ISSN:2348-2079

Volume-6 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

An Integrated Resource and Service Consolidation Approach using History Logs under Cloud Environment

[1]Mr. R. Vijayendren, [2]Ms. J. Thanuja, [3]Mr. P. Sathishkumar, [4]Mr. S. Santhosh Kumar, [5]Mr. S. Vadivel,

[1-4]UG Student, Department of Computer Science and Engineering,
K.S.Rangasamy College of Technology, Tiruchengode
[5]Assistant Professor, Department of Computer Science and Engineering,
K.S.Rangasamy College of Technology, Tiruchengode

Email id: [1]vijayendren.r@gmail.com

Abstract

Cloud computing provides all types of resources as services. Hardware and software resources are shared with reference to the user demands. Service components are deployed to support the enterprise applications. Virtual Machines (VM) are provided for the service components based on the capacity levels. Service component consolidation or VM consolidation refers the process of mapping the Virtual Machines to the service components. The service components associated with the same application is denoted as Dependant Service Component. Intelligent service management models identify the service dependencies to achieve the low network latency. The service consolidation with low communication time reduces the communication cost for the network applications.

Service components for enterprise applications are deployed in many hosts and network devices. Service components consolidation is carried out with the consideration of resource constraints, service dependency and network structure. CloudScout is a non-intrusive approach applied for automatically discovers dependent service components. The correlation among service components is analyzed with the time-series information from system monitoring logs. The log mining approach is used for the service dependency discovery with generality and privacy protection features. The iEntropy method is applied to dynamically determine the weight of each resource metric in the service distance calculation. The hierarchical and iterative k-means (HiKM) algorithm is adapted for the dependent service clustering process.The Queue network and network latency optimization are used in the VM consolidation process.

The CloudScout scheme is enhanced to support optimal service dependency discovery and service consolidation process. The service dependency discovery process is improved with sequential dependencies and complimentary dependencies. The Multi Objective Service Consolidation (MOSC) scheme is build to allocate the Virtual Machines for the service components. The service consolidation process is upgraded with resource level constraints.

Index Terms: Cloud services, Virtual Machine Consolidation, Service Dependencies, History Log Analysis and Service Consolidation.

1. Introduction

Performance is one of the most important aspects which data center owner should care about. Cloud computing architecture has scale adaptiveness and cost effectiveness advantage which gives a helpful way to provide high quality performance. Virtualization is the basic technology in cloud computing, it virtualizes the whole data center infrastructure as resource pool. Virtual machines with different kinds of workloads run above virtualization layer to get dynamic resources as soon as they need.

Workloads characteristics on a cloud data center wouldn’t change time by time, they emphasis the same kernel function and physical resource to handle requests in a period of time, which leads to that OS specialization becomes a reality. Workloads can be classified into five categories: CPU-Intensive, Memory Intensive, I/O-Intensive, Network-Intensive and compound according to different resource consuming patterns. Traditional computing and communicating patterns of OS restrict us from getting the most from cloud computing architecture.

On origin OS, users run these five kinds of workloads with the same OS kernel and configuration. There are conflicts between generality and performance, prime physical resources needed by different workloads are not the same and bottlenecks to system performance are different from one another, which leads to dominating function regions in OS kernel are different. Traditional suit-to all OS restricts resource utilization and processing efficiency, which makes uppermost needed processing ability insufficient and wastes those unnecessary functions. System performance will be penalized when confronting workloads with special resource consuming pattern while using common Linux as virtual machine OS. Workload-adaptive OS gives us a solution, it makes optimization to OS according to characteristics of the workloads running on virtual machines.

An intelligent cloud platform should realize workloads characteristics and provide corresponding specialized OS to achieve high performance. A cloud computing oriented network OS and service platform giving corresponding optimization strategy for virtual machines of four categories was design by our study. Related kernel functions and system parameters were specialized according to characteristics of each type of workload base on origin Linux operating system on this platform. Many researches into how to enhance processing ability of cloud architecture are taking, yet, most of the researchers concentrate on virtual machine consolidation, migration, snapshots and so on, merely do they put efforts on optimized virtual machine OS itself based on characteristics of cloud computing workloads to enhance performance of the whole cluster.

2. Related Work

Large data center operators try to find ways to decrease the operational power of data centers. In [1], the authors adopt the aforementioned policy and propose energy-aware heuristics that consolidate VMs onto the minimum number of servers. At the same time, the authors also explore solutions to minimize the SLA (Service Level Agreement) violations experienced within a cloud due to the workload consolidation strategies. The authors propose a workload consolidation algorithm that modifies the Best Fit Decreasing (BFD) approach. The proposed algorithm is named Modified Best Fit Decreasing (MBFD) and it works as follows. It sorts all VMs in decreasing order of their CPU demands, and allocates each VM to a host that provides the least increase of power consumption due to this allocation. The same problem is also tackled using Limited Look ahead Control (LLC). A Kalman filter is employed to estimate the number of future requests to predict the future state of the system and perform necessary reallocations. Ref [2] proposes adaptive heuristics to address the aforementioned problem in an online manner. The authors also conduct a competitive analysis and prove competitive ratios of optimal online deterministic algorithms for the single VM migration and dynamic VM consolidation problems. Gandhi et al. address the problem of allocating an available power budget among servers in a virtualized heterogeneous server farm, while minimizing the mean response time. Mertzios et al. [7] have considered the problem of minimizing the power consumption of a set of machines which can be measured by the amount of time the machines are switched on and processing jobs. Specifically, the above work tries to consolidate jobs whose processing times overlap, such that to minimize the busy time of machines. Another work on workload consolidation is that of [6], where the authors model the problem as an instance of the multi-dimensional bin-packing (MDBP) problem and design a distributed algorithm based on the Ant Colony Optimization. Last, Curino et al. [5] formalize the problemof consolidating multiple databases on fewer machines. They analyze the load characteristics of multiple dedicated database servers and pack their workloads into fewer physical machines, while they achieve near-zero performance degradation.

Network load may play a crucial role in performance degradation of communication-intensive applications. Sonnek et al. [8] considers VM migrations for minimizing the total communication overhead and improving in that way the application performance. In the above work, a distributed bartering algorithm was proposed that allows VMs to negotiate the placement on a physical server that is closer to the data required. The above problem is also tackled in [9], where the authors propose a fully distributed algorithm that results always in optimal placements in tree structured networks. The aforementioned algorithm is based on minimum-cut maximum-flow techniques. Ref [3] considers the online problem of minimizing the total network load between a service represented by one or more VMs and the clients requesting the corresponding service. The authors solve the problem by proposing an online strategy that migrates VMs towards the center of their communicational gravity. Their strategy is also accompanied with its competitive analysis.

In the following text, we mention the most relevant works we found in the literature in regards to the mechanisms of live virtual machine migrations and their effects on clouds. We must note that such works are complementary to our work, since our algorithm could adopt such mechanisms to implement virtual machine migrations in and of themselves. Specifically, study the problem of live virtual machine migrations across a Gigabit LAN. They facilitate the use of post-copy with adaptive pre-paging to eliminate all duplicate page transmissions. They also eliminate the transfer of free memory pages in both migration schemes through a dynamic self-ballooning mechanism. Remote Direct Memory Access (RDMA) to improve the efficiency of VM migration. Particularly, their methodology reduces the migration overhead: up to 80% on total migration time and up to 77% on application observed time. They presented a novel approach, by which it is possible to migrate groups among different clusters [4]. Another work focused on live virtual machine migrations is that provides a synchronization algorithm to orchestrate the migrating VM states. They showed through experimental evaluation that their algorithm can drastically reduce migration overheads compared with pre-copy algorithm. Finally, the effects evaluated of live migration of virtual machines performance on the performance of applications running inside Xen VMs. The authors of the paper showed that, in most cases, migration overhead is acceptable but cannot be disregarded, especially in systems where availability and responsiveness are governed by strict Service Level Agreements.

To the best of our knowledge, none of the above papers consider the problem of both minimizing the energy consumption and the total network load incurred within a cloud due to **(a)** VM communicational dependencies and **(b)** VM migrations. We compare our solutions with BFD and MBFD, since these algorithms are more closely to our work.

3. Cloud Service Dependency Discovery

Today's enterprise applications are designed with suites of independently deployable service components [10]. For instance, an enterprise web application commonly comprises a bunch of application servers, web servers, and database servers. We call the service components that belong to the same application as dependent service components. It is quite beneficial to identify the service dependency, since such knowledge provides a basis for intelligent service management to achieve low network latency of the corresponding applications.

Cloud computing technology aims to deliver everything as a service. With cloud computing, hardware and software are provided on demand. It is a common practice that an isolated virtual machine (VM) with appropriate capacity is provided for each service component. In this paper, we follow the assumption that each service component is assigned to an individual VM, which is widely used by the cloud computing community.

The reason we make this simplifying assumption is that the performance difference between one service component per VM and multiple service components per VM is relatively small, i.e., the intra-VM communications cost is close to inter-VM communications cost of 0.398 ms. Consequently, the two cases are sufficiently similar for this study. Hence, we will interchangeably use service component consolidation and VM consolidation in the rest of this paper.

With the ever-growing scale and complexity of enterprise applications, it is prevalent that a network application or a parallel application consists of thousands of dependent service components . Due to the capacity constraints of an individual physical host (PH), the dependent service components always span several hosts and network devices, making network efficiency a crucial issue. There are two approaches for achieving high network efficiency: (1) tuning the network structure according to the requirements of hosting VMs and (2) designing algorithms to consolidate the hosting VMs to make maximum use of the network. It is a cumbersome task to tune network structure after the infrastructure of a data center is settled. Therefore, it is a feasible and effective solution to consolidate VMs considering network efficiency, resource constraints, and network structure to promote the performance of network applications. In our work, we select VM consolidation as a representing example to evaluate the impact of service dependency on network efficiency. There are many aspects of network efficiency, here we focus on the data transmission cost and data communication cost.

Three network topologies are commonly adopted in cloud data centers: tree, fat-tree, and VL2. The structure of these three topologies all employs divide-and conquer strategies. Switches are divided into three layers, which relieves the data transmission pressure on a single switch. For example, in the tree topology, data is exchanged through the access switch when the source host and destination host are connected with the same access switch, such as host 1 and host 2. The data is transferred through both the access switch and the connected aggregation switch when the source and destination host are connected with different access switches, such as host 1 and host 3. This is a feasible way to reduce the data transmission cost, since the data transmission distance is reduced through locating dependent service components on hosts that are connected by switches at the lower layer.

We evaluated the average communication time in a large data center across two cities in China with mature network infrastructure: Suzhou City and Hangzhou City. Details of the experimental environment will be introduced in Section 6. According to the experimental results, we find that the average communication time between a pair of VMs that run on different PHs canvary considerably. The communication time between VMs on the same PH is the shortest (0.398 ms), since the data s directly transferred through the memory bus. The time cost between VMs on different PHs, which are connected by the same aggregation switch, is longer (0.604 ms). The worst case (17.2 ms) is when VMs connected by different aggregation switches are located in different cities. The above observations reveal that the communication cost of a network application can be reduced by consolidating its service components on hosts with ashorter communication time. Service dependency discovery is not trivial, which is capable of reducing the data transmission cost and communication cost by guiding to consolidate dependent services. To discover the service dependency, we adopt a log mining approach due to its advantages of generality and privacy protection. The key challenges in our work are twofold: service component distance calculation and dependent service component clustering. To solve these challenges, we propose a non-intrusive approach named CloudScout, which discovers the dependency by mining system monitoring data. In this paper, we describe our experiences in designing, implementing, and evaluating CloudScout. We make the following contributions:

1) We propose a method to calculate the distance between service components. We classify the state of a service component as dormant, stable, or active to calculate the correlations precisely. We develop a method named iEntropy to determine the weight of each resource metric dynamically in the service distance calculation.

2) We present an improved clustering algorithm named HiKM (short for hierarchical and iterative k-means). HiKM uses the hierarchical clustering results for the resource usage metrics and network connection number to determine the initial centroids for k-means. It improves the accuracy of service clustering compared to k-means and hierarchical clustering algorithms.

3) We adopt a queue network to verify the influence of service dependency on the network latency of network applications. It provides a foundation for consolidatin service components according to their dependency. 4) We conduct exhaustive experiments in real-world data centers. We deploy five applications on 290 VMs. The experimental results show that CloudScout can effectively discover service dependency and help to reduce the network latency of network applications and distributed applications.

4. Issues on cloud service dependency discovery schemes

Service components for enterprise applications are deployed in many hosts and network devices. Service components consolidation is carried out with the consideration of resource constraints, service dependency and network structure. CloudScout is a non-intrusive approach applied for automatically discovers dependent service components. The correlation among service components is analyzed with the time-series information from system monitoring logs. The log mining approach is used for the service dependency discovery with generality and privacy protection features. The iEntropy method is applied to dynamically determine the weight of each resource metric in the service distance calculation. The hierarchical and iterative k-means (HiKM) algorithm is adapted for the dependent service clustering process. The Queue network and network latency optimization are used in the VM consolidation process. The following issues are identified from the current cloud service dependency discovery schemes.

- Limited accuracy in service dependency discovery and service consolidation process
- Sequential dependencies are not focused in the analysis
- Complimentary dependency relations are not focused
- Resource conflict and availability constraints are not focused in the service consolidation process

5. Integrated Resource and Service Consolidation Approach

The service dependency discovery and service consolidation operations are carried out with log mining mechanism. Different relationship models are used in the service dependency discovery process. The service consolidation process is build with current resource status levels. The system is divided into six major modules. They are System monitoring process, Service dependency classification, Service distance estimation, Clustering process, VM consolidation process and Multi Objective Service Consolidation (MOSC) scheme.

The system monitoring module collects the service and resource usage details. The service dependency classification is used to categorize the services. Service components relationships are analyzed in the distance estimation module. The clustering process is build to group the similar service components. The virtual machine mapping operations are carried out under the VM consolidation process. The Multi Objective Service Consolidation (MOSC) scheme is constructed resource constraint based VM mapping process.

5.1. System Monitoring Process

The network and resource usage data values are collected under the system monitoring process. TCP and UDP connection details are collected to identify the connection number details.The CPU, memory, I/O and network resource details are collected as resource usage metrics. The system monitoring details are maintained under the history logs.

5.2. Service Dependency Classification

The service components and dependency categories are identified from the service dependency classification process. Service components are categorized for the correlation analysis process. Service dependencies are categorized with its operating flows. General, sequential and complimentary dependencies are used in the system.

5.3. Service Distance Estimation

The service distance estimation process is applied to estimate the relationship between the service components. The preprocess is performed to select the features from the system monitoring data values. The iEntropy method is used to assign the weights for the resource metrics. The distance estimation is performed between the pair of service components.

5.4. Clustering Process

The similar service components are grouped in the clustering process. Hierarchical and iterative k-means (HiKM) algorithm is used for the service component clustering process. The initial centroids are estimated using the hierarchical clustering results. The iterative clustering operations are carried out with the selected centroids.

5.5. VM Consolidation Process

The VM consolidation process is carried out to allocate the virtual machines to the service components. The queue network and network latencies are analyzed for the VM mapping process. The mapping process is performed using the service dependency information. Transmission cost and data communication cost parameters are also used in the mapping process.

5.6. Multi Objective Service Consolidation (MOSC) scheme

The Multi Objective Service Consolidation (MOSC) scheme is build to allocate the VMs for the service components with multiple criteria. The VM mapping process is build with parameter selection based model. Resource conflict and availability parameters are used in the VM allocation process. The service consolidation process is performed with different service dependency categories.

6. Conclusion

Enterprise applications are build with independently deployable service components provided under the cloud environment. CloudScout approach is employed to discover the dependent service components and service consolidation process. The service dependency discovery process is improved with sequential dependency and complimentary dependency relationships. The service consolidation scheme is enhanced with multi objective functions and resource constraints. The service dependency analysis is used in the virtual machine allocation process. The service dependency analysis is performed with general, sequential and complimentary dependencies. The service consolidation is achieved with current resource status information. Network latency and resource cost are minimized in the system.

REFERENCES

[1] A. Beloglazov, J. Abawajy, R. Buyya, “Energy-aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing,” In Journal Future Generation Computing Systems, Vol 28 (5), pp. 755-768, 2012.

[2] A. Beloglazov, R. Buyya, “Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers,” Concurrency and Computation: Practice and Experience, Vol. 24, pp. 1397-1420, 2012.

[3] M. Bienkowski, A. Feldmann, D. Jurka, W. Kellerer, G. Schaffrath, S. Schmid, J. Widmer, “Competitive Analysis for Service Migration in VNets,” ACM SIGCOMM workshop on Virtualized Infrastructure Systems and Architectures (VISA), 2010.

[4] Nikos Tziritas, Cheng-Zhong Xu, Thanasis Loukopoulos, Samee Ullah Khan and Zhibin Yu, “Application-aware Workload Consolidation to Minimize both Energy Consumption and Network Load in Cloud Environments”, 2013.

[5] C. Curino, E. P. C. Jones, S. Madden, H. Balakrishnan, “Workload-Aware Database Monitoring and Consolidation,” ACM International Conference on Managing of Data (SIGMOD), 2011.

[6] E. Feller, L. Rilling, C. Morin, “Energy-Aware Ant Colony Based Workload Placement in Clouds,” IEEE/ACM International Conference on Grid Computing (GRID), 2011.

[7] G. B. Mertzios, M. Shalom, A. Voloshin, P. W. H. Wong, S. Zaks, “Optimizing Busy Time on Parallel Machines,” International Parallel and Distributed Processing Symposium (IPDPS), 2012.

[8] J. Sonnek, J. Greensky, R. Reutiman, A. Chandra, “Starling: Minimizing Communication Overhead in Virtualized Computing Platforms Using Decentralized Affinity-Aware Migration,” International Conference on Parallel Processing (ICPP), 2010.

[9] N. Tziritas, S. U. Khan, C.-Z. Xu, J. Hong, “An Optimal Fully Distributed Algorithm to Minimize the Resource Consumption of Cloud Applications”, IEEE International Conference on Parallel and Distributed Systems, 2012.

[10] Jianwei Yin, Xinkui Zhao, Yan Tang, Chen Zhi, Zuoning Chen, Zhaohui Wu,“CloudScout A Non-Intrusive Approach to Service Dependency Discovery”, IEEE Transactions On Parallel And Distributed Systems, 2017