Long-Term Performance Evaluation of Hadoop Jobs in Public and Community Clouds

Kento AIDA†a), Omar ABDUL-RAHMAN†, Eisaku SAKANE†, and Kazutaka MOTOYAMA†, Members

SUMMARY Cloud computing is a widely used computing platform in business and academic communities. Performance is an important issue when a user runs an application in the cloud. The user may want to estimate the application-execution time beforehand to guarantee the application performance or to choose the most suitable cloud. Moreover, the cloud system architect and the designer need to understand the application performance characteristics, such as the scalability or the utilization of cloud platforms, to improve performance. However, because the application performance in clouds sometime fluctuates, estimation of the application performance is difficult. In this paper, we discuss the performance fluctuation of Hadoop jobs in both a public cloud and a community cloud for one to three months. The experimental results indicate phenomena that we cannot see without long-term experiments and phenomena inherent in Hadoop. The results suggest better ways to estimate Hadoop application performances in clouds. For example, we should be aware of application characteristics (CPU intensive or communication intensive), datacenter characteristics (busy or not), and time frame (time of day and day of the week) to estimate the performance fluctuation due to workload congestion in cloud platforms. Furthermore, we should be aware of performance degradation due to task re-execution in Hadoop applications.

key words: Cloud computing, performance evaluation, Hadoop

1. Introduction

Cloud computing is a widely used computing platform in business and academic communities. In particular, computing platforms in the business community are changing from traditional on-premises computing servers to cloud computing services, such as Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). Cloud services provide computing and storage resources on demand. The user can access resources operated in a datacenter via the Internet whenever the user requires additional resource capacity. Resources in clouds also sustain the performance of services running with the user’s local computing resources. The user can use the resources in clouds as backup or additional computing power when the user’s local resources are inadequate due to system faults or overloads.

Performance is an important issue when the user runs an application in the cloud. The user may want to estimate the application-execution time before running the application to guarantee the application performance. Performance estimation is also needed when the user (or resource broker—ing software) chooses the most suitable cloud to run the application in multiple cloud platforms. Moreover, the cloud system architect and the designer need to understand the application performance characteristics, such as the scalability or the utilization of cloud platforms, to improve performance. However, the application performance in clouds sometimes fluctuates. That is, the execution time of the same application can vary for different application runs, and this fluctuation complicates estimation of the application performance. Thus, further analysis of fluctuations is needed to improve the reliability of performance estimation in clouds.

The performance fluctuation is caused primarily by the virtualization factor and the software factor. A typical datacenter offering cloud services operates computing resources by using virtualization technology [1]. In other words, an application runs on virtual machines (VMs) running in physical computing servers. Virtual machines allocated to different applications may share resources, such as CPU, memory, and I/O devices, and the interference between VMs sharing resources could vary the performance of applications running on the VMs [2]–[4]. The software factor is related to the task behavior in an application running on the software platform. For example, MapReduce jobs often exhibit unpredictable performance for irregular tasks with long execution times [5].

The performance of applications running in clouds has been investigated in a number of studies [6]–[10]. These studies have investigated the performance by using benchmark programs and scientific application programs, and have discussed the capability of existing clouds to run scientific programs. Large performance fluctuations have been reported in public clouds [11]–[14]. However, the discussions are limited to the performance of micro benchmarks and VM startup times. To the best of our knowledge, fluctuation of the application-execution time in clouds through long-term experiments has not yet been examined.

In this paper, we present the results of experiments to evaluate the fluctuation of application-execution time in public and community clouds. We periodically ran applications in a public cloud and a community cloud for one to three months to investigate the fluctuation of the application-execution time. We used the Amazon Elastic Compute Cloud (EC2) [15] as the public cloud and the Hokkaido University Academic Cloud as the community cloud. We ran Hadoop benchmark programs to investigate the application-execution time. Hadoop [16] is an open-source software
framework for distributed computing and is widely used to develop large-scale data processing applications, e.g., big data applications, in clouds. However, fluctuation of the application-execution time often occurs [5]. The application performance fluctuates due to workload congestion, or the interference between VMs in clouds. Thus, we expect some fluctuation of the application performance in clouds depending on the time frame, e.g., time of day and day of the week, as observed in most Internet traffic patterns [17].

The contribution of this paper is to present the quantitative results of long-term experiments to investigate the performance fluctuation of Hadoop applications in clouds. The results indicate phenomena that we cannot see without long-term experiments and phenomena inherent in Hadoop. The obtained experimental results reveal the following.

- Contrary to previous studies using micro benchmarks, the fluctuation of the execution time among Hadoop applications is small and comparable to the performance fluctuation in the local PC cluster.
- The communication performance is affected by the workload congestion in cloud platforms, whereas the CPU performance is stable.
- The performance dependence on the time frame, the time of day and the day of the week is observed in an application with large communication in a busy data-center.
- Irregular jobs with a significantly long execution time are observed. The primary reasons for the performance degradation include not only workload congestion in the cloud platform but also task re-execution in Hadoop.

The above results suggest better way to estimate the Hadoop application performance in clouds. That is, we need to be aware of application characteristics (CPU intensive or communication intensive), datacenter characteristics (busy or not), and time frame (time of day and day of the week) to estimate the performance fluctuation due to workload congestion in cloud platforms. Furthermore, we need to be aware of the performance degradation due to task re-execution in Hadoop applications.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 describes the experimental setting and the benchmarks used in the experiments. Section 4 presents the experimental results. Finally, Sect. 5 summarizes the contributions of this paper and outlines future work.

2. Related Work

The performance of applications running in clouds has been investigated in numerous studies [6]–[10]. The performance of a scientific workflow application in Amazon EC2 has been investigated with a focus on the execution time and the cost [6]. Performance evaluations using micro-benchmark programs and scientific benchmarks in Amazon EC2 have also been reported in [9]. The performance of scientific applications have been compared for Amazon EC2 and local HPC clusters [7], [10]. The end-to-end networking performance in Amazon EC2 has been investigated, and reasons for the unstable network performance have also been discussed [8].

Further studies on performance interference between applications running on VMs in a physical computing server have been conducted, and experiments using benchmarks of Unix commands, compilation processes, Pov-Ray applications, micro-benchmark programs, and web server applications have been conducted [2], [3]. Approaches to create a performance model of application programs running on VMs in a physical computing server have also been reported, and experiments have been conducted using the vConsolidate benchmark, and a performance model has been discussed [18]. The performance interference between HPC applications has been investigated by the NAS Parallel Benchmarks [4].

Hadoop [16] is an open-source software framework for distributed computing that is widely used for large-scale distributed data processing, e.g., big data applications. Files (or data) processed in Hadoop are stored in the Hadoop distributed file system (HDFS). A file is decomposed into data blocks, which are stored in multiple servers, DataNodes, in a distributed manner. The operation of the distributed data blocks is executed by the master server, NameNode. Processing of data in Hadoop, or MapReduce, is executed in a master-slave manner and consists of three phases: the map phase, the shuffle phase, and the reduce phase. In the map phase, multiple map tasks read input data blocks from the HDFS and compute in parallel. Map tasks are executed on slave servers through a TaskTracker running on each slave server, and the assignment of map tasks to TaskTrackers is performed by a JobTracker running on the master server. The output of map tasks is sent to reduce tasks, which requires communication between slave servers. This communication phase is referred to as the shuffle phase. Finally, in the reduce phase, reduce tasks summarize the output of map tasks. MapReduce jobs often exhibit unpredictable performance due to irregular tasks with long execution times. Such outlier tasks in MapReduce jobs have been investigated, and a software system that reduces the impact of outlier tasks on application-execution time has been proposed [5].

While the abovementioned studies discuss the capability of public clouds for running applications, the performance fluctuation of applications running in public clouds was investigated in [11]–[14]. The experimental results using a micro benchmark show a large performance fluctuation. For example, the authors in [11] ran micro benchmarks in Amazon EC2 for one month. Their results show that the coefficient of variation in the CPU benchmark (Ubench) results is more than 20%. The experimental results in [12] show the coefficient of variation is more than 50%. The fluctuation of VM startup time in Amazon EC2 has been investigated [14]. However, discussions in the above works were limited to the performance of micro benchmarks or

AIDA et al.: LONG-TERM PERFORMANCE EVALUATION OF HADOOP JOBS IN PUBLIC AND COMMUNITY CLOUDS

1177
VM startup time. To the best of our knowledge, the fluctuation of the application-execution time in clouds through long-term experiments has not yet been discussed.

3. Experimental Settings

We investigated the fluctuation of application-execution time in clouds by periodically running applications from one to three months on both a public cloud and a community cloud. This section describes the experimental settings.

3.1 Cloud Platforms

We used a public cloud and a community cloud in our experiments. For the public cloud, we used Amazon EC2 in the Asia Pacific (Tokyo) region and the US EAST (N. Virginia) region. Amazon EC2 is one of the most popular IaaS cloud services in the world, and a number of studies have investigated the application performance in Amazon EC2. We used the Hokkaido University Academic Cloud as the community cloud. The Hokkaido University Academic Cloud is the largest community cloud that offers IaaS service to the academic community in Japan. Hereafter, “EC2” (the region name) and “Hokkaido” denote Amazon EC2 and the Hokkaido University Academic Cloud, respectively.

Amazon EC2 offers numerous options for instance types and regions. We chose the compute-optimized instance with numerous CPU cores, c1.xlarge. The instance consists of eight virtual CPU cores with 20 ECUs and 7 GB of memory. Considering the specific settings of the implemented application architecture in our experiments, we assumed that typical users favored a cost-effective instance to run their application programs. At the time of the experiments, c1.xlarge was the instance with the most CPU cores in the Asia Pacific (Tokyo) region and cost-effective compared with smaller instances. For example, the price of one c1.xlarge instance (with 20 ECUs) was cheaper than 20 m1.small instances (with 20 ECUs) in the price list at Jan. 13, 2015. We chose the Amazon Machine Image of the Red Hat Enterprise Linux 6.4 with an Elastic Block Store (EBS) boot and attached a 100-GB EBS formatted by ext4 to each instance. We installed Hadoop 1.1.0 for running benchmark programs on the instance.

We also chose the compute-optimized instance in Hokkaido for comparison with the experimental results in Amazon EC2. The instance consists of virtualized Intel Xeon E7-8870 (2.4 GHz) with 10 CPU cores, 30 GB of memory, and a 1-TB HDD formatted by ext4. Computing servers are connected through Gigabit Ethernet. We directly ran benchmark programs on physical computing servers; that is, no VM was used. We installed Ubuntu 12.04 and Hadoop 1.1.0 on the local PC cluster. While computing servers used in the experiments are dedicated to running our benchmark programs, the network is shared with other computing servers.

3.2 Benchmarks

We used Hadoop applications as the benchmark in our experiments. Hadoop is widely used as a distributed data processing framework in clouds. We deemed that the experimental results of Hadoop applications serve as a good performance indicator in clouds.

We launched four virtual servers (or instances) as slave servers and one virtual server as a master server. The master server hosts the NameNode process and the JobTracker process, and the slave server hosts the DataNode process and the TaskTracker process. Hadoop settings have many configuration options. We selected a well-balanced setting that we empirically obtained through preliminary experiments. In this setting, the number of slots in the TaskTracker, or the maximum number of map tasks simultaneously running on each TaskTracker, was eight. The total number of reduce tasks was four. The HDFS automatically replicates data blocks to improve reliability. We configured the replica setting such that the file system had a total of two copies of each block. Although more configuration options are possible, the study of the optimal Hadoop setting is beyond the scope of this paper.

We ran one micro benchmark, TestDFSIO, and two application benchmarks, Wordcount and Sort, as described in the following [16]:

3.2.1 TestDFSIO

TestDFSIO is a benchmark for investigating the read/write performance of the HDFS. In this benchmark, map tasks read/write data blocks on the HDFS and reduce tasks summarize the statistics. The results indicate the performance of the HDFS in the computing cluster used in the experiments. In our experiments, we conducted a write/read for eight 1-GB files on the HDFS.

3.2.2 Wordcount

Wordcount counts the number of occurrences of each word in a given input text file. Map tasks count the occurrences of words in data blocks of an input file, and reduce tasks collect and summarize the occurrences counted by map tasks. In our experiments, we used an archive of English Wikipedia pages [21] with a 40-GB input file.

3.2.3 Sort

Sort performs sorting of words in a given text file. Map tasks
read words in data blocks of an input file and the words are sorted in a shuffle phase. Then, reduce tasks collect and merge the sorted words. We again used an archive of English Wikipedia pages as the input file.

4. Experimental Results

In our experiments, we ran benchmark programs every four hours (2 AM, 6 AM, 10 AM, 2 PM, 6 PM, and 10 PM). We started/stopped the VMs before/after running every benchmark in Amazon EC2, so that different physical computing servers were allocated to VMs for every benchmark run. Our previous study [22] showed the results of experiments in which we kept VM instances running during the whole experimental period. The performance fluctuation in the previous study was smaller than the results presented in this paper. The results indicate the heterogeneity of physical computing servers in cloud platforms. The different physical computing servers sometimes differ in hardware characteristics and workload congestion. We conducted the experiments over three months in each cloud platform, except the experiment running Sort in EC2 (N. Virginia); that experiment was terminated after one month from the start due to the experimental error.

We observed server errors and Hadoop job execution errors during the experimental period. Server errors are due to the accidental shutdown of VMs. We observed four cases of Hadoop job execution errors: memory allocation failed in a reduce task, a map task failed to access HDFS, a slave server failed to access a master server, and a JobTracker entered safe mode. Server errors affected the execution of the benchmark programs; that is, benchmark programs were not started or failed during execution. When Hadoop job execution errors occurred, some jobs failed to run to completion, whereas other jobs did. Here, we omit the execution times of jobs that we observed as errors.

4.1 Overall Performance

Table 1 shows the summary of the average performance (avg.) and the fluctuation in the experimental results. We indicate the performance fluctuation by the coefficient of variation (CV) in this paper.

The results show that the performance fluctuation in three cloud platforms is comparable to the performance fluctuation in the local PC cluster (Local), where the computing resources are dedicated to our experiments. The results indicate that workload congestion in three cloud platforms do not significantly affect the performance fluctuation. The performance fluctuations in two application benchmarks, Wordcount and Sort, are small compared with the results in the previous studies using micro benchmarks. However, the performance of our micro benchmark, the read throughput in TestDFSIO, exhibits significantly larger fluctuation than the others, whereas the write operation exhibits small fluctuation. The performance of the read throughput is significantly affected by the cache effect. When the read operation is executed, some data are cached in the memory but others are not. This cache effect could account for the large fluctuation in the read throughput.

4.2 Performance Fluctuation

Figure 1 shows the throughput measured by TestDFSIO jobs in cloud platforms. The results show a large fluctuation of the read throughput, as presented in the CV of Table 1.

Figures 2 and 3 show the execution times of Wordcount and Sort jobs, respectively, through the overall experimental period in cloud platforms. The results consist of the total execution time (total exec.), the average execution time of a shuffle task (avg. shuffle), the average execution time of a reduce task (avg. reduce), and the average execution time of a map task (avg. map) for each job.

The results indicate two phenomena regarding the total execution time of jobs. First, we see long-term time dependency for total execution times of both the Wordcount job and the Sort job in EC2 (N. Virginia and Tokyo). For example, Figs. 2 and 3 show that the total execution times increase from mid-October 2013 to the end of the experimental period in EC2 (Tokyo). Execution times of the shuffle task and the reduce task also increase after mid-October, whereas the execution time of the map task, which is a computing intensive task, is stable during the whole experimental period. The shuffle task and the reduce task require communication between the slave servers, which are greatly affected by workload congestion in a cloud platform. The results

| benchmark          | EC2 (N. Virginia) | EC2 (Tokyo) | Hokkaido | Local |
|--------------------|------------------|-------------|----------|-------|
| TestDFSIO          | avg.[MB/s]       | 10.48       | 12.39    | 10.46 | 33.83 |
|                    | CV[%]            | 10.32       | 13.96    | 7.57  | 10.15 |
| Write throughput   | avg.[MB/s]       | 76.18       | 81.28    | 31.99 | 84.43 |
|                    | CV[%]            | 29.20       | 28.93    | 47.66 | 40.87 |
| TestDFSIO          | avg.[MB/s]       | 1143.36     | 1071.60  | 1129.71 | 659.78 |
| Read throughput    | CV[%]            | 7.15        | 10.03    | 8.28  | 10.18 |
| Wordcount          | avg.[s]          | 2750.39     | 2345.61  | 1662.39 | 916.62 |
| Execution time     | CV[%]            | 14.64       | 17.69    | 6.72  | 13.98 |
| Sort               | avg.[s]          | Sep. 30, 2013 | Aug. 6, 2013 | Dec. 10, 2012 | Apr. 16, 2013 |
| Execution time     | CV[%]            | Jan. 5, 2014  | Nov. 18, 2013 | Mar. 25, 2013 | Jul. 16, 2013 |
| experimental period| start (JST)      | *Nov. 2, 2013 | | | |
|                    | finish (JST)     | | | | |

* The experiments running Sort in EC2 (N. Virginia) has finished due to the experimental error.
indicate that the CPU performance in the cloud platforms is stable but the communication performance is affected by the workload congestion in the cloud platforms. For the same reason, we also see the increase of the execution time in EC2 (N. Virginia).

The above phenomenon is consistent with the current service level in EC2. We used the compute-optimized instance, c1.xlarge, in the experiments. The CPU performance is optimized in c1.xlarge but the communication performance is affected by workload congestion. Furthermore, the performance of the Sort job is more affected by workload congestion in comparison with the Wordcount job.
The reason is that the Sort job performs more communication in the shuffle phase in comparison with the Wordcount job. For example, 42 [GB] of data are shuffled in the Sort job, whereas 16 [GB] of data are shuffled in the Wordcount job. Table 2 shows the experimental results that explain the above discussion. The results indicate that the CVs of the average map task execution time are smaller than those of the shuffle and reduce tasks. The results also show that the CVs of the average task execution times of the Sort job are larger than those of the Wordcount job.

We also see two classes of jobs in the results: regular jobs with stable execution time and irregular jobs with significantly long execution time. For example, the total execution time of the regular job is approximately 1000 [sec] in EC2 (Tokyo), but that of the irregular job is approximately 1400 [sec] or longer. We discuss what happens in the irregular job in Sect. 4.4.

Figures 2 and 3 also show that the total execution time of the Wordcount job in Hokkaido increases continuously from December 30th, 2012 through January 4th, 2013. We observed that one slave server stopped during this period. Therefore, jobs were executed on three slave servers. Wordcount jobs were completed even on the three slave servers, because data blocks saved in the failed slave server were replicated in other slave servers. However, Sort jobs failed during this period, because the required disk space exceeded the disk capacity on the three slave servers. During this period, we also observed the performance degradation of TestDFSIO jobs for the same reason.

Hokkaido. Although both Hadoop 1.1.0 and Hadoop 0.20.2 are developed in the same development branch, one question may arise from the setting: How does the difference settings between Hadoop 0.20.2 and Hadoop 1.1.0 affect the performance fluctuation? In order to answer this question, we conducted the experiments to compare the performance fluctuation between Hadoop 0.20.2 and Hadoop 1.1.0 on Hokkaido. Table 3 shows the experimental results conducted by running benchmark programs every four hours from Dec. 12, 2014 through Jan. 13, 2015. While we see the performance (throughput and execution time) of Hadoop 1.1.0 is slightly better than Hadoop 0.20.2, the significant difference for CVs between Hadoop 0.20.2 and Hadoop 1.1.0 is not observed. Thus, we can say that the difference between Hadoop 0.20.2 and Hadoop 1.1.0 does not affect the performance fluctuation.

4.3 Performance Dependence on Time and Day

Next, we discuss the performance dependence on the time of day and the day of the week. The demand for computing resources in a datacenter might exhibit trends in time frames. We assume that the trends affect the performance of applications running in the datacenter.

Figure 4 presents the distribution of the total execution time (maximum/minimum execution time and the quantile) among Wordcount jobs, which are started at the time indicated on the horizontal axis in the cloud platforms. Moreover, Fig. 5 indicates the distribution of the total execution time among Wordcount jobs, which are executed on the day indicated on the horizontal axis. The results, or quantiles in

| benchmark | task  | EC2 (N. Virginia) | EC2 (Tokyo) |
|-----------|-------|------------------|-------------|
| Wordcount | map   | 2.10             | 1.65        |
|           | shuffle | 5.54             | 9.85        |
|           | reduce  | 7.98             | 8.71        |
| Sort      | map   | 5.42             | 6.96        |
|           | shuffle | 8.40             | 13.51       |
|           | reduce  | 10.74            | 17.06       |

Table 2: CVs [%] of average task execution time.

| benchmark | task  | Hadoop 0.20.2 | Hadoop 1.1.0 |
|-----------|-------|---------------|---------------|
| TestDFSIO | avg.[MB/s] | 25.33        | 27.13        |
|           | CV[%]   | 7.80          | 4.71          |
| TestDFSIO | avg.[MB/s] | 68.01          | 81.11        |
|           | CV[%]   | 18.48        | 22.48        |
| Wordcount | avg.[s]  | 1026.47       | 985.57       |
|           | CV[%]   | 9.05          | 10.17        |
| Sort      | avg.[s]  | 1722.12       | 1572.15      |
|           | CV[%]   | 7.93          | 8.25         |

Table 3: Performance comparison between Hadoop 0.20.2 and 1.1.0.
the figures, indicate that the performance fluctuation among Wordcount jobs is independent of both the time of day and the day of the week. However, we can see the dependence of Sort jobs on EC2 (N. Virginia). Figure 6 exhibits larger performance fluctuations during office hours (10 AM to 6 PM) and midnight (2 AM) in EC2 (N. Virginia). Moreover, Fig. 7 shows larger performance fluctuations on weekdays (Monday to Friday) in EC2 (N. Virginia).

Two questions arise from these results: Why does the performance fluctuation among Sort jobs show dependence? Why does the performance fluctuation in EC2 (N. Virginia) show dependence? The answer to the first question is that the communication performance is greatly affected by the workload congestion in a Sort job, as discussed in Sect. 4.2. For the second question, the answer is related to the workload congestion in datacenters of the Amazon Web Service. Although Amazon Web Service does not disclose the utilization in datacenters, EC2 (N. Virginia) is the oldest datacenter launched in 2006 and is known as a busy datacenter. On the other hand, both EC2 (Tokyo) and Hokkaido were
launched in 2011. We can say that the higher utilization in EC2 (N. Virginia) makes the application performance more sensitive to workload congestion. A similar phenomenon is also reported in [14].

4.4 Analysis of Irregular Jobs

We investigated the execution traces of irregular jobs. Here, we define an irregular job as a job with an execution time that is 30% longer than the average execution time over all jobs. The results show that irregular jobs are categorized into three groups: (a) jobs with reduce task re-execution, (b) jobs with long shuffle time, and (c) jobs with both reduce task re-execution and long shuffle time. The number of jobs categorized into (a), (b) and (c) are 8, 30 and 48, respectively on EC2.

A reduce task is re-executed in irregular jobs (a) by the Hadoop fault-tolerant mechanism. Hadoop automatically restarts a task when the task fails for some reason. We observed that the average reduce task execution time in irregular jobs in the worst case was 1.9 times longer than the average reduce task execution time in all jobs. Irregular jobs (a) were also observed among the jobs executed in the local PC cluster. This means that restarting a reduce task is not a special case that is observed only on a virtualization platform. The long shuffle time in irregular jobs (b) can be explained by the workload congestion in the cloud platform, as we discussed in Sect. 4.2. We observed that the average shuffle time in irregular jobs in the worst case was double the average shuffle time in all jobs.

We observed that all irregular jobs in the local PC cluster were categorized into (a). The performance degradation by irregular jobs are a primary cause that increases the performance fluctuation even on the mostly dedicated local PC cluster. We compared CVs between the results with irregular jobs (Table 1) and those without irregular jobs. The latter results, or CVs for the execution times of Wordcount and Sort jobs without irregular jobs, are 2.23% and 5.93%, respectively; while the former results are 10.18% and 13.98%, respectively. The network setting, where the network is shared with other computing servers, could be a cause for the performance fluctuation. However, we have not observed workload that has significantly affected the communication performance in the local PC cluster during our experiments. Thus, we conclude that the performance fluctuation on the local PC cluster is primarily caused by the irregular jobs with reduce task re-execution.

5. Conclusion

In this paper, we presented the experimental results obtained by investigating the fluctuation of the application-execution time in clouds. We investigated the performance fluctuation of Hadoop jobs in a public cloud and a community cloud for one to three months. The experimental results indicate phenomena that we cannot see without long-term experiments and phenomena inherent in Hadoop. The results give us the direction for estimating Hadoop application performances in clouds. We should be aware of the characteristics of applications, datacenters and time frames. We also should be aware that task re-execution greatly affects the performance of Hadoop applications.

We assumed a use case scenario of resource brokering that uses the results obtained in our experiments. Currently, we are developing a resource brokering architecture in inter-cloud platforms, where multiple cloud platforms connected via the Internet and jobs (or VMs) are migrated between the platforms to improve performance [23]. The results of the long-term experiments in cloud platforms help to optimize resource brokering strategies, e.g., the strategy to decide when and where a job should be migrated.

The experimental results reported in this paper, however, represent performance in a limited setting, i.e., a cluster of five virtual servers. The performance of applications in a larger setting with hundreds or thousands of servers may be more greatly affected by workload congestion and may fluctuate more. For example, a small number of virtual servers can be expected to be hosted on physical servers in a single rack. In this case, high networking performance is expected through a single network switch in the rack. However, hundreds or thousands of virtual servers are hosted on more physical servers distributed between multiple racks. The networking performance of virtual servers may be more greatly affected by external jobs. We selected a limited setting in our experiments due to budget limitations. We leave comprehensive experiments with more computing servers and more comprehensive benchmark suits, which address wider aspects of typical Hadoop applications, and real world applications like graph analysis or machine learning for future research.

Although our focus in this paper is on the execution time, cost is also an important issue when using clouds. Some users may require a performance guarantee for user applications with a limited budget. To reduce cost, the user may select more flexible instances, such as Amazon EC2 Spot Instance. In this case, the application performance may be more greatly affected by the balance between supply and demand in a cloud platform, and the performance may fluctuate more. The issue of the performance-cost tradeoff is more complex and is left for future work.

Acknowledgments

This work was supported in part by JSPS KAKENHI (Grant Number 24240006).

References

[1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, “Xen and the art of virtualization,” ACM SIGOPS Oper. Syst. Rev., vol.37, no.5, pp.164–177, Oct. 2003.
[2] Y. Koh, R.C. Knauerhase, P. Brett, M. Bowman, Z. Wen, and C. Pu, “An analysis of performance interference effects in virtual environments,” Proc. IEEE International Symposium on Performance
Analysis of Systems & Software (ISPASS 2007), pp. 200–209, 2007.

[3] X. Pu, L. Liu, Y. Mei, S. Sivathanu, Y. Koh, and C. Pu, “Understanding performance interference of I/O workload in virtualized cloud environments,” Proc. 2010 IEEE 3rd International Conference on Cloud Computing (CLOUD ’10), pp. 51–58, 2010.

[4] Y. Hashimoto and K. Aida, “Evaluation of performance degradation in HPC applications with VM consolidation,” Proc. 2012 Third International Conference on Networking and Computing (ICNC 2012), pp. 273–277, 2012.

[5] G. Ananthanarayanan, S. Kandula, A. Greenberg, I. Stoica, Y. Lu, B. Saha, and E. Harris, “Reining in the outliers in Map-reduce clusters using Mantri,” Proc. 9th USENIX Conference on Operating Systems Design and Implementation, pp. 1–16, 2010.

[6] G. Juve, E. Deelman, K. Vahi, G. Mehta, B. Berriman, B.P. Berman, and P. Maechling, “Scientific workflow applications on Amazon EC2,” Proc. 2009 5th IEEE International Conference on E-Science Workshops, pp. 59–66, 2009.

[7] K. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H.J. Wasserman, and N. Wright, “Performance analysis of high performance computing applications on the Amazon Web Services Cloud,” Proc. 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom 2012), pp. 159–168, 2010.

[8] G. Wang and T.S.E. Ng, “The impact of virtualization on network performance of Amazon EC2 data center,” Proc. 29th Conference on Information Communications (INFOCOM ’19), pp. 1163–1171, 2019.

[9] D. Knight, K. Shams, G. Chang, and T. Soderstrom, “Evaluating the efficacy of the cloud for cluster computation,” Proc. 2012 IEEE Aerospace Conference, pp. 1–10, 2012.

[10] A. Marathe, R. Harris, D.K. Lowenthal, B.R. de Supinski, B. Rountree, M. Schulz, and X. Yuan, “A comparative study of high-performance computing on the cloud,” Proc. 22nd International Symposium on High-Performance Parallel and Distributed Computing (HPDC ’13), pp. 239–250, 2013.

[11] J. Schad, J. Dittrich, and J.A. Quiñé-Ruiz, “Runtime measurements in the cloud: Observing, analyzing, and reducing variance,” Proc. VLDB Endow., vol. 3, no. 1-2, pp. 460–471, Sept. 2010.

[12] Y. Sudo and K. Hato, “Performance analysis of public clouds,” IEICE Technical Report, IN2013-60, 2013 (in Japanese).

[13] B. Dejun, G. Pierre, and C.H. Chi, “EC2 performance analysis for resource provisioning of service-oriented applications,” Proc. 2009 International Conference on Service-Oriented Computing, ICSOC/ServiceWave ’09, pp. 197–207, Springer-Verlag, Berlin, Heidelberg, 2009.

[14] M. Mao and M. Humphrey, “A performance study on the VM startup time in the cloud,” Proc. 2012 IEEE 5th International Conference on Cloud Computing (CLOUD), pp. 423–430, 2012.

[15] Amazon Elastic Compute Cloud (Amazon EC2), “http://aws.amazon.com/ec2/”

[16] Hadoop, “http://hadoop.apache.org/”

[17] K. Thompson, G.J. Miller, and R. Wilder, “Wide-area internet traffic patterns and characteristics,” IEEE Netw., vol. 11, pp. 10–23, 1997.

[18] O. Tickoo, R. Iyer, R. Illikkal, and D. Newell, “Modeling virtual machine performance: Challenges and approaches,” ACM SIGMETRICS Perform. Eval. Rev., vol. 37, no. 3, pp. 55–60, 2010.

[19] CITRIX, “http://www.citrix.com/”

[20] Apache CloudStack, “http://cloudstack.apache.org/”

[21] Wikipedia Database download, “http://en.wikipedia.org/wiki/wiki?wiki=database:download”

[22] K. Aida, O. Abdul-Rahman, E. Sakane, and K. Motoyama, “Evaluation on the performance fluctuation of hadoop jobs in the cloud,” Proc. 16th IEEE International Conference on Computational Science and Engineering (CSE 2013), pp. 159–166, 2013.

[23] O. Abdul-Rahman and K. Aida, “Multi-layered architecture for the management of virtualized application environments within inter-cloud platforms,” Proc. 5th International Conference on Cloud Computing Technology and Science (CloudCom 2013), pp. 238–243, 2013.