Distributed Shared Memory Programming in the Cloud

Ahmad Anbar, Vikram K. Narayana and Tarek El-Ghazawi
NSF Center for High-Performance Reconfigurable Computing (CHREC)
Department of Electrical and Computer Engineering
The George Washington University
{*anbar, vikram, tarek*}*@gwu.edu*

***Abstract*—Cloud computing is beginning to play a dominant role in scientific computing. However, there are still several challenges that need to be addressed, before data-intensive scientific applications make the transition to the cloud. Adoption of a distributed shared memory (DSM) programming paradigm will be one approach to ease the transition, through the use of Partitioned Global Address Space (PGAS) languages. This paper explores initial results from the adoption of a PGAS language, Unified Parallel C, in programming a representative private cloud based on Eucalyptus.**

***Keywords*-Parallel Programming; Cloud Computing; Distributed Shared Memory; PGAS; HPC; UPC**

I. Introduction

Cloud computing involves the use of the internet for providing on-demand access to shared resources, such as computational capabilities, data, storage and applications. Resources may be offered to users as services at different levels - software as a service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS) [1]. These levels respectively offer decreasing levels of abstraction, with IaaS allowing the greatest level of control over the operating system, storage and applications, and therefore best suited for custom scientific application development.

High-performance computing (HPC), based on parallel-programming paradigms such as message-passing or shared memory, has traditionally focused on using dedicated HPC platforms such as compute clusters or supercomputers. These platforms reside either within the organization, or are present in national labs and remotely accessed on a time-shared basis. Both of these solutions have limitations, such as high acquisition and maintenance costs (for in-house HPC hardware) and large turnaround times (for time-shared HPC resources). The use of cloud as a shared resource provides a sweet spot between these two options. Moreover, isolation between users through virtualization, and ability to deploy custom application environments through IaaS, makes cloud computing an attractive option for HPC. This is evidenced by Amazon recently opening up the "Cluster Compute Instance" in their cloud computing service [2].

This work was supported in part by the I/UCRC Program of the National Science Foundation under Grant No. IIP-0706352.

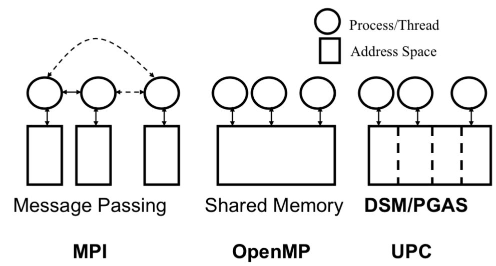

Figure 1. Parallel Programming Models in HPC

II. Motivation

In a typical usage scenario of HPC in the cloud, a virtual cluster is allocated to a user for developing and deploying parallel programs. Parallel programming on the virtual cluster exhibits the same programming complexity as traditional HPC. When we also consider the virtualization overheads, the cloud can become a rather unattractive solution for scientific computing if we do not address the programming issues. For instance, the widely used MPI [3] approach requires explicit two-sided communication between processes to exchange data, due to a distributed view of the memory address space (Figure 1). Programming efforts may be eased by using a shared-memory programming approach such as OpenMP [4], through a unified view of the address-space across processes. However, a shared memory view does not provide any distinction between local and remote data; this can cause performance degradation in a cloud when individual nodes could possibly be allocated in physically distant nodes. We believe that the distributed shared-memory (DSM) programming paradigm, embodied by Partitioned Global Address Space (PGAS) languages such as UPC (Unified-Parallel C, UPC [5]), is better suited for the cloud. DSM provides ease-of-programming due to one-sided communication afforded by the shared memory view, while enabling locality-awareness for increased performance. Based on this observation, we have initiated our explorations with UPC benchmarks deployed on a simple, private cloud configured in-house, described as follows.

Table I
CHARACTERISTICS OF THE TESTBED

	Compute Node	VM Instance
OS	Ubuntu Linux 10.04 LTS	
Processor	Intel Nehalem E5620 @ 2.4 GHz	
CPU cores	4	4
Memory	12 GB	2 GB
Storage	500 GB	20 GB

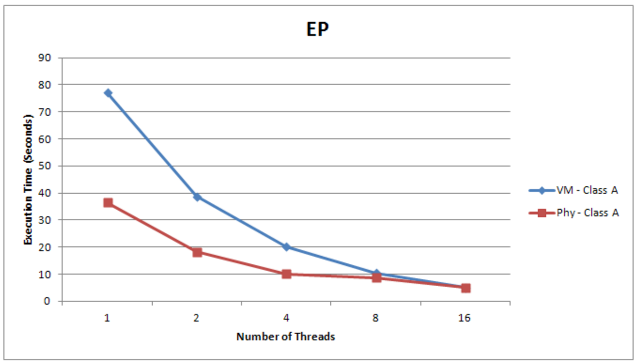

Figure 2. Results for NAS Parallel Benchmark: EP

III. TESTBED

We built a private cloud based on the open-source Eucalyptus [6] software platform available as part of the Ubuntu Linux operating system. Our initial prototype platform consists of 4 compute nodes, and a controller node that hosts the management components of Eucalyptus. Instances of KVM-based [7] virtual machines (VM) are launched on the compute nodes, based on user requests made to the Eucalyptus management system. The characteristics of the compute nodes and the VMs are summarized in Table I. We created a virtual cluster based on four instances of the VM launched on the compute nodes. For UPC programming, we use Berkeley UPC compiler version 2.14.0 [8].

IV. PRELIMINARY EXPERIMENTAL RESULTS

Our experiments are based on UPC versions of the NAS Parallel Benchmarks developed by us [9]. Figures 2 and 3 show the results for the EP and MG benchmarks respectively, comparing the execution within the cloud-based virtual cluster against native execution on the physical compute nodes. Threads were assigned to a node up to a maximum of four threads per node, before using a new compute node. As expected, we observed that execution on the cloud incurs some overheads, although it reduces substantially for 16 threads of the computationally intensive benchmark EP. For the MG benchmark, overheads were significantly larger with more nodes, due to greater communication among UPC threads, and also possibly due to UDP-based inter-node communication used in our UPC configuration.

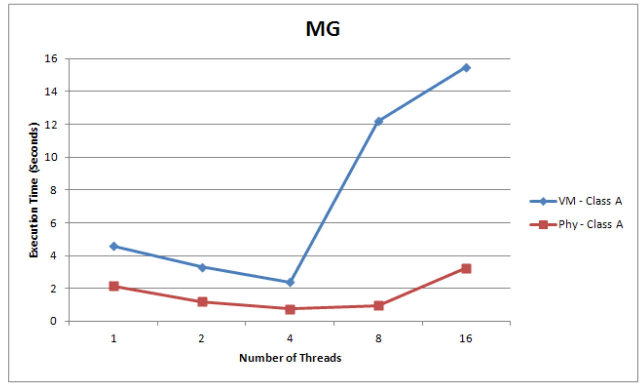

Figure 3. Results for NAS Parallel Benchmark: MG

V. CONCLUSION

Distributed shared memory (DSM) programming paradigm is a promising approach for HPC in the cloud. Further explorations are required in our experimental setup to obtain optimum configurations of UPC and the Eucalyptus-based platform.

REFERENCES

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," *NIST special publication 800-145*, 2011.

[2] Amazon. (2011, Nov. 15) *Announcement: Announcing a new Amazon EC2 Cluster Compute Instance*. [Online]. Available: https://forums.aws.amazon.com/ann.jspa?annID=1247

[3] M. P. I. Forum, "MPI: A Message-Passing Interface standard," *International Journal of Supercomputer Applications*, vol. 8, no. 3/4, pp. 165–414, 1994.

[4] L. Dagum and R. Menon, "OpenMP: an industry standard API for shared-memory programming," *Computational Science & Engineering, IEEE*, vol. 5, no. 1, pp. 46–55, 1998.

[5] T. El-Ghazawi, W. Carlson, T. Sterling, and K. Yelick, *UPC: Distributed shared memory programming*. John Wiley & Sons Inc., New York, May 2005.

[6] D. Nurmi, R. Wolski, C. Grzegorczyk, G. Obertelli, S. Soman, L. Youseff, and D. Zagorodnov, "The Eucalyptus Open-Source Cloud-Computing System," in *9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID)*, May 2009, pp. 124 –131.

[7] A. Kivity, Y. Kamay, D. Laor, U. Lublin, and A. Liguori, "KVM: The Linux Virtual Machine Monitor," in *Proceedings of the Linux Symposium*, vol. 1, 2007, pp. 225–230.

[8] *Berkeley UPC Compiler*. [Online]. Available: http://upc.lbl.gov

[9] The George Washington University High Performance Computing Laboratory. *UPC NAS Parallel Benchmarks*. [Online]. Available: http://threads.hpcl.gwu.edu/sites/npb-upc