March - April 2020
ISSN: 0193-4120 Page No. 4622 - 4629

A Cost Cap-based Dynamic Resource Allocation Approach in Heterogeneously Multi-Experimental Data Computing Environments

Seo-Young Noh[1], Syed Asif Raza Shah[2], Tae-Hyung Kim[*3]
[1]Assistant Professor, Department of Computer Science, Chungbuk National University, 1 Chungdae-ro Seowon-gu, Cheongju, 28644, Republic of Korea
[2] Assistant Professor, Department of Computer Science & CRAIB, Sukkur Institute of Business Administration University (SIBAU), Sukkur, Sindh, Pakistan
[*3]Principal Engineer, Samsung Research, Samsung Electronics, Seoul, Republic of Korea
rsyoung@cbnu.ac.kr[1], asif.shah@iba-suk.edu.pk[2], thkim4u@gmail.com[*3]

Article Info
Volume 83
Page Number: 4622 - 4629
Publication Issue:
March - April 2020

Article History
Article Received*: 24 July 2019*
Revised*: 12 September 2019*
Accepted*: 15 February 2020*
Publication*: 26 March 2020*

Abstract
In scientific research environments, improving resource sharing and utilization is important for scientific data processing. Computing environments in scientific communities are in general dedicated to specific and mission-oriented experiments. Such approaches will lead to under resource utilization. Cloud computing is being gradually and actively adapted by scientific research communities because of its flexibility and promptness by on-demand approach. In order to improve utilization, there are two main factors seriously considered, which are cost and processing time. Cost and processing time are mutually conflict such that one can negatively impact the other. Therefore, it is important to find a balancing point between these two factors. In this paper, we propose a simple virtual machine allocation approach which can reduce computing complexity of load balancing problem by simply capping the cost. In our experiment, results show that our approach can be used to improve resource utilization as well as cost efficiency in multiple scientific data processing environments.

Keywords: *Cloud Computing, Virtual Machine Allocation, Heterogeneous Scientific Environment, Data Computing*

1. Introduction

Data is getting more important nowadays in all aspects. Scientific community is also no exception. Such communities require significant computing performance in general. As research paradigm has been shifting toward data intensive scientific research, acquiring more as well as better computing power is one of keys in successful scientific discoveries[1]. In that sense, it is not difficult to observe different types of approaches to improve resource utilization in scientific communities. Grid computing[2], for instance, is one of popularly used examples. The main idea behind grid computing is to gather available computing resources through network[3]. It can be configured across multiple regions and even multiple continents. Grid computing has been writing a successful story in the high energy physics[4]. There are two main players in software viewpoint, Open Science

March - April 2020
ISSN: 0193-4120 Page No. 4622 - 4629

Grid(OSG)[5] and European Grid Infrastructure(EGI)[6]. OSG is popularly being used in United States while EGI used in European-based experiments. In high energy physics, U.S. based CDF(Collider Detector at Fermilab)[7] and D0[8] experiments, for example, used OSG based grid computing facility for data processing where data was produced from Fermi National Accelerator Laboratory(FNAL)[9]. In the other side, examples of EGI-based experiment are ALICE(A Large Ion Collider Experiment)[10] and CMS(Compact Muon Solenoid)[11] which have been actively conducting data analysis of high energy physics at CERN(European Organization for Nuclear Research)[12]. Such experiments produce tremendous experimental data and it is impossible for a single country to handle such amount of data by its own computing facility because it needs huge computing power and storage capacity which is in general leading to astronomical budget. Therefore, such big experiments tend to be organizing international collaborative consortium. An international research consortium naturally has to build up networks for data transmission[13-15]. For example, when processing data at CERN, acquired data from a detector has been stored at long-term storage and copies are transferred to Tier centers[16] scattered all around world through high speed network.

Cloud computing is now getting popular due to its flexibility and ease of use in data intensive researches[17,18]. Grid and cloud computing are both focusing on resource utilization, but they are big different in the underlying technology viewpoint. Grid computing is based on orchestration approach while cloud computing based on virtualization. In spite of the differences, many grid computing environments have been actively adapting cloud technology in their computing facilities. Therefore, cloud computing approach is now being attractive from data intensive science[19].

When processing data, there are two key factors that we must consider, which is cost and processing time. These two factors tend to move opposite directions, which means that if we need to improve processing time, cost tends to be going high. The other way is also true in general. If we reduce cost, then we need more processing time. Therefore, it is important to find a natural break-even point which balances the cost and the processing time. Finding such a balancing point between cost and time could be complex and require large amount of computation time. Scientific research communities using large scaled computing facilities handle quite many jobs submitted by scientists. In such environments, it is not an appropriate approach to find the finest accurate balancing point because computations for such results also require significant computing power, leading to the burden to the system. Therefore, we need a lightweight as well as simple approach.

In this paper, we will propose a simple virtual machine allocation algorithm to find a balancing point for maximizing resource utilization in limited computing power. Our approach is simple, but efficient in that it does not require big burden of computation time. In order to validate our approach, we applied our model to a real operational data produced from working computing environment.

The rest of this paper is organized as follows: In Section 2, we will discuss related works, mainly focusing on resource utilization perspective. In Section 3, we will introduce a simple resource allocation algorithm. In Section 4, we will discuss the experimental environment and results. Finally, we will give our insights in Section 5 as a conclusion.

March - April 2020
ISSN: 0193-4120 Page No. 4622 - 4629

2. Related Works

Under resource utilization is one of critical problems in data center. It found that many researchers have proposed various ways to improve resource utilization in data center and in scientific research communities. HTCondor[20] is one of spotlighted schedulers used in many scientific data computing. In order to utilize unused computing resources, HTCondor has a feature of flocking which can virtually combine remotely distanced computing resources[21]. Whenever there are over exceed jobs submitted, some jobs in the queue can be redirected to available computing resources which were flocked by HTCondor.

Just flocking unused computing resources are immediately preempted by owner's jobs. Occupied jobs should be repelled whenever the owner of the computing resources submits jobs, which means that running time of occupying jobs may require more execution time than their estimated time compared to the case that they were executed in their dedicated computing resources. In order to prevent such a circumstance, we need to find secure and reliable resources which cannot be preempted. One of simple and straightforward approach is to utilize cloud computing in scientific data analysis. Such an approach has been already proven by FNAL. FNAL conducted an experiment to combine their private cloud called FermiCloud and Amazone Cloud for CMS data processing[22]. Results show that the approach can achieve cost-effective and execution efficient in scientific data analysis by dynamically and elastically handling virtual machines in both cloud systems.

When adapting cloud computing in scientific research environment, it is also important to understand the overhead of virtual machines. Researchers at FNAL proposed a reference model which considers overheads of virtual machines when assigning virtual machines to physical machines[23,24]. Such an approach focuses on finding specific conditions in terms of number of virtual machines which can dramatically degrade the performance of host physical systems. There is an opposite approach compared to virtual machine viewpoint, which is focusing on the viewpoint of applications[25]. Since all applications embedded in virtual machines should be run on physical systems, computation performance heavily relies on types of applications running on the host systems. If I/O intensive applications are too much consuming the resource of the host machine, it is not a good approach to allocate another I/O intensive virtual machines to the same host because too many I/O intensive virtual machines are competing each other to acquire necessary resources of the host machine.

Beside virtualization approach, physical consolidation is another way to improve the utilization of dedicated computing resources[26]. It was common to split a big computing resources into multiple small size dedicated computing resources in traditional data center for high energy physics. Consolidating such divided resources can be done by using various schedulers like HTCondor. Although they are not even sharing common environments, they can be consolidated in cloud environment.

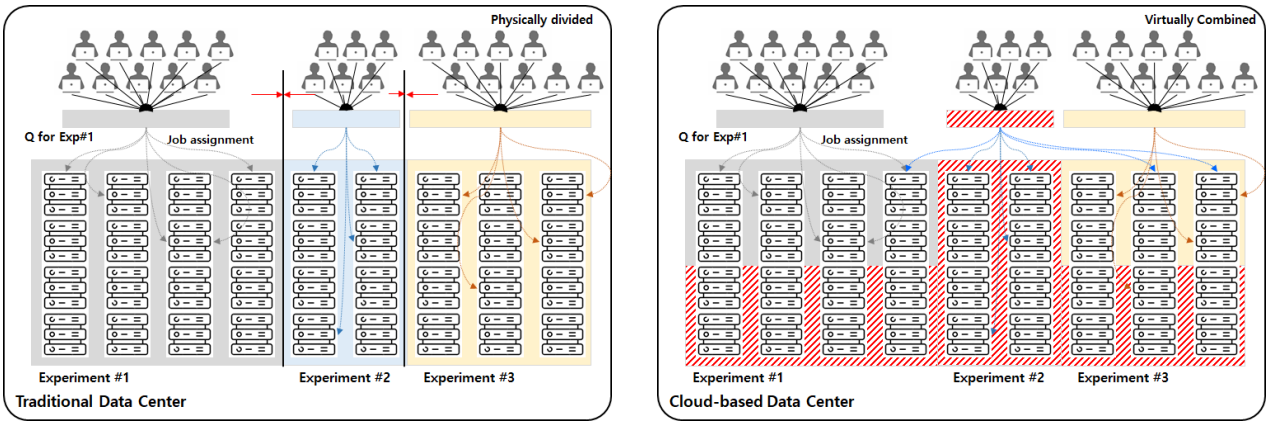
3. Cost Cap-based Allocation Algorithm

In traditional and simple approach, computing resources are dedicated to specific and mission-oriented experiments. Such configuration can be consolidated into multi-queue based resource sharing environment. Consolidating physically dedicated computing systems depends on the feature of preemption provided by schedulers. However, such approach may cause catastrophic results. Dedicated computing farms are configured with experiment specific requirements which might be big different compared to the configuration of the

March - April 2020
ISSN: 0193-4120 Page No. 4622 - 4629

other computing systems. Due to such job specific configuration requirements, directly applying to physical machines should be avoid as much as possible. In that sense, many scientific workflows nowadays are running in virtual machine-based or container-based infrastructures.

Figure 1 shows how to consolidate multiple scientific computing farms using virtualization technology. It is worth to note that traditional data centers for scientific data computing in general divide the total computing capacity into multiple dedicated computing farms to specific experiments. As shown in Figure 1-(a), physically dedicated computing farms to each experiment cannot be utilized by the other experiment even though there are available resources as idle status. Such circumstance is not appropriate and should be prevented in the resource utilization perspective. Therefore, it is useful if we can consolidate such physically divided systems into virtually single computing farm. In addition, it should be emphasized that whenever original owners want to preempt their resources by submitting jobs, those resource are immediately preempted by the original owners. Figure 1-(b) describes how the user group of experiment #2 utilizes idle resources dedicated in experiment #1 and experiment #3, respectively. Users submit their jobs as usual. However, in consolidated environment, jobs are redirected to unallocated resources in experiment #1 and experiment #3. Users are not aware they are using computing resources dedicated to different experiments, naturally leading to high resource utilization. In cloud-based data center, a resource controlling mechanism controls the virtual machines to be launched where and when in the consolidated computing environment.



(a) Physically Divided Traditional Data Center (b) Virtually Combined Cloud-based Data Center

Figure 1. Difference between Traditional Data Center and Cloud-based Data Center

Algorithm 1 shows the simple and straightforward virtual machine allocation algorithm in multiple experiment environment. It first checks if there are available slots which are not occupied by dedicated experiments.

If there are, we can reuse such available computing power for another resource consuming experiments. Once available slots are found, C_i which is dedicated resources for experiment i, we can redirect the available resource C_i to experiment j. In order to find resource consuming experiment, the algorithm calls getFullQueue(). If starving queue, q, has been found, then we redirect jobs in q to

March - April 2020
ISSN: 0193-4120 Page No. 4622 - 4629

available computing cluster C_i by calling redirectJobs().

Algorithm 1: Simple Virtual Machine Allocation Algorithm

```
procedure vm-allocation
    while true do
        /* check if there is available slots */
        if isAvailableSlot() == true then
            /* acquiring slots from cloud Ci */
            Ci ← getSlot()
            /* get a queue which is full */
            q ← getFullQueue()
            /* redirect jobs to available slots in Ci */
            redirectJobs(q, Ci)
        else
            if isWaitingJobs(qi) == true then
                if isQccupiedBy(Ci, qi) then
                    waiting()
                else
                    rePreempt(Ci, qi)
                end if
            end if
        end if
    end while
end procedure
```

If there is no available slot in the computing farms, we have to check if a computing farm, C_i, is occupied by another experiment j. In this case, the computing farm C_i must be reassigned to its correspondence experiment i. By calling rePreempt(), occupied resource C_i is reallocated to experiment i. Running jobs at C_i will be repelled from C_i and resubmitted to q_j, where q_j is the queue for experiment j. If there are available slots in C_j, repelled jobs will be immediately assigned to C_j; otherwise, it has to wait until there are available slots at C_j.

4. Experiments and Results

In this paper, we have taken the experimental data for computation time and cost from three dedicated scientific computing farms called CDF, HCP(Hadron Collider Physics), and STAR(Solenoidal Tracker at RHIC)[27] for 31 days. The three different experiments have been measured in the computing facility at Global Science experimental Data hub Center of Korea Institute of Science and Technology Information[28].

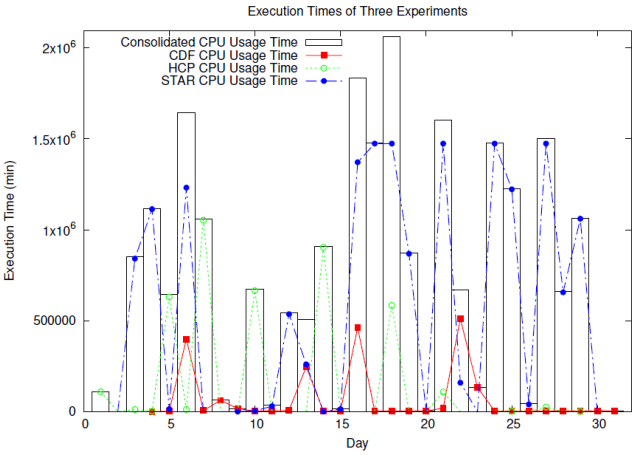


Figure 2. Execution Times of Three Experiments and Consolidated Time

As shown in Figure 2, daily execution time is represented in different line colors and their consolidated execution time has been expressed in a bar graph. It is easy to note that STAR experiment requires the most computing resources while the others have relatively available computing resources. When measured, each experiment has its own dedicated cluster system. CDF, HCP, and STAR have 100, 200, and 300 CPU cores, respectively. In our experiment, we map each core to each virtual machine, that is, core and virtual machine are one-to-one correspondence. For our analysis, we need to introduce the cost of a virtual machine. In this paper, we refer to the AWS pricing model[29], the most popular and widely used cloud platform.

Table 1 shows the allocated CPUs and average execution time in the three experiments. The costs of computing capacity has been derived from AWS pricing model. In our experiment, we adopted single core virtual machine price model of AWS, which is t2.small type roughly $0.023 per hours.

March - April 2020
ISSN: 0193-4120 Page No. 4622 - 4629

Table 1: Resources, average running time, and costs in the three experiments

Exp.	CPU	Average time(mins)	Cost($)
CDF	100	30	55.2
HCP	200	45	110.4
STAR	300	60	165.6
	SUM (600)	AVG(45)	SUM (331.2)

Figure 4 shows the combined execution time and the cost at statically configured cluster system. Daily execution time has been accumulated and the cost of the computing farm has been indicated in empty-circle line which is constant. Accumulated execution time is depending on the number of dedicated CPU cores and average execution time.

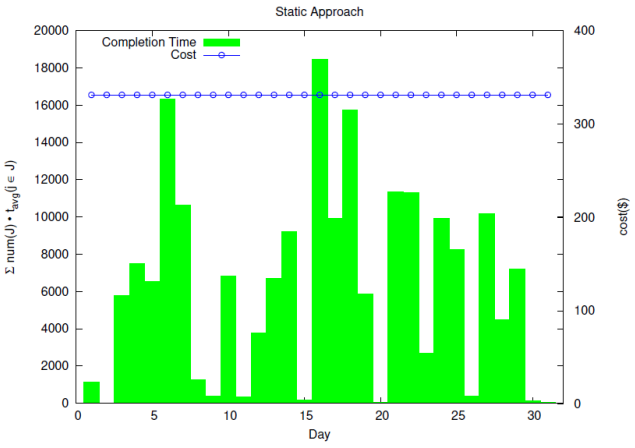


Figure 4. Static Combined Execution Time and Cost

Now we can apply Algorithm 1 to our example. In our experiment, we have the projected cost, which is $321.1. Since the cost has been capped, computing time has to be adjusted according to the capped cost. Figure 5 shows experimental results when capping the cost. Although completion time has relatively higher than the static approach, we can save the cost by capping. If an accumulated execution time is under the boundary of the capped cost, then there is no affection. However, if such time is beyond the cost, then we can expect the duration of data processing will be extended proportionally as much as the amount of reduced cost. It should be noted that our simple and straightforward approach reduces the computation time to calculate the optimal balancing point between the cost and the processing time

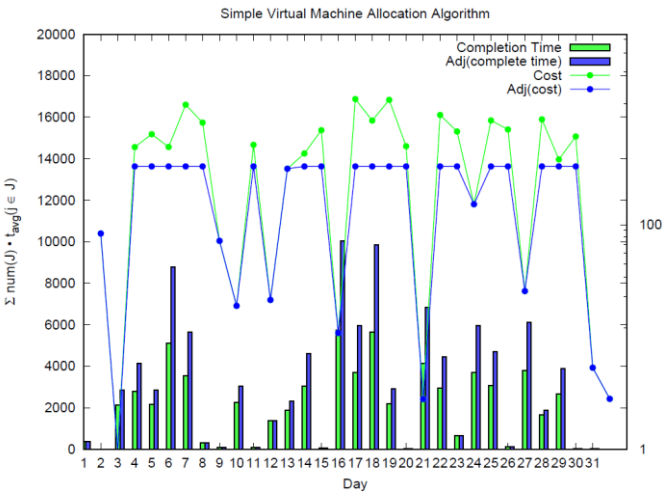


Figure 3. Cost-capped Approach

5. Conclusion

Computing power is getting more important for scientific discoveries. More computing power, better and faster research outputs can be expected. In that sense, it is important to find the best utilization in available computing resources. Utilization of existing computing resources are always challenging tasks due to its scale, security, and networking burdens. Grid is one of examples which can be the representative of computing utilization by loosely coupling approach. Although grid computing has been successfully supporting

March - April 2020
ISSN: 0193-4120 Page No. 4622 - 4629

many big facility based data analysis such as ALICE, CMS, CDF, and STAR experiments, loosely coupled approach is not appropriate for those environments where responsiveness is most considerable.

Cloud computing is also considered same as Grid computing. However, there is fundamental difference between two approaches. Underlying technology on which cloud computing is based is *virtualization* while grid computing relies on *orchestration*. In such computing environments that responsiveness is important and exclusive, virtualization technology can play important role for resource utilization. In this paper, we have analyzed three different experiments for computing usages, execution times, and costs. As discussed, *cost* and *execution time* are mutually conflict. Two factors are tending to move opposite directions, that is, if cost is going down, then execution time would move toward high. Therefore, it is important to find an appropriate point, which is called *interesting point*.

In this paper, we have proposed a simple virtual machine allocation algorithm which can be used in multiple experiment supportable computing facilities. Rather than finding the complex break-even point between the cost and the performance, our approach simply caps either the cost or the processing time. Such a lightweight and simple approach can reduce the computation time as well as improve resource utilization. As shown in our results, our resource allocation algorithm can be used in the internally dedicated system and shows the improvement of reducing execution time when capping the cost. It is worth to note that our approach can be used to cap either the cost or the execution time although we capped the cost in this paper. Considering various factors which can find more accurate optimized point is also important, but such an approach is not practicable in data intensive computing environment as discussed in this paper. Therefore, our lightweight and straightforward approach will be useful when simplifying the resource utilization problem in scientific data processing environments.

6. Acknowledgment

This work has been supported by the research grant of the Chungbuk National University in 2019 and the Korea Institute of Science and Technology Information(KISTI).

7. References

[1] Tony Hey, Stewart Tansley, and Kristin Tolle, editors. The Fourth Paradigm: Data-Intensive Scientific Discovery. Microsoft Research, 2009.

[2] Ian Foster and Carl Kesselman. The Grid 2: Blueprint for a New Computing Infrastructure. 2003.

[3] Ian T. Foster and Peter Kacsuk. Editors' message. J. Grid Comput., 1(1):1-2, 2003.

[4] Dagmar Adamov and Pablo Saiz. Grid computing in high energy physics experiments. In Soha Maad, editor, Grid Computing, chapter 9. IntechOpen, Rijeka, 2012.

[5] OSG. Open Science Grid. Available from https://opensciencegrid.org/

[6] EGI. European Grid Infrastructure. Available from https://www.egi.eu/

[7] CDF. Collider Detector at Fermilab. Available from https://www-cdf.fnal.gov/

[8] FNAL. Fermi National Accelerator Laboratory. Available from https://www.fnal.gov/

[9] ALICE. A Large Ion Collider Experiment. Available from https://home.cern/science/experiments/alice

[10] CMS. Compact Muon Solenoid. Available from https://home.cern/science/experiments/cms

[11] CERN. European Organization for Nuclear Research. Available from https://home.cern/

[12] Qiming Lu, Liang Zhang, Sajith Sasidharan, Wenji Wu, Phil DeMar, Chin Guok, John Macauley,

[13] Inder Monga, Se-Young Yu, Jim Hao Chen, Joe Mambretti, Jin Kim, Seo-Young Noh, et al. BigData express: Toward schedulable, predictable, and high-performance data transfer. In Proceedings of the 5th IEEE/ACM International Workshop on Innovating the Network for Data-Intensive Science, INDIS@SC 2018, Dallas, TX, USA, November 11, 2018, pages 75-84. IEEE, 2018.

[14] Syed Asif Raza Shah, Wenji Wu, Qiming Lu, Liang Zhang, Sajith Sasidharan, Phil DeMar, et al. Amoebanet: An sdn-enabled network service for big

March - April 2020
ISSN: 0193-4120 Page No. 4622 - 4629

data science. J. Network and Computer Applications, 119:70-82, 2018.

[15] Syed Asif Raza Shah and Seo-Young Noh. A dynamic programmable network for large-scale scientific data transfer using amoebanet. Applied Sciences, 9(21), 2019

[16] S Fuess, G Garzoglio, B Holzman, R Kennedy, A Norman, S Timm, et al. The HEP-Cloud facility: elastic computing for high energy physics - the NOvA use case. Journal of Physics: Conference Series, 898:052014, Oct 2017.

[17] S Timm, G Garzoglio, P Mhashilkar, J Boyd, G Bernabeu, N Sharma, et al. Cloud services for the fermilab scientific stakeholders. Journal of Physics: Conference Series, 664(2):022039, Dec 2015.

[18] Seo-Young Noh, Steven Timm, and Haengjin Jang. vcluster: a framework for auto scalable virtual cluster system in heterogeneous clouds. Cluster Computing, 17(3):741-749, 2014.

[19] HTCondor. Htcondor manuals. Available from https://research.cs.wisc.edu/htcondor/

[20] I. Gable et al. A batch system for HEP applications on a distributed IaaS cloud. J. Phys. Conf. Ser., 331:062010, 2011.

[21] Burt Holzman, Lothar A. T. Bauerdick, Brian Bockelman, Dave Dykstra, Ian Fisk, Stuart Fuess, et al. HEPCloud, a new paradigm for HEP facilities: CMS amazon web services investigation. Computing and Software for Big Science, 1(1):1, Sep 2017.

[22] Hao Wu, Shangping Ren, Gabriele Garzoglio, Steven Timm, Gerard Bernabeu, Keith Chadwick et. A reference model for virtual machine launching overhead. IEEE Trans. Cloud Computing, 4(3):250-264, 2016.

[23] Hao Wu, Shangping Ren, Gabriele Garzoglio, Steven Timm, Gerard Bernabeu, and Seo-Young Noh. Modeling the virtual machine launching overhead under fermicloud. In 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2014, Chicago, IL, USA, May 26-29, 2014, pages 374-383.

[24] Hui Zhao, Qinghua Zheng, Weizhan Zhang, Yuxuan Chen, and Yunhui Huang. Virtual machine placement based on the VM performance models in cloud. In 34th IEEE International Performance Computing and Communications Conference, IPCCC 2015, Nanjing, China, December 14-16, 2015, pages 1-8.

[25] Byungyun Kong, Geonmo Ryu, Sangwook Bae, Seo-Young Noh, and Heejun Yoon. An efficient approach to consolidating job schedulers in traditional independent scientific workflows. Applied Sciences, 10(4), 2020.

[26] Seo-Young Noh and Sangho Ha. mvcluter: Multiple experiments supportable virtual cluster system. Indian Journal of Science and Technology, 8(S8):582-587, 2015.

[27] STAR. Solenoidal Tracker at RHIC. Available from https://www.bnl.gov/rhic/STAR.asp

[28] GSDC. Global Science experimental Data hub Center at Korea Institute of Science and Technology Information. Available from https://www.kisti.re.kr/eng/

[29] Amazon. Amazon web service pricing model. Available from https://aws.amazon.com/ec2/pricing/on-demand/