www.ijiarec.com

ISSN:2348-2079

Volume-7 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

Big file transfer over multiple paths in cloud computing

A.Sinthiya, A.Gopalakrishnan

M.E. (Final year Student), Department of CSE College of Gnanaman

Asst.Prof /CSE, College of Gnanamani

ABSTRACT

We derive the tail distribution of the workloads for each server and the approximation for the tail sojourn times based on large deviation analysis. Furthermore, we optimize the cluster sizes that fulfill the requirements of target tail sojourn times. Extensive simulation experiments show very good matches to the derived analysis in a variety of scenarios, i.e., large numbers of servers experiencing a high number of different execution speeds, under various traffic intensities, workload variations and cluster sizes. Finally, we apply our proposed analysis on estimating the tail sojourn times of a Wikipedia system hosted in a private cloud, and the tested results strongly confirm the applicability and accuracy of our analysis. Our research here aims to address the challenging question of how to best dimension the size of a cluster deployed in a cloud, in terms of number of VMs experiencing varying execution speeds, so that the target value of tail sojourn times can be met. We particularly take an analytical perspective and focus on deriving the tail response times for any given cloud cluster size using various key system parameters.

INTRODUCTION

With ever-rapid development of social networking, science and e-business, enormous amounts of data, now frequently termed as 'big-file' is generated by these applications in the Internet recently. For instance, more than one Tara Byte of data is generated during many modern high energy physics experiments every day, such as Zero, etc.; some famous social network websites, e.g., Facebook, Twitter serve billions of page views every month, store a large number of new photos every month, and manage billions of pieces of content. Frequent data storage and transfer are indispensable to support data intensive operations, which results in high data processing costs. Recently, the emerging cloud computing has attracted considerable attention from both academia and industry, and is considered to be a powerful technology to perform large-scale and complex computing. In order to support cloud computing, efficient file transfer scheduling schemes are critical for file downloading or uploading. Sometimes, the files (related to different applications) are required to be transferred with different Quality of Service (QoS) [3]–[5], e.g., a tight latency constraint. This requires the network administrator to be capable of a global view and schedule the file transfer by computing feasible path set with efficient flows, when the time constraint with different level is considered. However, this is not available in the traditional IP networks since the centralized administration is not supported in the existing IP networks [1].

We propose two algorithms for addressing the SFTS problem: one is a heuristic (SFTS-H), and another one is an exact algorithm (SFTS-A). Especially, the SFTS-A adopts the auxiliary graph technique to construct a corresponding time expanded network which can solve the problem exactly. Simulation results show that both of the algorithms can solve the SFTS problem, at the cost of sacrificing some performance [2].

Author for correspondence:
Department of CSE College of Gnanaman,

We prove the flow computation issue can be addressed by solving a corresponding maximum flow over time problem. Especially in SFTS-A, we propose an approach based on Linear Programming (LP) technique for solving the maximum flow over time problem on the auxiliary graph. Moreover, we also propose a LP formulation for addressing the traffic distribution issue [3-5].

We propose a heuristic with an intelligent scheme to solve the MFTS issue. Our scheme can reconfigure the multi file flow at the right time when the transfer efficiency from the overall is considered. Simulation results show that our scheme can schedule the multi-file transfer efficiently and achieve high network utilization.

The *Constraints Validator* is used to filter the resource set (*Time space*) by ensuring that all application constraints are satisfied. The engine makes a *move* by choosing a resource from the *Time space*, and checking if that particular resource meets the application constraints; if the constraints are satisfied, then the *move* is valid. A solution path is formed by combining (chaining) each valid move . A set of solution paths (solution space), which satisfy the application constraints, will be generated. If none of the available resources satisfy the constraints, no solution is generated. A "degraded" solution path, which only partially satisfies the application constraints (e.g. with lower hardware specifications) could be offered. The use of "degraded" solution paths is not considered in this paper [6-8].

Map Reduce is emerging as an important programming model for large-scale data-parallel applications such as web indexing, data mining, and scientific simulation. Hardtop is an open-source implementation of Map Reduce enjoying wide adoption and is often used for short jobs where low response time is critical. Hardtop's performance is closely tied to its task scheduler, which implicitly assumes that cluster nodes are homogeneous and tasks make progress linearly, and uses these assumptions to decide when to speculatively re-execute tasks that appear to be stragglers.

In practice, the homogeneity assumptions do not always hold. An especially compelling setting where this occurs is a virtualized data center, such as Amazon's Elastic Compute Cloud (EC2). We show that Hadoop's scheduler can cause severe performance degradation in heterogeneous environments. We design a new scheduling algorithm, Longest Approximate Time to End (LATE),that is highly robust to heterogeneity. LATE can improve Hadoop response times by a factor of 2 in clusters of 200 virtual machines on EC2.

EXISTING SYSTEM

Recently, some high-performance networks (HPNs) have been emerging, which makes real-time big-file transfer and intelligent traffic engineering (TE) possible. Among these HPNs, SURFnet, UltraScience Net (USN), On-demand Secure Circuits and Advance Reservation System (OSCARS) of ESnet, and ION of Internet2 are anticipated to be deployed. Moreover, the emerging paradigm, Software-Defined Networking (SDN) is considered to be a pivotal technique which allows the network to be controlled in a systematic way, and SDN is able to coordinate the facilities in HPNs by a management framework, namely the control plane, for example, the Google B4 network is built based on SDN technologies. These HPNs allow network administrator to change the routing and bandwidth allocation dynamically, so that the QoS-enabled big file transfer can be achieved [9].

PROPOSED SYSTEM

First, we derive the workload distribution for hard to- analyze systems that capture the key characteristics of today's Server systems, i.e., renewal arrivals, highly varying job sizes, Markov-modulated execution speeds, processor sharing. Second, we develop an approximation scheme for the times, which are one of the critical SLA parameters, and further optimize the cluster size based on that. Last but not least, we evaluate our proposed analysis via extensive simulation and a small Wikipedia prototype system. Assuming the path set for each file transfer is obtained, we relate the multi-file flow scheduling problem to the **M**aximum **M**ulti-commodity **F**low over **T**ime (MMFT) problem, for solving the MFTS problem.

We conduct comparison between the proposed multipath transfer approach (Here, we select SFTS-H algorithm) and the single-path transfer approaches including the **W**idest- **S**hortest **P**ath routing (WSP) algorithm and **S**hortest- **W**idest **P**ath routing (SWP) algorithm. We evaluate their performance by comparing the probability of addressing the SFTS problem, i.e., the success-rate. Afterwards, we conduct performance comparisons between the SFTS-H and SFTS-A in terms of the aggregated-bandwidth, the number of the *s – Youtube Video Link* paths, the required total-delay and the running time [10].

PROPOSED APPROACH

We propose a heuristic with an intelligent scheme to solve the MFTS issue. Our scheme can reconfigure the multi file flow at the right time when the transfer efficiency from the overall is considered. Simulation results show that our scheme can schedule the multi-file transfer efficiently and achieve high network utilization.
It have proven the intermediate flow issues affect the computation of the optimal solution to the multi-commodity flow over time problem and the complexity of the intermediate-flow issues.

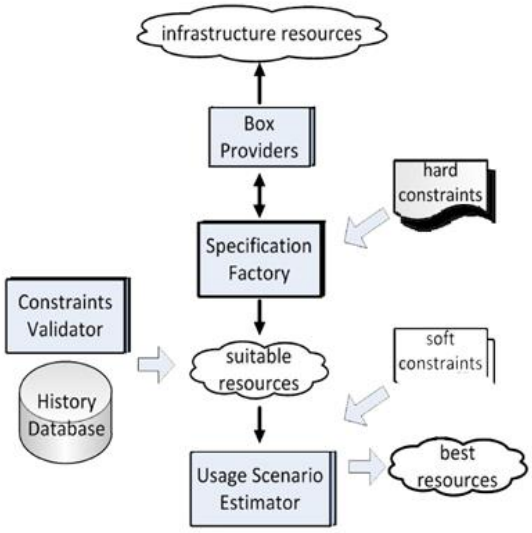
A mobile app consists of a central server that collects and processes data coming from multiple instances of client applications running on smart phones, tablets or smart devices with metering or remote-control capabilities. The data flows between these two sides of the app are conveyed by the FCM that provides messaging capabilities as topic-based publish/subscribe services. From the client side, the Firebase Messaging API and Android Studio 1.4 or higher with Gradle can be used to design and implement an application to send and receive notifications from FCM belonging to a given set of topics of interest.

Server Workload Analysis for Power Minimization using Consolidation

Server consolidation has emerged as a promising technique to reduce the energy costs of a data center. In this work, we present the first detailed analysis of an enterprise server workload from the perspective of finding characteristics for consolidation. We observe significant potential for power savings if consolidation is performed using off-peak values for application demand. However, these savings come up with associated risks due to consolidation, particularly when the correlation between applications is not considered.

We also investigate the stability in utilization trends for low-risk consolidation. Using the insights from the workload analysis, two new consolidation methods are designed that achieve significant power savings, while containing the performance risk of consolidation. We present an implementation of the methodologies in a consolidation planning tool and provide a comprehensive evaluation study of the proposed methodologies.

Business and temporal dependence in service processes are often found in multi-tier architectures and storage devices and must be captured accurately in capacity planning models as these features are responsible of significant performance degradations. However, existing models and approximations for networks of first-come first served (FCFS) queues with general independent (GI) service are unable to predict performance of systems with temporal dependence in workloads. To overcome this difficulty, we define and study a class of closed queuing networks where service times are represented by Markova Arrival Processes (MAPs), a class of point processes that can model general distributions, but also temporal dependent features such as business in service times.



METHODOLOGY

Allocation Resources

The multi-file multi-path transfer problem is usually translated into the well-known multi-commodity (file) flow over time problem. The multi-commodity flow or multi-commodity flow over time problem has been extensively studied and an effective tool for modeling routing issues in many telecommunication networks, e.g., traditional computer network, ad-hoc network, SDN-enabled network, etc. It has been proved that the multi-commodity flow or multi-commodity flow over time problem is weakly NP-hard even to the case of only two commodities. However, the MFTS problem becomes more challenging when the continuous flow stream (similar to the intermediate-flow issues) is considered, where the residual-flow in the network needs to be calculated to achieve exact scheduling.

Maximizing the Throughput for the Multi-File Flow

Assuming the path set for each file transfer is obtained, we relate the multi-file flow scheduling problem to the **M**aximum **M**ulti-commodity **F**low over **T**ime (MMFT) problem, for solving the MFTS problem.

A static flow called $\tau - length - bounded$ flow is proposed, to translate the MMFT problem to the static flow problem. In this way, the MMFT problem can be formulated based on the $\tau - length - bounded$ path set Pj , where Pj is equal

Simulations for SFTS-H and SFTS

We conduct comparison between the proposed multipath transfer approach (Here, we select SFTS-H algorithm) and the single-path transfer approaches including the **W**idest- **S**hortest **P**ath routing (WSP) algorithm [42] and **S**hortest- **W**idest **P**ath routing (SWP) algorithm [43]. We evaluate their performance by comparing the probability of addressing the SFTS problem, i.e., the success-rate. Afterwards, we conduct performance comparisons between the SFTS-H and SFTS-A in terms of the aggregated-bandwidth, the number of the s – *Youtube Video Link* paths, the required total-delay and the running time. Especially, the $k$max is set to 10 when executing the SFTS-H algorithm.

CONCLUSION

In this paper, we have studied the SFTS and MFTS problems in cloud computing. For SFTS problem, we adopted maximum flow over time issue and auxiliary graph technique, to propose a heuristic and an exact algorithm, respectively. Simulation results show that both of our proposed algorithms can solve the SFTS problem. Especially, the heuristic can solve the SFTS problem efficiently although the exact algorithm can achieve better QoS, e.g., the transfer delay. For

MFTS problem, we proposed a heuristic with an intelligent scheme which can maximize the throughput and schedule the multi-file flow dynamically, by solving the maximum multi-file flow over time problem. Simulation results show that our algorithm schedules the multi-file flow dynamically and can achieve high network utilization. Possible future work that can be derived from this work is designing traffic engineering strategy for files with different priorities.

FUTURE WORK

In the current design we are targeting the reservation of a single transfer window per request, i.e., of a window with constant bandwidth (height) and specific time duration (width). Given a busy network, there can always be a possible solution if such a window is scheduled into the future beyond current commitments. A more attractive solution is, of course, to allow for multiple windows per request, filling available reservation gaps. This problem is much more challenging to optimize and manage. We allowed for such a possibility by having multiple windows per request in the APIs, and plan to address this possibility in the future, after the single window strategy is fully implemented.

REFERENCES

[1]. "Dzero.". [Online]. Available: https://www-d0.fnal.gov/, 8, 2016.
[2]. "Facebook.". [Online]. Available: https://www. facebook.com/, 12, 2016.
[3]. S. Kandula, I. Menache, R. Schwartz, and S. R. Babbula, "Calendaring for wide area networks," ACM SIGCOMM Comput. Commun. Rev., 44(4), 2015, 515–526.
[4]. H. Zhang et al., "Guaranteeing deadlines for inter-data center transfers," IEEE/ACM Trans. Netw., 25(1), 2017, 579–595.
[5]. G. Han, L. Liu, S. Chan, R. Yu, and Y. Yang, "HySense: A hybrid mobile crowdsensing framework for sensing opportunities compensation under dynamic coverage constraint," IEEE Commun. Mag., 55(3), 2017, 93–99.
[6]. "Surfnet." [Online]. Available: ttps://www.surf.nl/en/aboutsurf/ subsidiaries/surfnet, 2016.
[7]. N. S. Rao, W. R. Wing, S. M. Carter, and Q. Wu, "Ultrascience net: Network testbed for large-scale science applications," IEEE Commun. Mag., 43(11), 2005, 12–17.
[8]. "ESnet, OSCARS: On-demand secure circuits and advance reservation system." [Online]. Available: http://www.es.net/oscars, 2016.
[9]. "Internet2.",. [Online]. Available: http://www.internet2.edu/ion, 2016.
[10]. S. Jain et al., "B4: Experience with a globally-deployed software defined WAN," ACMSIG COMM Comput. Commun.4 3(4), 2013, 3–14.