ISSN XXXX XXXX © 2019 IJESC

Research Article Volume 9 Issue No.4

Towards Predictable Deadline-Mixed Match and Transactional Workloads for Cloud Computing Jobs

Stenni Martin[1], S. Shanthi, M.E, (Ph.D)[2]
Student[1], Assistant Professor[2]
Sri Eshwar College of Engineering, India

Abstract:
We present a technique that enables existing middleware to moderately manage mixed workloads, long running jobs and transactional applications. In this technique job scheduler is assigned to manage the different workloads. A job scheduler is a computer application for controlling unattended background program execution. Our proposed system is determined by complex application goals and takes into account the application satisfaction with how well the goals are met. We have exhibited, both using a real-system and a simulator, that this approach improves satisfaction fairness across applications compared to existing system. The implementation plan has new aspect. It allows diverse workloads to be collected on any server machine so that we can reduce the decision making process of resource allocation. Our technique permits collocation of the workload types on the same physical hardware, and leverages virtualization control mechanisms to perform online system reconfiguration. In our experiments, including simulations as well as a prototype system built on top of state-of-the-art commercial middleware, we demonstrate that our technique maximizes mixed workload performance while providing service differentiation based on high-level performance goals.

Keywords: Cloud Computing, Job scheduler, Resource allocation, Virtualization.

1. INTRODUCTION

The proliferation of virtualization technologies and rich Internet connectivity have brought an explosive growth in cloud computing. Tenants outsource their computation and storage to public cloud providers, and pay for the service usage on demand. This model offers unprecedented advantages in terms of cost and reliability, compared with traditional computing model that uses dedicated, in house infrastructure. Despite the tremendous momentums it grows, the future viability of cloud computing, however, still depends on its offered performance to tenants. Amazon EC2. Amazon EC2 is a representative public cloud service, it requires a tenant to specify the number (type) of virtual machines (VMs) needed, which are then assigned to available physical servers by the cloud provider. VMs establish TCP connections for data transfer. The bandwidth of a VM depends on the number of contending TCP flows along the path. As a result, the network performance of a tenant is not guaranteed at all, and a selfish tenant can get more bandwidths than others by simply establishing more TCP connections, even under the constraint of bandwidth capping mechanism. It not only makes the task finish time of cloud computing jobs unpredictable, but also affects the cost a tenant pays, since tenants are charged based on the duration of VM occupation in EC2. Hard bandwidth guarantee. In order to achieve predictable performance for cloud computing jobs, hard bandwidth guarantee is proposed to provide to tenants, represented by SecondNet, Oktopus, etc. In this kind of solutions, a tenant specifies both the number of VMs and associated bandwidth of each VM (for hose-model bandwidth abstraction) or VM pair (for matrix-model bandwidth abstraction). Cloud provider then guarantees the tenant's bandwidth requirement by allocating the reserving bandwidth in related links. Although the application performance is predictable by this way, the cloud resource may not be efficiently utilized for two reasons. First, bandwidth fragmentation will naturally exist during bandwidth allocation. Second, theVMand bandwidth requests specified by the tenants may not well match the residual resource in the cloud. The works in and then improve the bandwidth utilization by considering variable bandwidth requirements across time and across VMs, respectively. But both solutions still require the tenants to specify their resource demands, and thus cannot fundamentally solve the problem. Bazaar tries to adjust the number of VMs and the associated bandwidths allocated to a tenant, so as to better match theVMand networking resource in the cloud. However, the solution is designed specifically for MapReduce jobs. If considering a broader range of cloud applications, arbitrarily varying the number of VMs may break the application semantics. Minimum bandwidth guarantee. To improve bandwidth utilization besides providing predictable network performance, some very recent works propose minimum bandwidth guarantee to tenants, while letting the residual bandwidth shared by tenants in a best-effort manner. By this way, the network performance a tenant gets is no worse than the minimum guaranteed bandwidth, and is thus predictable. Meanwhile, the available bandwidth can be fully utilized by tenants with higher traffic demands. Although this approach significantly improves bandwidth utilization in the cloud, they may not be able to meet the demand of cloud computing jobs with deadline requirements. The reason is obvious: the minimum bandwidth guaranteed may be less than the bandwidth required by a job to meet its deadline, while there is no guarantee on the share of extra bandwidth. To guarantee jobs finishing before deadlines, tenants need to accurately set the minimum bandwidth the job needs, which is a great burden for them. In this work we propose DCloud, which is a new interface between tenants and provider for cloud computing with deadline requirements. In today's cloud infrastructure, we have found that data analytic jobs account for a large proportion of cloud jobs, such as web logs analysis, weather forecast analysis, finance analysis, scientific simulation, machine learning, etc.,. A great part of these jobs have deadline requirements since results of them may be useless if they do not finish in time. Cloud resource allocation for these jobs are the focus of DCloud. DCloud requires a tenant to specify both the required resource and the job deadline when submitting a job request to the cloud. The required resource is quantified by the number of VMs and associated bandwidth, as well as the referenced job running duration profiled under the requested VMs and bandwidths. When the cloud provider allocates the resource, she can leverage the time interval between the job running duration and the job deadline to reshape the resource request, which leaves room to efficiently utilize the residual cloud resource without violating the job's deadline. We develop a novel resource allocation algorithm to exploit the room, which uses time sliding to smooth out the peak demand and bandwidth scaling to balance the usage of network resource and non-network resource in the cloud. Table 1 compares DCloud with other bandwidth sharing solutions. Although the concepts above are intuitive and promising, many challenges exist in transferring the basic ideas into a practical system. We employ a number of mechanisms and algorithms to address the challenges. First, when reshaping the tenants' requests, we depend on an inversely proportional rule to conservatively estimate the job running duration after bandwidth scaling, without knowledge of the application semantics. Second, we use the metric of dominant resource utilization (DRU) to perform joint optimization of different types of cloud resources when allocating them to the tenants. Thirdly, we introduce a profiling relax index to mask the possible profiling errors from the tenants. Finally, we design a strategy-proof and job-based charging mechanism to encourage tenants to submit true deadline and resources.

PROBLEM STATEMENT

Problem analysis is used to find the causes of a positive or a negative deviation. In the existing system, the user request is not analyzed properly to allocate the suitable server. Different types of workload require different control mechanisms for management. Different workloads are allotted manually to the server. It will create a problem of overloading. The resources are not utilized properly. Workload takes important consideration while doing the large scale application. There is no scheduling strategy in the existing system to reduce the overhead. To avoid this overhead problem, the new plan is implemented.

2. RELATED WORKS

In cloud computing environments, mutually non-trusted tenants deploy their services in a shared datacenter infrastructure. Each tenant consists of a collection of one or more virtual machines (VMs) placed on one or more physical machines. Cloud environments have a strong requirement to enforce performance isolation among tenants that share a datacenter, but currently mechanisms are lacking to provide performance isolation for datacenter network I/O resources. Effective management of network bandwidth will be crucial to handle the growing range of service workloads that stress local area network resources in the datacenter. Data-intensive applications on scalable frameworks like MapReduce can be highly network-intensive. Also, future datacenters will merge traditional messaging traffic with network storage traffic onto a single converged datacenter fabric, using new network standards and distributed storage and file systems. This paper introduces properties that multi-tenant network performance isolation solutions should provide to meet the practical needs of both cloud users and cloud datacenter providers. We show that previous techniques fall short of meeting all of these requirements, and we report on our significant progress in building an I/O virtualization control system called Gatekeeper that is intended to fulfill these needs. Isolation must work at these large scales. That require per-tenant or per-VM state to be maintained at each switch are impractical if the need to manage a large amount of state at high speed renders the switches prohibitively expensive for cloud computing infrastructure.

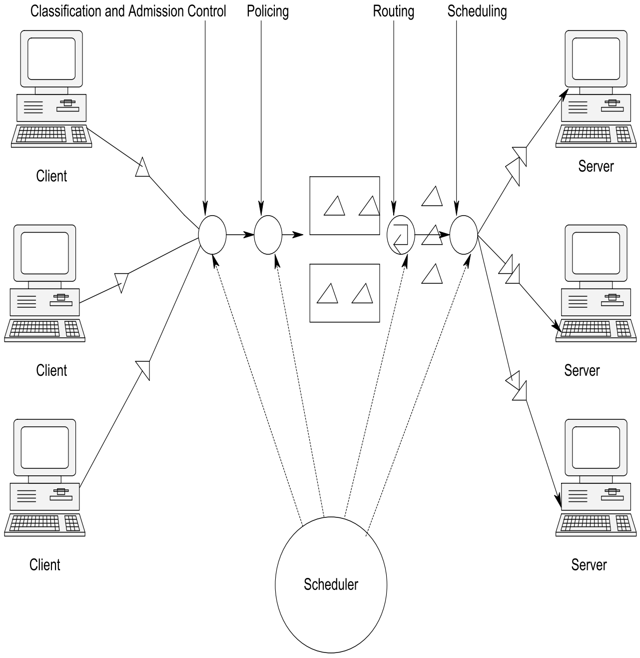
3. CHALLENGES AND CONTRIBUTIONS

3.1 Profiling Relax Index

We start with the allocation parameter of profiling relax index g, which is used for masking the job profiling inaccuracy. We vary g from 0 to 12 percent (as suggested in, the existing profiling method can limit the prediction error below 12 percent), and set all the other parameters as the default. Note that though we reserve more resource than requested for each job, we do not add the actually required resource to run the job. Under different settings we compute the four metrics and show in Fig. 4. We observe that the profiling relax index has only minor impact on all four metrics. For example, when g increases from 0 to 12 percent, the percentage of successful jobs drops by 2.8 percent and the provider's revenue drops by 4 percent. The drops on the VM and bandwidth utilization are 3.5 and 2 percent respectively. The fundamental reason is that, although we conservatively reserve more resource for a target job, the resource is released to be used by other jobs when the target job finishes earlier than expectation. In all the following simulations, we set g ¼12% (worst case) for the DCloud allocation.

3.2 Deadline Extension Ratio

The larger the expected deadline extension ratio r is, the larger room there is for a cloud provider to reshape the request. Though this setting may depend on application types, we vary r from 1 to 20 to understand its general impact. From Fig. 5, as expected, we can see that the deadline extension ratio has almost no impact on VC, as VC does not consider the job deadline at all. Hence in what follows we focus on discussing BL and DCloud. DCloud complete more jobs with larger r. For DCloud, the gain comes from larger room for both time sliding and bandwidth scaling. For BL, it is due to the higher probability that a job can finish before a longer deadline. When r ¼ 1, there is no room for either time sliding or bandwidth scaling, so DCloud performs the same as VC. BL performs the worst among the three, because there is no bandwidth guarantee and some accepted jobs cannot finish before the deadlines. VC can provide bandwidth guarantee and thus the allocated jobs can finish within the deadlines, but its accepted jobs are much fewer than DCloud. When r is as high as 20, DCloud can successfully finish about 40 and 80 percent more jobs than VC and BL, respectively. A good allocation algorithm should make efficient utilization of VM slots. We show the results. For this metric, VC and BL are not affected by the deadline extension ratio. BL has the highest VM utilization (close to 100 percent), because it accepts jobs as long as there are any available VM slots, regardless of the bandwidth demands. DCloud performs better than VC, because more jobs can be accepted. The VM utilization of DCloud also grows with higher r, because the more bandwidth scaling prolongs the VM occupation. When the deadline extension ratio is 20, DCloud's VM utilization is almost three times that of VC. The server link utilization indicates how well the network is utilized. In Fig. 5c, all three models do not result in high utilization of server links, due to the oversubscription in higher-level links. VC and BL have similar results, both less than DCloud. It is because the VC results in bandwidth fragmentation, while the competition based bandwidth sharing in BL may cause bandwidth waste if the numbers of flows in server links are imbalanced. We also consider using the total amount of traffic transferred among VMs as another metric. Due to the limited space, we do not include the graphs. DCloud performs even better under this metric. This is because DCloud can make more localized VM allocation by bandwidth scaling as explained, and thus more traffic are transmitted among VMs located in the same physical server.



Requested Bandwidth per VM

Another important parameter in the resource request is the amount of bandwidth consumed per VM. It reflects the balance between the VM resources and network resources. Intuitively, DCloud has larger advantage with larger bandwidth demand, because we consider balancing the usage of the two resources by bandwidth scaling. Fig. 6 shows the comparison on four metrics with the varied bandwidth. With a larger bandwidth demand, each VM consumes more resource and thus the percentage of successful jobs will be smaller, as is confirmed in Fig. 6a. For DCloud and VC, it is more difficult to allocate VMs for jobs with higher bandwidth requests. For BL, higher bandwidth requests also mean longer job running durations because of bandwidth competition, so the probability for a job to violate the deadline is also higher. With per VM consuming more bandwidth, the overall VM utilization becomes smaller in DCloud and VC, shown, because fewer jobs can be accepted. However, for DCloud, we find that it is not linearly decreasing. When the bandwidth goes beyond 450 Mbps, the VM utilization in DCloud even increases, because of larger room for bandwidth scaling and prolonged VM occupancy. Similarly, DCloud also outperforms the other two, when evaluating using the server link utilization in Fig. 6c. The server link utilization in DCloud slowly decreases with higher requested bandwidth, because higher bandwidth per VM indicates more difficulty in scaling the bandwidth to meet the job deadline. When the bandwidth requirement is as low as 50 Mbps, Dcloud can earn 63.8 and 926 percent more than VC and BL, respectively. At the other extreme when the bandwidth requirement is as high as 500 Mbps, the gap is even more obvious, i.e., DCloud earns 87.0 and 2,170 percent more than VC and BL, respectively.

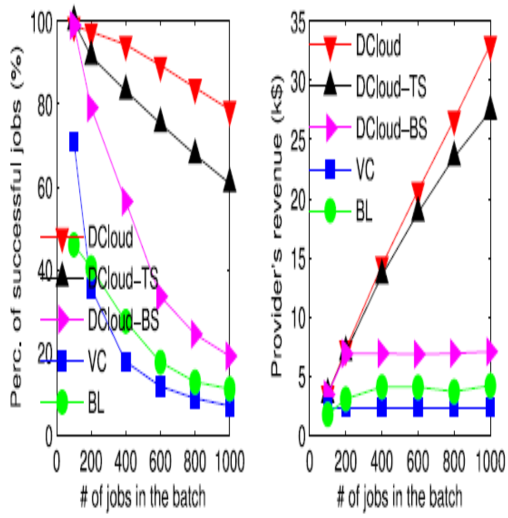
4. METHODOLOGY OVERVIEW

Batch Processing

Another possible model of job arrival is the batch processing, in which all the jobs simultaneously arrive at t ¼ 0. We also study DCloud resource allocation by time sliding only (we call DCloud-TS) and by bandwidth scaling only (we call DCloud-BS), to understand where the benefit comes from. The percentage of jobs meeting deadlines against the number of jobs in the batch. DCloud can accept significantly more jobs than VC and BL in batch processing, especially when the number of jobs in the batch is large. Besides, both time sliding and bandwidth scaling help accept more jobs in DCloud. The revenue of the cloud provider is illustrated in Fig. 9b. The revenue of DCloud increases with more jobs in the batch, because it can use time sliding and bandwidth scaling to accept more jobs within deadlines. But VC and BL allocation cannot earn more profit even when there are more jobs (>200), because the instantaneous cloud resource is used up. In the batch processing, time sliding in Dcloud allocation contributes more than bandwidth scaling in smoothing the peak demand. But by bandwidth scaling only, DCloud can also earn 176 percent more than VC, and 68.2 percent more than BL.

Summary of the Simulation Results

We summarize our key observations from the simulations as follows. First, the effectiveness of DCloud is not very sensitive to the selection of the profiling relax index. Second, Although BL has the highest VM utilization, the jobs are not guaranteed to finish before deadlines. DCloud has much higher VM utilization than VC and has the highest network utilization among the three, in all scenarios. Third, compared with VC, in most scenarios DCloud can complete more than twice jobs and earns more than 50 percent for the cloud provider, even with less costs for individual jobs. The gap between DCloud and BL is much more significant. Finally, from analyzing different parameters, we conclude that DCloud has better performance under the settings of larger deadline extension ratio, smaller requested bandwidth per VM, smaller network over subscription ratio, and higher job arrival rate.



5. CONCLUSION AND FEATURE WORKS

In this paper we designed DCloud, a resource allocation approach for cloud computing jobs to both meet their deadlines and efficiently utilize the cloud resource. By requiring a tenant to submit both her job deadline and the resource demand, DCloud provides predictable performance to applications and also leaves room for shaping the resource requests to better match the residual resource. DCloud uses time sliding and bandwidth scaling to determine the most appropriate time interval to launch each job, as well as the VM locations and reserved link bandwidth. A charging mechanism to encourage selfish tenants to submit the actual required resource, which makes the resource allocation algorithms work more effectively. Extensive simulations and testbed experiments show that, compared with the baseline allocation and recently proposed VC allocation, DCloud can finish significantly more jobs within deadlines, make better utilization of the VM and network resource, and gain much more revenue.

6. REFERENCES

[1]. Amazon Elastic Compute Cloud [Online]. Available: www.aws.amazon.com/ec2 Q2

[2].Google App Engine [Online]. Available: www. code. Google. com/appengine/

[3]. Windows Azure Platform [Online]. Available: ww. Micro soft. com/windowsazure/

[4]. H. Rodrigues, J. R. Santos, Y. Turner, P. Soares, and D. Guedes, "Gatekeeper: Supporting bandwidth guarantees for multi-tenant datacenter networks," in Proc. 3rd Conf. I/O Virtualization, 2011, p. 6.

[5]. C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang, Secondnet: A data center network virtualization architecture with bandwidth guarantees," in Proc. 6th Int. Conf., 2010, pp. 1–12.

[6]. H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," in Proc. ACM SIGCOMM Conf., 2011, pp. 242–253.

[7]. N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. Ramakrishnan, and J. E. Van der Merwe, "Resource management with hoses: Point-to-cloud services for virtual private networks," IEEE/ACM Trans. Netw., vol. 10, no. 5, pp. 679–692, Oct. 2002.

[8]. D. Xie, N. Ding, Y. C. Hu, and R. Kompella, "The only constant is change: Incorporating Time-varying network reservations in data centers," in Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun., 2012, pp. 199–210.

[9].J. Zhu, D. Li, J. Wu, H. Liu, Y. Zhang, and J. Zhang, "Towards bandwidth guarantee in multi-tenancy cloud computing networks," in Proc. IEEE Int. Conf. Netw. Protocol, 2012, pp. 1–10.

[10].V. Jalaparti, H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Bridging the tenant-provider gap in cloud services," in Proc. 3rd ACM Symp. Cloud Comput., 2012. pp 1–14.

[11].L. Popa, G. Kumar, M. Chowdhury, A. Krishnamurthy, S. Ratnasamy, and I. Stoica, "Faircloud: Sharing the network in cloud computing," in Proc. ACM SIGCOMM Conf. Appl., Technol., Archit., Protocols Comput. Commun., 2012, pp. 187–198.

[12].L. Popa, P. Yalagandula, S. Banerjee, J. C. Mogul, Y. Turner, and J. R. Santos, "Elastic Switch: Practical work-conserving bandwidth guarantees for cloud computing," in Proc. ACM SIGCOMM Conf., 2013, pp. 351–362.