Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Uploading multiply deferrable big data to the cloud platform using cost-effective online algorithms

Baojiang Cui [a,b], Peilin Shi [a,b,*], Weikong Qi [c], Ming Li [c]

[a] School of Computer Science, Beijing University of Post and Telecommunications, Beijing, China
[b] Nation Engineering Laboratory for Mobile Network Security, China
[c] Institute of Telecommunication Satellite, CAST, China

HIGHLIGHTS

- A basic single-ISP case was analyzed and the MHSA was designed.
- MDSA was designed to optimize the cloud scene based on MHSA.
- The simulation experiments show total cost was reduced by 12% base on MDSA.

ARTICLE INFO

Article history:
Received 27 January 2016
Received in revised form
28 March 2016
Accepted 1 May 2016
Available online xxxx

Keywords:
Cloud computing
Data center
Multiple delay
ISP
Charge model
Bandwidth cost

ABSTRACT

Cloud computing consists of processing big data and provides convenient, on-demand network access to a shared pool of configurable computing resources. Cloud data center costs have become a hot topic in recent years. To minimize bandwidth costs, a better solution for uploading multiply deferrable big data to a cloud computing platform for processing using a MapReduce framework was studied. The multiply deferrable big data, which have its own delay window sizes, are produced by local cloud users, and the bandwidth charging model in this paper is the Max contract pricing scheme adopted by Internet service providers (ISPs). A basic single-ISP case was analyzed. We then extended the study to the cloud scene. The Multi-Heuristic Smoothing Algorithm for the single case was designed, and we proved that the worst-case competitive ratio of the Multi-Heuristic Smoothing Algorithm falls between $2(1-(1-1/D_{\max})^{D_{\max}})$ and 2, where $D_{\max}$ is the maximum delay window size. In addition, the Multi-Dynamic Self-Adaption Algorithm (MDSA) was designed to optimize the cloud scene based on the Multi-Heuristic Smoothing Algorithm. The simulation experiments demonstrated that the total cost was reduced by 12% when the Multi-Dynamic Self-Adaption Algorithm was adopted.

1. Introduction

When cloud computing was initially proposed, people called it into question; however, its extensive development has made it a hot topic in academia and in industry and has driven increasingly more experts to study it. Cloud Computing consists of both the applications delivered as services over the Internet and the hardware and system software housed in data centers that provide those services [1]. With the widespread popularity and rapid development of the information industry, data centers as information service providers have played an important role in the foundation, which includes the "islands of information" consolidation, server consolidation, and the integration of information technology and industrialization. With the rapid development of IT, network and business applications are increasingly moving toward the use of a large number of servers and storage clusters contained in large data centers, and large-scale data centers are rapidly expanding [2,3].

Cloud computing platform costs continue to increase the already high costs related to data centers' energy consumptions, and bandwidth management has received substantial attention. In recent years, the explosive growth in data and in the analysis of big data has resulted in the increased cost of cloud computing platforms. In addition, the bandwidth cost optimization of data centers has become a challenge and hot spot for cloud computing platform research in recent years. Cost optimization can result in annual savings of millions of dollars [4].

* Corresponding author at: School of Computer Science, Beijing University of Post and Telecommunications, Beijing, China.
E-mail addresses: cuibj@bupt.edu.cn (B. Cui), noahstone@163.com (P. Shi).

http://dx.doi.org/10.1016/j.future.2016.05.001
0167-739X/

The transmission of big data is priced by Internet service providers (ISPs). Many ISPs have adopted a charging approach called the 95th percentile charge model. In this model, the charge for the use of metrics is close to the corresponding peak for the charging month and excludes extreme cases, namely, eliminating the top 5% [5,6]. In this model, ISPs divide the charge period into many time slots such as every slot being 5 min in duration. In addition, the ISP determines the volume of every time slot used by users. The ISP chooses the 95th percentile in the charge period to represent the amount of bandwidth delivered to the users and charges accordingly [7,8]. To demonstrate exactly how this scheme works, let us consider the case of a 30-day month. To calculate usage, the ISPs measure the monthly bandwidth (consumed megabits in a 5-minute statistical interval; in this example, a total of 8640 statistical intervals) and then find the largest section, namely, 432 (95th 8640 percentile) [5,9]. Subsequently, ISPs can calculate the bandwidth cost based on this value. The monthly billing basis depends on the percentile charge scheme. In this paper, we consider a percentile model that remains in good agreement with infrastructure provisioning costs [9].

In the past, research based on the 0-percentile charge model has been limited to conditions wherein a job must be uploaded to the cloud when it is generated by the local user. However, in the real world, the uploading of data is not very urgent. Data upload tasks can be delayed for a period of time. For example, uploading of the biological information of the human genome project and astronomical observation data to the cloud computing platform can be postponed [9]. Without considering the delay of the data, the data transmission process will lead to an unreasonable higher peak under the MAX model if a large number of data tasks are uploaded during one time slot or during a very short period of time [10].

Because of the uncertainty of the arrival time of the data task, the greatest challenge is to design an optimal scheduling scheme toward achieving cost reductions. Currently, researchers can only design online algorithms based on the current state for smoothing of the volume of uploading tasks toward reducing the peak. Only [6,9] have studied uploading deferrable big data to the cloud in an attempt to reduce bandwidth costs. In [9], the author's research is based on a cloud platform that applies a MapReduce-like framework and uniform delay size for each workload, which are generated by local cloud users.

The research of Zhang can be divided into two main parts. The first part is for a single-ISP case, where the author studied and designed a Heuristic Smoothing Algorithm. The results of their experiments demonstrated that the peak volume of the Heuristic Smoothing Algorithm is 28% smaller than that of the Simple Smoothing algorithm. The second part of the research is in the cloud, where the author designed a Randomized Uploading Scheme (RUS) based on the Heuristic Smoothing Algorithm to accommodate cloud computing platforms with multiple ISPs. When the delay time is increased, more space will become available to smooth the peak, and the effect of the Randomized Uploading Scheme in terms of cost savings will be improved. In [9], the maximum delay time of data tasks is uniform. However, in the real world, the condition concerning the uniform delay size is idealistic; some task data can be delayed for upload to the cloud. In addition, the delay window size can be determined by some agreement, such as SLA or some other situation whereby, for example, the user may determine the window size based on their business requirements or a time limit. The delay time should be diverse because the actual data tasks are also diverse. In addition, the core idea of the Randomized Uploading Scheme design in [9] is that each mapper with one ISP is treated as a single case, and each workload is randomly assigned to the one mapper for processing. Clearly, the Randomized Uploading Scheme is not the most reasonable method. Based on the research of [9], this paper makes the following contributions to the field:

- This paper analyzes an optimization model for bandwidth costs under workloads with multiple delay sizes.
- We analyze a single case with multiple delay sizes and redesign the Simple Smoothing algorithm [9] to adapt to data tasks with multiple delay sizes. We prove that the competitive ratio of the Simple Smoothing algorithm is 2.
- Under the single case with multiple delay sizes, we build queues representing the scheme of each corresponding delay window size before analyzing the mixed sizes that contain all queues. In addition, inspired by the Heuristic Smoothing Algorithm, we prove that the optimized solution in multiple-delay-size cases is related to both the maximum delay size and the available space for each queue. Moreover, we design the Multi-Heuristic Smoothing Algorithm for obtaining improved performance. We prove that the competitive ratio of the Multi-Heuristic Smoothing Algorithm is between $2(1-(1-1/D_{\max})^{D_{\max}})$ and 2.
- To analyze the cloud case, first, we consider the optimization of a single delay size. We propose two improvements, whereby the first type of improvement is a simple anticipation, mainly to avoid some data center idle waste, and the second type is the Single-Dynamic Self-Adaption Algorithm, which is based on the cost function Φ for selecting the ISP. The cost function can be designed to be diversified according to the actual situation, thus increasing the flexibility of the Single-Dynamic Self-Adaption Algorithm. This paper simply designs a function based on the charges.
- We extend the Single-Dynamic Self-Adaption Algorithm to the Multi-Dynamic Self-Adaption Algorithm for use when considering multi-delay-size cases for the cloud. The algorithm can schedule data task uploading sequences according to the redesigned cost function, therein reducing the cost of the data center for the cloud computing platform.

In the remainder of this paper, we discuss related work in Section 2 and present the mathematical model in Section 3. The single case and cloud case are analyzed in Sections 4 and 5, respectively. The evaluation results are presented in Section 6, and we conclude the paper in Section 7.

2. Related work

Our work is mainly concerned with the scheduling sequence for uploading workloads toward reducing the bandwidth cost when subject to multiple delay sizes. Scheduling toward achieving reduced CPU energy consumption the work [11] has been studied by considering dynamically varying the clock speed when the operating system is monitoring the idle time to minimize the idle time. Weiser et al. studied three types of scheduling methods: OPT, FUTURE, and PAST algorithms. Ref. [12] further studied the power consumption in NOCs for reducing power consumption by first adopting closed-loop control. G. Chen et al. studied mapping applications into NOCs and choosing an energy-efficient path (between the memory and the processor) for data transfer [13]. The work [14] studied VLSI computation for the optimization of processor execution times and the author established a model under limited power conditions and demonstrated that minimizing the execution time under energy constraints is not equivalent to minimizing the operation count or computation depth. Similarly, a few other scholars have conducted investigations on the optimization of the network. N. Bansal provided online algorithms to maximize the throughput for jobs with arbitrary sizes and deadlines and optimized the total time and energy consumed [15]. M. Markovitch et al. studied online algorithms for maximizing the transaction capacity of delay-bounded traffic by designing the PROACTIVE GREEDY and GREEDY algorithms [16]. Ref. [17] provided DCLB to balance the network load and minimize bandwidth