



Scalability of Distributed Engineering Computation over Cloud of Virtual Machines

By Han Gyoo Kim

Hongik University

Abstract - It is investigated to verify the scalability aspects of the distributed engineering computation on the cloud computing. In the study, a parallel virtual machine program distributed over a network of cloud computers is used in solving a finite difference version of a typical complicated engineering differential equation. It is found that there exist a pseudo-optimal number of virtual machines, which does not necessarily coincide with the number of tasks and the pseudo-optimal number depends on various overheads over the network of virtual machines. Increasing the number of machines in the cloud beyond certain threshold one does not improve computing performance due to the communication overhead between the task processes distributed over the network.

Keywords : *Cloud computing, networked parallel computing, task decomposition, scalability.*

GJCST-B Classification: C4



Strictly as per the compliance and regulations of:



Scalability of Distributed Engineering Computation over Cloud of Virtual Machines

Han Gyoo Kim

Abstract - It is investigated to verify the scalability aspects of the distributed engineering computation on the cloud computing. In the study, a parallel virtual machine program distributed over a network of cloud computers is used in solving a finite difference version of a typical complicated engineering differential equation. It is found that there exist a pseudo-optimal number of virtual machines, which does not necessarily coincide with the number of tasks and the pseudo-optimal number depends on various overheads over the network of virtual machines. Increasing the number of machines in the cloud beyond certain threshold one does not improve computing performance due to the communication overhead between the task processes distributed over the network.

Keywords : Cloud computing, networked parallel computing, task decomposition, scalability.

I. INTRODUCTION

With the advent of cloud computing [6], parallel processing, the method of having many small tasks solve one large problem, has been receiving attention with ever increasing demand for higher performance, lower cost and sustained productivity. In general, solving a large distributed engineering problem has been facilitated by two major techniques: massively parallel processors and network distributed computing.

Massively parallel processors [7] provide the most powerful environment under which high computational power can be attained, combining a few hundred to a few thousand CPUs connected to hundreds of gigabytes of memory. However, MPPs suffer from two drawbacks: economy and availability.

On the other hand, distributed computing over networked cloud computing environment [3, 10] is gaining its edge over MPPs because of its widespread availability of today's cloud computing with its economic advantage. Distributed computing provides well defined APIs easily applicable to various programs as well as effective compilers over a network of workstations, resulting in cost effective solutions.

Despite its widespread acceptance as one the best candidates for distributed computing paradigms, distributed virtual machines over cloud computing environment are yet to be verified whether they can offer scalable solutions to solving the distributed engineering problems in efficient and economical ways. While it is

Author : Department of Computer Engineering Hongik University. This work was supported by 2011 Hongik Research Fund.
E-mail : hgkim@hongik.ac.kr

believed that employing more virtual machines from the cloud would help to expedite the execution of distributed problems, it is not clear whether we can obtain overall performance increase proportional to the number of virtual machines employed in the cloud [2].

The objective of this study is not in any specific engineering problem itself but to perform a parametric numerical experiment on parallel virtual machines by solving typical engineering differential equations of which solution can be obtained by way of PVM over a cloud computing network.

PVM (Parallel Virtual Machine) [1,8] is one of the most promising distributed computing systems available and may be applied to a network of heterogeneous workstations such as network of cloud computing. PVM provides easy to program APIs through which complex and CPU intensive scientific problems, such as global climate modeling and new drug design, can be solved without relying on expensive MPPs by decomposing them into a set of simple tasks manageable on a network of virtual machines of a cloud system.

In this paper, the experience with PVM is presented in pursuit of finding a pseudo-optimal decomposition of PVM tasks of which set are assigned to individual virtual machines in a cloud system. It will only be a *pseudo*-optimal since various types of overhead are affected by nondeterministic factors such as actual data transfer rate and the network load at time of execution. We will investigate how total execution time of a finite difference program processed in parallel varies as number of homogeneous virtual machines on a network of a cloud system is varied along with the effects of discretization size in both space and time.

II. EXPERIMENT

Distributed computing using PVM may be approached from three fundamentally different viewpoints based on the organization of computing tasks [4].

The first and the most common model for PVM applications can be termed *crowd* computing, where a collection of loosely related processes performs computations on different portions of the workload usually involving periodic exchanges of intermediate results. The second model supported by PVM is termed a *tree* computation. In this scenario, processes are dynamically spawned in a tree-like manner. This paradigm is a natural fit to applications where the total

workload is not known a priori, for example as in recursive divide-and-conquer algorithms. The third model termed *hybrid* is a combination of the tree and the crowd model.

As mentioned above, PVM provides various methods of task decomposition that fit into modeling of diverse scientific problems. The choice of model will be application dependent and should be selected to best match the natural structure of the distributed program while taking into account the communication overhead.

For the purpose of studying the scalability of distributed virtual machine solutions, it is presented a PVM program that calculates heat diffusion in a thin wire by solving the finite difference version of a one dimensional heat equation. To accomplish the aim, as mentioned in the introduction, it suffices to choose to analyze a most simple problem preferably to which an analytic solution is also available.

Consider a thin wire of length L , density ρ , specific heat c and thermal conductivity k with the ends of the wire maintained at a fixed temperature of T_e and an initial temperature profile of;

$$T(z, \tau = 0) = T_e + T_0 \sin(\pi \frac{z}{L}) \quad (1)$$

With the conventional non-dimensionalization, where, $A = (T - T_e)/T_0$, and $t = \tau L (\frac{\rho c}{k})$, and $x = z/L$, the temperature in the wire is described by the following heat equation;

$$\partial^2 A / \partial x^2 = \partial A / \partial t \quad (2)$$

With the initial and boundary conditions of;

$$A(x, t = 0) = \sin(\pi x) \quad (3a)$$

$$A(x = 0, t) = A(x = 1, t) = 0 \quad (3b)$$

The exact solution of Eq. (2) subject to (3a) and (3b) can be found by the method of separation of variables [5] as;

$$A(x, t) = e^{-\pi^2 t} \sin(\pi x) \quad (4)$$

Finite-difference solution to the above problem will be sought via parallel processing and in particular through distributed computing using the PVM program. We will adopt an explicit scheme with forward differencing in time and central differencing in space, and hence the solution can be obtained by marching in time from a given initial temperature distribution.

If we denote $A(x_i, t_j)$ as $A_{i,j}$, temperature at position x_i and at time t_{j+1} can be expressed as;

$$A_{i,j+1} = \beta(A_{i+1,j} + A_{i-1,j}) + (1-2\beta)A_{i,j} \quad (5)$$

where $\beta = \Delta t / (\Delta x)^2$ and the stability criterion for the explicit scheme requires that $\beta \leq 1/2$.

For our problem, where the solution over the whole space can be obtained through the same equation as provided by Eq. (5), natural choice of

programming structure would be the master-slave method of the crowd programming paradigm where the slaves, spawned by the master program, perform the actual computations. We choose to divide the wire into 50 subsections and the solutions to each subsection will be obtained separately by each slave programs, although it will be required that the right most and the left most temperature information be exchanged with its right and left neighboring subsections. The workload of the slaves is allocated by the master through data decomposition whereby the initial temperature distribution of each subsection is sent to respective slaves.

Overall structure of the parallel processing adopted in this paper is as follows:

The master program spawns 50 copies of the same slave program, each of which handles a subsection of the wire. After receiving initial temperature distribution, each slave computes heat diffusion in the corresponding wire subsection. At each time step, each slave program needs to communicate boundary information with its left and right neighboring slaves. When a specified final time is reached, all 50 slave programs send its final temperature profile to the master, who then terminates the spawned slaves and ends the program.

In order to study the scalability aspects, same program was executed on 1 to 60 separate processes on the virtual machines available over the network of 1Gbps Ethernet. When more than one virtual machine is utilized, the master and the slave programs are allocated to each machine such that an even distribution of workload is achieved.

The finite difference PVM program was executed on six different configurations with four values of Δx ranging from 1/50 to 1/1000 and four Δt 's for each Δx ranging from 2.0×10^{-4} to 6.25×10^{-8} until a preset final time is reached. These various processing configurations were set to investigate to find the pseudo-optimal number of virtual machines that preserve the scalability of the whole system.

The final time was chosen such that for each different Δx , the largest Δt results in 750 iterations and the smallest 6000. Hence for a given Δx , total number of iterations on Eq. 5 is inversely proportional to the size of Δt . The final temperature profile for all of the above parameter values resulted in essentially the analytic solution of Eq. (4). The total computing time elapsed in carrying out this numerical experiment on a network of virtual machines were recorded and are analyzed from the perspective of whether the cloud computing environment could effectively provide scalable solutions to such highly distributed problems.

III. RESULTS AND DISCUSSIONS

Results analyzed in this section were obtained by running a group of PVM processes on a set of virtual machines running on VMware [9]. Total computing time, including data communication overhead between machines, will definitely depend on the degree of network traffic at the time of execution. Results used in the analysis below are only one out of many trials executed under similar working environment.

a) Data Communication Overhead

To investigate the effect of data transfer over the network on the total computing time, time taken solely in data transfer between machines were investigated and the results are presented in Fig. 1.

This includes time taken for PVM setup plus network latency and data transmission over the network of virtual machines. It can be seen that the increase in the amount of data will not necessarily result in a larger communication overhead but will be strongly dependent on the number of machines on the network.

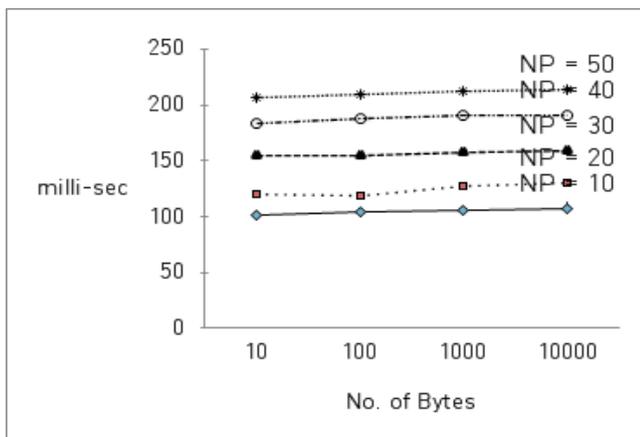


Fig. 1: Communication time for data transfer

However the results of Fig. 1 clearly show that data communication time is bounded. One can also deduce that the PVM setup time is about 100 ms for our experimental environment and the data transfer time exhibits some randomness reflecting the characteristics of some other uncontrollable glitches in the network over the execution of repeated computation.

The behavioral aspects of communication overhead are inferred to be strongly dependent on the characteristics of how much distributed the particular problem and how much data are to be transferred between the machines over the network.

However, what is important from the experimental observation is that the communication overhead does not diverge unboundedly, but is rather bounded once the distributed solution is given.

This observation justifies that virtual machines in a cloud computing environment can definitely provide scalable computational means to a certain types of

distributed problems such as complicated engineering differential equations.

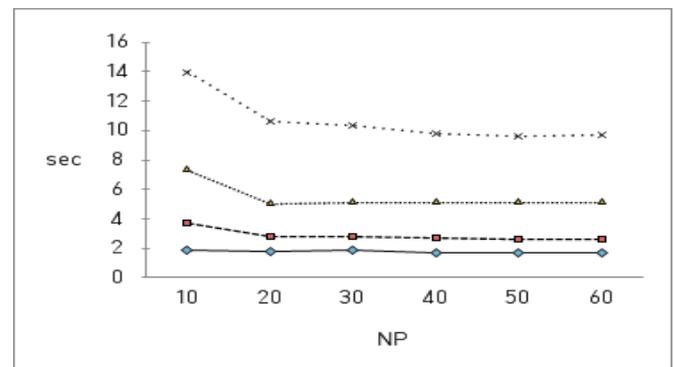
b) Total Computation Time

Figs 2 below describes the total computation time versus the number of machines utilized by the PVM program.

The most important result is that for all cases, it seems to exist a pseudo-optimal configuration. This pseudo-optimum becomes more pronounced as computation load increases (see Fig. 2d). This can be explained by the fact that total computation time is comprised of actual CPU time, which decreases as more virtual machines are utilized and workload is more evenly distributed, and communication overhead, which increases with NP as more data transfers are required between more machines. Hence total computation time is the least when gains obtained from work distribution minus the increase in communication overhead is the largest.

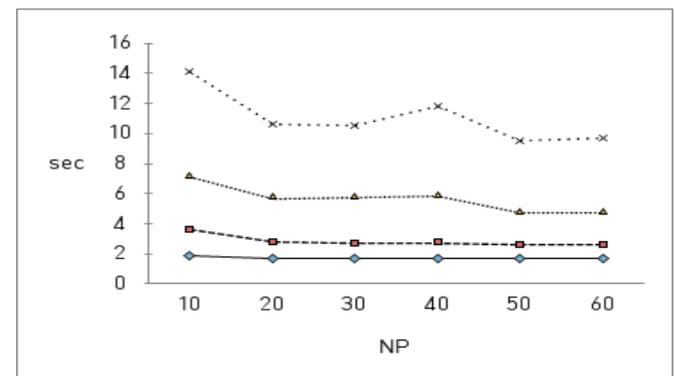
(a) $\Delta x = 0.02$

$\Delta t = 2.0 \times 10^{-4}, 1.0 \times 10^{-4}, 5.0 \times 10^{-5}, 2.5 \times 10^{-5}$ respectively



(b) $\Delta x = 0.01$

$\Delta t = 5.0 \times 10^{-5}, 2.5 \times 10^{-5}, 1.25 \times 10^{-5}, 6.25 \times 10^{-6}$ respectively



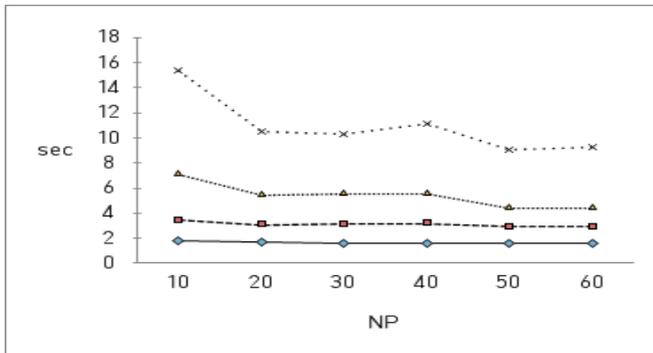
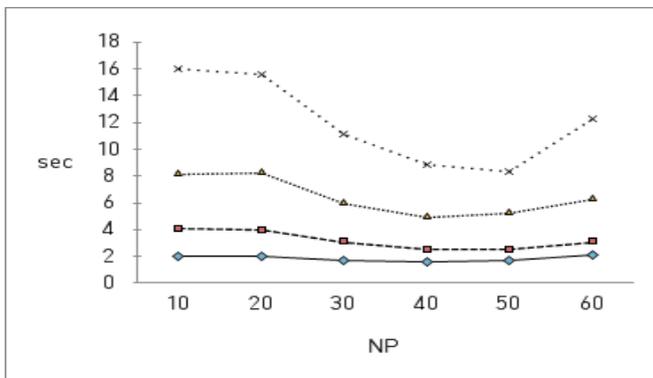
(c) $\Delta x = 0.005$ $\Delta t = 2.0 \times 10^{-6}, 1.0 \times 10^{-6}, 0.5 \times 10^{-7}, 2.5 \times 10^{-7}$ respectively(d) $\Delta x = 0.001$ $\Delta t = 5.0 \times 10^{-7}, 2.5 \times 10^{-7}, 1.25 \times 10^{-7}, 6.25 \times 10^{-8}$ respectively

Fig. 2 : Total computing time, T , as a function of number of virtual machines, NP

For our problem it can be concluded that the optimum configuration is obtained when 45 to 55 machines are utilized. That is, about 50 virtual machines are shown to be pseudo-optimal number of machines that our cloud system can provide virtual machines to our specific problem in scalable way, i.e., the total computing time is basically proportionally decreased as the number of virtual machines is increased to about 50. It should, however, be noticed that these pseudo-optimal numbers of scalable virtual machines that deduced from our experiments should be different for other types of problems as other types of problems should demand different amount of resources such as CPU time and the amount of data communicated among the virtual machines.

It should also be remembered that this analysis is based on the fact that the programs were executed where data transfer time is almost constant. In addition, we can conclude that our problem is such that communication overhead is of comparable order to actual CPU time, especially when computation burden is not low (e.g. for small Δx) and hence different types of configuration do play an important role regarding total computation time.

The observations from the experiments fortifies the common belief that scalability is more prominent in a network of virtual machines where most of the computational powers of the machines are required to solve the problem than in a network where less powers are required.

IV. CONCLUSION

In this paper, a numerical computational experiment were performed over a virtual machine cluster connected by 1Gbps Ethernet using PVM to solve a one dimensional differential equation with the purpose of investigating how scalable the network of virtual machines are.

When the data transfer time is relatively constant, there exists a pseudo-optimal task decomposition set by decrease in CPU time of individual virtual machines and increase in communication overhead as more machines are utilized thus preserving the scalability of the system.

However, when the traffic is heavy with data transfer being more random, total computation time shows a certain degree of random ness with less systematic improvement as workload is distributed among more machines.

However, it is clear that in most of the cases studied, distributed computation over the cloud of virtual machines is no worse than serial computation in the worst case and shows highly scalable improvement over serial computation in many other cases.

REFERENCES RÉFÉRENCES REFERENCIAS

1. Geist, A et al., *PVM: Parallel Virtual Machine - A User's Guide and Tutorial for Networked Parallel Computing*, MIT Press, Cambridge, MA., 1994.
2. Hayes, B., "Cloud Computing," *Communications of the ACM*, Vol. 51, No.7, July 2008, pp.9-11.
3. Hwang, K., Fox, G. C., and Dongarra, J., *Distributed and Cloud Computing 1st ed.*, Morgan Kaufman, October 2011.
4. Kim, H. G., Seong, K. J., and Kim, S. H., "A Numerical on Network Parallel Computing Using PVM," *Proc. Of KISS*, Vol. 23, No. 2, 1996, pp. 1031-1034.
5. Meyers, G. E., 1971, *Analytical Methods in Conduction Heat Transfer*, McGraw-Hill, New York, pp.271-274.
6. Miller, M. *Cloud Computing*, QUE, 2008.
7. Noda, H. et al, "The Design and Implementation of the Massively Parallel Processors based on the Matrix Architecture," *IEEE Journal of Solid State Circuits*, Vol. 42, Issue 1, 2007, pp.183-192.
8. www.csm.ornl.gov/pvm, 2012.
9. www.vmware.com, 2012

10. Zhang, Q., and Boutaba, R., "Cloud Computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, Springer, Vol. 1, No. 1, 2010, pp.7-18.

