign jobs, a set of jobs assigned by users, a set of resource providers who provide resources and a set of resources. Global scheduler does not schedule jobs dynamically as they come to the system but statically in batch of *N* jobs. Local resource managers inform global scheduler about properties and reservation times of resources they are responsible for.

There will be a multicriteria mathematical model for scheduling tasks in grid environment presented as a tool to determine efficiency of multicriteria scheduling algorithms – in this paper improved strong pareto evolutionary algorithm. Both, mathematical model and SPEA2 uses different approach how to solve problem. In a case of mathematical model there is a priori optimisation and the result is an optimal solution. Disadvantage of solving mathematical models with methods of mathematical programming is solving time. On the other hand in case of algorithms like SPEA2 can be solving time extremely shorter but solution is not certain optimal. SPEA2 uses a posterior optimization and contractor gets a set of solutions where can choose one which is best for his preferences.

In a commercial sphere a contractor often asks how long does it take to complete the task and how much does it cost. Assuming that faster calculation would be more expensive it is quite logical question. Presented model contains two criteria – computation time and computation cost for each job but more criteria can be simply added. Users can set their preferences in different ways, but generally we can distinguish two ways: 1. preferences can be determined explicitly or 2. preferences are discovered based on previous experience. In the first case, users determine the importance of each criterion by ranking it (where first is the most important and last is the least) or weight is assigned to each criterion explicitly. In the second case, in phase of learning the user gets a group of possible solutions and evaluates them. The global scheduler can learn user preferences. According to the knowledge thus obtained global scheduler can then decide automatically. The choice of method depends on two factors: 1. whether users are aware of their preferences and know how to express them and 2. whether their preferences are stable. If user changes preferences with each task, the automatic learning for the global scheduler is very complicated. And this is very common situation in grids where user criteria vary according to personal plans, budget and various deadlines. Especially for this reason will be in this model chosen approach, where users can enter their preferences explicitly.

Compared algorithm will be SPEA2. It is one of the most important multi-objective evolutionary algorithms (MOEA) that use elitism approach. In the design of SPEA2, the goal was to eliminate the potential weaknesses of its predecessor and to incorporate most recent results in order to create a powerful and up-to-date MOEA. Concerning fitness assignment scheme of SPEA2, for each individual is assigned a raw fitness calculated on basis of the strength value of solutions who dominate it. To discriminate between individuals having identical raw fitness values additional density information is incorporated. To form the next generation, SPAE2 combines offspring and current population. Subsequently, the best individuals in terms of non-dominance and diversity are chosen and added to archive. SPEA2 also uses a truncation and adding procedure to achieve chosen size of archive.

References

1. I. Foster and C. Kesselman. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, 1998. ISBN: 1-55860-475-8.
2. Michael Pindeo. Scheduling: Theory, Algorithms and Systems. Springer, 3-rd edition, July 2008. ISBN: 978-0-387-78934-7.
3. Krzysztof Kurowski, Jarek Nabrzyski, Ariel Oleksiak, and Jan Weglarz. Scheduling jobs on the grid – multicriteria approach. Computational Methods in Science and Technology, 12(2):123–138, 2006.
4. Abdullah Konak, David W. Coit, Alice E. Smith. Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering & System Safety Volume 91, Issue 9, September 2006, Pages 992-1007 Special Issue - Genetic Algorithms and Reliability. ISSN: 0951-8320
5. Eckart Zitzler, Marco Laumanns, and Stefan Bleuler. A Tutorial on Evolutionary Multiobjective Optimization. Metaheuristics for Multiobjective Optimisation. Springer, 2004. ISBN: 9783540206378.

Application Multi-Model in Simulation of Mathematical Models

Igor Kvasnica[1] and Viera Šipková[2]

[1] Alexander Dubček University of Trenčín
Faculty of Mechatronics, Department of Informatics
Študentská č. 2, 911 50 Trenčín, Slovak Republic
kvasnica@tnuni.sk
[2] Ústav informatiky, Slovenská akadémia vied
845 07 Bratislava, Dúbravská cesta 9, Slovak Republic
viera.sipkova@savba.sk

Abstract. In the last years most processors have been constructed as a processing system composed of multiple independent cores, and computing clusters as a collection of multi-core nodes. The composition of cores in multi-core architecture shows a great variety. Homogeneous multi-core systems include only identical cores, unlike heterogeneous, which may include a mixture of different special-purpose cores with complex memory hierarchies. Multi-core systems promise further performance and efficiency gains, however, only in case the application code is designed for exploiting all of the available hardware components simultaneously in effective way. Therefore, the investigation of new parallel algorithms and programming models is fundamental. This paper describes various parallel software techniques and programming models used in mathematical modeling of simulation applications.

1 Introduction

In the last several years most processors have been constructed as a processing system composed of multiple independent cores, and the number of cores is predicted to be increasing exponentially over time. The forecast is to include hundreds or thousands of them within a few years. The composition of cores in multi-core architecture shows a great variety. Homogeneous multi-core systems include only identical cores, unlike heterogeneous, which may include a mixture of different special-purpose cores with complex memory hierarchies. Cores can be coupled together tightly or loosely; for example, they may or may not share caches, and they may implement message passing or shared memory inter-core communication. Multi-core systems promise further performance and efficiency gains, however, only in case the application code is designed for exploiting all of the available hardware components simultaneously in effective way. The second important observation is that managing memory hierarchies is critical too, memories are to be managed so that cores are not starved for data. Regarding the forthcoming clusters and grids, there is a trend of building powerful clusters consisting of a collection of complex many-core nodes equipped eventually with GPU or other accelerators. In the process of the software development the selection of an appropriate programming model which is capable to abstract the underlying architecture, is significant. New methodologies and mechanisms enabling the highly parallel execution of applications need to be investigated.

This work outlines some of the best practices applied in the development of parallel applications implementing distributed mathematical models. They originated mostly from the area of high performance computing, but they can be combined with modern technologies to obtain a powerful solution and performance.

2 Principles of Parallel Programming in Mathematical Models

According to opinions of S. Akhter and J. Roberts [1]: "*The entire concept of parallel programming centers on the design, development and deployment of threads within an application and the coordination between threads and their respective operations*". Threads enable multiple operations to proceed simultaneously. Breaking a program down into individual tasks which can be executed in parallel, including also the definition of task dependencies, is known as *program decomposition*. Basic forms how to decompose a program are: the *decomposition by task*, *decomposition by data*, and *decomposition by data flow*.

The use of threads can improve performance substantially, however, managing the simultaneous processes and their possible interactions is far from a simplicity. Major challenges which are always required to solve are: *communication*, *synchronisation*, *load balancing*, and *scalability*.

Within the flight simulator [2] mathematical models were modified to satisfy the given conditions as stated above. The equation (1) rates the speed increment, it defines the change speed from the fuel supply (2), and increment speed from the attack angle (3).

$$\Delta V(s) = -G_{V/\delta_T}(s)\Delta\delta_T(s) - G_{V/\delta_B}(s)\Delta\delta_B(s) \ . \tag{1}$$

For fuel supply:

$$G_{V/\delta_T}(s) = 5\frac{s^3 + 1,12s^2 + 62,78s + 25,32}{s^4 + 1,13s^3 + 62,80s^2 + 28,66s + 4,09} \ . \tag{2}$$

For elevator:

$$G_{V/\delta_B}(s) = \frac{-0,11 \cdot (9,81s + 620,97) - 0,42 \cdot (-9,81s - 10,01)}{s^4 + 1,13s^3 + 62,80s^2 + 28,66s + 4,09} \ . \tag{3}$$

The increment attack angle rewrites equation (4), defines its from change fuel supply (5) and increment speed from change of position elevator (6):

$$\Delta\alpha(s) = -G_{\alpha/\delta_T}(s)\Delta\delta_T(s) - G_{\alpha/\delta_B}(s)\Delta\delta_B(s) \ . \tag{4}$$

For fuel supply:

$$G_{\alpha/\delta_T}(s) = 5\frac{0,002s^2 - 0,2518s - 0,1}{s^4 + 1,13s^3 + 62,80s^2 + 28,66s + 4,09} \ . \tag{5}$$

For elevator:

$$G_{\alpha/\delta_B}(s) = \frac{-0,11 \cdot (-s^3 + 0,89s^2 + 0,012s - 2,45) - 0,42 \cdot (-s^2 - 0,41s - 0,025)}{s^4 + 1,13s^3 + 62,80s^2 + 28,66s + 4,09} \ . \tag{6}$$

BENEFITS IN WORKLOAD VIRTUALIZATION AND INFORMATION VIRTUALIZATION FOR SIMULATION IN GRID

Igor KVASNICA[1], Peter KVASNICA[2]

[1] Regional Department for Environment Issues of Trenčín,
Hviezdoslavová č. 3, 911 33 Trenčín, Slovak Republic,
kvasnica.igor@tn.kupz.sk
[2] Alexander Dubček University in Trenčín, Center of information technologies and Faculty of Mechatronics, Department of Informatics,
Študentská č. 2, 911 50 Trenčín, Slovak Republic,
kvasnica@tnuni.sk

Abstract. The paper deals with the method of workload and information virtualization for simulation in grid computing. Policy driven resource management and datacenter architecture has been proposed how service oriented approach in emission air induce their additional utilization. Availability this architecture is certainly an area, which is essential to grid computing. This essential is complicated to manage the running processes of the model in grid architecture. The result of a change to the emission factor that reflects the existing situation in modeling methods and technologies. Users priorities associated with the a workload virtualization requiring them is essential to maximize software payment return on investment. Commercial grids today are mostly deployed at the enterprise level (i.e. within the intranet).

Keywords: grid technologies, workload virtualization, mathematical model, information virtualization, air emissions, proactive license.

1 INTRODUCTION

Grid computing, in the commercial space, builds upon a set of management disciplines, which aims at mapping available resource capabilities to application workloads. This paper illustrates innovative technologies, used at environment research, that address key issues found in commercial grid environments. The workload virtualization technologies have four main areas, information virtualization, policy driven resource management, datacenter architecture, and policy management datacenters.
The simulation of mathematical model of the climate changes can be created using parallel and network computer architecture or grid computing.

2 GRID COMPUTING BACKGROUND

Grid computing allows for flexible, secure, and coordinated resource sparing among a dynamic collection of individuals, institutions, and resources [1]. Companies today use grid computing for better and faster decision making in emission air and climate change. The management of living environment requires to create, manage, operate and exploit commercial grids, are enabled through a set of layered components, depicted in Figure1 [2].

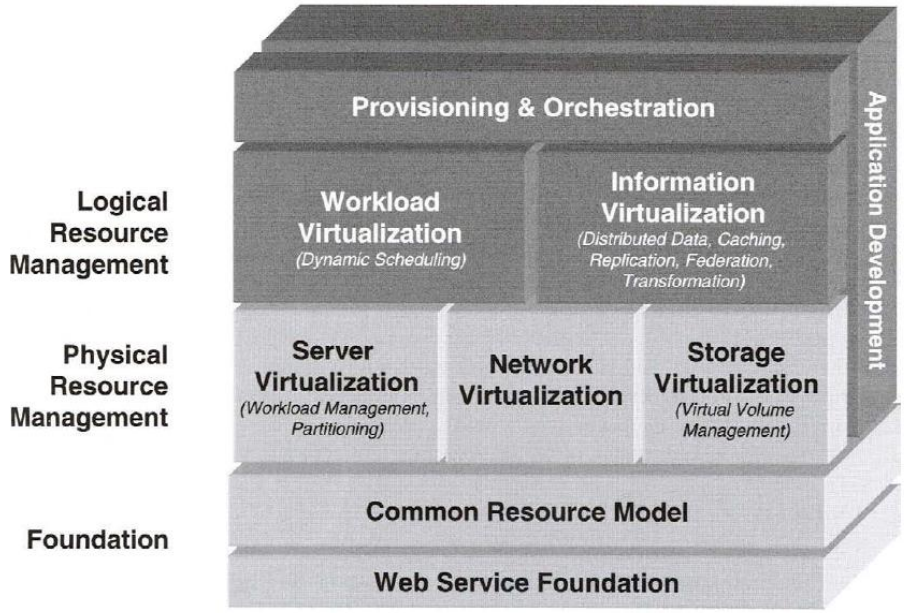

Figure 1. Commercial grid layered components [Prost, 2005]

At the core are the web services foundation, based on emerging standards, which specify how services are described, and choreographed [3], based on the Common Information Model (CIM) [4] specifications.
The next level up is the management layer of physical IT resources (servers, network, storage devices). At this level, we find services to define and configure virtual server partitions, virtual volumes of storage, virtual networks, etc. Some important are the workload virtualization, informatic datacenter architecture with policy their using.

3 WORKLOAD VIRTUALIZATION OF APPLICATIONS

We are currently focusing on advancing the application-flow execution management technology in support of both scientific workflow execution management and living environment process performance management needs in grid environments.

Rendering a Large Set of Graphical Data in Cluster Environment

František Hrozek, Branislav Sobota, Csaba Szabó, Štefan Korečko

Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 041 20 Košice, Slovakia
frantisek.hrozek@tuke.sk, branislav.sobota@tuke.sk,
csaba.szabo@tuke.sk, stefan.korecko@tuke.sk
http://kpi.fei.tuke.sk

In creation of various 3D visualizations and animations is necessary to process a large amount of graphical data. Rendering clusters are used to speed up this processing. These rendering clusters can speed up visualization rendering from several weeks to few days.

DCI KPI TU of Košice has been working on project in which was needed to process large amount of graphical data. This project was 3D city agglomeration visualization [1] and visualized city agglomeration area was the areal of Technical University of Kosice. Project has three parts:

- Modeling – in this part was created 3D model of visualized city agglomeration using,
- Presentation – in this part was created script which allows the smooth redraw of 3D model given by the transition path. Also serves for panoramic images creation,
- Visualization – in this part was created application, which uses created panoramic images for information displaying.

For panoramic images creation, in presentation part, was needed to render 14 480 images with photorealistic lighting and resolution 1280x1024 pixels. Rendering of all images on one PC (hardware configuration: CPU AMD Athlon(tm) 64 Processor 3700+, RAM 1GB DDR, GPU 256 MB GeForce 7300 GT) would take 202 days which was unacceptable and therefore was used cluster rendering.

Cluster-based rendering [2] in general can be described as the use of a set of computers connected via a network for rendering purposes, ranging from distributed non-photorealistic volume rendering over raytracing and radiosity-based rendering to interactive rendering using application programming interfaces like OpenGL.

Each cluster needs application that manages and monitors data which flows through network between connected computers. For this project was used Autodesk Backburner which is part of Autodesk 3ds Max [3]. Autodesk Backburner is the set of applications used to manage and monitor the Autodesk Backburner Distributed Queueing System.

The Autodesk Backburner Distributed Queueing Systems consists of the following components:

- An Autodesk application that sends jobs to the Distributed Queueing System (the Render Client)

- At least one Linux or Windows computer that does the rendering (the Render Node)
- A workstation that distributes and manages the jobs running on the Distributed Queueing System (the Backburner Manager)
- At least one workstation that monitors the jobs running on the Distributed Queueing System (the Backburner Monitor)

Rendering cluster used in project was built from 20 PC (Fig. 1.), with earlier mentioned configuration. Rendering time was 10.2 day, which was approximately 20 times faster then with one computer. Saved time was used for work on new projects and also for improving existing projects.

Fig. 1. Photo of rendering cluster.

Acknowledgment: This work is supported by VEGA grant project No. 1/0646/09: Tasks solution for large graphical data processing in the environment of parallel, distributed and network computer systems.

References

1. Hrozek, F., Sobota, B., Janošo, R.: 3D city agglomerations visualization. In: ICETA 2009 : 7th International Conference on Emerging eLearning Technologies and Applications : Information and communications technologies in learning : Conference proceedings : November 19-20, 2009, Stará Lesná, The High Tatras, Slovakia. Košice : Elfa, 2009. ISBN 978-80-8086-089-9
2. Sobota B., Straka M., Perháč J.: A visualization in cluster environment, Grid Computing for Complex Problem 2007, Bratislava, 22.10-23.10.2007, Bratislava, Ústav Informatiky SAV, 2007, tretia, pp. 68-73, ISBN 978-80-969202-7-3
3. 3ds Max 9 User Reference. URL: http://usa.autodesk.com/adsk/servlet/item?siteID=123112&id=10175188&linkID=9241177

Course

Course on Development of Grid Applications

Miroslav Dobrucký, Viera Šipková, Viet Dinh Tran

Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia
Miroslav.Dobrucky@savba.sk, Viera.Sipkova@savba.sk,
upsyviet@savba.sk

Programme schedule:

- Introduction to EGI (European Grid Initiative)
- Principles of cluster and grid computing
- Development of grid applications
- Overview of high-level grid tools
- Introduction to cloud computing