VILNIUS UNIVERSITY

Aleksandr Igumenov

ELECTRICAL ENERGY AWARE PARALLEL AND DISTRIBUTED COMPUTING

Summary of Doctoral Dissertation

Technological Sciences, Informatics Engineering (07T)

Vilnius, 2012

Doctoral dissertation was prepared at the Institute of Mathematics and Informatics of Vilnius University in 2007–2012.

Scientific Supervisor

Prof. Dr. Julius Žilinskas (Vilnius University, Technological Sciences, Informatics Engineering – 07 T).

The dissertation will be defended at the Council of the Scientific Field of Informatics Engineering at the Institute of Mathematics and Informatics of Vilnius University:

Chairman

Prof. Dr. Habil. Gintautas Dzemyda (Vilnius University, Technological Sciences, Informatics Engineering – 07 T).

Members:

Prof. Dr. Juozas Augutis (Vytautas Magnus University, Technological Sciences, Energetics and Power Engineering – 06 T),
Prof. Dr. Eduardas Bareiša (Kaunas University of Technology, Technological Sciences, Informatics Engineering – 07 T),
Prof. Dr. Romas Baronas (Vilnius University, Physical Sciences, Informatics – 09 P),
Assoc. Prof. Dr. Olga Kurasova (Vilnius University, Technological Sciences, Informatics Engineering – 07T).

Opponents:

Dr. Sergėjus Ivanikovas (Lithuanian University of Educational Sciences, Physical Sciences, Informatics – 09 P),
Prof. Dr. Dalius Navakauskas (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering – 07 T).

The dissertation will be defended at the public meeting of the Council of the Scientific Field of Informatics Engineering in the auditorium number 203 at the Institute of Mathematics and Informatics of Vilnius University, at 1 p. m. on 28th of September 2012.

Address: Akademijos st. 4, LT-08663 Vilnius, Lithuania.

The summary of the doctoral dissertation was distributed on 27th of August 2012.

A copy of the doctoral dissertation is available for review at the Library of Vilnius University.

VILNIAUS UNIVERSITETAS

Aleksandr Igumenov

LYGIAGRETIEJI IR PASKIRSTYTIEJI ELEKTROS ENERGIJĄ TAUSOJANTYS SKAIČIAVIMAI

Daktaro disertacijos santrauka

Technologijos mokslai, informatikos inžinerija (07T)

Vilnius, 2012

Disertacija rengta 2007–2012 metais Vilniaus universiteto Matematikos ir informatikos institute.

Mokslinis vadovas

prof. dr. Julius Žilinskas (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07T).

Disertacija ginama Vilniaus universiteto Matematikos ir informatikos instituto Informatikos inžinerijos mokslo krypties taryboje:

Pirmininkas

prof. habil. dr. Gintautas Dzemyda (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07T),

Nariai:

prof. dr. Juozas Augutis (Vytauto Didžiojo universitetas, technologijos mokslai, energetika ir termoinžinerija – 06T),
prof. dr. Eduardas Bareiša (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija – 07T),
prof. dr. Romas Baronas (Vilniaus universitetas, fiziniai mokslai, informatika – 09P),
doc. dr. Olga Kurasova (Vilniaus universitetas, technologijos mokslai, informatikos inžinerija – 07T).

Oponentai:

dr. Sergėjus Ivanikovas (Lietuvos edukologijos universitetas, fiziniai mokslai, informatika – 09P),
prof. dr. Dalius Navakauskas (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija – 07T).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties tarybos posėdyje 2012 m. Rugsėjo mėn. 28 d. 13 val. Vilniaus universiteto, Matematikos ir informatikos instituto 203 auditorijoje.

Adresas: Akademijos g. 4, LT-08663 Vilnius, Lietuva.

Disertacijos santrauka išsiuntinėta 2012 m. rugpjūčio mėn. 27 d.

Disertaciją galima peržiūrėti Vilniaus universiteto bibliotekoje.

1. Introduction

The relevance of the problems, the scientific novelty of the results and their practical significance as well as the aim and tasks of the work are described in this chapter.

Relevance of the Problem

The growth of electricity consumption raises problems related to electricity production and imports (when there not enough internal resources in the country). The majority of new technologies require larger and larger quantities of electricity. It can be seen everywhere, but especially the problem highlights when speaking about the modern information computer technologies. Newer devices are released and consume less electric power, but the amount of such equipment is increasing, which increases consumption of electric power. Consumption growth also concerns the distributed and parallel computing. In the world, every year, the number of supercomputers, clusters, grid and cloud computers is growing up. All these resources require a lot of electricity. Same applies to the most powerful computer in the world, "K computer". It requires 12 659.89 kW of electricity when running at full power. That much electricity would be sufficient to maintain about 60 000 usual households.

Parallel computing has been developed and evolved rapidly during the last 30 years. Various technologies of parallel computing have been implemented. All technologies are different and require relevant knowledge from modern scientists. Scientists can use different technologies: supercomputers, cluster systems, grid’s, cloud computers and other computing resources. Depending on the technical implementation of computing resources, they may be provided with different software. New libraries are created as well as new programming standards for parallel computing. For example, MPI, OpenMP, gLite, XMPP-MPI, CUDA, OpenCL and other parallel programming standards have been implemented.

When talking about parallel computers, most often we have in mind several personal computers connected by a network into a single computing resource. If the resource is small, then its computational power is limited. If the computational power is growing up, so does the number of computers of a computing resource. This has influence also on electric power consumption. At present, it is a big problem that requires attention from scientists and manufacturers of computer equipment. In 2005 this problem was named - Green Computing. San Murugesan was the first to define the term of green computing as “the study and practice of designing, manufacturing, using, and disposing of computers, servers, and associated subsystems – such as monitors, printers, storage devices, and networking and communications systems – efficiently and effectively with minimal or no impact on the environment.” At the present time, we can also find other names of this problem: green information technologies (green IT), environment-friendly IT communications and computing. The topicality of this problem is stressed by the list of most efficient energy aware supercomputers named - Green500 (existing since 2005). Optimization methods can help to solve the electric energy consumption problems. In addition, solution of this problem can minimize the use of available computing resources to reduce their electricity consumption. This fact is emphasized by the green computing expression – “Higher computing power with the smaller influence on the environment.”

The Aim and Tasks

The key aim of the dissertation is to propose an environment-friendly method, in order to minimize electric energy consumption in parallel and distributed computing.

To achieve the aim, it was necessary to solve the following tasks:

- to analyze the green computing technologies and electric energy consumption problem;
- to investigate the available parallel and distributed computing technologies;
- to propose serial, parallel and distributed algorithms for topology optimisation of a systems of trusses;
- to investigate electric energy consumption of parallel and distributed systems while solving optimization and other problems;
- to develop a software for minimizing electric energy consumption of parallel and distributed systems at work.

The Objects of Research

The object of research of the dissertation is environment-friendly technologies that can minimize electric energy consumption of parallel and distributed computers, when solving optimization problems.

Scientific Novelty

1. The branch and bound algorithm of combinatorial optimization to find the best solution to a system of trusses has been proposed and investigated.
2. Consumption of electrical power of parallel and distributed systems when solving optimization problems has been experimentally investigated.
3. The strategy for saving electrical energy consumption by parallel and distributed systems has been proposed and tested experimentally.

Practical Significance of the results

A method for saving electrical energy of parallel and distributed computers has been developed. This method is installed in the real cluster – hpc.mii.vu.lt. The method can also be used in different parallel computing systems such as supercomputers, computers with GPU and others.

The research was partly supported by:

- The programme "Parallel and distributed computing and e-infrastructure network (LitGrid)" supported by Lithuanian Ministry of Education and Science, 2007-2009;
- The project B-03/2007 of Lithuanian State Science and Studies Foundation through the Programme for Higher Technologies: “Global optimization of complex systems using high performance computing and grid technologies”, 2007–2009;
- Research supported by Research Council of Lithuania: "Global Optimization with Simplicial Partitions", 2010–2011;
- COST action IC0805 “Open European Network for High Performance Computing on Complex Environments”, from 2008 until now.
- The proposed electric energy saving method is installed in hpc.mii.vu.lt cluster.

- In collaboration with "*Poznan Supercomputing and Networking Center* (PSNC)" scientist's new parallel and distributed calculations standard – XMPP-MPI has been tested and improved.

The Defended Statements

1. The method proposed for saving electrical energy consumption of parallel and distributed systems can reduce energy consumption up to 90 %.
2. The proposed parallel Branch-and-Bound combinatorial algorithm for optimizing truss structures allows obtaining the optimum design of trusses up to 9 nodes.

Approbation and Publications of the Research

The main results of the dissertation were published in 5 scientific papers: 4 articles in the periodical scientific publications; 1 article in the proceedings of scientific conference. The main results of the work have been presented and discussed at 9 national and international conferences.

The Scope of the Scientific Work

The dissertation is written in Lithuanian. It consists of 5 chapters and the list of references. There are 122 pages of the text, 30 figures, 10 tables, and 125 bibliographical sources.

2. Electrical Energy Consumption and the Parallel and Distributed Computing

In this chapter, the classification of parallel and distributed computing technologies as well as the main methods of green computing are presented and described.

Electrical Energy Consumption

Recently, optimization of electrical energy consumption has become a very important task. Scientific studies use different computer resources: supercomputers, clusters, grids and cloud computing. These resources consume much electrical energy, so a very important task now is to reduce electrical energy costs without compromising computing power.

Recently the term "green computing" has emerged. San Murugesan defines the field of green computing as "the study and practice of designing, manufacturing, using, and disposing of computers, servers and associated subsystems – such as monitors, printers, storage devices and networking and communications systems – efficiently and effectively with minimal or no impact on the environment" (Murugesan 2008). At the present time, we can also find other titles of this problem: green information technologies (green IT) and environment-friendly IT communications and computing.

Green500 (www.green500.org) is the analogue of the Top500 list (the list of most powerful supercomputers in the world), that classify clusters and supercomputers according to electrical energy power. The list appeared in 2005 after Dr. Wu-chun Feng conference presentation and the article "IEEE IPDPS Workshop on High-Performance, Power-Aware Computing". Following this report the scientific community began to pay more attention to this problem.

Currently there are no available public solutions to optimize electrical energy consumption of cluster systems. Mostly commercial tools can be used for cluster monitoring and administration. In Lithuania scientists pay too little attention to this problem, despite that the number of clusters and supercomputers is increasing every year. Since most of today parallel computing clusters is based on the personal computers, we can considered the problem of electrical energy consumption reduction of clusters as a single computer energy consumption reduction problem. However the decisions should be applied globally to all the computers that compose a cluster. There are several methods how to reduce electrical energy consumption for a single personal computer:

1. Turn off devices when these are not in use.
2. The use of tools to minimize computer electrical energy consumption:
   - 2.1. To use screensavers.
   - 2.2. To use the monitor hibernation.
   - 2.3. To use the sleep mode of hard disk.
   - 2.4. To use the standby mode of the computer (computer is turned off but the memory is powered).
   - 2.5. To use hibernation of the computer (the content of memory is saved in hard disk drive and the computer is turned off).
3. Eliminate Phantom load (power consumption when a device is turned off).
4. Upgrade a computer with the new energy-efficient components (change TFT LCD to LED LCD monitors).
5. Upgrade a computer with the new components to extend its lifetime (memory upgrade, hard disks upgrade, or replace hard drive disks by solid-state disks).
6. Buy efficient and electrical energy-aware equipment when buying new.
7. It is not only useful to buy the existing efficient equipment, but also to develop new electric power-aware technologies.

All these recommendations well reflect the concept of green computing. These solutions should be applied not to one computer but to all computer resources, that make up a cluster or supercomputer. And it is not so simple to do that.

Hardware for Parallel and Distributed Computing

There are several parallel and distributed computing execution systems: supercomputers, cluster systems, grids, graphic processing units, and other computers.

Supercomputers

The era of supercomputers began in 1976. That year the first vector system supercomputer Cray-1 was developed. Lithuania also has several supercomputers. One of them was presented in Vilnius University on March 13 2012, the Vilnius University supercomputer.

Top500 (www.top500.org) is an official list of the most powerful supercomputers in the world. The list was published in 1993 and updated twice a year in June and November. The actual performance of a computer is measured by the Linpack test.

However supercomputers have a huge issue, high electrical energy consumption. Same applies to the most powerful computer in the world, “K computer”. In the Green500 list it took only the 32nd place, according to the MFlop/s per W ratio, and the total electrical power, when it is operating at full load, is 12 659.89 kW. Furthermore, such computers are needed to be cooled with powerful cooling systems. Many researchers and hardware developers try to solve this problem. The researchers are looking for the ways to run tasks so that those systems would use less cooling system. Others suggest abandoning the use of expensive supercomputers and making calculations with a small, but powerful cluster. In addition, some scientists formulate new guidelines for creating new green supercomputers (use the quick and fast solid-state drives, using supercomputers with graphics processing unit technology, use more virtualization technology in computing, and other recommendations).

Computing Clusters

Cluster is a group of computers connected by a local network into one virtual computer which can work as a single computing resource. It is assumed that clusters are more efficient than personal computers and cheaper than other high-performance computing systems. Currently, clusters are very widespread between supercomputers. In the list Top500, of November 2011, cluster systems represent a large proportion of supercomputers 410 clusters of 500 computers. In Lithuania, there are several clusters as well, most of which are of the major universities: the clusters of Vilnius University CLUSTER.MII.VU.LT and HPC.MII.VU.LT, the cluster "Vilkas" of Vilnius Gediminas Technical University, cluster at Kaunas University of Technology and other clusters.

Clusters have drawbacks just like supercomputers. Electricity consumption grows with the size of cluster. As a result, most of these cluster administrators also face the electrical energy consumption problem. Often, the methods for supercomputers can be used, but there are differences. Some researchers propose to use the clusters computer-workers of which have no hard-disks; others suggest using of virtualization technologies. In addition general advice has been formulated how and from which components new or improved old clusters should be assembled.

Grid Computing

Distributed computing is a computational method to solve a large number of tasks. A “virtual supercomputer” is used, connected into one computing resource by the network, out of computers working in parallel at the same time. These technologies are used to solve scientific, mathematically complex calculations that require large computing resources. In addition, grid computing is used in commercial organizations to solve such difficult tasks as: economic forecasting, seismic analysis, drug development and testing, and others. A grid consists of many different computers, using different processor architectures with a wide range of software and connected by computer networks. The grid may be connected from very distant computers. In 2005 Lithuania, Latvia, Estonia, Poland, Sweden, Switzerland and Belarus created a combination of clusters (Baltic Grid) for general usage of distributed computing resources. Each of these countries has their own Grid: Estonian Grid, Latvian Grid, LitGrid and others.

Since most of the grids are made up from simple clusters, all grids have the same electrical energy consumption problems as the clusters, but there are some differences. For example, it is advisable to use a multicore processor; avoiding the use of air conditioners, to use self-conditioning server rooms; to more effectively implement programs, so that they less heat processors of computers.

Computers with Graphic Processing Units

Over the past few years, graphics processing units have greatly improved and became powerful computing devices. Today, there are graphics processors with several cores and very high memory bandwidth. They have incredible power in graphics and non-graphics computing. Currently, they are widely used in experimental studies and to process the research data. In Top500, the list of supercomputers, the number of clusters with graphics processing units (GPU) is increasing every year.

However, clusters and supercomputers, consisting of the components with GPU, have significant disadvantages compared with the classical systems. Their power consumption is extremely high. Since GPU is usually very hot at work it must be cooled. When running a single computer with a GPU it is slightly noticeable, but when we have a lot of them it is necessary to use the powerful air conditioning systems. Electrical energy consumption is also significant for these air conditioning systems. So far there are no easy solutions to this problem, but most authors suggest the ways associated with the efficient allocation of tasks among computers with GPU.

Parallel Programming Technologies

All this equipment needs different programming standards or libraries to make calculations.

OpenMP, Open MultiProcessing

OpenMP (Open MultiProcessing) implements the parallel computing using the threads (parts of the program running in parallel) and shared memory technology. The main thread during execution creates a few extra threads, among which the tasks are allocated. Threads are run in the computer with the shared memory. Currently, the latest version is the OpenMP 3.1 (OpenMP 2011). OpenMP is implemented in different compilers such as GCC, Sun Studio compilers, Visual C++ (since 2005 version), Intel C++ Compiler, IBM XL compiler, PGI and others.

MPI, Message Passing Interface

The development of the MPI (Message Passing Interface) standard involved not only scientists, but also manufacturers of supercomputers, whose main goal was to make an efficient use of the existing and future computer resources. Using MPI functions, each program can find out how many processes are running, what the process number is, as well as exchange data with other processes. MPI is the standard for data transmission implemented by an explicit model (data are transferred from one procesor to another processor’s address area in the network by cooperating functions). With regard to the MPI and C++ compatibility, the MPI 1.1 standard was defined only for C and Fortran languages, but soon C++ interface (for example, OOMPI) has appeared. The MPI 2.0 standard has been defined with the C++ interface.

All these technologies and implementations have been useful to run parallel and distributed tasks. But now parallel and distributed computing has issues. Scientists and IT manufacturers try to solve such problems. In this dissertation several of them are presented.

3. Ways of Reducing Electrical Energy Consumption in the Parallel Computing Clusters, Parallel Branch and Bound Algorithms

In the world, computer electrical energy consumption problem has been particular importance since 2005. This year the concept of green computing has been formulated. In Lithuania insufficient attention was paid to this problem. Various authors offer different electrical energy consumption reduction methods, but not always they are available publicly. Usually computing clusters can have electrical energy saving tools installed only in the host computer, or not installed at all. In this chapter, a way to investigate electrical energy consumption is proposed. Also the use of electrical energy-saving techniques for parallel and distributed computing is proposed.

Electrical Energy Consumption Investigation

At presents, optimization of electrical energy consumption is a very important task. Scientists in their studies used different computer resources: supercomputers, clusters, grid and cloud computing. All the resources above use a lot of energy and very important task is to reduce electrical energy consumption without reducing the capacity of computations.

Parallel tasks were implemented, electric power and total electrical energy consumption of the computations were recorded, in addition, the overall load of cluster has been captured by the cluster load monitoring system. Electric power shows how much electricity is required by a cluster at the time of measurement. The total electrical energy consumption shows how much electricity is consumed by a cluster over time according to energy meter indications.

To realize those objectives we need: the cluster environment to run the task, the power meter “UPM Energy Meter MP300”, a remote monitoring system consisting of camera “ACME PC Camera CA09, 1.3 Mpixels” (Figure 1 shows the camera monitoring “Energy Meter”) and a cluster load monitoring system “GANGLIA” (Figure 2). The scripts to create and playback online video broadcast translation have been created.

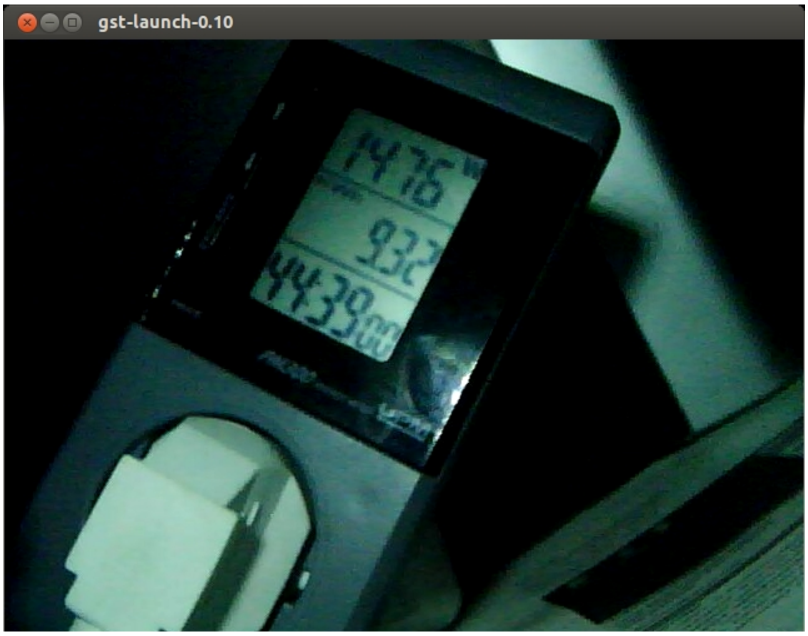


Figure 1. *GSTlaunch-0.10 - monitoring system of readings of UPM Energy Meter MP300*

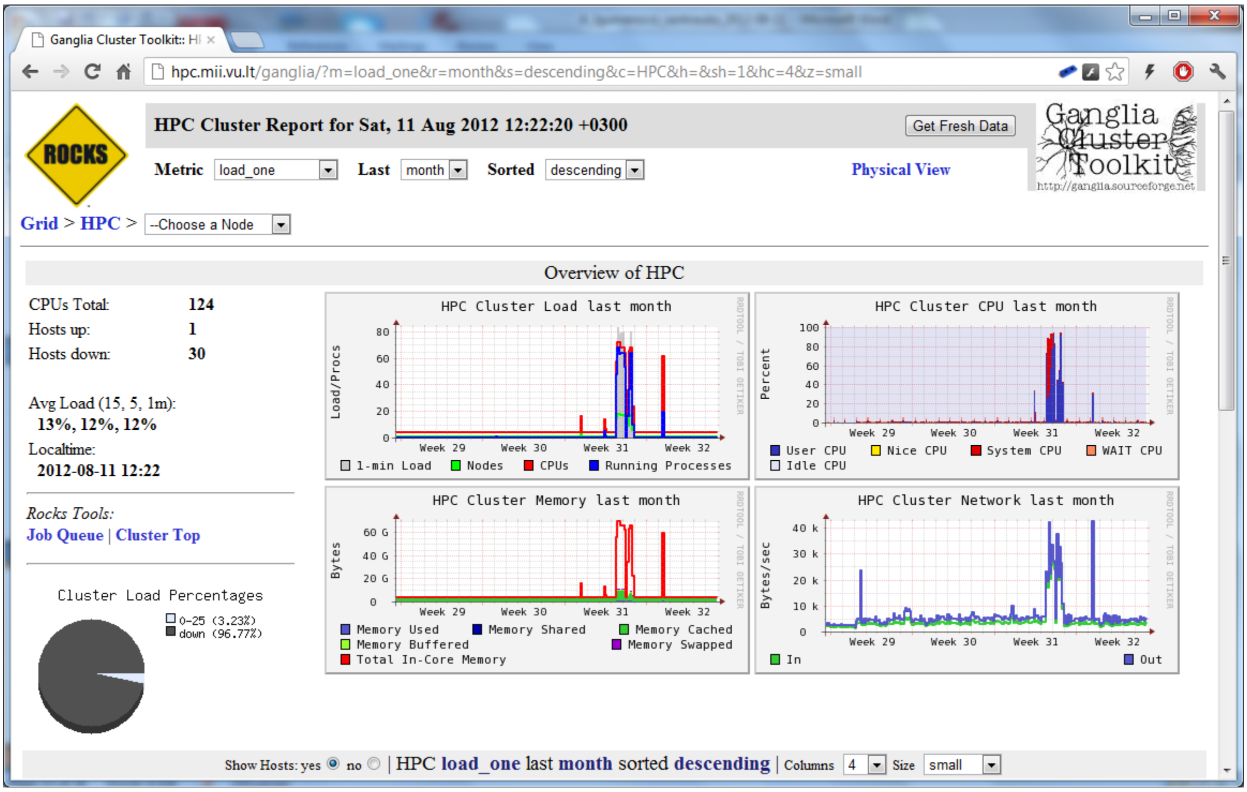


Figure 2. *The cluster load monitoring system "GANGLIA"*

Reduction of Electrical Energy Consumption by Clusters

Whole carrying out the experiments in grid it has been noticed that grid clusters and their computers are not always loaded in full. The tasks are not optimally distributed by grid infrastructure among the computers, so computers are not fully loaded. For example tasks are performed not by one multicore computer, but with a few computers (though one is enough). Other scientists' researches have proved that the tasks are ineffectively distributed in a grid.

Systems of clusters are efficient, because they do not use monitors. Only the Internet connection is needed for cluster administration and maintenance, a monitor is required for installation or setup, and this happens very rarely. Most users connect to the host computer of the cluster by the Internet and submit their tasks. It has been noticed that the cluster works with a low load of tasks, while compute nodes of the cluster are loaded only by system processes for a long time (Idle State), which means without any meaningful computations. Thus, the compute nodes of the cluster consume electrical energy in vain.

After observing the operation of clusters we propose a new working way for a cluster system. It is proposed to automatically turn off compute nodes when they are not needed for computations. In this way you can reduce consumption of electrical energy of cluster systems during the cluster downtime (when compute nodes are waiting for the tasks from the network). The only requirement for cluster computers is that BIOS (Basic Input-Output System) must be set-up so that the network card must be active all the time. That should be done because the network card has to wait for commands to wake up from the host computer.

To achieve these objectives, scripts 1 and 2 have been developed and installed in computers. Files are created using the Linux Bash commands. As a result, they are relatively universal and can be used in all the clusters with Linux operating system (OS). In order that the script worked in the host computer an additional package WAKEONLINE must be installed. The package WAKEONLINE allows us to wake any network computer knowing only the physical address of the network card (MAC address). In addition, to launch the scripts it had be enabled in the compute nodes and host computer the CRON service (scheduled work starting service). Also, as mentioned above, BIOS of compute nodes must be set-up, so that the network cards were enabled all the time. This is a necessary condition for script 1 operation.

Script 1. C*ompute nodes waking script*

```
#!/bin/bash        // work environment
#export PATH=/bin:/bin:/opt/gridengine/bin/lx26-x86     //path to additional environment files;
. /etc/profile.d/sge-binaries.sh                         //path to the other additional environment files;
if [ `qstat -u '*' | grep -c 'queue of waiting jobs'` != '0' ];   //bash if find how many tasks are in a queue;
then
        echo "Run the PC's" >> /tmp/log_pal.log;         //output in the log file;
        wakeonlan XX:XX:XX:XX:XX:XX;     //Wakeonline command with physical addresses of computers;
else
        echo "Addition PC's not needed" >> /tmp/log_pal.log;    //output in the log file;
fi
```

Script 2. *Compute nodes turning off script*

```
#!/bin/sh
if [ -z "$(ps ax | grep -v grep | grep 'program'" ];            // function if has to check if there exists or not task "program", if not, the computer is turned off; parameter -v grep | grep 'program' can reject a line with command grep from the results;
then
         shutdown -h now ;                 // command to turn off the computer;
fi
```

The proposed working way of cluster is useful only when the clusters are low-loaded which means they have a few users. Highly loaded clusters, applying such a strategy, may not always help reduce electrical energy consumption.

Topology Optimization of Truss Systems

Structures of trusses are the objects that consist of nodes interconnected by various parameter beams. Construction may be varied, with movable and with fixed supports, and may have one or more points of force. As examples of these structures are bridges, cranes, roof overlay and support systems, billboards with a side attachment and others.

To optimize such systems, different algorithms can be used: genetic, simulated annealing, and branch-and-bound algorithms.

Mathematically the optimization of the truss structures problem is formulated as follows:

$$\min\ f(X) = \sum_{i=1}^{n} x_i L_i \rho_i A_i\,, \qquad (1)$$

$$g(X) = 0,$$

$$h(X) \leq 0,$$

$$x_i \in \{0,1\},$$

where the objective function $f(X)$ corresponds to the total mass of the structure; binary variables x_i code the presence of the i-th beam in the structure; L_i is the length of the i-th beam; ρ_i is density of beam material; A_i is a cross-sectional area of the same i-th beam; n is the number of possible beams, $n = m(m - l)/2$, where m is the number of nodes. Equality constraints defined by $g(X)$ ensure that the total sum of forces at each node would be zero, and then the system would be in the static equilibrium state; and the inequality constraints defined by $h(X)$, ensure that the stresses in beams would not exceed the critical values and the system be locally stable. Constraint functions $g(X)$ and $h(X)$ are modelled by a finite element method.

Checking each structure, we face a system of linear equations and perform matrix multiplication. Systems of linear equations and matrix multiplication can also be implemented by parallel technologies.

Finding the optimal structure by a complete enumeration method, you must check all possible structures. The number of possible structures (N) grows up exponentially with the number of the nodes (m) and potential bars (n) $N = 2^{\mathrm{n}} = 2^{\mathrm{m}\times(\mathrm{m}-1)/2}$. Parallel computing can be used to speed up computations.

Examples of truss structures are presented in Figure 3.

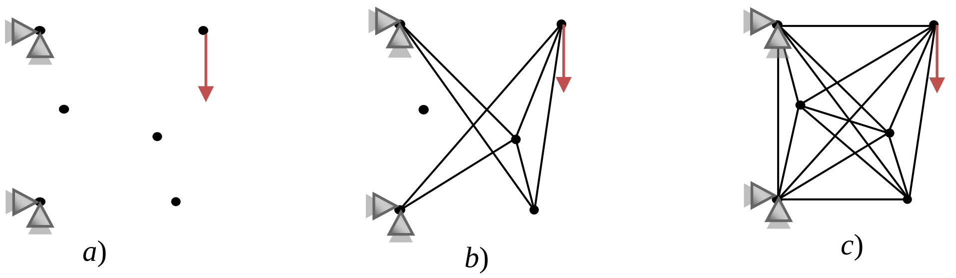


Figure 3. *Examples of truss structures a) the structure without beams, only 6 nodes, b) the optimal design with 7 beams and c) the structure with all available beams*

Complete Enumeration Algorithm for the Optimization of Truss Structures

The simplest algorithm, which allows us to solve combinatorial optimization tasks, is an enumeration of all possible solutions. For the truss structures, the structure is coded in a binary form and combinatorial algorithms are very useful for optimization of such structures. The lightest structure satisfying the constraints is optimal.

Algorithm 1 is implemented by a complete enumeration method for optimizing truss structures. During the computation constraints are checked. If the constraints are satisfied and the objective function value is lower than that previously known, then this value and the binary vector X are saved. At the end of the algorithm, the minimum of the objective function value is f^*, corresponding to the structure coded by a vector X^*.

One way to speed up solution of the algorithm is its parallelization. The loop “for” can be parallelized and the various constructions of trusses can be tested by different independent processors (or a multicore processor) at the same time. Frequently complete enumeration algorithms are well parallelizable.

Algorithm 1. *Sequential algorithm of complete enumeration for optimizing the truss system*

$f^* = \infty$
for $\forall X \in \{(0, \dots ,0,0), (0, \dots ,0,1), \dots , (1, \dots ,1,1)\}$ **do**
  if $f(X) < f^*$ **and** $g(X) = 0$ **and** $h(X) \leq 0$ **then**
    $f^* = f(X)$
    $X^* = X$
  end if
end for

Algorithm 2. *Parallel OpenMP algorithm of complete enumeration for optimizing the truss system*

$f^* = \infty$
Include: *omp.h*
#pragma omp parallel for
for $\forall X \in \{(0, \dots ,0,0), (0, \dots ,0,1), \dots , (1, \dots ,1,1)\}$ **do**
  if $g(X) = 0$ **and** $h(X) \leq 0$ **then**
    #pragma omp critical
    if $f(X) < f^*$ **then**
      $f^* = f(X)$
      $X^* = X$
    end if
  end if
end for

Algorithm 3. *Parallel MPI algorithm of complete enumeration for optimizing the truss system*

$f^* = \infty$
Input: *rank*, *size*
Include: *mpi.h*
  for $\forall X \in \{(0, \dots ,0,0), (0, \dots ,0,1), \dots , (1, \dots ,1,1)\}$ **do**
    if ((*int*) *X* % *size* = *rank*) **and** $f(X) < f^*$ **and** $g(X) = 0$ **and** $h(X) \leq 0$ **then**
      $f^* = f(X)$
      $X^* = X$
    end if
  end for
  if ($rank <> 0$) **then MPI_SEND** (f^*, X^* **to** 0)
  else
    for 1 **to** *size*-1 **do MPI_RECV** ($f^{*\prime}$, X' **from** ANY)
      if $f^{*\prime} < f^*$ **then**
        $f^* = f^{*\prime}$
        $X^* = X'$
      end if
    end for
  end if

Algorithm 2 implements a parallel version of the complete enumeration algorithm. The OpenMP pragma “# pragma omp parallel for” is used to identify a parallel loop, “# pragma omp critical” is used to protect the variable f^* during the check and change in the run of the algorithm. More efficient versions of the algorithm use local variables f^* and X^* to avoid a possible conflict when data saving. In this case, the results are collected only at the end of the computation. For example, in the MPI version of complete enumeration algorithm (algorithm 3), all variables are local, but at the end the results are collected and summarized by the processor 0.

Branch-and-bound algorithm for the optimization of truss structures

One of the first branch-and-bound algorithms has been proposed by Horst and Tuy in their works. Their branch-and-bound algorithm is used to realize the covering global optimization and combinatorial optimization methods. Branch-and-bound algorithms can be used for selecting candidates and managing their disposal. A branch-and-bound algorithm consists of initialization, selecting, subdivision, and bound computation rules.

A branch-and-bound algorithm consists of:

1. The initialization phase.

   A feasible set of solutions is divided into a finite number of subsets:

   $$D = \bigcup_{i=1}^{m} D_i. \qquad (2)$$

   The minimum search algorithm of the function $f(X)$ is convenient to represent by a search tree. Tree branches define the feasible solution set D. Branch-children define subsets D_i.

2. From a set of candidates a subset is selected and divided; in the newly obtained subsets, the lower (LB) and upper (UB) bounds of the minimum of the function are computed. The algorithm defines how to efficiently compute the lower (LB) and upper (UB) bounds of the minimum of the function $f(x)$ in the set D_i:

   $$LB(D_i) \leq \min_{X \in D_i} f(X) \leq UB(D_i). \qquad (3)$$

   Then, the $UB(D_i)$ are known, we can calculate the upper bound $UB(D)$ of the minimum value of the $f(x)$ over the set D:

   $$UB(D) = \min_{1 \leq i \leq m} UB(D_i). \qquad (4)$$

3. If we know these bounds we can often greatly reduce the amount of computations. If, in the set D_i, the lower bound $LB(D_i)$ is higher than $UB(D)$:

   $$LB(D_i) > UB(D), \qquad (5)$$

   then we can diminish such set because it does not contain the global minimum.

The algorithm is aimed at a fast reduction of a set of candidates and at the fast convergence to the solution.

Algorithm 4 presented is the branch-and-bound algorithm for optimizing truss structure. Its construction is coded as a vector X, which depends on the variable p. The algorithm can be an isolated branch of solutions, depending on the objective function, after which the branches are not better than the already known optimum. If the solution is greater than the known f^*, then the branch of further solutions should not be considered. Because each additional truss in the structure can only increase the mass of the structure.

Algorithm 4. *Sequential branch-and-bound algorithm for optimizing the truss system*

```
f* = ∞; x_i = 0, i = 1, 2, ..., n; p = 1;
while p > 0 do
        if f(X) < f* and g(X) = 0 and h(X) ≤ 0 then
                f* = f(X)
                X* = X
        end if
        if f(X) < f* then p = n
        while X_p = 1 and p ≥ 1 do
                x_p = 0
                p = p − 1
        end while
        if p > 0 then x_p = 1
end while
```

Algorithm 5. *Distributed branch-and-bound algorithm for optimizing the truss system*

```
Input: rank, size
k = log(size) / log (2)
(x_1, x_2, ..., x_k) = binary_code(rank)
x_i = 0, i = k + 1, ..., n
f* = ∞; p = k + 1;
while p > k do
        if f(X) < f* and g(X) = 0 and h(X) ≤ 0 then
                f* = f(X)
                X* = X
        end if
        if f(X) < f_min then p = n
        while X_p = 1 and p ≥ k + 1 do
                x_p = 0
                p = p − 1
        end while
        if p > k then x_p = 1
end while
```

The branch-and-bound algorithm for truss structure optimisation using distributed computing is presented as algorithm 5. During the run of the algorithm the computation are executed by several computers independently. Depending on the unique number of a process the vector X is generated. The first positions of x_i, $i = 1, \dots, k$, the number of process (rank) are coded, the number of such positions is equal to binary expression of the length of the total number of processes ($k = \log(size)/\log(2)$). After this x_i, $i = k + 1, \dots, n$, components of vector X can be freely changed by algorithm.

Parts of the vector for distributed computations are shown in Figure 4.

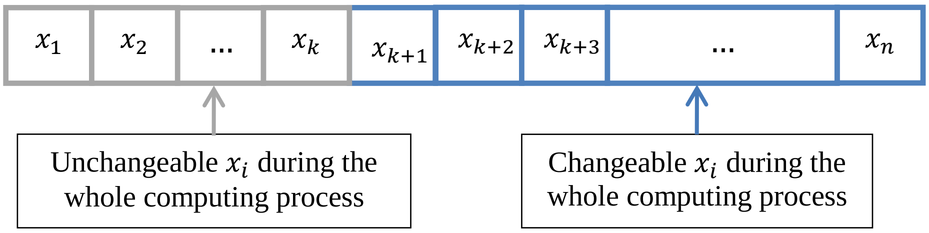


Figure 4. *The structure of vector X for distributed computing*

4. Experimental Investigations

In this chapter, the results of experiments, intended to demonstrate the advantage of the proposed method for electrical energy saving, are presented. Experimental tests were performed using grid and cluster computing. To investigate electrical energy consumption the research methodology is used, described in Section 3. Also in this section the experiments and other results are described: computer network bandwidth impact on parallel computing performance, optimization of truss structures investigated in grid and cluster computers.

Computer Network Influence on the Performance of Parallel Computing

When reviewing parallel computing, we decided to analyze the performance of existing computer networks. The typical computing algorithms were used in experiments (matrix multiplication, solving of a system of linear equations (SLE) by the Gaussian method). The experiments were done with 100 Mbps and 1 Gbps computer networks. During them one or two processes were used in computers. Experiments were repeated 100 times and the results averaged. Speedup of the algorithm is shown in Figure 5.

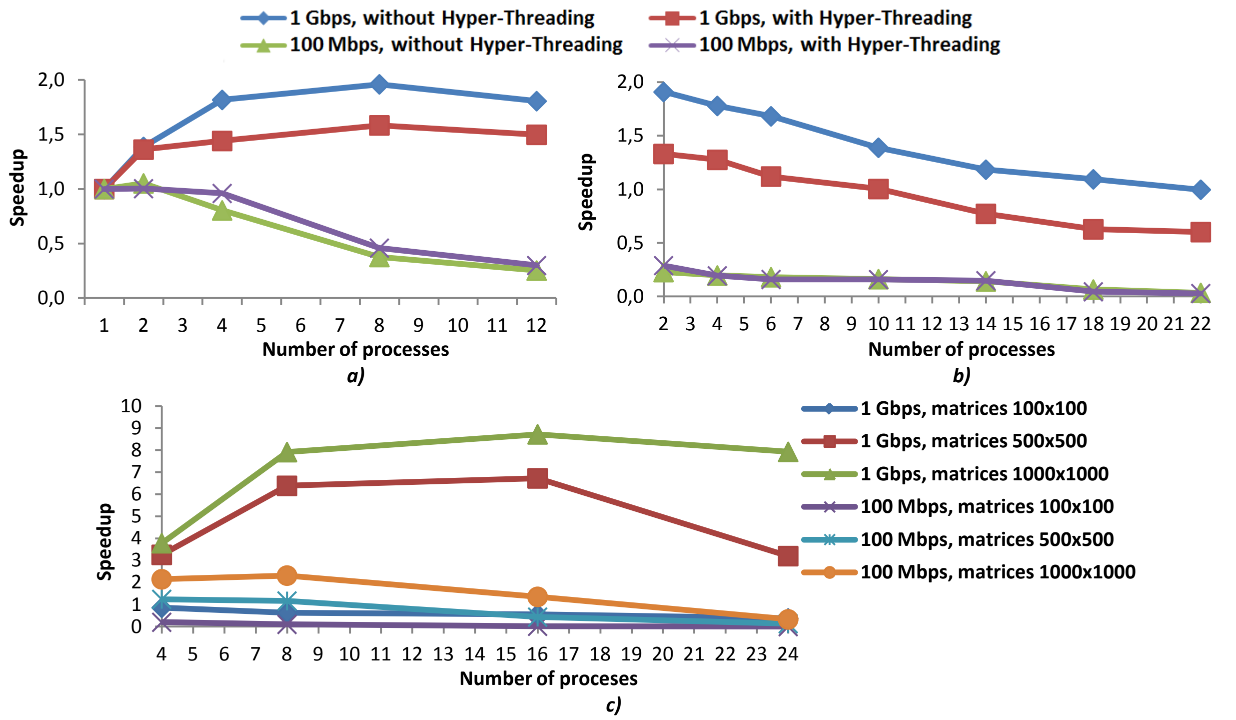


Figure 5. *Speedup of solutions depending on network bandwidth and usage of Hyper-Threading technology for the running tasks: a) solving a system of linear equations; b) matrix and vector production; c) multiplication of matrices*

The results obtained with 1 Gbps network are much better than that with 100 Mbps network. This is due to a highly intensive exchange of data among processes. Uselessness of the Hyper-Threading technology with the 1Gbps network is also seen for these algorithms. But sometimes the use of Hyper-Threading technology is helpful for a 100 Mbps network (Figure 5a, b). In Figure 5b, it is easy to notice that the algorithm performance is slowing down, because of excessive exchange of messages among compute nodes. Matrix multiplication (Figure 5c) results of the size 100 x 100 in the 1 Gbps network are worse due to a small size of the task. With larger matrices the result is much better, speedup increase up to 8.5 times.

Investigation of Electrical Energy Consumption

The dissertation investigated the cluster energy consumption of various parallel computations. One of the algorithms is a parallel branch-and-bound algorithm for multidimensional scaling with the city-block distances; the other is a parallel branch-and-bound algorithm for optimization of truss structures. Both the optimization algorithms were used to investigate electrical energy consumption of clusters. Electrical energy power of one computer was also investigated, depending on its load (table 1).

The measurements were made by using “UPM Energy Meter MP300” (Figure 6). This meter has several measurement modes. Only electrical power (W) and consumption of electrical energy (kWh) were used for the study. In addition, package “gst-launch-0.10” was installed into the host computer for interactive monitoring of the meter readings over the Internet. Two bash scripts 3 and 4 were created for meter monitoring. Script 3 is installed in the Cron daemon and regularly ran in host computer. Script 4 can be run on any computer with Linux OS with installed module “gst-launch-0.10”.

Figure 6. *UPM Energy Meter MP300*

Table 1. *Multicore computer electrical power dependence on its load*

Load	Electrical power, W			
	For computer	Without Idle	For one core	To additional core
grid.mii.lt	Intel Quad Core Q9400, Scientific Linux 4.7 OS			
Idle	50 ± 5			
25 %	80 ± 5	30	30	30
50 %	93 ± 5	43	22	13
75 %	110 ± 5	60	20	17
100 %	120 ± 5	70	18	10
cluster.mii.lt	AMD Athlon 64 x2 Dual Core 5000+, Rocks Cluster 5.1 OS			
Idle	59 ± 5			
50 %	91 ± 5	32	32	32
100 %	115 ± 5	56	28	24
hpc.mii.vu.lt	Intel Quad Core Q9400, Rocks Cluster 5.4 OS			
Idle	46 ± 5			
25 %	67 ± 5	21	21	21
50 %	81 ± 5	34	17	13
75 %	89 ± 5	43	14	9
100 %	97 ± 5	51	13	7
hpc.mii.vu.lt	Intel Core I5-760, Rocks Cluster 5.4 OS			
Idle	50 ± 5			
25 %	78 ± 5	28	28	28
50 %	90 ± 5	40	20	12
75 %	105 ± 5	55	18	15
100 %	122 ± 5	72	18	7

The study revealed that even and without computations (idle state, when the computer waited for the tasks), the electrical power for one computer is not small (average 50 watts, Table 1). When the tasks are assigned to a computers processor, electric power increases gradually, depending on processor's load, but slightly compared to idle. In addition, the operating system influences on electric power consumption of compute node. Grid.mii.lt and hpc.mii.vu.lt use the same processors, Intel Quad Core Q9400, but their power consumption is different. The grid.mii.vu.lt cluster has been installed with Scientific Linux 4.7 OS, which is installed with a graphic environment. Hpc.mii.vu.lt cluster has installed with Rocks cluster 5.4 OS and Rocks installs the compute nodes without a graphic environment.

Table 2. *Electrical energy consumption of grid computers depending of the task allocation, with idle*

Number of process-workers	Electrical energy consumption, kWh					
	8 nodes task			9 nodes task		
	Manual distribution of tasks	gLite distribution of tasks	Consumption increase, times	Manual distribution of tasks	gLite distribution of tasks	Consumption increase, times
1	0.0141	0.0141	1.00	3.157	3.157	1.00
2	0.0090	0.0090-0.0145	1.00-1.61	2.499	2.499-3.689	1.00-1.48
4	0.0088	0.0088-0.0173	1.00-1.97	1.797	1.798-4.053	1.00-2.26
8	0.0080	0.0100-0.0120	1.25-1.50	2.580	2.902-4.195	1.12-1.62
16	0.0106	0.0120-0.0160	1.13-1.51	3.037	3.147-3.405	1.04-1.12

Depending on these results (Table 1) it was concluded, that for efficient use of electricity (to minimize power consumption) computer's processor must be loaded as much as possible, rather than tasks divided among several computers (without loading them completely). As shown in Table 2, the grid infrastructure does not make this during the computations, task load cluster's computers partly; due to this the cluster consumes more power. Rocks Cluster 5.4 operating system distributes tasks with maximum load on computer's processors.

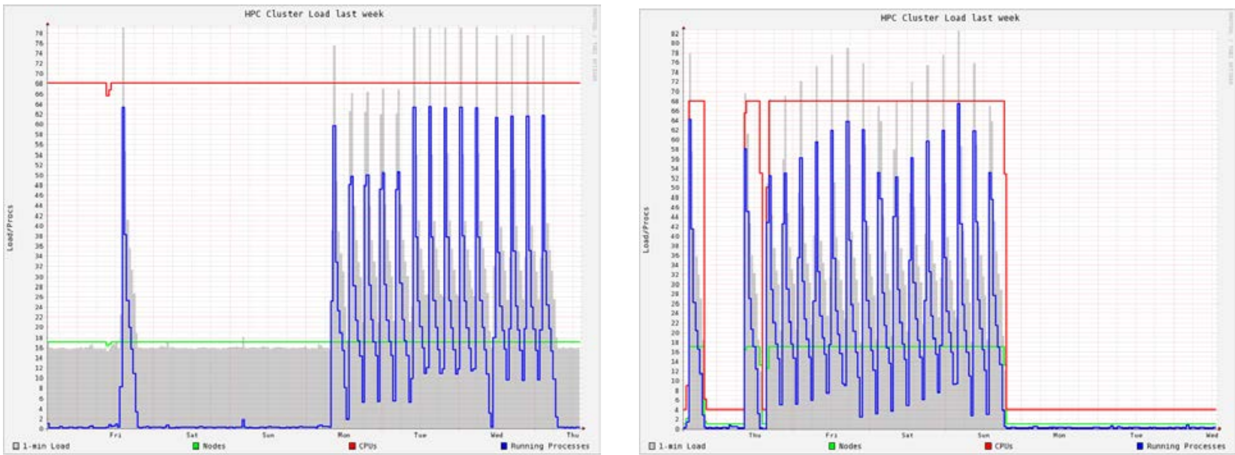
Investigation of the Electrical Energy-Saving Strategy

The dissertation proposes a new clusters working strategy. It has the advantage, because computers are switched off when they not in use. The measurements showed that the proposed electrical energy-saving way is useful in use with clusters and grids. Table 3 below shows electrical energy consumption of a cluster of 16 computers with Intel i5-760 processor depending on the use of new strategy.

Table 3. *The electrical energy-saving mode in comparison to with the normal work of the cluster*

	Time, hours	The total electrical energy consumption, kWh	Electrical energy consumption per hour, Wh	Saved, %
The normal work of the cluster	171	189.07	1 106	–
The work of the cluster with new strategy	171	132.12	777	30

This table shows that during the same time the cluster, which uses the proposed method of work, consumes 30% less energy than the cluster working as usual. Saved electrical energy per year is (1 106 – 777) * 24 * 365 = 2 882 kWh. Figure 7 shows the illustrations taken from the GANGLIA cluster monitoring system.



a) *b)*

Figure 7. *The cluster's work: a) normal working, b) working with new strategy*

You can see in Figure 7 that the cluster's computers have been working in proposed manner it is switched on and off according to need. Green line shows how many computers are turned on, red how many core are available, blue how many processes (tasks) are currently running. The system always shows one more computer than the number of compute nodes, it is the host computer.

General Conclusions

The dissertation presents the area of green computing. In the world it is a very important problem, but in Lithuania too little attention is paid to it. Power demand grows up every year, because new technologies are developed, the number of clusters and supercomputers is growing. The dissertation also deals with parallel computers and shows their respect to green computing: distribution of tasks among the computing nodes, excessive electrical energy consumption. The impact of parallel computing on the environment should be reduced and other problems of grid and cloud computing, supercomputers and clusters should be solved.

The electrical energy consumption reduction method proposed for parallel and distributed systems gives good results. The results of experiments allowed us to draw the following conclusions:

1. The proposed branch-and-bound algorithm for optimizing truss structures is faster than the complete enumeration algorithm (up to 2.4 times with 7 nodes and up to 3.6 times with 8 nodes tasks). The proposed algorithm, in a reasonable period of time, can solve optimization problems of truss structures with 8 nodes.
2. The proposed optimization of truss structures by the parallel branch-and-bound algorithm method in a grid enables distributed calculations without installed libraries of message passing, and it also not limited by only one cluster. A disadvantage of the proposed method is that the compute nodes cannot exchange information among them which can increase the amount of necessary calculations. The proposed distributed algorithm, in a reasonable period of time, can solve optimization problems of truss structures with 9 nodes.
3. The experiments have showed that the grid distributed computing systems have no optimal distribution of tasks; often clusters inefficiently allocate tasks among compute nodes. Therefore these computations consume much more electricity (up to 2,5 times), than when the allocation of tasks is optimized.
4. Clusters for parallel computing use electricity about 50 Wh per computer even in the idle state.
5. The dissertation has proposed a way for saving electrical energy. It is to automatically switch off the compute nodes while they are waiting for computing tasks. The proposed method allows the cluster to reduce electrical energy consumption up to 90 % when the computer is not needed. When a cluster of 16 computers with Intel Core i5 CPU 760@2.80GHz processors is loaded all time about 50%, it can save about 2 882 kWh of electricity per year with the new electrical energy-saving way.

List of the Author's Publications on the Subject of the Dissertation

Articles in the reviewed scientific periodical publications:

1. Igumenov A.; Petkus T. 2008. Analysis of Parallel Calculations in Computer Network. *Information Technology and Control*, ISSN 1392-124X, 37(1), 57-62. (Abstracted/Indexed in ISI Web of Science, Inspec).
2. Igumenov A.; Žilinskas J. 2009. Combinatorial Algorithms for Topology Optimization of Truss Structure. In: *Information technologies 2009 (IT 2009) 15th International Conference on Information and Software Technologies*, April 23-24 2009, Kaunas, Lithuania, ISSN 2029-0063, 2029-0055. pp. 229-234. [Abstracted/Indexed in ISI Web of Science (Conference Proceedings Citation Index)].
3. Igumenov A.; Žilinskas J. 2010. Combinatorial Topology Optimization of Truss Structures Using Distributed Grid Computing. *Jaunųjų mokslininkų darbai*, ISSN 1648-8776, 1(26); Supplement, 277-280.
4. Igumenov A.; Žilinskas J. 2011. Power Consumption Optimization with Parallel Computing. *Jaunųjų mokslininkų darbai*, ISSN 1648-8776, 4(33); 119-122.

Articles in other editions:

1. Igumenov A.; Žilinskas J.; Kurowski K.; Mackowiak M. 2010. Optimization of Topology of Truss Structures Using Grid Computing. In: *2010 IEEE International Conference on Cluster Computing Workshops and Posters (CLUSTER WORKSHOPS)*, Heraklion, Crete, Greece, September 20-24, 2010, ISBN: 978-1-4244-8395-2. doi:10.1109/CLUSTERWKSP.2010.5613101 (Abstracted/Indexed in IEEE Xplore).

Short description about the author

Aleksandr Igumenov received a Bachelor’s degree in mathematics from Vilnius Pedagogical University in 2005 and Master’s degree in informatics in 2007. 2007-2012 – PhD studies at Vilnius University, Institute of Mathematics and Informatics, Systems Analysis Department.

LYGIAGRETIEJI IR PASKIRSTYTIEJI ELEKTROS ENERGIJĄ TAUSOJANTYS SKAIČIAVIMAI

Tyrimų sritis ir problemos aktualumas

Elektros energijos suvartojimo didėjimas kelia problemas, susijusias su elektros energijos gamyba ir importu (kai neužtenka vidinių šalies resursų). Dažniausiai naujos technologijos reikalauja vis didesnių elektros energijos kiekių. Tai galima pastebėti visur, bet ypač ši problema išryškėja kalbant apie šiuolaikines informacines technologijas. Išleidžiami vis naujesni įrenginiai, kurie neimlūs elektros energijai, tokių įrenginių skaičius didėja, o tai didina elektros energijos suvartojimą. Suvartojimo didėjimas liečia ir lygiagrečiuosius bei paskirstytuosius skaičiavimus. Pasaulyje kiekvienais metais didėja superkompiuterių, klasterių, gridų, bei debesų kompiuterių skaičius. Visi šie resursai reikalauja daug elektros energijos. Pats galingiausiais pasaulyje kompiuteris, „K computer“, veikiantis pilnu pajėgumu, reikalauja 12 659,89 kW elektros galios. Tokios elektros galios užtektų išlaikyti vidutiniškai apie 60 000 įprastų namų ūkių.

Lygiagretieji skaičiavimai per pastaruosius 30 metų labai išsivystė. Atsirado įvairios technologijos įgyvendinančios lygiagrečiuosius skaičiavimus. Visos technologijos yra skirtingos ir reikalauja iš šiuolaikinio mokslininko atitinkamų žinių. Mokslininkai gali naudoti skirtingą techniką: superkompiuterius, klasterines sistemas, gridus (skaičiuojamuosius tinklus), debesų kompiuterius ir kitus resursus. Priklausomai nuo techninės skaičiavimo resursų realizacijos gali skirtis ir su jais teikiamos programinės technologijos. Taip atsiranda naujos bibliotekos skirtos atlikti lygiagrečiuosius skaičiavimus, taip pat nauji programavimo standartai. Tarp tokių galima paminėti MPI, OpenMP, gLite, XMPP-MPI, CUDA, OpenCL ir kiti. O tai reiškia, kad šiuolaikinis mokslininkas turi mokėti teisingai realizuoti savo algoritmus su viena ar keliomis technologijomis.

Kalbant apie lygiagrečiąsias technologijas, dažniausiai turima omeny daug kompiuterių, sujungtų kompiuteriniu tinklu į vieną skaičiavimo resursą. Jei toks skaičiavimo resursas nedidelis, tai ir jo pajėgumai nedideli. Jei auga pajėgumai, auga ir kompiuterių skaičius, iš kurių susideda skaičiavimų resursas. Tai įtakoja ir elektros energijos suvartojimą. Šiuo metu tai yra didelė problema, reikalaujanti dėmesio iš mokslininkų bei kompiuterinės įrangos gamintojų. Taip 2005 metais atsirado terminas, įvardinęs šią problemą – „green computing“ (liet. – *žalioji kompiuterija*). Pirmas terminą įvedė San Murugesan, pateikęs tokį apibrėžimą: žalioji kompiuterija yra mokslas bei praktiniai patarimai apie tai, kaip reikia projektuoti, gaminti, naudoti ir utilizuoti kompiuterius ir serverius, kad jie veiktų kuo efektyviau, minimaliai įtakojant arba visiškai neįtakojant aplinkos. Dabartiniu metu įvairių autorių straipsniuose galima sutikti ir kitokius šios problemos įvardinimus: žalios informacinės technologijos (žaliosios IT), aplinką tausojantys skaičiavimai ir IT. Šią problemą akcentuoja ir sąrašas Green500.com (egzistuoja nuo 2005 metų). Optimizavimo metodai gali padėti sprendžiant elektros energijos suvartojimo mažinimo uždavinius. Be to, įvairių uždavinių sprendimui nebūtina naudoti visų prieinamų skaičiavimo resursų, galima panaudoti tik jų dalį, siekiant mažinti jų elektros energijos suvartojimą. Šį faktą pabrėžia efektyvios žaliosios kompiuterijos posakis – „kuo didesnis kompiuterių skaičiavimo pajėgumas su kuo mažesne įtaka aplinkai“.

Darbo tikslas ir uždaviniai

Darbo tikslas – pasiūlyti elektros energijos suvartojimą mažinančius būdus, skirtus lygiagretiesiems ir paskirstytiesiems skaičiavimams.

Siekiant iškelto tikslo buvo sprendžiami šie uždaviniai:

- analitiškai apžvelgti žaliosios kompiuterijos technologijas ir elektros energijos suvartojimo problemą;
- ištirti esamas lygiagrečiųjų ir paskirstytųjų skaičiavimų technologijas;
- pasiūlyti nuoseklųjį, lygiagretųjį ir paskirstytųjų strypinių konstrukcijų tikslaus optimizavimo algoritmą;
- ištirti lygiagrečiųjų ir paskirstytųjų skaičiavimų sistemų elektros energijos suvartojimą, sprendžiant lygiagrečiųjų skaičiavimų ir optimizavimo uždavinius;
- pasiūlyti ir sukurti programinę realizaciją, skirtą mažinti lygiagrečiųjų ir paskirstytųjų skaičiavimų sistemų elektros energijos suvartojimą.

Tyrimo objektas ir metodai

Disertacijos tyrimo objektas yra technologijos, mažinančios kompiuterių elektros energijos sąnaudas, sprendžiant optimizavimo uždavinius, pasitelkiant lygiagrečiuosius ir paskirstytuosius skaičiavimus.

Analizuojant mokslinius ir eksperimentinius pasiekimus lygiagrečiųjų skaičiavimų srityje naudoti informacijos paieškos, sisteminimo, analizės, lyginamosios analizės ir apibendrinimo metodai. Remiantis eksperimentinio tyrimo metodu, atlikta statistinė duomenų ir tyrimų rezultatų analizė, kurios rezultatams įvertinti naudotas apibendrinimo metodas.

Darbo mokslinis naujumas

1. Ištirtas šakų ir rėžių algoritmo taikymas strypinėms konstrukcijoms optimizuoti.
2. Eksperimentiškai ištirtas lygiagrečiųjų ir paskirstytųjų skaičiavimo sistemų elektros energijos suvartojimas sprendžiant optimizavimo uždavinius.
3. Pasiūlyta ir eksperimentiškai ištirta lygiagrečiųjų ir paskirstytųjų skaičiavimo sistemų darbo strategija, skirta elektros energijos suvartojimui mažinti.

Darbo rezultatų praktinė reikšmė

Sukurtas elektros energijos mažinimo būdas skirtas lygiagretiems ir paskirstytiems kompiuteriams. Ši būdas gali būti pritaikyta ir kitiems kompiuteriniams sistemoms, pvz., superkompiuteriams, kompiuteriniams sistemoms su GPU ir kitoms.

Tyrimai atlikti moksliniuose projektuose:

- Lietuvos Respublikos švietimo ir mokslo ministerijos programa „Lygiagrečių ir paskirstytų skaičiavimų ir e-paslaugų tinklas (LitGrid)“ 2007–2009;
- Lietuvos valstybinio mokslo ir studijų fondo Aukštųjų technologijų plėtros programos projektas B-03/2007 „Globalus sudėtingų sistemų optimizavimas naudojant didelio našumo skaičiavimus ir grid technologijas“ 2007–2009;
- Lietuvos mokslo tarybos finansuojamas mokslininkų iniciatyva vykdomas mokslinis tyrimas „Globaliojo optimizavimo algoritmai su simpleksiniais posričiais“ 2010–2011;

- COST veikla IC0805 “Open European Network for High Performance Computing on Complex Environments” nuo 2008 iki dabar.

Praktinė darbo reikšmė:

- Pasiūlyta lygiagrečiųjų ir paskirstytųjų skaičiavimo sistemų darbo strategija buvo ištirta ir pritaikyta HPC.MII.VU.LT klasteryje.
- Bendradarbiaujant su Lenkijos „*Poznan Supercomputing and Networking Center*“ (*PSNC*) mokslininkais buvo testuojamas ir šiuo metu tobulinamas naujas lygiagrečiųjų ir paskirstytųjų skaičiavimų standartas XMPP-MPI.

Ginamieji teiginiai

1. Sukurta strategija leidžia sumažinti klasterių ir gridų elektros energijos suvartojimą iki 90 %.
2. Šakų ir rėžių kombinatorinis algoritmas leidžia gauti optimalią strypinę konstrukciją iki 9 mazgų topologijos optimizavimo uždaviniams.

Darbo rezultatų aprobavimas

Tyrimų rezultatai publikuoti 5 moksliniuose leidiniuose: 4 periodiniuose recenzuojamuose mokslo žurnaluose, 1 konferencijos pranešimų medžiagoje. Tyrimų rezultatai buvo pristatyti ir aptarti 9 nacionalinėse ir tarptautinėse konferencijose Lietuvoje ir užsienyje.

Darbo apimtis

Disertaciją sudaro 5 skyriai ir literatūros sąrašas. Disertacijos skyriai: Įvadas, Elektros energijos suvartojimas bei lygiagretieji ir paskirstytieji skaičiavimai, Elektros energijos suvartojimo mažinimo būdai lygiagrečiųjų skaičiavimų klasteriams, lygiagretieji šakų ir rėžių algoritmai, Eksperimentiniai tyrimai, Bendrosios išvados. Papildomai disertacijoje pateikti naudotų žymėjimų ir santrumpų sąrašai. Disertacijos apimtis 114 puslapiai, kuriuose pateikti 22 paveikslai ir 14 lentelių. Disertacijoje remtasi 125 literatūros šaltiniais.

Bendrosios išvados

Disertacijoje pristatyta žaliosios kompiuterijos sritis. Pasaulyje tai labai aktuali problema, bet Lietuvoje jai skiriamas per mažas dėmesys. Elektros energijos poreikis kiekvienais metais auga, nes vystomos technologijos, didėja klasterių ir superkompiuterių skaičius.

Disertacijoje pristatyti lygiagretieji kompiuteriai ir parodyta, kad žaliosios kompiuterijos atžvilgiu jie yra netobuli: neoptimalus užduočių paskirstymas tarp skaičiuojamųjų mazgų, pernelyg didelis elektros energijos suvartojimas. Reikia mažinti lygiagrečiųjų kompiuterių įtaką aplinkai ir spręsti kitas gridų, debesų kompiuterių, superkompiuterių bei klasterių problemas.

Siūlomas lygiagrečiųjų ir paskirstytųjų sistemų elektros suvartojimo mažinimo būdas pasižymi gerais rezultatais. Eksperimentinių tyrimų rezultatai leido daryti šias išvadas:

1. Siūlomas šakų ir rėžių algoritmas strypinių konstrukcijų optimizavimui veikia sparčiau (nuo 2,37 kartų 7 mazgams iki 3,6 kartų 8 mazgams) už pilnojo perrinkimo algoritmą. Pasiūlytu algoritmu per priimtiną laiką galima išspręsti iki 8 mazgų strypinių konstrukcijų optimizavimo uždavinius.

2. Pasiūlytas strypinių konstrukcijų optimizavimo gride šakų ir rėžių algoritmu būdas leidžia vykdyti paskirstytuosius skaičiavimus ir be įdiegtų pranešimų siuntimo bibliotekų, o taip pat ir neapsiribojant vienu klasteriu. Siūlomo būdo trūkumas – procesai-darbininkai neapsikeitinėja informacija tarpusavyje, dėl to gali išaugti reikalingų atlikti skaičiavimų kiekis. Paskirstytuoju algoritmu per priimtiną laiką galima išspręsti iki 9 mazgų strypinių konstrukcijų optimizavimo uždavinius.
3. Eksperimentų metu buvo nustatyta, kad grido paskirstytųjų skaičiavimų sistemos užduočių skirstymas nėra optimalus, dažnai klasteriai apkraunami neefektyviai skirstant uždavinius tarp mazgų-darbininkų. Dėl to skaičiavimams atlikti suvartojama kur kas daugiau elektros energijos (iki 2,5 karto) negu reikėtų skirstant užduotis optimaliai.
4. Lygiagrečiųjų skaičiavimų klasteriai bei grido klasteriai vartoja elektros energiją net ir nevykdydami skaičiavimų – 50 Wh per valandą vienam kompiuteriui.
5. Disertacijoje pasiūlytas elektros energijos taupymo būdas – automatiškai išjunginėti mazgus-darbininkus kol nėra skaičiavimo užduočių. Siūlomas būdas leidžia sumažinti klasterio elektros energijos suvartojimą iki 90 %, kai kompiuteriai laukia užduočių. Kai 16 kompiuterių klasteris su Intel Core i5 CPU 760 @ 2.80GHz procesoriais yra apkrautas apie 50 %, sutaupoma apie 2882 kWh elektros energijos per metus.

Aleksandr Igumenov

ELECTRICAL ENERGY AWARE PARALLEL
AND DISTRIBUTED COMPUTING

Summary of Doctoral Dissertation

Technological Sciences,
Informatics Engineering (07T)

Aleksandr Igumenov

LYGIAGRETIEJI IR PASKIRSTYTIEJI ELEKTROS
ENERGIJĄ TAUSOJANTYS SKAIČIAVIMAI

Daktaro disertacijos santrauka

Technologijos mokslai,
Informatikos inžinerija (07T)