International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Deadline Constraint Heuristics based Workflow Scheduling in Cloud Computing

Ranjitha R[1], P. Krishnamoorthy[2]

[1]M.E Student, Computer Science Department & Kingston Engineering College, Katpadi, Vellore (DT), TamilNadu, India

[2]M. E (CSE), Assistant Professor, Computer Science Department & Kingston Engineering College, Katpadi, Vellore (DT), TamilNadu, India

Abstract: ***Cloud computing allows business customers to scale up and down their resource usage based on needs. Many of the touted gains in the cloud model come from resource multiplexing through virtualization technology. In this paper, we present a system that uses virtualization technology to allocate data center resources dynamically based on application demands and support green Computing by optimizing the number of servers in use. We introduce the concept of "Time Slot Filtering" to measure the unevenness in the multidimensional resource utilization of a server. By minimizing time, we can combine different types of workloads nicely and improve the overall utilization of server resources. We develop a set of heuristics that prevent overload in the system effectively while saving energy used. Trace driven simulation and experiment results demonstrate that our algorithm achieves good performance.***

Keywords: Workflow, Scheduling, Time slots, Cloud computing

1. Introduction

Nowadays much attention has been paid on workflow scheduling in service computing environments (cloud computing, grid computing, Web services, etc). Resources are generally provided in the form of services, especially in cloud computing. There are two common ways for service delivery: (i) An entire application as a service, which can be directly used with no change. (ii) Basic services are combined to build complex applications, e.g., Xignite and Strik Iron offer Web services hosted on a cloud on a pay-per-use basis . Among a large number of services in cloud computing, there are many services which have same functions and supplied by different cloud service providers (CSPs). However, these services have different non-functional properties. Basic services are rented by users for their complex applications with various resource requirements which are usually modelled as workflows. Better services imply higher costs. Services are consumed based on Service-Level Agreements, which define parameters of Quality of Service in terms of the pay-per-use policy. Though there are many parameters or constraints involved in practical workflow scheduling settings, deadline and time slot are two crucial ones in cloud computing, a new market oriented business model, which offers high quality and low cost information services . However, the two constraints have been considered separately in existing researches. It is necessary to consider both of the constraints jointly because: (i) Deadlines of the workflow applications needs to be met. (ii) Unreserved time slots is crucial for resource utilization from the perspective of service providers. (iii) Utilization of time slots in reserved intervals should be improved to avoid renting new resources (saving money). In this paper, we consider the workflow scheduling problem with deadlines and time slot availability (WSDT for short) in cloud computing. To the best of our knowledge, the considered problem has not been studied yet. Service capacities are usually regarded to be unlimited in cloud computing, which can be used at any time. However, from the CSP's perspective, service capacities are not unlimited. Available service capacities change with workloads, i.e, they cannot satisfy user's requests at any time when a cloud service is shared by multiple tasks. Only some available time slots are provided for new coming users by CSPs in terms of their remaining capacities . For example, each activity in has different candidate services with various execution times, costs and available time slots. For activity 4, there are two candidate services with different workloads. If service 0 is selected for activity 4, the execution time is 4 with the price 6 and available time slots [0,4)S[9,14). Time slot [4,9) is unavailable because there is no remaining capacity. The considered WSDT problem is similar to the Discrete Time/Cost Trade-off Problem (DTCTP) to some extent. We can modify existing algorithms for the latter to the problem under study with less than 200 activities and no more than 20 candidate services in the service pool, spending thousands of seconds. However, the number of activities is usually far more than 200 in practical workflow applications which makes the modified versions are not suitable for the problem under study.

2. Proposed System

In this paper, we present the design and implementation of an automated resource management system that achieves a good balance between the two goals. Two goals are Overload Avoidance and Green Computing

A. Overload avoidance

The capacity of physical machines should be sufficient to satisfy the resource needs of all VMs running on it. Otherwise, the PM is overloaded and can lead to degraded performance of its VMs

B. Green computing

The number of physical machines used should be minimized as long as they can still satisfy the needs of all VMs. Idle PMs can be turned off to save energy.

Licensed Under Creative Commons Attribution CC BY

Advantages
We develop a resource allocation system that can avoid overload in the system effectively while minimizing the number of servers used.

Modules:
- Cloud computing
- cloud resource provisioning
- Virtual machine placement
- Quality of Services (QoS)
- Time Slot Filtering

3. Modules Description

Cloud Computing
Cloud computing is a general term for anything that involves delivering hosted services over the Internet. These services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). The name cloud computing was inspired by the cloud symbol that's often used to represent the Internet in flowcharts and diagrams. A cloud service has three distinct characteristics that differentiate it from traditional hosting. It is sold on demand, typically by the minute or the hour; it is elastic -- a user can have as much or as little of a service as they want at any given time; and the service is fully managed by the provider (the consumer needs nothing but a personal computer and Internet access). Significant innovations in virtualization and distributed computing, as well as improved access to high-speed Internet and a weak economy, have accelerated interest in cloud computing.

Optimal Cloud Resource Provisioning (OCRP)
An optimal cloud resource provisioning (OCRP) algorithm is proposed by formulating a stochastic programming model. The OCRP algorithm can provision computing resources for being used in multiple provisioning stages as well as a long-term plan, e.g., four stages in a quarter plan and twelve stages in a yearly plan. The demand and price uncertainty is considered in OCRP. In particular, an optimal cloud resource provisioning (OCRP) algorithm is proposed to minimize the total cost for provisioning resources in a certain time period. To make an optimal decision, the demand uncertainty from cloud consumer side and price uncertainty from cloud providers are taken into account to adjust the tradeoff between on-demand and oversubscribed costs. This optimal decision is obtained by formulating and solving a stochastic integer programming problem with multistage recourse. Benders decomposition and sample-average approximation are also discussed as the possible techniques to solve the OCRP algorithm. Extensive numerical studies and simulations are performed, and the results show that OCRP can minimize the total cost under uncertainty.

Virtual Machine Placement
When a virtual machine is deployed on a host, the process of selecting the most suitable host for the virtual machine is known as virtual machine placement, or simply placement. During placement, hosts are rated based on the virtual machine's hardware and resource requirements and the anticipated usage of resources. Host ratings also take into consideration the placement goal: either resource maximization on individual hosts or load balancing among hosts. The administrator selects a host for the virtual machine based on the host ratings. Virtual machine placement is the process of mapping virtual machines to physical machines. In other words, virtual machine placement is the process of selecting the most suitable host for the virtual machine. The process involves categorizing the virtual machines hardware and resources requirements and the anticipated usage of resources and the placement goal. The placement goal can either be maximizing the usage of available resources or it can be saving of power by being able to shut down some servers. The autonomic virtual machine placement algorithms are designed keeping in mind the above goals.

VM Migrations
We aim to migrate away the VM that can reduce the server's Usage the most. In case of ties, we select the VM whose removal can reduce the skewness of the server the most. For each VM in the list, we see if we can find a destination server to accommodate it. The server must not become a hot spot after accepting this VM. Among all such servers, we select one whose skewness can be reduced the most by accepting this VM. Note that this reduction can be negative which means we select the server whose skewness increases the least. If a destination server is found, we record the migration of the VM to that server and update the predicted load of related servers. Otherwise, we move onto the next VM in the list and try to find a destination server for it. As long as we can find a destination server for any of its VMs, we consider this run of the algorithm a success and then move onto the next hot spot. Note that each run of the algorithm migrates away at most one VM from the overloaded server. This does not necessarily eliminate the hot spot, but at least reduces its temperature. If it remains a hot spot in the next decision run, the algorithm will repeat this process. It is possible to design the algorithm so that it can migrate away multiple VMs during each run. But this can add more load on the related servers during a period when they are already overloaded.

Quality of Services (QoS)
QoS (Quality of Service) refers to a broad collection of networking technologies and techniques. The goal of QoS is to provide guarantees on the ability of a network to deliver predictable results. Elements of network performance within the scope of QoS often include availability (uptime), bandwidth (throughput), latency (delay), and error rate. QoS involves prioritization of network traffic. QoS can be targeted at a network interface, toward a given server or router's performance, or in terms of specific applications. A network monitoring system must typically be deployed as part of QoS, to insure that networks are performing at the desired level. QoS is especially important for the new generation of Internet applications such as VoIP, video-on-demand and other consumer services. Some core networking technologies like Ethernet were not designed to support prioritized traffic or guaranteed performance levels, making it much more difficult to implement QoS solutions across the Internet.

Licensed Under Creative Commons Attribution CC BY

Cloud resource provisioning

Cloud resource provisioning is a challenging job that may be compromised due to unavailability of the expected resources. Quality of Service (QoS) requirements of workloads derives the provisioning of appropriate resources to cloud workloads. Discovery of best workload-- -resource pair based on application requirements of cloud users is an optimization problem. Acceptable QoS cannot be provided to the cloud users until provisioning of resources is offered as a crucial ability. QoS parameters-based resource provisioning technique is therefore required for efficient provisioning of resources.

Time Slot Filtering

Time Slot Filtering Though there are many available time slots, not all of them meet requirements of activities of workflow instances. Some available time slots might not be available for an activity even before the service assignment. For example, Eft(n)>D in the fastest schedule, or the duration of a time slot is less than the execution time of the activity,or the start or finish time is beyond the earliest start or the latest finish time of the activity.

By filtering out all impossible time slots, remaining time slots are eligible for activities of the instance, which make workflow scheduling much more efficient.

4. Proposed Algorithms

The service assignment for each activity in the WSDT depends on both finish times of all predecessors and available time slots of the service. In this paper, an Iterated Local Adjusting Heuristic framework (ILAH) is proposed for the problem under study. ILAH consists of four components: Time Slot Filtering, Initial Solution Construction, Solution Improvement and Perturbation. ILAH starts from an initial solution π. Improving and perturbing operations are performed on π iteratively until the termination criterion is satisfied. The termination criterion is set as α, the number of consecutive iterations without improvement. Let C(π) be the total cost of π. The high level procedure of ILAH is described in Algorithm 1.

Algorithm 1: ILAH

1 begin
2 Time Slot Filtering;
3 Generate the initial solution π by an initial solution construction strategy;
4 π best← π , C(π best)←C(π);
5 while (termination criterion not met) do 6 π ← Improve(π);
7 if (C(π best)>C(π)) then
8 π best← π , C(π best)←C(π);
9 Perturbation(π);
10 return πbest

For the last three components, we present three initial solution construction strategies, two improvement methods and one perturbation strategy.

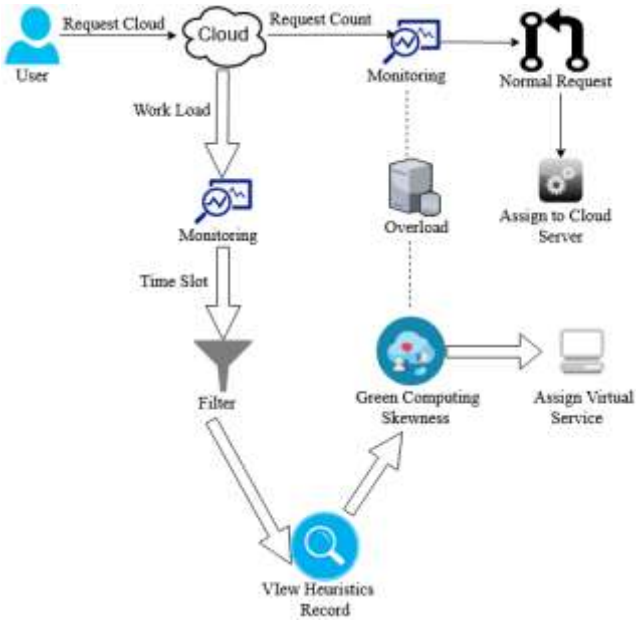

Figure 1: Architecture Diagram

Time Slot Filtering

Though there are many available time slots, not all of them meet requirements of activities of workflow instances. Some available time slots might not be available for an activity vi even before the service assignment. For example, Eft(n)>D in the fastest schedule, or the duration of a time slot is less than the execution time of the activity, or the start or finish time is beyond the earliest start or the latest finish time of the activity. By filtering out all impossible time slots, remaining time slots are eligible for activities of the instance, which make workflow scheduling much more efficient .The Time Slot Filtering procedure is given in Algorithm 2.

Algorithm 2: Time Slot Filtering

1 begin
2 for (each vi∈V) do
3 Calculate Est(i), Eft(i), Lft(i), Lst(i) using equations (8), (9), (10), (11);
4 if (Eft(n)>D) then
5 return NULL; /* infeasible problem */
6 for (each vi∈V) do
7 for (each service Mj i ∈Mi) do 8 for k = 0 to Ns ij−1 do
9 if Fijk−Bi,j,k<eik or Bi,j,k>D or Bijk>Lft(i) or Fijk<Est(i) then
10 Remove sijk from Sij;
11 if (Ns ij=0) then
12 Remove Mj i from Mi;
13 for (each vi∈V) do
14 Generate the service pool Mi by sorting all candidate services in non-increasing order of costs;
15 return {Mi}.

5. Conclusions

We have presented the design, implementation, and evaluation of a resource management system for cloud computing services. Our system multiplexes virtual to physical resources adaptively based on the changing demand. We use the skewness metric to combine VMs with different

Licensed Under Creative Commons Attribution CC BY

resource characteristics appropriately so that the capacities of servers are well utilized. Our algorithm achieves both overload avoidance and green computing for systems with multi resource constraints. Three initial solution construction strategies were developed among which the MCARF and the MACF showed more effective than the EFTF on initial solution construction. Two improvement strategies, the FIH and the GIH, were introduced which had similar influences on the solution improvement. The FIH was very effective for improving poor solutions. By integrating the worst and best initial solution construction strategies (EFTF and MCARF) with the two improvement strategies, four ILAH-based algorithms were developed. Though the EFTF was the worst initial solution construction strategy, it was strange that the EFIG showed the best performance. However, the EGIH obtained the worst performance. In addition, the EFTF was not sensitive to instance parameters while the EGIH was affected by most of the parameters.

For future research, the impact of different pricing interval lengths on work flow scheduling is worth studying. Instance intensive work flows is also a desirable area of study for future work.

6. Acknowledgment

This work has been supported by Mr.P Krishnamoorthi M.E (CSE, Assisstend Professor) Kingston Engineering College, Vellore.

References

[1] C.Weinhardt,A.Anandasivam,B.Blau,andJ.St¨oßer,"Businessmodels in the service world." IT professional, vol. 11, no. 2, pp. 28–33, 2009

[2] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities," in High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on. IEEE, 2008, pp. 5–13.

[3] E. L. Demeulemeester, W. S. Herroelen, and S. E. Elmaghraby, "Optimal procedures for the discrete time/cost trade-off problem in project networks," European Journal of Operational Research, vol. 88, no. 1, pp. 50–68, 1996.

[4] X. Zhang, L. Yang, C. Liu, and J. Chen, "A scalable two-phase top-down specialization approach for data anonymization using mapreduce on cloud," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 2, pp. 363–373, 2014.

[5] M. Menzel, R. Ranjan, L. Wang, S. Khan, and J. Chen, "Cloudgenius: A hybrid decision support method for automating the migration of web application clusters to public clouds," IEEE Transactions on Computers, vol. 64, no. 5, pp. 1336–1348, 2015.

[6] W. Dou, X. Zhang, J. Liu, and J. Chen, "Hiresome-ii: Towards privacyaware cross-cloud service composition for big data applications," IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 2, pp. 455–466, 2015.

[7] C. Liu, R. Ranjan, C. Yang, X. Zhang, L. Wang, and J. Chen, "Mur-dpa: Top-down levelled multi-replica merkle hash tree based secure public auditing for dynamic big data storage on cloud," IEEE Transactions on Computers, vol. 64, no. 9, pp. 2609–2622, 2015.

[8] A. Verma and S. Kaushal, "Deadline constraint heuristic–based genetic algorithm for workflow scheduling in cloud," International Journal of Grid and Utility Computing, vol. 5, no. 2, pp. 96–106, 2014.

[9] S.Abrishami,M.Naghibzadeh,andD.H.Epema,"Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds," Future Generation Computer Systems, vol. 29, no. 1, pp. 158–169, 2013.

[10] S. Abrishami and M. Naghibzadeh, "Deadline-constrained workflow scheduling in software as a service cloud," Scientia Iranica, vol. 19, no. 3, pp. 680–689, 2012.

[11] A. G. Delavar and Y. Aryan, "Hsga: a hybrid heuristic algorithm for workflow scheduling in cloud systems," Cluster computing, vol. 17, no. 1, pp. 129–137, 2014.

[12] C. Akkan, A. Drexl, and A. Kimms, "Network decomposition-based benchmark results for the discrete time–cost tradeoff problem," Euro pean Journal of Operational Research, vol. 165, no. 2, pp. 339–358, 2005.

Licensed Under Creative Commons Attribution CC BY