

Effective utilization of processors in computer cluster when solving problems of various parallelizable scales

Tomas Petkus¹, Olga Kurasova¹, Ernestas Filatovas²,
Aleksandr Igumenov², Sergejus Ivanikovas²

¹*Faculty of Science and Technology, Lithuanian University of Educational Sciences*
Studentų str. 39, LT-08106 Vilnius, Lithuania

²*Faculty of Fundamental Science, Vilnius Gediminas Technical University*
Saulėtekio al. 11, LT-10223 Vilnius, Lithuania

E-mail: tomas.petkus@leu.lt, olga.kurasova@leu.lt, ernestas.filatovas@vgtu.lt

E-mail: aleksandr.igumenov@vgtu.lt, sergejus.ivanikovas@vgtu.lt

Abstract. In this paper, effective utilization of processors in computer cluster is investigated. Generally, a computer cluster is treated as a specially designed system only for solving specific computationally intensive and time consuming problems. Commonly, computers in offices and universities are idle for a considerable proportion of the time. Therefore, in this research, we focus on exploitation of such computational resources. The experimental investigation shows that when executing multiple processes in each CPU core allows to solve computationally intensive problems more precisely compared to the case, where the number of processes is equal to the number of CPU cores, meanwhile the same amount of energy is consumed in both cases.

Keywords: parallel computing, computer cluster, processor utilization, power consumption and savings.

Introduction

Recently, high performance computing (HPC) centres with huge resources are constantly established and their computational power continually increases. According to the concept of such centres, they are designed for solving large scale problems, where extremely high performance computational resources are required. However, there are many types of problems that could be solved by using computer clusters consisted of serial CPU and GPU due to the fact that computing performance of the current processors enables to solve complicated problems with high enough computational burden. Moreover, personal computers (PCs) in offices and universities are idle for a considerable proportion of the time. Thus, we can exploit these PCs for computations on demand. The aim of this paper is to investigate effectiveness of processor utilization in a PC cluster when solving problems of different parallelizable scales.

1 Effective utilization of processors

At the time, when PCs in offices and universities are idle, they can be utilized effective for research purposes related to parallel computing or even for solving real-world time consuming problems [2, 5, 7]. If a PC cluster is devoted only for high performance computing, usually Linux is used in the PCs. However, often in offices and universities, MS Windows is used, thus, parallel computing has to be organized considering peculiarities of this operating system [2, 3, 6]. Moreover, nowadays, energy consumption and savings are ones of the main issue of green computing when drawing up computing resource usage strategies and developing energy efficient algorithms [1]. Even a typical desktop PC can consume sufficiently high amount of energy while idling without any gain in productivity [4].

Thus, a PC cluster needs to be used as effective as possible. A concept of one of the usage strategies is based on volunteer computing, also known as public-resource computing, where distributed computing relies on public donating the processing power for solving one of the purposeful problems.¹ Majority of such computing uses BOINC infrastructure. However, if an institution has a specific computational burden problem to be solved and do not have specialized computational resources, a solution for this task could be the employment of PCs available in the institution. Only requirement is that the PCs must be connected to a local network and have task scheduling software. Such a computer network composes a computer cluster that could be used for computing on demand independently whether the PCs are used by users for everyday works or stay idle. It is necessary to utilize these resources effectively, therefore, strategies of problem parallelizing should be investigated. One of ways of effective utilization of a PC cluster is based on executing multiple processes simultaneously in each CPU core.

2 Experimental investigation

The experimental investigation aims to show the dependence of computing results on the number of processes run parallel to solve problems of various parallelizable scales – sizes of tasks to be solved in a PC simultaneously. The following hardware and software of PC cluster and local network are used in experiments: Intel® Core™ i5-2400 CPU @ 3.4 GHz with 4 cores; 4 GB DDR3 RAM; Intel DH61CR Motherboard; Integrated Intel®82579V Gigabit Ethernet LAN controller; HP 1810-48G Switch, 48 10/100/1000 and 4 SFP 100/1000 Mbps ports; MS Windows 7 OS; MPICH [x86] 1.4.1p1, Visual Studio C++ 2010 Express; CoreTemp 1.0 RC6 for CPU watt and temperature measurement; Brennenstuhl PM 231 E as a energy cost gauge. The classic master–slave communication model under MPI standard is used in the PC cluster.

There are many real-world problems where during the solving process a master and slaves must constantly exchange the results obtained. The characteristics of such problems are as follows:

- the whole problem can be divided into tasks that can be effectively solved in a parallel way;

¹ <http://www.distributedcomputing.info/>.

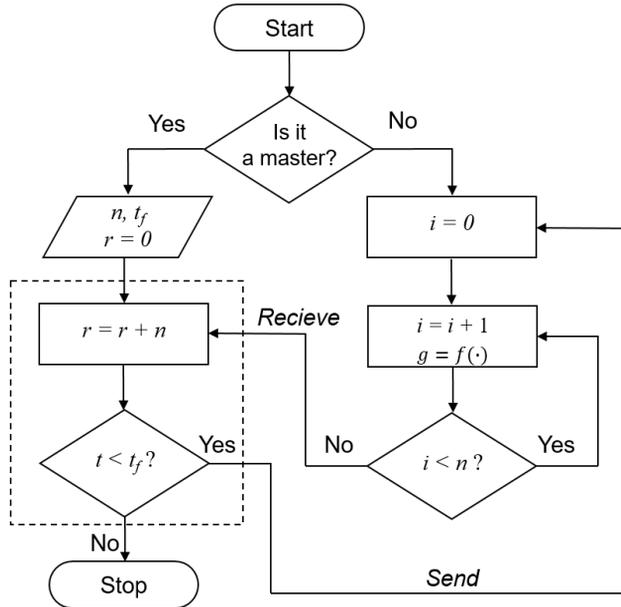


Fig. 1. Used parallelizing scheme.

- each task consists of series of function evaluations, therefore the size of a task n can be easily controlled;
- the smaller size of a task n , the more sending/receiving operations are performed, however a more precise solution of the problem is found faster.

For the experimental investigation, a problem with the described characteristics has been simulated and an parallel algorithm to solve this problem has been developed (see Fig. 1). The dividing of the problem into tasks of various sizes n simulates problems of various parallelizable scales. The result of the simulated problem is a number of function $f(\cdot)$ evaluations r , this number indicates accuracy of the result (the higher value, the more precise result). The computational process is iteratively repeated until the current time t reaches the finish time t_f .

For simplicity, the size of a task n corresponds its computation time in seconds. The experiments with various task sizes have been performed: $n = 0.0001, 0.001, 0.01,$ and 0.1 . The time of solving the whole problem has been limited to 60 s. A series of experiments has been conducted, where various number of processes p has been run in a single CPU. As a slight randomization exists in function $f(\cdot)$ calculation, each experiment has been repeated 10 times. The minimal, maximal and average number of function evaluations are fixed. Confidence intervals of the averages are estimated with a confidence level 0.9.

The dependence of the number of function evaluations r on the task size n and the number of processes p is presented in Fig. 2. We see the larger task size n , the higher number of function evaluations r are obtained. It should be noted that the number of processes p also influences the results obtained. By increasing p , the number r are also increasing, although the number of physical CPU cores is the same and equal to four. We see that the worst results are got when running four processes ($p = 4$),

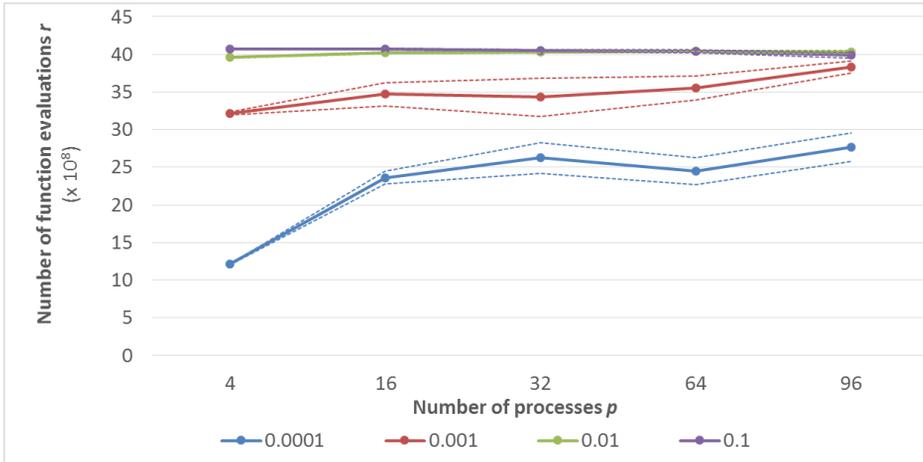


Fig. 2. Dependence of the number of function evaluations r (averages and their confidence intervals) on the task size n and the number of processes p .

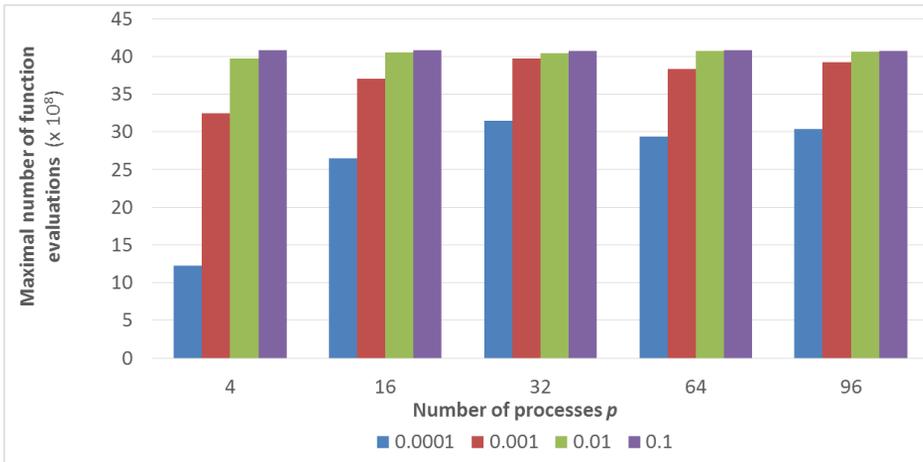


Fig. 3. Dependence of the maximal number of function evaluations on the task size n and the number of processes p .

as same as physical cores. When the number of running processes p is increased up to 16 processes, the number of function evaluations r is almost doubled in the case, when the task size $n = 0.0001$.

Often when solving real-world problems, it is necessary to get not only precise averaged results, but also the most precise results as possible. This corresponds the maximal number of function evaluations of the simulated problem that are shown in Fig. 3. Table 1 shows that it is worth to run more than one process on each CPU core. If the task size is small enough ($n = 0.0001$ or 0.001) the maximal numbers of function evaluations are achieved with 32 processes. A greater number of processes does not allow getting better results. When the task size is larger and equal to 0.01 or 0.1 , running more processes does not influence the results obtained so much (Fig. 3).

Table 1. Percentage increase of the number of function evaluations r compared to the case of four processes.

$n \setminus p$	16	32	64	96
0.0001	115%	156%	139%	147%
0.001	14%	22%	18%	21%

It is worth to point out that nowadays energy consumption and savings are ones of the main factors when drawing up computer cluster usage strategies and developing energy efficient algorithms. Most of the time PCs in offices and university computer labs do not perform any tasks, however even idling PCs consume electricity. Thus, the power consumption has been measured when solving the computationally intensive problem in a PC cluster: a idle computer uses 25 watts, a computer (without monitor) – 89 watts, CPU – 49 watts, and this consumption does not depend on the number of running processes p , if this number is greater than the number of cores. Therefore, it is purposeful to run multiple processes parallelly also for the purpose of energy consumption and savings, because more precise results can be obtained consuming the same amount of energy.

3 Conclusions

In this paper, the utilization of processors in a PC cluster has been investigated by solving the simulated computationally intensive problem. The problem has been parallelized and examined by varying the task sizes and number of processes dedicated to each CPU core. The experimental research allows to draw such conclusions:

- The multi-core processors are more effectively utilized, if more than one process is dedicated to each CPU core. The number of processes influencing effective utilization of processors depends on problem parallelizable scales. The greatest effectiveness is observed, if a task size is sufficiently small. In this case, the number of function evaluations increases to 156% compared to the case where the number of processes coincides with the number of CPU cores.
- A CPU uses the same amount of energy without reference to the number of executing processes, if the number is greater than the number of CPU cores. Thus, it is also purposeful to execute multiple processes in a core for a purpose of energy savings.

Moreover, when carrying out experiments, it has been determined that more precise results are obtained, each process executes approximately the same number of tasks on a fixed time. Thus, further researches should be related to developing cluster usage strategy considering to this fact.

References

- [1] J. Carretero, S. Distefano, D. Petcu, D. Pop, T. Rauber, G. Runger and D.E. Singh. Energy-efficient algorithms for ultrascale systems. *Supercomp. Front. Innov.*, **2**(2):77–104, 2015.

- [2] S. Chien, G. Makinabakan, A. Ecer and H.U. Akay. Parallel computing on network of Windows based PCs. In *Parallel Computational Fluid Dynamics 2007*, pp. 155–162. Springer, 2009.
- [3] T. He, J. Ni and G. Wang. A heterogeneous Windows cluster system for medical image reconstruction. In *First International Multi-Symposiums on Computer and Computational Sciences, 2006. IMSCCS'06*, volume 1, pp. 410–415. IEEE, 2006.
- [4] K. Kurowski, A. Oleksiak, M. Witkowski and J. Nabrzyski. Distributed power management and control system for sustainable computing environments. In *Green Computing Conference, 2010 International*, pp. 365–372. IEEE, 2010.
- [5] H. Lin, O. Sirisaengtaksin and P. Chen. The establishment of a cluster computing environment at a small institution. In *Proceedings of the 14th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI 2010)*, volume 1, pp. 161–166, 2010.
- [6] J. Qiu and S.-H. Bae. Performance of windows multicore systems on threading and MPI. *Concurrency Comput. Pract. Exp.*, **24**(1):14–28, 2012.
- [7] S.-T. Wang, Y.-C. Chen and H.-Y. Chang. Enabling automated deployment for cluster computing in distributed PC classrooms. *World Acad. Sci., Eng. Techn.*, **65**:993–997, 2012.

REZIUMÉ

Efektyvus procesorių panaudojimas kompiuterių klasteryje įvairaus išlygiagretinamumo uždaviniams spręsti

T. Petkus, O. Kurasova, E. Filatovas, A. Igumenov, S. Ivanikovas

Straipsnyje nagrinėjamas efektyvus procesorių panaudojimas kompiuterių klasteryje. Įprastai kompiuterių klasteris yra sukonstruojamas kaip vieningas unifikuotas kompiuterinis resursas, skirtas skaičiavimams imliems ir daug laiko reikalaujantiems uždaviniams spręsti. Tačiau dažnai įmonių biuruose, taip pat ir universitetuose kompiuteriai žymią paros dalį nėra naudojami. Tikslinga juos tuo metu išnaudoti sudėtingiems skaičiavimams atlikti. Eksperimentinis tyrimas parodė, kad vykdant keletą procesų viename procesoriaus branduolyje galima išspręsti skaičiavimams imlius uždavinius daug tiksliau, lyginant su atveju, kai vykdomų procesų skaičius lygus branduolių skaičiui. Be to, abiem atvejais išnaudojamas vienodas elektros energijos kiekis.

Raktiniai žodžiai: lygiagrečiai skaičiavimai, kompiuterių klasteris, MPI, procesorių apkrovimas, energijos suvartojimas ir taupymas.