# IMPROVING RESOURCE USAGE IN HPC CLOUDS

**V. Antonenko [1, a], A. Chupakhin [1, b], I. Petrov [1, c], R. Smeliansky [1, d]**

[1] *Lomonosov Moscow State University, 1 Leninskiye Gory, Moscow, 119991, Russia*

E-mail: [a] anvial@lvk.cs.msu.ru, [b] andrewchup@lvk.cs.msu.ru, [c] ipetrov@cs.msu.ru, [d] smel@cs.msu.ru

Nowadays many supercomputer users are dissatisfied with a long waiting time for their jobs in the supercomputer queue. Therefore, to reduce the queue of jobs to the supercomputer, we suggest use cloud resources (HPC-as-a-service). Our main goal is to decrease wait time plus execution time for jobs in supercomputer.

One of the key drawbacks associated with HPC-clouds is low CPU usage due to the network communication overhead. Instances of HPC applications may reside on different physical machines separated by significant network latencies and network communications may consume significant time and thus result in CPU stalls.

In this paper we present and check hypothesis: "MPI programs that don't require a lot of computing resources can effectively share the same set of resources". It's possible when network in the cloud is slow or MPI programs can intensively use the network resources and not intensively use computational resources. Thus, such programs can run simultaneously without significant slowdown, because when one program is waiting to receive data over the network, CPU stalls and can execute another program.

We checked our hypothesis on popular MPI benchmarks – NAS Parallel Benchmarks (NPB). The experiments have shown that we can improve the CPU usage in the cloud with negligible performance degradation of HPC-applications execution (in terms of time spent).

Keywords: High Performance Computing, Cloud, HPC-as-a-service, Message Passing Interface

Vitaly Antonenko, Andrey Chupakhin, Ivan Petrov, Ruslan Smeliansky

# 1. Introduction

During the past decade public clouds have attracted tremendous amount of interest from academic and industrial audiences as an effective and relatively cheap way to get powerful computing infrastructure for solving a lot of problems in different areas. One such area is High Performance Computing (HPC). Even though clouds are less powerful than server clusters or supercomputers [1], they are becoming more popular as a platform for HPC due to the low cost and easy to access. Several papers [2, 3] have shown that one of the main performance bottlenecks in HPC-clouds issues from communication delays within the DC (data center) network. While supercomputers use fast interconnections like InfiniBand or GE (gigabit ethernet) [4, 5], HPC-clouds mostly rely on slow Ethernet network. This performance bottleneck could also lead to CPU underutilization with network-intensive applications, since such applications may spend a lot of time waiting for their messages to pass through the network. In this paper we analyze how network communication overhead affects the CPU utilization in HPC-clouds. We also present and check the following hypothesis applied to HPC-clouds: network-intensive HPC-applications could share CPU cores among each other with negligible performance degradation. Such behaviour could be used to improve CPU utilization and to increase the effectiveness of HPC-application execution. The hypothesis was checked in a cloud environment using popular HPC benchmark – NPB [6]. The paper is organized as follows. Section 2 presents *Related Work*. Section 3 contains *Problem description*. Section 4 presents the *Experiments*. Section 5 contains *Conclusions and Future Work*.

# 2. Related Work

Authors in [2] used CloudSim [7] to analyze the possibility of running HPC-applications in the cloud. They improved performance of HPC-clouds by adjusting cloud virtualization mechanisms and HPC-application's settings. The authors have also shown that some HPC-applications underutilize CPU for almost half the time in HPC-clouds. The paper [3] shows that cloud network creates a significant bottleneck due for HPC-applications due to low communications speeds and large delays. The authors show that cloud can be used for a subset of HPC-applications, specifically low communication-intensive applications with high CPU count and communication-intensive applications with low CPU count. According to the article [8] about half of the MPI jobs in supercomputers use less than 120 cores. It's very important because it's not a very large value for modern clouds even with that now the idea of micro DC is gaining popularity [9].

# 3. Problem Description

The current situation with supercomputers is as follows:

- Low user experience when working with supercomputers due to the fact that users often wait for a long time until their jobs start to execute;
- Scheduler in supercomputer allocates entire computing node with multiple CPUs and cores, rather than individual cores. At the same time on each core can be executed only one MPI process at one time;
- Due to the allocation of entire compute nodes, as well as badly written MPI programs, there is resources fragmentation leads to resource underutilization.

Our main goal is to reduce (wait time + execution time) for jobs in supercomputer queue. One possible solution to fix the problem of a large wait time is to use additional resources. We suggest to use additional cloud resources. If you have additional cloud resources you can send some jobs for execution to the supercomputer and some to the cloud. But you need to send to the cloud programs of a certain type. We assume that these are programs that have good ability of sharing resources with other programs. We investigated this problem in this article.

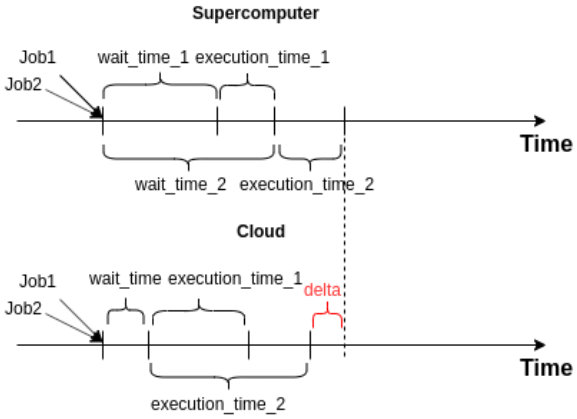



Figure 1. Supercomputer and cloud perform MPI jobs in different ways

In our work we check the following hypothesis: "MPI programs that don't require a lot of computing resources can effectively share the same set of resources".

In the Fig. 1 our hypothesis is demonstrated. In the supercomputer jobs are often executed sequentially and because of this they have a large wait time. It is important to understand that execution time of MPI programs in supercomputer is less than in the cloud. Additional cloud resources could help reduce wait time for MPI jobs in supercomputer's queue. Also sharing the same cloud resources between MPI programs could help reduce wait time even more and at the same time sharing resources could allow to keep execution time in the cloud not very big compared to execution time in the supercomputer. Thus, a couple of jobs in the cloud can have (wait time + execution time) less than in the supercomputer, see Fig. 1.

We conducted some experiments to check our hypothesis. We checked our hypothesis on MPI programs from NPB because they are very similar to the real MPI programs. NPB consists of programs with different nature and different resource usages [6]. We use the following tasks: CG – Conjugate Gradient; EP – Embarrassingly Parallel; FT – discrete 3D fast Fourier Transform; IS – Integer Sort; LU – Lower-Upper Gauss-Seidel solver.

# 4. Experiments

This section presents an experimental evaluation of network influence on CPU utilization in the clouds and evaluation of resources sharing ability for MPI programs.

## 4.1. Testbed

All experiments were performed on a single rack consisted of 7 heterogeneous physical servers all connected to a single switch (*star* topology) with optical fibres. The specification of servers: head server – Intel Xeon CPU E5-2650 v4 @ 2.20GHz with 48 cores with 64 Gb RAM and 6 workers – Intel Xeon CPU E5-2667 v4 @ 3.20GHz with 16 cores with 32 Gb RAM. Each physical link had the maximum bandwidth equal to 10 Gbits/sec.

## 4.2. Methodology

We have created using QEMU/KVM hypervisor 64 virtual machines (VMs) (Ubuntu 16.04, 1 vCPU, 1024 Mb RAM). MPI version was 3.2. Head server contained 16 VMs, other servers contained 8 VMs per each. Average RTT between different VMs was near 400 μs. Bandwidth between VMs at the same server – 18.2 Gbits/sec, on different servers – 5.86 Gbits/sec.

During the experiments we measured characteristics of MPI programs: CPU with perf linux utility and network usage with netstat linux utility. Also we configured bandwidth and delay on the interfaces in each VM using traffic control utility. When we launched MPI programs each MPI process was running on a separate VM. NPB programs has different sizes, we use size B.

### 4.1. Experimental Results

#### *4.1.1. CPU Utilization*

In this experiment we have checked how network bandwidth influences the CPU utilization. We launched sequentially 5 NPB MPI programs with 2, 4, 8, 16, 32, 64 MPI processes, each process on separate VM. In this experiment we considered three bandwidth speed: 100 Mbits/sec, 1000 Mbits/sec, 10000 Mbits/sec. In Fig. 4 you can see that for MPI programs from NPB when the number of MPI processes increases, CPU usage drops, because different MPI processes run on different virtual machines and data is transferred over the network between the different physical servers and so the delay increases. Also CPU usage drops when MPI program run in one physical servers (2, 4 and 8 CPU number). This CPU usage decrease allows share the same CPU between different MPI programs.

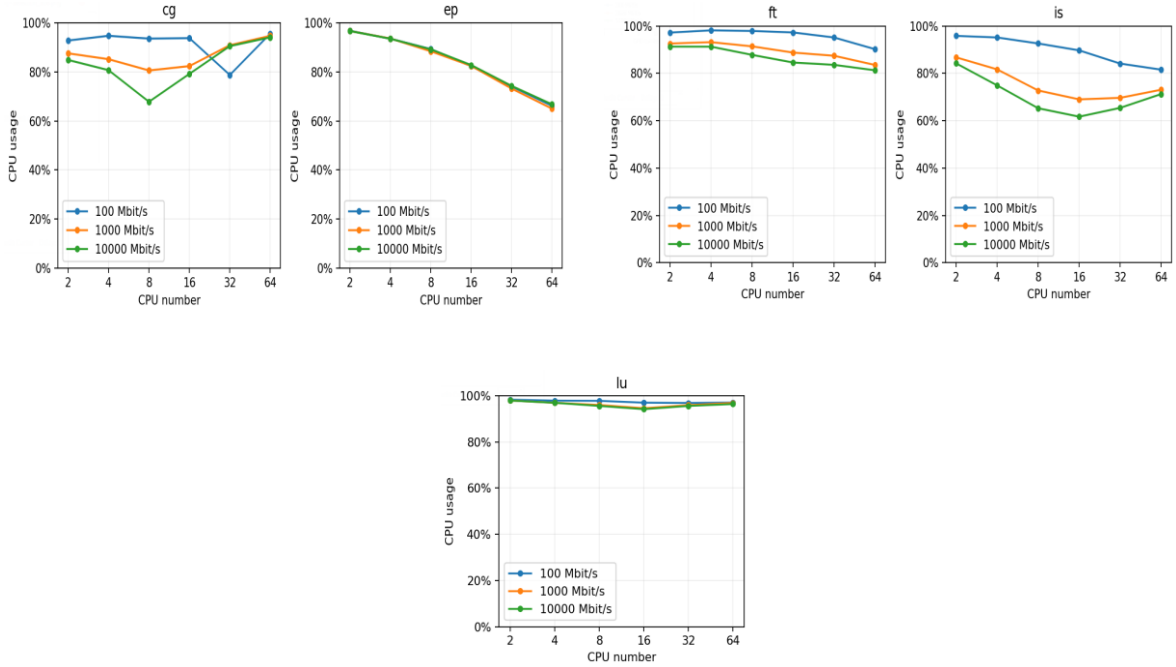
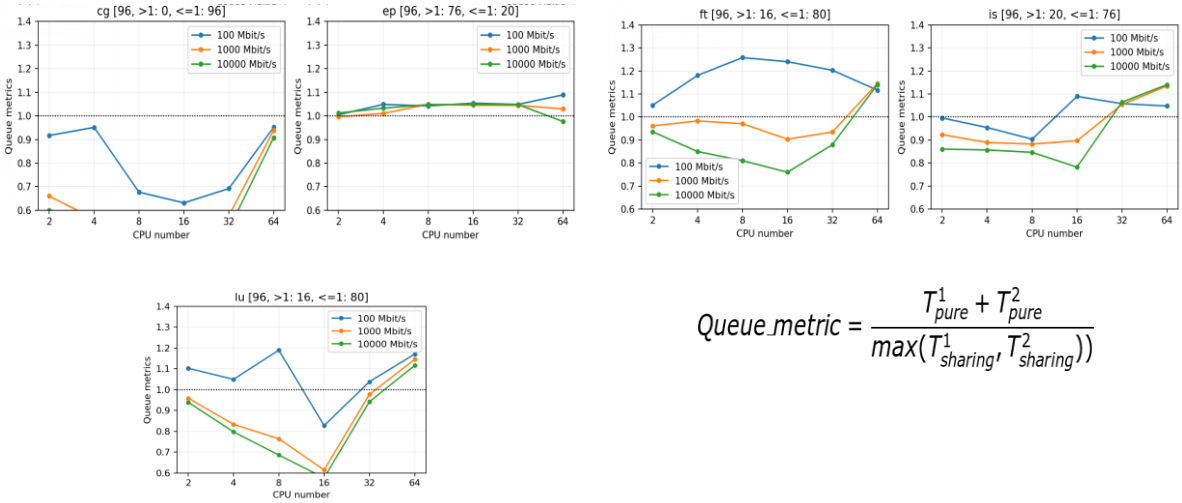



Figure 4. CPU utilization for NPB

#### *4.1.2. Core Sharing*




$$Queue\_metric = \frac{T^1_{pure} + T^2_{pure}}{max(T^1_{sharing}, T^2_{sharing}))}$$

Figure 5. Queue metric

In this experiment we investigated the ability to share CPU cores between different HPC-applications, see Fig. 5. The experiment was performed as follows. We launched sequentially 5 pair of

NPB MPI programs (each pair contained two identical programs) on *N* VMs (2, 4, 8, 16, 32, 64) (*N* MPI processes from one MPI program and *N* MPI processes from another MPI program). To understand how well MPI programs can be shared, we calculated the *queue metric*, see Fig. 5, where *pure time* is execution time without resources sharing, *sharing time* is execution time when two MPI programs use the same CPUs and cores. If value of *queue metrics* is more than *1* therefore two programs run simultaneously take less time to complete than in sequential order. According to the Fig. 5 in the cloud with slow network (100 Mbits/sec) we can get up to 20 percent execution time acceleration. Also you can see that not all MPI programs can effectively share resources with other MPI programs.

## 5. Conclusions and Future Work

In this research we present the experiments which show that MPI programs can utilize not all provided CPU resources in the cloud with slow network and thus underutilized resources could be used to implement other MPI programs. Second experiment shows that we can get up to 20 percent execution time acceleration when we run in the cloud two MPI programs simultaneously in contrast of sequential run. Such behaviour could be used to improve CPU utilization and to increase the effectiveness of HPC-application execution.

Our further research – develop scheduler for the cloud which can share resources according special metrics for MPI programs – also our further research will be related to the problem of prediction the execution time of MPI programs on a supercomputer. Predicted time can help us to understand where to send task: to the supercomputer or to the cloud.

## Acknowledgement

This work is supported by Russian Ministry of Science and Higher Education, grant #05.613.21.0088, unique ID RFMEFI61318X0088.

## References

[1] Netto, M. A., Calheiros, R. N., Rodrigues, E. R., Cunha, R. L., & Buyya, R. (2018). HPC cloud for scientific and business applications: Taxonomy, vision, and research challenges. ACM Computing Surveys (CSUR), 51(1), 8.

[2] Gupta, A., Faraboschi, P., Gioachin, F., Kale, L. V., Kaufmann, R., Lee, B. S., ... & Suen, C. H. (2016). Evaluating and improving the performance and scheduling of HPC applications in cloud. IEEE Transactions on Cloud Computing, 4(3), 307-321.

[3] Gupta, A., & Milojicic, D. (2011, October). Evaluation of hpc applications on cloud. In 2011 Sixth Open Cirrus Summit (pp. 22-26). IEEE.

[4] Infiniband in supercomputer systems. https://www.businesswire.com/news/home/20181112005379/en/Mellanox-InfiniBand-Ethernet-Solutions-Accelerate-Majority-TOP500

[5] Gigabit Ethernet in supercomputer systems. https://www.mellanox.com/solutions/high-performance-computing/top500.php

[6] NAS Parallel Benchmarks. https://www.nas.nasa.gov/publications/npb.html

[7] Goyal, T., Singh, A. and Agrawal, A. (2012). Cloudsim: simulator for cloud computing infrastructure and modeling. Procedia Engineering, 38, pp.3566-3572.

[8] A. Prabhakaran and L. J., "Cost-Benefit Analysis of Public Clouds for Offloading In-House HPC Jobs," *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, 2018, pp. 57-64.

[9] Blesson Varghese, Rajkumar Buyya, "Next generation cloud computing: New trends and research directions", Future Generation Computer Systems, Volume 79, Part 3, 2018, Pages 849-861, ISSN 0167-739X