International Journal of Interdisciplinary Innovative Research &Development (IJIIRD)
ISSN: 2456-236X
Vol. 05 Issue 02 | 2021

SLA-Based Resource Scheduling for Big Data Analytics as a Service in Cloud Computing Environments

[1]Preethi.S, [2]Ganeshan M
[1]MCA Scholar, School of CS & IT, Dept. of MCA, Jain (Deemed-to-be University)-069
[2]Assistant Professor, School of CS & IT, Dept. of MCA, Jain (Deemed-to-be University)-069

ABSTRACT

Data analytics plays a significant role in gaining insight of big data that can benefit in decision making and problem solving for various application domains such as science, engineering, and commerce. Cloud computing is a suitable platform for Big Data Analytic Applications (BDAAs) that can greatly reduce application cost by elastically provisioning resources based on user requirements and in a pay as you go model. Big Data Analytic Applications (BDAAs) are typically catered for specific domains and are usually expensive. Moreover, it is difficult to provision resources for Big Data Analytic Applications (BDAAs) with fluctuating resource requirements and reduce the resource cost. As a result, Big Data Analytic Applications (BDAAs) are mostly used by large enterprises. Therefore, it is necessary to have a general Analytics as a Service (AaaS) platform that can provision BDAAs to users in various domains as consumable services in an easy-to-use way and at lower price. To support the AaaS platform, our research focuses on efficiently scheduling Cloud resources for BDAAs to satisfy Quality of Service (QoS) requirements of budget and deadline for data analytic requests and maximize profit for the AaaS platform. We propose an admission control and resource scheduling algorithm, which not only satisfies QoS requirements of requests as guaranteed in Service Level Agreements (SLAs), but also increases the profit for AaaS providers by offering a cost-effective resource scheduling solution. We propose the architecture and models for the AaaS platform and conduct experiments to evaluate the proposed algorithm. Results show the efficiency of the algorithm in SLA guarantee, profit enhancement, and cost saving. We evaluate the algorithm performance by adopting a data splitting method to process smaller data samples as representatives of the original big datasets. We conduct extensive experiments to evaluate the proposed admission control and profit optimization scheduling algorithms. Experimental evaluation shows the algorithms perform significantly better compared to the state-of-the-art algorithms in enhancing profits, reducing resource costs, increasing query admission rates, and decreasing query response times.

Keywords: Big Data Analytic Applications (BDAAs), Analytics as a Service (AaaS), Quality of Service (QoS), Service Level Agreements (SLAs).

1. INTRODUCTION

As we enter into the big data era, data analytics becomes critical in decision making and problem solving for various domains, such as science, engineering, commerce, and industry. Big data is referred to a massive volume of structured, unstructured, semi-structured, or real-time data, which is difficult to manage and process with traditional database techniques Huge amount of data is generated from various sources, i.e., social media, sensors, websites, and mobile devices every day. The key enablers of big data are the increased capability of storage and computing resources enabled by recent advances in Cloud computing and the increased availability of data and the ease of access to data supported by various companies like Amazon, Google, IBM, and Facebook. Cloud computing is a suitable platform for big data analytics by providing various resources, which are:

(1) Cloud infrastructures, such as Amazon EC2, which offer on demand virtualized resources, i.e., Virtual Machines (VMs) and storages

(2) Cloud platforms, such as Google App Engine, which manage underlying infrastructure resources and allow users to deploy and run applications

(3) Application services, such as IBM Social Media Analytics which are provided to users through web browser portals. Analyzing big data to find potential insights in the data is essential for individuals, organizations, and governments to make better decisions, i.e., product trend prediction, business strategy making, and disaster prediction and management. However, big data analytics has several challenges. Obtaining analytic solutions is usually expensive as it requires large data storage systems to store voluminous data and considerable computing resources to analyze the data. Furthermore, licenses of Big Data Analytic Application (BDAA) are often expensive. As a result, big data analytics is mostly used by a small number of large enterprises. Moreover, scheduling resources for BDAAs requires expert knowledge. For example, in order to reduce the time of generating analytic solutions, BDAAs should be deployed on various machines for parallel

processing on large datasets, while in order to save energy and cost, resources should be scaled up and down for varied resource demands of BDAAs.

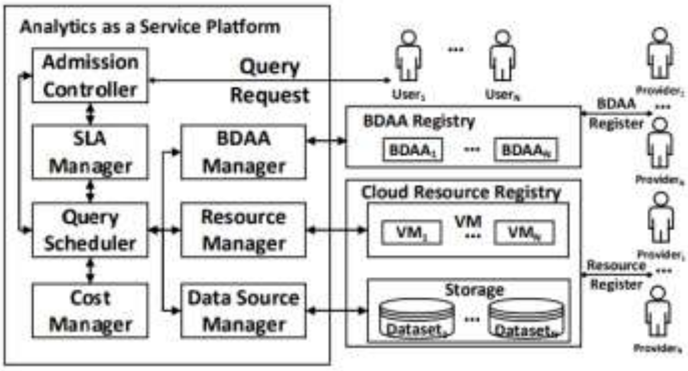
Analyzing big data to find potential insights in the data is essential for individuals, organizations, and governments to make better decisions, product trend prediction, business strategy making, and disaster prediction and management. However, big data analytics has several challenges. Obtaining analytic solutions is usually expensive as it requires large data storage systems to store voluminous data and considerable computing resources to analyze the data. Furthermore, licenses of Big Data Analytic Application (BDAA) are often expensive. As a result, big data analytics is mostly used by a small number of large enterprises Moreover, scheduling resources for BDAAs requires expert knowledge. For example, in order to reduce the time of generating analytic solutions, BDAAs should be deployed on various machines for parallel processing on large datasets, while in order to save energy and cost, resources should be scaled up and down for varied resource demands of BDAAs. The lack of such expertise makes resource scheduling for BDAAs difficult for many users. The above challenges illustrate the need for an Analytics as a Service (AaaS) platform, which aims at serving on-demand analytic requests and delivering AaaS to users as consumable services in an easy to use way and at lower cost. The delivered AaaS should satisfy Quality of Service (QoS) requirements of requests with Service Level Agreement (SLA) guarantees. SLA is the agreement negotiated between service users and providers, which defines the metrics, expected QoS, and penalties during service delivery

To support the AaaS platform, our research focuses on efficiently scheduling Cloud resources for BDAAs and provisioning BDAAs to users as consumable services. We aim to serve the interests of AaaS providers to enhance profit by proposing cost-effective resource scheduling solutions, increase market share by accepting more requests, and improve reputation by delivering satisfactory services to users with SLA guarantees. BDAAs that typically process read-only query requests on given datasets and leave out data consistency and security issues are our targeted applications

The key contribution of our work is an admission control and resource scheduling algorithm that admits queries based on QoS requirements, delivers analytic solutions to users with SLA guarantees, and provides a cost-effective resource scheduling solution to create higher profit for AaaS providers. To achieve this, we

- propose the architecture and models of the AaaS platform;
- formulate the resource scheduling problem based on mixed Integer Linear Programming (ILP) model and propose the admission control and resource scheduling algorithm;
- conduct experiments to evaluate the performance of the algorithm in SLA guarantee, profit enhancement, and cost saving

2. SYSTEM ARCHITECTURE



We propose the architecture of the AaaS platform that provisions general AaaS to various domains of users as consumable services, as shown in Fig. 1. The architecture is composed of the following components:

Admission controller makes decisions about whether to accept queries submitted by users. Admission controller first searches exhaustively in BDAA registry for requested BDAAs and the Cloud resource registry for all possible resource configurations. Then, it estimates query execution time and cost as basis to decide whether to accept or reject queries. Accepted queries are submitted to SLA manager.

SLA manager builds SLAs for accepted queries. SLA violations not only decrease user satisfaction but also generate additional penalty cost; thus, they should be avoided.

Query scheduler is the core component of the AaaS platform, which makes scheduling decisions and coordinates the other components. Scheduling decisions mainly contain

Resource configuration: the scheduler decides which type of Cloud resource to use and how many to use for each resource type

VM control: the scheduler sends VM controlling commands to the resource manager to control VMs, i.e., create VM, terminate VM, and migrate VM

BDAA selection: the scheduler selects requested BDAA to execute a query

Query execution sequence: queries that request the same BDAA are submitted to the waiting queues of VMs running

Query management: the scheduler monitors and manages status of queries during their lifecycles. Query status can be one of submitted, accepted, rejected, waiting for execution, being executed, succeeded, and failed. Query status is monitored to allow the scheduler to take appropriate actions, i.e., send query with failed status to cost manager to generate penalty cost.

Cost manager manages all the cost occurred in the AaaS platform, i.e., query cost and resource cost. It also aims at providing pricing policies that can attract more users to enlarge market share and generate higher profit.

BDAA manager manages BDAAs provided by different BDAA providers and keeps up-to-date BDAA information.

Data source manager manages datasets that are to be processed. As big data has high volume, we move the compute to the data to save data transferring time and network cost.

Resource manager keeps a catalog of all available Cloud resources from different providers and monitors status of VMs to allow the scheduler to take actions such as terminating idle VMs at the end of billing period to save cost.

2.1 Models

he models of the AaaS platform are shown as below:

Query request model: Query requests should contain the following information in query specifications:

- QoS requirements, which contain budget and deadline to fulfil
- required resources to execute a query
- the requested BDAA
- data characteristics including data size, data type, and data location
- user who submits the query and
- query type, i.e., aggregation query and join query.

Cloud resource model contains a set of datacenters and a matrix showing the network bandwidth between the datacenters. Each datacenter contains a set of hosts and data storages that pre-store datasets. Each host contains a set of VMs.

BDAA profile model contains resource configurations, data processing time, data size, data type, and costs for different queries. BDAA profiles are the basis for AaaS platform to estimate query time and cost to make admission and scheduling decisions. Obtaining the profiles requires expert knowledge. Thus, BDAA profiles are assumed to be provisioned by BDAA providers and are reliable.

Cost model contains sub-models of resource cost, penalty cost of SLA violations, query cost (query income) that charges users for using AaaS platform, and profit that is created for AaaS platform providers. Profit is the difference of query cost and the sum of resource cost and penalty cost. Query cost has different cost policies, which are:

- cost based on urgency of deadline
- proportional cost, i.e., proportional to BDAA cost
- a combination of (a) and (b).

Resource cost contains BDAA cost and Cloud resource cost. BDAA cost policies are:

- fixed cost, i.e., annual contract
- usage period-based cost, i.e., hourly basis
- per request-based cost.

Penalty cost policies are:

- fixed cost
- delay dependent cost and
- proportional cost

3. ADMISSION CONTROL AND RESOURCE SCHEDULING

We propose the admission control and resource scheduling algorithm that only admits queries whose QoS can be satisfied and schedules resources to execute queries with SLA guarantees to maximize the profit of AaaS providers. We adopt proportional query cost model, which calculates the query cost proportional to BDAA cost and fixed BDAA cost model, which is fixed cost under annual contract. We aim to make scheduling decision that can give 100% SLA guarantee. Thus, penalty cost of violating SLAs can be avoided if scheduling decision is well made. Since query cost and BDAA cost are fixed and penalty cost can be avoided, our goal of maximizing profit of the AaaS providers is to minimize the Cloud resource cost

3.1 Admission Control Algorithm

The admission controller makes query admission decisions. It first searches the BDAA registry to check whether a query requested BDAA exists. If the BDAA exists, admission controller obtains the BDAA profile. Based on the profile, expected execution time and query cost for different resource configurations are calculated. The expected query cost is calculated based on the adopted query cost model. The expected finish time is the sum of estimated execution time, specified timeout (the maximum time the scheduling algorithm can run), VM creation time (the time to create new VM when needed), submission time (the time when the query is submitted), and waiting time (the time till the query is scheduled). Admission controller makes query acceptance decision if estimated finish time and cost of queries can satisfy QoS requirements of queries using any resource configuration. Afterwards, SLA manager builds SLAs for accepted queries

3.2 Scheduling Algorithms

After the admission controller accepts queries, the query scheduler makes the scheduling decision for each BDAA. We develop scheduling algorithms that support different scheduling scenarios, which are periodic scheduling that schedules queries for each Scheduling Interval (SI) and nonperiodic (real time) scheduling that schedules queries whenever they arrive. For periodic scheduling, we set different SI parameters to study how SI can influence the performance of the scheduling algorithms. We detail the study of algorithm performance for different scheduling scenarios in performance evaluation section.

We aim at scheduling and provisioning resources for BDAAs that can maximize profit for AaaS providers by minimizing resource cost and guarantee SLAs of queries on deadline and budget. The scheduling algorithms adopt a twophase scheduling policy that provision resources in a scalable and elastic way. They scale resources down by releasing resources when the provisioned resource capacity is more than required to reduce cost and scales resources up by leasing new resources when provisioned resources do not have sufficient capacity to execute queries to avoid SLA violations.

4. REVIEW OF LITERATURE

In today's competitive world, the potential business values of these applications depend a lot on the quality of service (QoS) offered by providers. Hence, to gain competitive advantages, providers should focus on the needs of their customers and respond proactively to their marketing strategies, not only to build and raise customers awareness of their services but also meet customers' best expectations for service quality. That is to say, providers must provide the required and promised services to their customers, and these services must achieve the requirements of users (ex. availability, elasticity and scalability).

Given these circumstances, it is very important and necessary for efficient methods to manage and guarantee the QoS promised. Service Level Agreement (SLA) represents a formal contract among service providers and customers, which captures agreements in the sense of QoS. SLAs play an integral role in governing the relationships between providers and customers in the context of Cloud-hosted BDAAs. Furthermore, SLAs can be considered as a strong differentiator, which allows service providers to provide various levels of guarantees for services offered to customers as well as to distinguish itself from competitors. Therefore, it has become a critical task to manage the SLAs thus a high QoS of Cloud-hosted BDAAs may then be guaranteed. Existing surveys focus on SLAs management in grid computing (for Sim) or Cloud computing Internet of Things (IoT) emerges with the recent advancements in computing. Mubeen et al. investigated the existing work on SLAs management for IoT-based applications in Clouds. To the best of our knowledge, there exists only one preliminary review with a simple taxonomy of SLA management for Cloud computing and Cloud-based BDAAs and it does not suffice in any in-depth understanding of managing SLAs or the trends for future research in this area. SLA-specific management for BDAAs in Clouds has largely been ignored.

To bridge gap in this field and spot out the trends for future work, this study performs a systematic literature review (SLR) of the state-of-the-art of SLA-specific management for Cloud-hosted BDAAs. The review mainly concerns the requirements and characteristics across Cloud computing stacks. In particular, a taxonomy-based study is emphasized for the following reasons:

- The existing works on SLA management for Cloud-hosted BDAAs manifest a wide range of thematic perspectives (e.g., techniques used, Cloud models deployed, and layers considered). Further, in each different perspective, various subcategories have been discussed. Take the technique perspective as an example, researchers proposed multiple techniques to address SLA management for Cloud-hosted BDAAs (e.g., simulation and machine learning). It is important to form a hierarchy of these categories for a comprehensive understanding.
- This paper aims not only to present researchers an outlook on SLA-specific management for BDAAs in Clouds, but also to give new insights through a global thematic taxonomy in this research area. The main contributions of this survey include:
- A systematic literature review of SLA Management for Cloud-hosted BDAAs with build-up thematic taxonomy covering six core dimensions including actors, Service layers, techniques, Cloud service and deployment models, SLA metrics and conceptualization;
- A unified SLA model for Cloud-hosted BDAAs from a layer-based perspective to link different types of SLAs in a vertical motion;
- A multi-dimensional categorization scheme regarding SLA metrics dedicated for Cloud-hosted BDAAs, which allows systematically categorization of both common metrics and niche metrics for each layer with respect to requirements of BDDAs; An SLA template is provided for a representative Cloud-hosted BDAA to aid understanding SLAs conversation across its different layers.
- Identification of open issues and future directions of Cloud-hosted BDAA based on the systematic literature review.

5. PERFORMANCE EVALUATION

To evaluate the proposed admission control and scheduling algorithm, we build the framework of AaaS in CloudSim which is a discrete event simulator. CloudSim enables repeatable and controllable experiments and offers comprehensive environment for resource scheduling and provisioning studies. We conduct experiments on the AaaS framework to evaluate the effectiveness and efficiency of the proposed admission control and resource scheduling algorithm for SLA guarantee, profit enhancement, and cost saving. ILP and AGS are utilized as baseline algorithms for the evaluation of AILP scheduling algorithm.

5.1 Resource Configuration

We simulate a datacenter that consists of 500 physical nodes. Each node has 50 CPU cores, 100GB memory, 10TB storage, and 10GB/s network bandwidth. We simulate five types of memory optimized VMs, which are based on Amazon EC2 VM model and managed by the resource manager. The VMs are r3.large, r3.xlarge, r3.2xlarge, r3.4xlarge, and r3.8xlarge. VMs in Amazon EC2 are charged by an hourly basis with unit of dollar/hour. The unit for memory is GiB while for storage is GB.

5.2 Workload

The workload generated in this study is based on the Big Data Benchmark, which runs query applications using different data analytic frameworks on large datasets with given data size, data type, and data location. The benchmark uses Amazon EC2 VMs and details query processing time and the VM configurations. Based on the information, resource requirements of queries requesting different BDAAs under different resource configurations are modeled in CloudSim. Each query request contains the following information: Query submission time is generated using a Poisson process with 1 minute mean Poisson arrival interval. Query class contains 4 types, which are: scan query, aggregation query, join query, and User Defined Function (UDF) query. BDAAs: in our experiment, 4 types of BDAAs are considered, which are built on Impala (disk) (BDAA1), Shark (disk) (BDAA2), Hive (BDAA3), and Tez (BDAA4). Query resource is modeled based on the resource requirements of queries. As the performance of queries vary, we model 10% performance variation by introducing a variation coefficient, which is generated from a Uniform Distribution with upper bound 1.1 and lower bound 0.9. Query user: we simulate 50 users who submit queries and request for BDAAs. Query deadline is generated as the benchmark does not contain QoS requirements on deadline (the same for budget).

5.3 Result Analysis

We generate an approximate 7 hours query workload that contains 400 queries. The requested BDAAs by queries are managed by the BDAA manager. We use lp_solve 5.5 as the integer linear programming solver. Queries whose QoS requirements on deadline and budget can be satisfied are admitted by admission controller. To avoid deadline violation and ensure the performance of queries, the number of queries assigned to VMs by the query scheduler is less than the number of VM cores to avoid time sharing between queries as time sharing may cause deadline violations.

6. RELATED WORK

Our research focuses on provisioning effective and efficient algorithms to support the AaaS platform to deliver on-demand AaaS for various domains of users with SLA guarantees and at lower cost in order to enhance profit, enlarge market share, and improve user satisfaction. Sun et al. propose a general-purpose analytic framework to provision cost-effective analytic solution with multi-tenancy support. They propose an SLA customization mechanism to satisfy diverse QoS requirements of tenants. However, their work does not consider admission control; thus, SLAs are at risk of violations. Moreover, they do not address resource scheduling algorithms with SLA guarantees on deadline and budget and also do not address profit enhancement purpose of AaaS providers. Mian et al. focus on resource provisioning for data analytic workloads in a public Cloud that aims to determine the most cost-effective resource configuration using greedy search heuristic. Their work schedules queries for one application, which does not serve general data analytic purpose. Moreover, the SLA they considered is average query response time and SLA violation is allowed for a cheaper resource cost, which is different from our SLA guaranteeing perspective. We believe that SLA violations can significantly decrease user satisfaction and reputation of AaaS providers. Therefore, SLA violations should be avoided. Garg et al. manage resources for heterogeneous workload in a datacenter. They focus on resource scheduling problem for mixed workloads instead of data analytic workloads. They propose an admission control and resource scheduling algorithm, which considers deadline-constrained scheduling while we consider budget-constrained scheduling as well, which is the market feature of delivering analytics as a service in Cloud computing environments. Zulkernine et al. propose the conceptual architecture of Cloud-based Analytics as a Service, which does not consider admission control. Their work focuses on data analytics for workflow applications. Scheduling and resource provisioning for MapReduce tasks is a well-explored area, which targets at a specific application model and does not serve general data analytic purpose. Alrokayan et al. propose cost-aware and SLAbased algorithms to provision Cloud resources and schedule MapReduce tasks under budget and deadline constraints. They adopt Lambda architecture and focus their current work on the batch layer. Mattess at al. propose an algorithm to dynamically provision resources to meet soft deadline of MapReduce tasks while minimizing the budget. Both of the above works do not consider admission control for SLA guaranteeing purpose. There are some works focus on delivering specific analytics as a service and provisioning query processing techniques, which give support to the AaaS platform and enrich the BDAAs that the AaaS platform can provide to users. Chen et al.address the challenges in delivering continuous analytics as a service to analyze real-time events. Barga et al. propose Daytona to offer scalable computation for large scale data analytics, which focuses on spreadsheet applications. Agarwal at al. propose BlinkDB, which enables approximate query processing on data samples of large datasets with bounded response time and bounded errors

7. CONCLUSIONS AND FUTURE WORK

Data analytics has significant benefits in decision making and problem solving for various domains. To support the AaaS platform that offers data analytics as a service to various domains of users as consumable services, our research focuses on proposing effective and efficient admission control and resource scheduling algorithms. The aim is to allow AaaS providers delivering AaaS with SLA guarantees to increase market share, improve reputation, and enhance profit. We have proposed the architecture and models of the AaaS platform and conducted experiments to evaluate the performance of the admission control and resource scheduling algorithm. Experiments are conducted based on different scheduling scenarios, which are non-periodic (real time) scheduling and periodic scheduling with varied scheduling intervals. Results have shown that our admission control algorithm successfully admits queries based on QoS requirements to allow scheduling algorithms giving SLA guarantees. Moreover, AILP, which is an integration of AGS and ILP, can save more resource cost and create higher profit while overcoming the limitation of ILP in ART. As part of the future work, we will

- continue working on proposing efficient admission control and resource scheduling algorithms;
- study the effect of application profiling in the performance of algorithms; and
- study data sampling techniques that allow query processing on sampled datasets for quicker response time and higher cost saving.

8. REFERENCES

[1] M. Mao and M. Humphrey, "A performance study on the VM startup time in the Cloud," in Proc. of 2012 IEEE 5th International Conference on Cloud Computing (CLOUD), pp. 423-430, 2012.

[2] F. Zulkernine, P. Martin, Y. Zou, M. Bauer, F. Gwadry-Sridhar, and A. Aboulnaga, "Towards Cloud-Based Analytics-as-a-Service (CLAaaS) for big data analytics in the Cloud," in Proc. of 2013 IEEE International Congress on in Big Data (BigData Congress), pp. 62-69, 2013.

[3] M. Alrokayan, A. Vahid Dastjerdi, and R. Buyya, "SLA-aware provisioning and scheduling of Cloud resources for big data analytics," in Proc. of 2014 IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), pp. 1-8, 2014.

[4] M. Mattess, R. N. Calheiros, and R. Buyya, "Scaling MapReduce applications across hybrid Clouds to meet soft deadlines," in Proc. of 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), pp. 629-636, 2013.

[5] Q. Chen, M. Hsu, and H. Zeller, "Experience in Continuous analytics as a Service (CaaaS)," in Proc. of the 14th International Conference on Extending Database Technology, pp. 509-514, 2011.

[6] R. S. Barga, J. Ekanayake, and W. Lu, "Project daytona: Data analytics as a Cloud service," in Proc. of 2012 IEEE 28th International Conference on Data Engineering (ICDE), pp. 1317-1320, 2012.

[7] S. Agarwal, A. P. Iyer, A. Panda, S. Madden, B. Mozafari, and I. Stoica, "Blink and it's done: interactive queries on very large data," in Proc. of the VLDB Endowment, vol. 5, pp. 1902-19

[8] H. Wada, P. Champrasert, J. Suzuki, K. Oba, in IEEE Congress on Services-Part I, 2008. Multiobjective optimization of sla-aware service composition (IEEE, Honolulu, 2008), pp. 368–375

[9] S. Chaisiri, B. S. Lee, D. Niyato, in IEEE Asia-Paciffic Services Computing Conference (APSCC), 2009. Optimal virtual machine placement across multiple cloud providers (IEEE, Singapore, 2009), pp. 103–110

[10] [J. Wang, P. Varman, C. Xie, in 2010 International Conference on High Performance Computing (HiPC), 2010. Avoiding performance ?uctuation in cloud storage (IEEE, Dona Paula, 2010), pp. 1–9

[11] H. Goudarzi, M. Pedram, in 2011 IEEE International Conference on Cloud Computing (CLOUD), 2011. Multi-dimensional sla-based resource allocation for multi-tier cloud computing systems (IEEE, Washington, DC, 2011), pp. 324–331

[12] C. C. Lin, P. Liu, J. J. Wu, in 2011 IEEE International Conference on Cloud Computing (CLOUD), 2011. Energy-aware virtual machine dynamic provision and scheduling for cloud computing (IEEE, Washington, DC, 2011), pp. 736–737

[13] A. Beloglazov, J. Abawajy, R. Buyya, Energy-aware resource allocation heuristics for e?cient management of data centers for cloud computing. Futur. Gener. Comput. Syst. 28(5), 755–768 (2012)

[14] A. Gupta, L. V. Kale, D. Milojicic, P. Faraboschi, S. M. Balle, in 2013 IEEE International Conference on Cloud Engineering (IC2E), 2013. Hpc-awarevm placement in infrastructure clouds (IEEE, Redwood City, 2013), pp. 11–20

[15] S. Singh, I. Chana, Qos-aware autonomic resource management in cloud computing: a systematic review. ACM Comput. Surv. (CSUR). 48(3), 42 (2016) [16]. K. Xiong, X. Chen, in 2015 IEEE 35th International Conference on Distributed Computing Systems Workshops (ICDCSW), 2015. Ensuring cloud service guarantees via service level agreement (sla)-based resource allocation (IEEE, Columbus, 2015), pp. 35–41

[16] X. Zheng, Z. Cai, H. Gao, A Study on Application-Aware Scheduling in Wireless Networks. IEEE Trans. Mob. Comput. 16(7), 1787–1801 (2017) 05, 2012