Journal of Network and Computer Applications 59 (2016) 14–27

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

A new energy-aware task scheduling method for data-intensive applications in the cloud

Qing Zhao [a], Congcong Xiong [a,*], Ce Yu [b], Chuanlei Zhang [a], Xi Zhao [a]

[a] School of Computer Science and Information Technology, Tianjin University of Science and Technology, 300222 Tianjin, China
[b] School of Computer Science and Technology, Tianjin University, 300072 Tianjin, China

ARTICLE INFO

Available online 21 May 2015

Keywords:
Energy aware scheduling
Data-intensive application
SLA violation rate
Data correlation

ABSTRACT

Maximizing energy efficiency while ensuring the user's Service-Level Agreement (SLA) is very important for the purpose of environmental protection and profit maximization for the cloud service providers. In this paper, an energy and deadline aware task scheduling method for data-intensive applications is proposed. In this method, first, the datasets and tasks are modeled as a binary tree by a data correlation clustering algorithm, in which both the data correlations generated from the initial datasets and that from the intermediate datasets have been considered. Hence, the amount of global data transmission can be reduced greatly, which are beneficial to the reduction of SLA violation rate. Second, a "Tree-to-Tree" task scheduling approach based on the calculation of Task Requirement Degree (TRD) is proposed, which can improve energy efficiency of the whole cloud system by reducing the number of active machines, decreasing the global time consumption on data transmission, and optimizing the utilization of its computing resources and network bandwidth. Experiment results show that the power consumption of the cloud system can be reduced efficiently while maintaining a low-level SLA violation rate.

1. Introduction

With the arrival of Big Data, many applications with a large amount of data have been abstracted as scientific workflows and run on a cloud platform. Cloud computing has been envisioned as the next-generation computing paradigm because of its advantages in powerful computing capacity and low application cost (Buyya, 2008; Armbrust et al., 2010; Pedram, 2012). However, the growing quantity of cloud data centers has greatly increased the total energy consumption in the world, which has become a critical environmental issue because of high carbon emissions. On the other hand, high power consumption is also a big problem in terms of economic cost from the perspective of cloud service providers. Researchers (Qureshi et al., 2009) have found, a 3% reduction in energy cost for a large company like Google can translate into over a million dollars in cost savings. High energy consumption not only translates to high cost but also leads to high carbon emissions, which are not environmentally friendly. Therefore, the problem of energy-aware performance optimization has attracted significant attention.

In literature, many existing works (Gorbenko and Popov, 2012; Fard et al., 2012) have shown that a task scheduling strategy is crucial for the overall performance of cloud workflow systems' energy efficiency. The high ratio of under-loaded machines is the main reason for low energy efficiency. Researchers (Chen et al., 2008) have found that even during idle periods, most of today's servers consume up to 50% of their peak power. Virtualization technology (Barham et al., 2003) is a significant technology for improving the utilization of resources, and is also the technical foundation of cloud computing. Therefore, the virtual machine (VM) is the basic deployment unit in this paper. A reasonable task-scheduling strategy based on VM placement can make hosts work in a proper load so as to achieve the objective of high energy efficiency.

With the development of precision instrument technologies, many scientific research fields have accumulated vast amounts of scientific data, such as astronomy, meteorology, and bioinformatics. To solve the energy problem of these data-intensive applications, a special-task scheduling method will be presented in this paper. The scheduling principles can be derived according to the characteristics of the data-intensive application as follows:

1. Decrease network traffic through rational data layout and task scheduling.
   I/O operations are the most time consuming part for data-intensive applications in the cloud. A bad task scheduling strategy can increase the amount of data transmission greatly and would directly result in an increased task response time. Then, to satisfy the user's service-level agreement (SLA), the

* Corresponding author.

http://dx.doi.org/10.1016/j.jnca.2015.05.001
1084-8045/

cloud system would have to allocate more computing resources to applications to decrease their time consumptions on computing. An indirect result is that the energy consumption on the server side would be increased.

On the other hand, frequent data transmission will also lead to a large amount of power consumption. As shown in (Cavdar and Alagoz, 2012), network devices consume up to 1/3 of the total energy consumption (excluding cooling equipments), and network conflicts are the main reason for this high consumption. The frequent and large amounts of data movement will inevitably lead to the prolonging of the total network transmission time consumption and an increase in the risk of network conflict.

Therefore, we believe it is important to reduce the amount of data transmission for data intensive applications in the cloud. Fortunately, there are data dependencies between tasks, and reasonable data placement and task scheduling based on these dependencies can decrease the amount and time consumption of data transfers. This is the first optimization objective of this paper.

2. Improve the utilization of servers in the cloud, and reduce the generation of inefficient energy.

As mentioned above, the underloaded machine is the main cause of low energy efficiency. Therefore, if the utilization of one machine is low, two strategies can be performed. One is to allocate more tasks to this host in order to improve its resource utilization situation. The other is to migrate the tasks on it to other machines so as to make it closed or turn to sleep state.

On the other hand, the overloaded status also needs to be changed. This is because the error rate will increase greatly under the condition of overload, hence lead to a high increase of power consumption. In the literature (Srikantaiah et al., 2009), the optimal CPU utilization of today's servers is about 70% in terms of energy efficiency. Therefore, making the active servers work at a balanced energy efficient utilization rate, and turning the underloaded servers off should be an intelligent strategy. This is another optimization objective of this paper.

For the conveniences of the readers, the symbols defined in this paper are illustrated in Table 1.

The remainder of the paper is organized as follows. Section 2 presents related works. Section 3 builds the user workflow model. Section 4 builds the cloud environment model. Section 5 shows our energy consumption model. Section 6 gives the detail of the task scheduling method. Section 7 presents and analyzes the simulation results. Finally, Section 8 addresses conclusions and future work.

2. Related works

The great amounts of energy consumed by supercomputers and computing centers have been a major resource and environmental concern facing today's society. In data centers, large amounts of energy are wasted by leaving computing and networking devices—such as servers, switches, and routers—powered on in a low

Table 1
Symbols illustration.

$\lvert D \rvert$	The number of global data items
$d_i(1 \le i \le \lvert D \rvert)$	The i th data item
$D_{in}(t)$	The set of input data items of the task t
$Size_i$	The size of the data item d_i
$\lvert S \rvert$	The number of physical servers in the cloud environment
$s_j(1 \le j \le \lvert S \rvert)$	The j th physical server
$SC^j_{storage}$	The storage capacity of server s_j
D_j	A set of data items stored on server s_j
$VM_x(1 \le x \le m)$	The x th type of VMs in the cloud platform
VC^x_{cpu}	The type of VM VM_x.
VC^x_{mem}	The memory c the CPU capacity of apacity of the type of VM VM_x.
t_k	The k th task
$WCET^k$	The Worst-Case execution time of the task t_k runs on a specific type of VM
$T^k_{deadline}$	The deadline restrict of the whole application the task t_k belongs to
$b_{i,j}$	The network bandwidths between server $s_i$and servers_j
SC^j_{cpu}	The CPU capacity of the physical server s_j
SC^j_{mem}	The memory capacity of the physical servers_j
$n^j_{VM_x}$	The number of the type of VM VM_xthat are allocated into the server s_j.
$Util^j_t$	The CPU utilization rate of $s_j$at time t
Opt_j	The optimal utilization level in terms of performance-per-watt for the server s_j
$EffecUtil^j_t$	The effective CPU utilization of the server s_j at the time t
$n^j_{VM_x\_runningTask}(t)$	The number of VM_xthat is running task on the server s_j at the time t
$\lvert D_{in}(t_k) \rvert$	The number of data items in the input set of the task t_k
$c^{initialData}_{i,j}$	The data correlation between$d_i$and$d_j$derived from the initial data items.
$c^{int\ ermediateData}_{i,j}$	The data correlation between$d_i$and$d_j$derived from the intermediate data items.
$c_{i,j}$	The integrated correlation between $d_i$andd_j.
T_i	The set of tasks which need the data item $d_i$as its input data
TRD	Task Requirement Degree
$OOED$	On-off expectation degree at time t
$TaskMargin_j(t)$	How much additional workload is needed to increase the machine's utilization to Opt_j in order to improve the energy efficiency of this server
$aveUtil_p$	The average CPU utilization in the period t
$Power^j$	The overall power consumption of the server s_j
$Power^j(t)$	The power consumption of the server s_j at time t.
$Total\_Power$	The total power consumption the cloud system