Taming Power Peaks in MapReduce Clusters

Nan Zhu*
Dept. of Computer Science
Shanghai Jiaotong University
Shanghai, China
zhunan@sjtu.edu.cn

Lei Rao*
School of Computer Science
McGill University
Montreal, Canada
leirao@cs.mcgill.ca

Xue Liu
School of Computer Science
McGill University
Montreal, Canada
xueliu@cs.mcgill.ca

Jie Liu
Microsoft Research
Microsoft Corp.
Redmond, USA
jie.liu@microsoft.com

Haibin Guan
Dept. of Computer Science
Shanghai Jiaotong University
Shanghai, China
hbguan@sjtu.edu.cn

ABSTRACT

Along with the surging service demands on the cloud, the energy cost of Internet Data Centers (IDCs) is dramatically increasing. Energy management for IDCs is becoming ever more important. A large portion of applications running on data centers are data-intensive applications. MapReduce (and Hadoop) has been one of the mostly deployed frameworks for data-intensive applications. Both academia and industry have been greatly concerned with the problem of how to reduce the energy consumption of IDCs. However the critical power peak problem for MapReduce clusters has been overlooked, which is a new challenge brought by the usage of MapReduce. We elaborate the power peak problem and investigate the cause of the problem in details. Then we design an adaptive approach to regulate power peaks.

Categories and Subject Descriptors

H.m [**Information Systems**]: Miscellaneous; D.0 [**Software**]: General

General Terms

Algorithms, Design, Performance

1. MOTIVATION AND PROBLEM STATEMENT

The energy consumption of Internet Data Centers (IDCs) is an important aspect of data center efficiency. It is reported [4] that IDC operators pay extremely large bill for the energy consumption for their IDCs. As computational needs increase, this trend will continue or even be intensified. Recently both academia and industry are greatly concerned about the problem of energy management for IDCs. Among these research, regulating high power peaks have received considerable attentions. With high power peaks, Power Distribution Units (PDUs) and other power provisioning infrastructures should be subscribed with high level configurations, which greatly increase the capital cost for building and operation of IDCs. Power capping technique is a regulated way to enable additional machines to be hosted and prevent overload situations in IDCs [3]. MapReduce [2] is one of the mostly deployed frameworks for large scale data-intensive applications. However the usage of MapReduce brings new challenges for IDC energy management problems. As we will show in this paper, the decision of when and where to schedule the map and reduce tasks significantly affects the peak power in IDCs.

We observed this problem from Microsoft Production Servers. The power consumption of computing servers running on MapReduce framework does not keep in a stable level and temporary peaks appear at certain moments. We use the Hadoop official software [1] to elaborate this problem and replicate the phenomenon similar to the production system. We elaborate the power peak problem on a 200-nodes Hadoop cluster. The power consumption of the cluster is shown in Figure 1 (More details are in Section 2). The power consumption curve starts at around 10 kilowatts, which is close to the idle power of the nodes in the cluster. It takes about 210 virtual minutes for the cluster to finish the calculation of the workload. During the simulation time, the curve surges at a few time intervals. According to the power consumption distribution, there are less than 8% time intervals where the cluster is operating close to its peak power. If we could remove these few intervals we might be able to further increase the number of machines hosted within a given power budget or decrease the power budget to a lower value, which can both greatly improve the energy efficiency of the IDCs. In this paper, we systematically study the cause of these temporary power peaks in MapReduce clusters and propose an adaptive approach to address the power peak problem.

2. ANALYSIS ON THE POWER CONSUMPTION AND SCHEDULING

In this section, we elaborate the power peak problem on a 200-nodes Hadoop cluster, in which each node has 8 processing cores with the frequency of 2.4 GHz. We use the workload trace file in Mumak provided by Yahoo!.

We first show that the scheduling of the map and reduce tasks affects the power peaks of the Hadoop cluster. We trace both the power consumption and the number of arrived

*The first two authors contribute equally for the work in this paper.

SIGCOMM'11, August 15–19, 2011, Toronto, Ontario, Canada.
ACM 978-1-4503-0797-0/11/08.

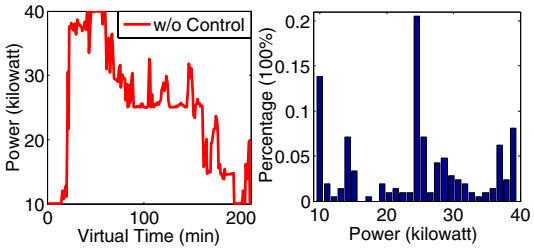

Figure 1: Power Consumption and its Distribution of the Hadoop Cluster

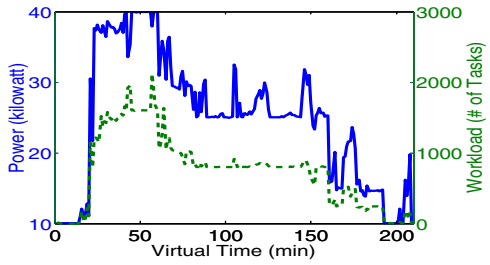

Figure 2: Power Consumption vs. Workload Scheduling Along with Virtual Time in Mumak

tasks with the sampling frequency of 1 minute and present the power consumption and workload scheduling trace in Figure 2. We can observe that the power consumption surges with the increase of the arrived task number for most cases. There are some interesting points in the figure. At the 44th virtual minute, the power consumption goes down when the workload surges. This is because the sizes of jobs at the 42nd and 43rd minutes are large and during these two minutes the system utilization is very high. At the 44th virtual minute, these large size jobs have left the system and only a number of jobs with small sizes have been invoked. The system utilization becomes lower. It is important to remind that the power consumption is linear in the processor utilization (for fixed processor frequency) and is not directly related to the workload arrival rate.

From the above observations and corresponding analysis, we can conclude that the default scheduling in the Hadoop system leads to poor system performances: at a few time intervals the cluster is operating at high power, and during most of the total time interval the cluster operates at low power. This indicates that the system utilization is very low. It is desirable that the system is utilized more evenly with the high power to be relative lower.

3. THE DESIGN OF ADAPTIVE APPROACH

The design of our adaptive approach mainly consists of two modules: The model building module and the controller module. The model building module dynamically estimates the input-output model of the Hadoop cluster. The input is the workload arrival rate and the output is the power of each node in the Hadoop cluster. Based on the dynamic model, the controller module adjusts the input in order to regulate power peaks.

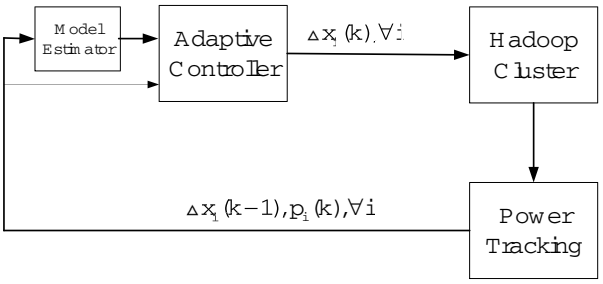

Figure 3: Architecture of the Adaptive Approach

The architecture of the adaptive approach is shown in Figure 3. We consider a general power consumption model of each node i in the Hadoop cluster as following

$$p_i(k) = A_i p_i(k-1) + B_i \Delta x_i(k), \quad (1)$$

where A_i and B_i are the unknown system parameters and these parameters may vary due to the varying workload; Δx_i represents the change to the arrival rate threshold for node i. For $k = 1, 2...,$ the kth control point represents the time k. At each time point, the model estimator gets the change to the workload arrival rate threshold Δx_i form the adaptive controller and the real nodes' power from the power tracking. Based on them, the model estimator calculates the system parameters. We use the recursive least square (RLS) estimator with exponential forgetting to identify the system parameters A_i and B_i for all node i. Then the model estimator sends the updated system parameters to the adaptive controller. The adaptive controller decides the system input, i.e., the workload arrival rate for each node in the Hadoop cluster, according to the power budget.

To achieve the control objective with power capping, we define the following cost function for each server i:

$$J_i(k) = \left(p_i(k+1) - p_i^{cap}(k+1)\right)^2. \quad (2)$$

This stands for the differences between the consumed power and the power cap value should be as small as possible. It can ensure the power consumed is limited around the expected level. We want to optimize the above cost function so as to regulate the power consumption of the Hadoop cluster.

4. REFERENCES

[1] Hadoop. *http://hadoop.apache.org/*.

[2] J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *OSDI 04'*, pages 137–150, 2004.

[3] X. Fan, W.-D. Weber, and L. A. Barroso. Power provisioning for a warehouse-sized computer. In *ISCA*, 2007.

[4] J. Koomey. Worldwide electricity used in data centers. *Environmental Research Letters*, Sept. 2008.