Future Generation Computer Systems 36 (2014) 91–101

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

A new multi-objective bi-level programming model for energy and locality aware multi-job scheduling in cloud computing

Xiaoli Wang, Yuping Wang*, Yue Cui

School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi, 710071, China

HIGHLIGHTS

- We propose a new multi-objective bi-level programming model based on MapReduce to improve energy efficiency of servers.
- The relationship between performance and energy consumption of severs is taken into account in the proposed model.
- Data locality can be adjusted dynamically according to current network state.
- A new effective multi-objective genetic algorithm based on MOEA/D is proposed to solve the above large-scale scheduling model.

ARTICLE INFO

Article history:
Received 14 December 2012
Received in revised form
13 August 2013
Accepted 3 December 2013
Available online 28 December 2013

Keywords:
Energy aware
Data locality
Multi-job scheduling
Cloud computing
MapReduce

ABSTRACT

How to reduce power consumption of data centers has received worldwide attention. By combining the energy-aware data placement policy and locality-aware multi-job scheduling scheme, we propose a new multi-objective bi-level programming model based on MapReduce to improve the energy efficiency of servers. First, the variation of energy consumption with the performance of servers is taken into account; second, data locality can be adjusted dynamically according to current network state; last but not least, considering that task-scheduling strategies depend directly on data placement policies, we formulate the problem as an integer bi-level programming model. In order to solve the model efficiently, specific-design encoding and decoding methods are introduced. Based on these, a new effective multi-objective genetic algorithm based on MOEA/D is proposed. As there are usually tens of thousands of tasks to be scheduled in the cloud, this is a large-scale optimization problem and a local search operator is designed to accelerate convergent speed of the proposed algorithm. Finally, numerical experiments indicate the effectiveness of the proposed model and algorithm.

1. Introduction

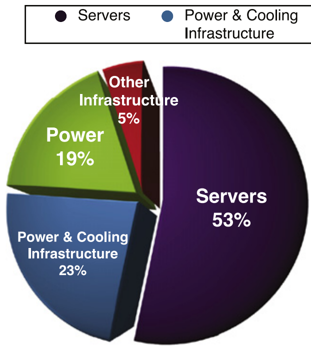
As cloud computing [1] platforms are growing in popularity, soaring power usage of data centers has drawn increasing attention. Reducing energy consumption will not only cut down the operational costs of data centers, but also reduce the amount of greenhouse gases emissions. It is estimated that data centers consumed approximately 1.5% of all electricity worldwide in 2011, which was about 56% higher than that of the preceding five years [2]. What is more, according to Amazon's CEMS project [3], energy-related cost amounts to 42% of the total budget as shown in Fig. 1, inclusive of both direct power consumption (19%) and the investment of the supporting infrastructure for cooling and power distribution (23%). Yet average data center energy efficiency is merely 50% [4].

Among all the approaches trying to reduce energy consumption of data centers, the most direct and intuitive one is to decrease energy consumed by supporting systems, including power distribution equipment and cooling systems. First, an overall power distribution loss only accounts for 8% of the total energy consumption for a data center with a PUE (Power Usage Effectiveness) of 1.7. That is to say, even with better technology, the reduction will not exceed 8% [3]. Second, Google's "free cooling" mode sets a successful example of reducing the energy consumption of cooling systems. It removes heat from servers by evaporating water or low temperature ambient air [5]. Although "free cooling" mode has proved to be useful, it has a key prerequisite that providers must have sufficient financial and technical strength to run several data centers around the world and the data should be backed up across them with seamless migration of computing loads.

Powering off idle devices when possible is regarded as another way of reducing energy consumption, especially during off-peak traffic hours. There exist a great number of solutions in the literature, which basically can be divided into two categories: designing energy-proportional servers or networks [6] and establishing energy aware virtualization over servers [7]. (1) Engineers from Google try to design energy-proportional servers that consume energy in proportion to the amount of work performed as they

* Corresponding author. Tel.: +86 15829630075.
E-mail addresses: wyp9000@yahoo.com.cn, ywang@xidian.edu.cn (Y. Wang).

0167-739X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.future.2013.12.004



	cost	%
Servers	2,997,090	53%
Power & Cooling Infrastructure	1,296,902	23%
Power	1,042,440	19%
Other Infrastructure	284,686	5%

Fig. 1. Monthly cost of Amazon's data center.

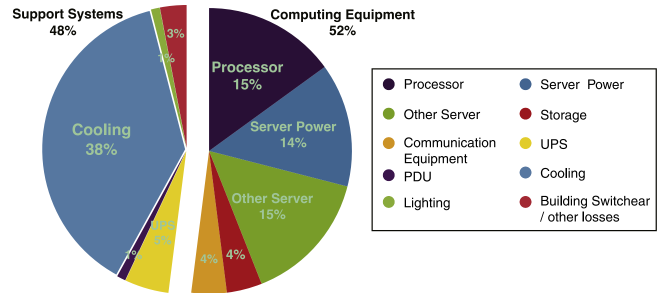


Fig. 2. Energy consumption within a data center.

noticed that even an energy efficient server still consumes about half its full power when doing virtually no work [8]. Meanwhile, [9] proposes energy proportional networks whose power consumption is more proportional to the amount of traffic it is moving. (2) Virtualization abstracts away the details of physical hardware and provides virtualized resources for high-level applications [10–12]. Services that only need a small fraction of computational resources can be virtualized and run within a virtual machine (VM). Several VMs with low resource utilization can run on a single hardware unit. Therefore, unused servers can be hibernated or turned off to save energy. However, the virtualization process will result in huge energy consumption because VMs are repeatedly created, terminated, cloned or moved from one host to another host [13,14].

The last approach of equal importance is to improve servers' energy efficiency. Fig. 2 shows how energy is used within a data center according to Emerson Network Power's analysis [15]. It was found that energy consumed by servers accounts for 52% of the total consumption, while support systems consume the remaining 48%. Furthermore, every Watt of savings that could be achieved on servers created approximately 2.84 W of savings in all. Therefore, it becomes critical to put forward an effective way to improve servers' energy efficiency. Certain literature [16,17] tries to achieve this goal by adjusting servers' CPU through task-scheduling strategies based on a given data deployment.

Different from the previous approaches, in this paper, first, appropriate task-scheduling strategies together with reasonable data placement policies are designed first; then, the relationship between performance and energy consumption of servers is taken into account; furthermore, task data locality can be adjusted according to current network state and cloud workload. Based on these, a new energy and locality aware multi-job scheduling model is proposed to further improve servers' energy efficiency. As the basic knowledge, MapReduce framework and Apache Hadoop are introduced in Section 2. Section 3 describes the energy and locality aware multi-job scheduling problem in mathematics, followed by relevant definitions and notations; next comes its corresponding large-scale integer bi-level optimization model. In order to solve this model, an effective genetic algorithm based on specific-design genetic operators is presented in Section 4. Simulation experiments are given in Section 5. Finally, the conclusions are made in Section 6.

2. Background knowledge

MapReduce, Google's massive data processing framework, rapidly processes vast amounts of data in parallel [18]. It finishes the computation by mapping and reducing data under cluster environment. Many real world jobs are expressible in this model, such as large-scale machine learning problems, large-scale graph computations, extracting specific properties from web pages of new experiments and products, etc.

In the MapReduce framework, any application is specified by jobs. For one job, the input files are first split into M pieces. Then M *map* tasks and N *reduce* tasks will be scheduled to process these data splits. It is worth noticing that network bandwidth is a relatively scarce resource in cloud computing environment. The master will attempt to schedule a *map* task on a server that contains a replica of its corresponding input split. This task-scheduling scheme based on data location is referred to as data locality, which can avoid a large-scale data movement.

Apache Hadoop is an open-source software framework that supports data-intensive distributed applications and implements MapReduce [19]. In Hadoop MapReduce framework, each worker has a number of slots for *map* tasks and *reduce* tasks. For example, a worker node may be able to run 100 *map* tasks and 100 *reduce* tasks simultaneously. Every active *map* or *reduce* task takes up one slot. For the Hadoop default scheduler, the scheduling of tasks is very simple: for a *reduce* task, the default scheduler simply takes the next in its list of yet-to-be-run *reduce* tasks, since there are no data locality considerations; for a *map* task, however, it takes account of the data location and picks a task whose input split is as close as possible to the worker node. In the optimal case, the task is data-local, that is, running on the same node that the split resides on. Some of the tasks are not data-local and they have to retrieve their data from a different rack from the one they are running on. Typically, each job would use the whole cluster, so jobs had to wait their turn.

There is also a multiuser scheduler called the Fair Scheduler. It aims to give every user a fair share of the cluster capacity over time. If a single job is running, it gets all of the cluster. As more jobs are submitted, free task slots are given to the jobs in such a way as to give each user a fair share of the cluster. A short job belonging to one user will complete in a reasonable time even while another user's long job is running, and the long job will still make progress. Jobs are placed in pools, and by default, each user gets their own pool. A user who submits more jobs than a second user will not get any more cluster resources than the second, on average. It is also possible to define custom pools with guaranteed minimum capacities defined in terms of the number of map and reduce slots.

3. A new multi-objective bi-level programming model

It is known that server's resource utilizations, especially for CPU and hard disk utilizations, exert a direct influence on its energy efficiency. Certainly, server's energy efficiency may also be affected by other resources such as memory, bandwidth, etc., but in order to simplify the model in the first place, only the most weighted resources are considered here. The problem of how to improve servers' energy efficiency, however, cannot be solved simply by balancing loads among servers so that each server's resource utilization reaches 100%, which is because we have to optimize both the performance of servers and energy savings.

The following gives a more detailed description of the relationship between the performance of servers and energy savings. When server's resource utilizations are low, idle power is not amortized effectively and hence the level of energy efficiency turns