

Energy Conservation In Computational Grids

Monika Yadav¹ and Sudheer Katta² and M. R. Bhujade³

¹Department of Computer Science and Engineering, IIT Bombay
monika@cse.iitb.ac.in

²Department of Electrical Engineering, IIT Bombay
ksudheer@ee.iitb.ac.in

³Faculty at Department of Computer Science and Engineering, IIT Bombay
mrb@iitb.ac.in

Abstract

With the increasing development of high-speed wide-area networks and powerful though low cost computational resources, grid computing has emerged as an attractive computing paradigm. With recent advances in computing, the scope of grid computing has been expanded to include mobile and pervasive devices. Grids have become one of the highest consumers of energy. High performance computing is achieved at the cost of energy consumption. Increasing the performance of computer systems requires increasing number of resources which leads to higher power consumption and it creates negative impact on environment. Idle sites in grid are direct sources of energy waste. Saving energy causes two direct advantages: financial and environmental cost reduction and sustainability. With the Grid becoming a viable high-performance alternative to the traditional supercomputing environment, various aspects of Grid resource utilization which can help in energy conservation are gaining significance. In this paper we explore the possibility of reduction in energy consumption of a grid of heterogeneous computers by making use of different power states of cpu.

KEYWORDS

Grid Computing, load balancing

1. INTRODUCTION

The rapid development in computing resources has enhanced the performance of computers and reduced their costs. Grid is a type of distributed system which supports sharing and coordinated use of resources independently from their physical type and location in dynamic virtual organisations that share same goal[1]. Grids have a lot of specific characteristics like autonomy, heterogeneity, scalability, adaptability and resources computation data separation. Previous researches on grid computing have shown load balancing , improvement of performance and security as major issues. Little has been done on energy conservation in computational grids. The consideration of energy as a primary concern in operating environments was proposed almost a decade ago [2]. To enhance the performance of computer systems requires increasing number of resources which leads to increase in power consumption. A great deal of energy used in high performance computing [3] could be saved. High energy consumption has numerous negative effects ,one of them being increase in heat production which leads to increase in costs because of cooling required.

When cluster nodes consume and dissipate more power, they must be spaced out and aggressively cooled; otherwise un-dissipated power causes the temperature to increase rapidly [3]. Hsu and Feng [3] found that cost of cooling server rooms in some institutions is as high as 70% of the energy used to power the equipment that is being cooled. Power consumption of computing nodes in idle state is a high proportion of their peak power consumed. All servers

draw 60% or more of their peak power even when idle [4]. So power consumption and heat dissipation are two important problems when number of machines are housed together.

The most common method employed to reduce energy consumption is voltage and frequency scaling (VFS) [5],[6],[3]. This involves reducing the processor's power consumption by adjusting its voltage and frequency either manually [6] or dynamically DVFS [3],[7],[5],[8] by means of processor's technologies such as Intel's speedstep [9]. The methods employed to energy consumption can cause load unbalancing. Unbalancing is the strategic use of load balancing to concentrate the processing on fewer processors so as to completely unload other servers. The energy saving is achieved by unloaded computers idling, being switched to low power state or even being turned off. We try to raise the issue of conserving energy for computational grids through power aware scheduling. Energy aware techniques involve management of workloads mainly in schedulers.

2. SUGGESTED SYSTEM MODEL

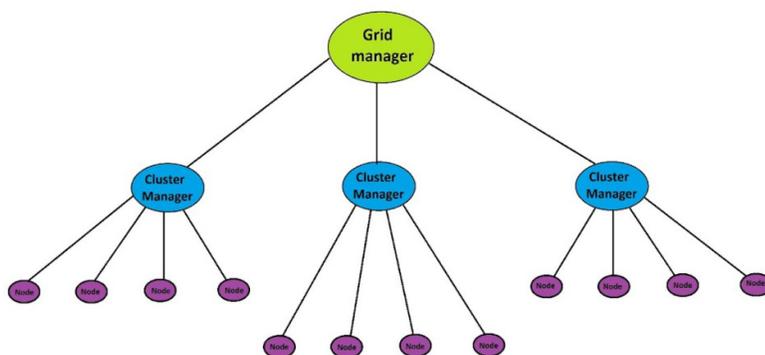


Figure 1: Grid Topology

We regard grid computing as a set of clusters where each cluster owns a set of worker nodes present in a local domain i.e. a LAN. Clusters are connected together using WAN. Grid is represented as a three level tree. Consider the root as Grid manager at level 0. Each cluster represents a subtree. Leaves of the sub tree are nodes of the cluster. Cluster manager is the root of each cluster. Subtrees corresponding to each cluster are aggregated to form a three level tree. Cluster manager is a virtual node and is responsible to each cluster. Assume that the nodes within the cluster are connected using high speed LAN.

Level 0 : It represents grid manager. It receives the task information from the cluster manager of the cluster where the task is submitted and sends this information to every other cluster manager. It receives the computed energy values from all the clusters and compares them.

Level 1 : It represents cluster manager. It does energy calculations on all nodes in its cluster for the submitted task and finds the minimum energy consuming node in the cluster.

Level 2 : The lowest level corresponds to the computing node linked to their respective cluster. It receives the tasks from users. It is the execution or the worker node where the task will be executed.

3. ENERGY CALCULATIONS AND PROPOSED ALGORITHM

3.1 Power states of CPU

Power state of cpu is the capability by which a processor switches between different operating frequencies and voltages. This property enables power saving. The Advanced Configuration and power Interface Specification defines the CPU P-states power management states. If configured properly according to system workload, this feature provides power savings. Many technologies offer the clock speed of processor to change dynamically by software according to requirement. This feature enables the task to meet the needs it requires for processing along with minimisation of power consumed and reduces heat dissipation. Power consumed by cpu depends on capacitance, voltage and frequency. Running a processor at high clock speeds allows for better performance. However, when the same processor is run at a lower frequency (speed), it generates less heat and consumes less power. In many cases, the core voltage can also be reduced, further reducing power consumption and heat generation. Power consumption is lower at lower processing speeds. Capability of an idle processor to turn off unused components to save power can be achieved.

There is a series of dynamic frequency scaling built into some Intel microprocessors which can provide an excellent balance between providing power when you need it and conserving it when you don't. It allows the system to dynamically adjust processor voltage and core frequency, which can result in decreased average power consumption and decreased average heat production. Of course the most energy-efficient approach would be to allocate all tasks to the most energy-efficient resource and turn off all the other resources. However, that approach would considerably slow the execution of tasks sent to the grid, reducing its utility, as each task would need to wait its turn before being allowed to process.

P_a : power state of cpu during ALU operations.

P_m : power state of cpu during memory operations.

P_t : power state of cpu during migration of task to / from the node.

P_I : power state of cpu during idle condition.

Assume power consumed at P_I is negligible and the other power states in terms of power consumed by them are arranged as $P_a > P_m > P_t$. The lowest power state is P_I when the cpu is in idle mode. So the cpu can be switched off in this state. Analysing the machine code will give number of cpu instructions and memory instructions.

3.2 Energy Equations

The energy equation at a node if the task is executed by the node where the task is submitted or some other node in the same cluster.

$$E = N_m * P_m / S_m + N_a * P_a / S_a + K \quad (1)$$

Here

N_m corresponds to number of memory instructions.

S_m is speed of memory operations.

P_m is power state which refers to power consumed by cpu during memory operations.

N_a is number of ALU operations.

S_a is speed of ALU operations.

P_a is power state which refers to power consumed by cpu during ALU operations.

K is the constant energy that corresponds to sum total energy for compilation of the task and

transmission and reception energy if at all by the system which has got the job.

The energy equation at a node if the task is submitted to a node in some other cluster is given as follows.

$$E = N_m * P_m / S_m + N_a * P_a / S_a + 2 * P_t * T_s + K \quad (2)$$

Here

P_t corresponds to power state which refers to power consumed by cpu during transmission of the task to/ from the node.

T_s is the time taken for task migration which can be computed given size of task, size of memory if at all being transferred and link speed.

3.3 Analysis of the Algorithm

Tasks are submitted to nodes at lowest level. Task information corresponds to number of cpu instructions and number of memory instructions in the task . The node sends this task information to its cluster manager. The cluster manager computes the energy values if the task was executed at some node within the same cluster. It sends the minimum computed value to the grid manager along with task information. The grid manager sends this information to every other cluster and compares the energy values. The task is scheduled to the node which consumes minimum energy given that the node satisfies task's requirements.

For job j_x arriving at node N_i

N_i sends task information to its cluster manager C_a .

C_a calculates energy consumption at each node in the cluster using equation (1) given that the job queue of node is not full and memory requirements are met and sends the minimum value to grid manager along with task information.

Grid manager sends this task information to every other cluster manager.

For each cluster C_i receiving task information from grid manager

For each node N_j in cluster C_i

C_i calculates energy that will be consumed on executing the task on node N_j using equation (2) given that the job queue of node is not full and memory requirements are met.

C_i sorts the calculated energies received from all the nodes in cluster and sends the minimum value E_x to grid manager.

Grid manager sorts computed energy values E_x received from all the cluster managers along with that received from C_a . It picks the minimum value and sends the cluster_Id of the cluster corresponding to this minimum value to C_a .

If (cluster_Id $\neq C_a$)

Schedule the task to the cluster corresponding to cluster_Id

Else

Schedule the task to C_a itself.

4. Conclusion

With the increasing demand of global grids, the energy consumption of grid infrastructure has escalated to the degree that grids are becoming a threat to the society rather than an asset. There is a limited energy budget for grid applications. We have tried to explore energy constrained scheduling to maximize the total utility of all grid applications with limited energy budget. This is an efficient approach which will give a good comparison of energy consumption at each node in all clusters for a given task. We have explored the effect on energy consumption if the task was migrated to some node within same cluster or a node in some other cluster in the grid. The cluster manager associated with each cluster keeps information of each node's speed of cpu and memory operations. Implementing this algorithm in high performance computing or simulations will decrease the energy consumption of the system. There won't be much change if we implement it on system which have IO operations or working on real time applications.

References

- [1] Foster, I., C. Kesselman and S. Tuecke, "The anatomy of the grid: Enabling scalable virtual organisations." *Intl. J. High performance Computing Applications*.
- [2] A. Vahdat, A. Lebeck and C. S. Ellis, "Every joule is precious: the case for revisiting operating system design for energy efficiency" in *EW 9: Proceedings of the 9th workshop on ACM SIGOPS European workshop*, New York, USA.
- [3] "A Feasibility analysis of power awareness in commodity based High Performance Clusters", in *7th IEEE International Conference on Cluster computing (CLUSTER'05)*, Boston, Massachusetts, Sept'2005.
- [4] J. Chase and R. Doyle, "Energy management for server clusters," in *HOTOS '01: Proceedings of the Eighth Workshop on Hot Topics in Operating Systems*. Washington, DC, USA: IEEE Computer Society, 2001, p. 165.
- [5] E. Elnozahy, M. Kistler, and R. Rajamony, *Energy-Efficient Server Clusters*, ser. *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2003, ch. *Power-Aware Computer Systems*, pp. 179–197.
- [6] V. W. Freeh, F. Pan, N. Kappiah, D. K. Lowenthal, and R. Springer, "Exploring the energy-time tradeoff in mpi programs on a power-scalable cluster," in *IPDPS '05: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)*. Washington, DC, USA: IEEE Computer Society, 2005, p. 4.1.
- [7] C. Hsu, W. Feng, and J. S. Archuleta, "Towards Efficient Supercomputing: A Quest for the Right Metric," in *1st IEEE Workshop on High-Performance, Power-Aware Computing (in conjunction with the 19th International Parallel & Distributed Processing Symposium)*, Denver, Colorado, April 2005.
- [8] S. U. Khan, "A game theoretical energy efficient resource allocation technique for large distributed computing systems," in *The 2009 International Conference on Parallel and Distributed Processing Techniques and Applications*. Las Vegas, NV, USA: CSREA Press, 2009.
- [9] B. Khargharia, S. Hariri, and M. S. Yousif, "Autonomic power and performance management for computing systems," *Cluster Computing*, vol. 11, no. 2, pp. 167–181, 2008.

Contact Information

Your comments and questions are valued and welcome. Contact the authors at:

Monika Yadav (2nd year, MTech , Department of Computer Science and Engineering, IIT Bombay)

Mumbai , Maharastra , India 400076

Phone: (+91)- 9167563942

E-mail : monika@cse.iitb.ac.in

M. R. Bhujade(Faculty at Department of Computer Science and Engineering, IIT Bombay) Mumbai

, Maharastra , India 400076

E-mail : mrb@iitb.ac.in

Sudheer Katta (3rd year , Dual Degree , Department of Electrical Engineering with specialisation in
Communication and signal processing , IIT Bombay)

Mumbai,Maharastra , India 400076

Phone: (+91)- 9004669664

E-mail: ksudheer@ee.iitb.ac.in