ANALYSIS OF PERFORMANCE BOTTLENECK OF P2P GRID APPLICATIONS

MAREK ŠIMON, LADISLAV HURAJ AND VLADIMÍR SILÁDI

Abstract

The OurGrid distributed environment is an open source grid middleware that enables the creation of P2P computational grids to speed up the execution of bag-of-tasks applications. Bag-of-tasks is an application model composed of several independent tasks not communicating with each other during their execution, tolerating network delays and faults. The paper shows that the main advantage to avoid bottlenecks is if the data gets larger or computations become more complex and the conclusion is made based on the presented measurements. Understanding such patterns sheds an insightful light on performance bottlenecks and design tradeoffs for P2P Grid applications.

General Terms: parallel computing, peer-to-peer, OurGrid
Additional Key Words and Phrases: grid computing system, performance of parallel applications

---

1. INTRODUCTION

Foster et al. [2001] define grid as controlled and coordinated resource sharing and resource use in dynamic, scalable virtual organizations. The thousands, or even millions of processors that may be accessible within a Virtual Organization represent a significant source of computational power, if they can be harnessed in a useful fashion.

Peer-to-peer (P2P) applications have been originally used for single services e.g. file sharing, and are now considered to a larger scope from video streaming to system update and distributed data base. P2P applications have found new usability based on recent advances in microprocessors architecture and networks in High Performance Computing (HPC) [Nguyen et al., 2010].

The P2P computing distributes computational tasks among the member nodes instead of concentrating it at a single server, in an autonomous, heterogeneous, and decentralized nature of participating peers. However, current P2P schemes offer limited data management facilities.

In the paper two different pending time-consuming applications and the possibility of their parallelization through special P2P computing environment OurGrid are tested. First one deals with interpolation methods for determination of the snow cover depth from large geo-data kept from raingauge stations [HURAJ et al., 2010a; HURAJ et al., 2010b; SILADI et al., 2011].

The second application researches stimulus responsible for radioactive decay generated by physical vacuum; therefore it seems that knowledge characterizing this decay could bear certain information on the structure of the physical vacuum and this structure is very probably indeed fractal [Húšťava et al., 2013].

So the aim of the present paper is to show an implementation on distributed architectures based on P2P parallel computing systems for the solution of snow cover computing as well as of radioactive decay characteristics' problem by parallel computations.

The feasibility of two time-consuming applications based on implementation of a P2P computational grid OurGrid is presented as well as the comparison of the parallel algorithms' performance with the sequence ones.

OurGrid is an open source grid middleware that enables the creation of P2P computational grids to speed up the execution of bag-of-tasks applications. P2P environment OurGrid can simultaneously submit job to node and execute job submitted by other nodes. In that sense each node can be both resource provider as well as consumer, which conforms to the design principle of P2P applications and so to establish an ad hoc grid application.

Bag-of-tasks (BoT) is an application model that is especially suitable for execution of grid applications since it is composed of several uniprocessor tasks. The tasks demand no communication during its execution, tolerating network delays and faults. These characteristics facilitate OurGrid powerful resources available in a grid for expedition of the execution of BoT applications [ROSE et al., 2008].

The rest of the paper is organized as follows. The next section reviews principles of P2P computing, including the OurGrid environment for parallelization. Then the performance of the parallel experiments is evaluated in next section. Finally, a conclusion is drawn in the last section.

2. PEER TO PEER GRID COMPUTING

Since peer-to-peer and grid computing systems bring together distributed resources, both systems have numerous similarities. On the whole, peer-to-peer systems are designed to achieve a single task (e.g. file sharing), while grids are multi-purpose and aim to offer more flexibility for distributed applications. Moreover, the installation, configuration and administration of peer-to-peer systems are generally easier [SMITH et al., 2004].

ANALYSIS OF PERFORMANCE BOTTLENECK OF P2P GRID APPLICATIONS

The OurGrid is a P2P system that addresses the issue of grid assembly for applications that can benefit from best-effort resource allocation. Each user can interact with particular OurGrid community through a client broker, which hides the heterogeneous grid structure and provides fast application turn-around for end user.
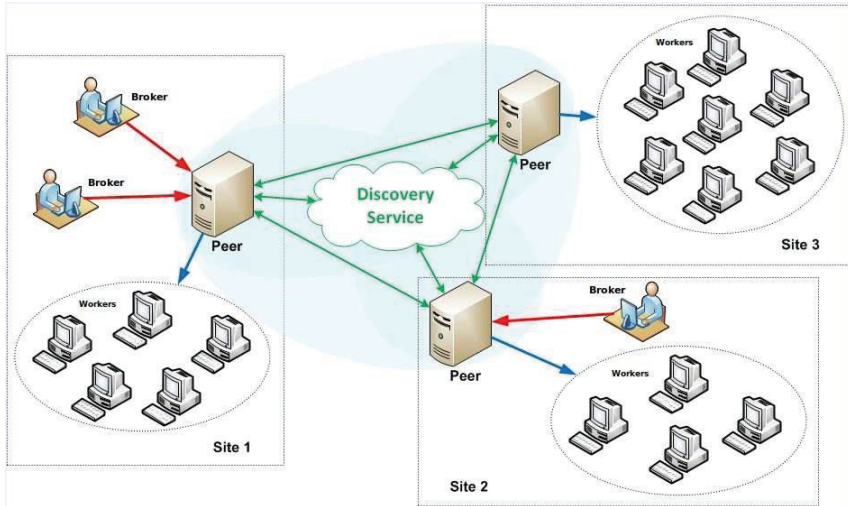

Fig. 1 Main structure of the OurGrid (www.ourgrid.org/images/OurGrid2.JPG)

OurGrid provides two modules for resource access and scheduling called Peer and MyGrid. Peer is responsible for connecting to other Peers in particular community and discovering resources to users (through the MyGrid broker and scheduler). Each user can be connected to only one Peer in the community. The Peer controls resource distribution among other Peers and local users. MyGrid broker schedules tasks using all available resources provided by its Peer. The goal of this strategy was to use opportunistic computing techniques providing idle resources from space-shared resources to grid users [ROSE et al., 2008; CALHEIROS et al., 2008; ANDRADE et al., 2005; ZHAO et al., 2010].

Nevertheless, the applications' parallelism is in the program module, and the application is parallelized in the Single-Program-Multiple-Data (SPMD) style.

In OurGrid a task is executed in three steps described in jdf file based on OurGrid's protocol for resource description and job management:

1. the initial step performs the upload of input data and executables;

2. a remote step performs the execution of one or more programs at the remote machine;

3. a final step concludes the execution of the task, usually collecting any output generated in the previous step [SANTOS et al., 2007].

Preparing of the parallel job execution in our two cases is detailed with two steps: division to the input data vector for computational tasks and further collecting to the results of the computational tasks.

In general, running a computational job in a P2P grid middleware requires a management work. A job must be divided into smaller tasks. Each task has to be allocated to worker peer and its status and progress needs to be tracked. Finally the individual results must be collected, verified and merged to the overall solution of the job [WANDER et al., 2010].

3. PERFORMANCE OF EXPERIMENTS

We made two sets of experiments on OurGrid scheme. Both of them ran on PC workers with CPU Intel Core2 QUAD 2.66GHz with Ubuntu 1204 as the operating system. The OurGrid jobs were managed by a central Peer and a Broker running on server with OS Ubuntu. Currently we can pursue the high performance with OurGrid only in a local area network. For the parallelization we used the previously designed C program which is relatively time consuming due to large data matrices in snow cover computing experiment and due to extensive loops in the radioactive decay experiment.
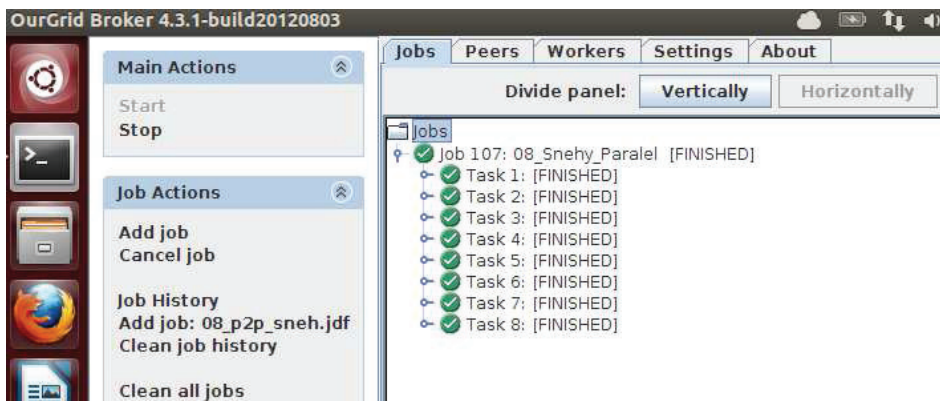

Fig. 2 A Broker in OurGrid system with a job and 8 tasks

The first set of computational experiments has been carried out on 1, 2, 4 and 8 machines/workers. The sizes of the matrices varied from 1000 × 1000 to 10 000 ×

10 000. All experiments on OurGrid scheme were compared to the sequential experiment on single CPU.

As could be seen from Figure 3, there is no increase of efficiency compared to sequential algorithm. On the contrary, the overhead for data distribution and collection and for task management was exceedingly high for all amounts of workers compare to clear time of sequential algorithm.

However, if we compare particular times of performance of the program execution at the remote workers' machines without overhead, the result times are corresponded to particular time ratio of the sequential algorithm execution time. The final time is strongly dependent on task management and OurGrid environment is insufficient for this type of tasks.
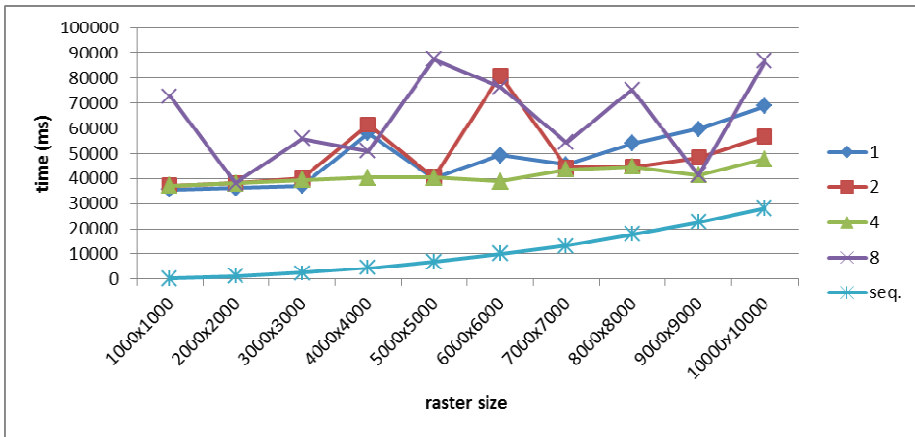

Fig. 3 Chart of execution time depending up problem size (raster size) and number of workers

The second set of experiments for the radioactive decay was performed for specific parameters: 60 000 points, min and max difference 9767 and dimension 22. The experiment was carried out on 20 machines/workers.

Duration time of whole job was 309 973 707 msec, i.e. 3.588 days. If we only consider the time of the CPU execution for the remote step, the slowest task duration was 309 531 000 msec, i.e. about 3.583 days. The overhead difference is 442707 msec, what is only 0.14% from whole time.

The duration of sequential version of the application was guessed depending on its loops and mathematical operations to 1.24E+16 msec, i.e. about 57.29 days. The

efficiency factor of the second experiment for the radioactive decay is 15.97 that indicates OurGrid environment as appropriate paralleling tool for such kind of tasks.

4. CONCLUSIONS

In this paper, possible techniques used for manual optimization and improvement of the performance of parallel applications on OurGrid environment are presented.

Based on the presented measurements, the following two facts can be concluded: utilizing of OurGrid is meaningful if the data gets larger or computations become more complex; in case of time-consuming but not compute-intensive jobs, the increasing of number of tasks/workers will not increase parallel efficiency, in our case for application of the spatial interpolation method of snow cover depth. On the contrary, it will increase the performance bottleneck between the source and the remaining nodes of the network.

There are other advantages and disadvantages of OurGrid scheme computing. Pros are e.g. donating a worker is extremely easy, setting up and maintaining a OurGrid scheme is much easier than installing the server software of utility grids. On the other hand, the various workers' areas do not have high-speed connections. Another current limitation of OurGrid scheme is that it can only be used for bag-of-tasks parallel applications. However BoT-parallel applications are useful in many fields and we will have a good and practical solution for running BoT application on P2P environment in a couple of years.

The performance improvement of experimental evaluations of other parallel computing techniques for spatial interpolation not based on P2P computing (GPGPU, multi-core CPU design, system with distributed memory) were described in our previous articles [HURAJ et al., 2010a; HURAJ et al., 2010b; SILADI et al., 2011].

In conclusion, the experiments indicate that the P2P computing will be efficient for compute-intensive parallel projects where few minutes' overhead does not have a significant impact on the overall computation time.

REFERENCES

ANDRADE, N., AND COSTA, L., AND GERMOGLIO, G., AND CIRNE, W. 2005. Peer-to-peer Grid Computing with the OurGrid Community. In *Proceedings of the SBRC*, 2005.
CALHEIROS, N., AND FERRETO, AND T., ROSE, C. D. 2008. Scheduling anf management of virtual resources in grid sites: the site resource scheduler. In *Parallel Processing Letters*, 2008.
FOSTER, I., AND KESSELMAN, C., AND TUECKE, S. 2001. The anatomy of the grid: Enabling scalable virtual organizations. In *International Journal of High Performance Computing Applications 15* (3) (2001) 200–222.

HURAJ, L., AND SILÁDI, V., AND SILÁČI, J. 2010a. Design and Performance Evaluation of Snow Cover Computing on GPUs, In *Proceedings of the 14th WSEAS International Conference on Computers: Latest Trends on Computers*, Corfu Island, Greece, July 2010, pp. 674-677, ISBN: 978-960-474-213-4.

HURAJ, L., AND SILÁDI, V., AND SILÁČI, J. 2010b. Comparison of Design and Performance of Snow Cover Computing on GPUs and Multi-core processors, In *WSEAS TRANSACTIONS on INFORMATION SCIENCE and APPLICATIONS*, Issue 10, Volume 7, October 2010, pp. 1284-1294, ISSN: 1790-0832

HÚŠŤAVA, Š., AND DIRGOVÁ LUPTÁKOVÁ, I., AND ÖLVECKÝ, M., AND ŠIMON, M., AND VADKERTIOVÁ, A., AND JURINOVÁ, J. 2013. Investigation of Chaotic Dynamics at 241Am Radioactive Decay. In *Aktuálne problémy súčasnej vedy*. ISSN 2227-1392, Issue 2(3), 2013, pp. 32-36.

NGUYEN, T. T., AND EL BAZ, D., AND SPITERI, P., AND JOURJON, J., AND CHAU, M. 2010. High performance peer-to-peer distributed computing with application to obstacle problem, In *Proceedings of IEEE IPDPS*, Workshop HOTP2P, 2010, USA.

ROSE, C. A. F. D., AND FERRETO, T., AND CALHEIROS, R. N., AND CIRNE, W., AND COSTA, L. B., AND FIREMAN, D. 2008. Allocation strategies for utilization of space-shared resources in bag of tasks grids. In *Future Gener. Comput. Syst.*, 24(5):331–341, 2008.

SANTOS, R., AND ANDRADE, A., AND CIRNE, W., AND BRASILEIRO, F., AND ANDRADE, N. 2007. *Relative Autonomous Accounting for Peer-to-Peer Grids. Concurrency and Computation: Practice and Experience*, September 2007.

Siládi, V., and Siláči, J., and Huraj, L. 2011. Performance Evaluation and Comparison of MPI and OpenMP paradigms for interpolating computation, In *Aktuaľnyje problemy i innovacii v ekonomike, upravlenii, obrazovanii, informacionnych technologijach*, Issue 6, Volume 1, May 2011, pp. 185-188, ISSN: 2074-1685.

SMITH, M., AND FRIESE, T., AND FREISLEBEN, B. 2004. Towards a service oriented ad-hoc grid. In *Proceedings of 3rd International Symposium on Parallel and Distributed Computing*, 2004.

WANDER, M., AND WACKER, A., AND WEIS, T. 2010. Towards Peer-to-Peer-based Cryptanalysis. In *6th IEEE Workshop on Security in Communication Networks*, SICK 2010, Denver, USA, 2010.

ZHAO, H., AND LIU, X., AND LI, X. 2011. A Taxonomy of Peer-to-Peer Desktop Grid Paradigms. *Cluster Computing: The Journal of Networks, Software Tools, and Applications*, Springer, 2011.

Marek Šimon and Ladislav Huraj
Department of Applied Informatics and Mathematics, University of SS. Cyril and Methodius,
J. Herdu 2, Trnava, SK-917 01,
Slovak Republic

Vladimír Siládi
Department of Computer Science, Matej Bel University,
Tajovského 40, 97401 Banská Bystrica,
Slovak Republic