Computers & Industrial Engineering 117 (2018) 249–261

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Efficient jobs scheduling approach for big data applications

Yanling Shao[a,c], Chunlin Li[a,*], Jinguang Gu[b], Jing Zhang[a], Youlong Luo[a]

[a] Department of Computer Science, Wuhan University of Technology, Wuhan 430063, PR China
[b] Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan University of Science and Technology, PR China
[c] College of Computer and Information Engineering, Nanyang Institute of Technology, Nanyang 473000, PR China

A R T I C L E I N F O

Keywords:
Big data
Dynamic scheduling
Energy efficiency
MapReduce
Resource allocation

A B S T R A C T

The MapReduce framework has become a leading scheme for processing large-scale data applications in recent years. However, big data applications executed on computer clusters require a large amount of energy, which costs a considerable fraction of the data center's overall costs. Therefore, for a data center, how to reduce the energy consumption becomes a critical issue. Although Hadoop YARN adopts fine-grained resource management schemes for job scheduling, it doesn't consider the energy saving problem. In this paper, an Energy-aware Fair Scheduling framework based on YARN (denoted as EFS) is proposed, which can effectively reduce energy consumption while meet the required Service Level Agreements (SLAs). EFS not only can schedule jobs to energy-efficiency nodes, but also can power on or off the nodes. To do so, the energy-aware dynamic capacity management with deadline-driven policy is used to allocate the resources for MapReduce tasks in terms of the average execution time of containers and users request resources. And then, Energy-aware fair based scheduling problem is modeled as multi-dimensional knapsack problem (MKP) and the energy-aware greedy algorithm (EAGA) is proposed to realize tasks fine-grained placement on energy-efficient nodes. Finally, the nodes which have been kept in idle state for the threshold duration are turned off to reduce energy costs. We perform extensive experiments on the Hadoop YARN clusters to compare the energy consumption and executing time of EFS with some state-of-the-art policies. The experimental results show that EFS can not only keep the proper number of nodes in on states to meet the computing requirements but also achieve the goal of energy savings.

1. Introduction

Although the big data applications and analytics have made great progress on large cluster within cloud datacenter, there are still some challenges in processing data intensive applications (Chen, Mao, & Liu, 2014). Among them, how to reduce the energy costs on large clusters is a key one. By the US Environmental Protection Agency report (Brown et al., 2007), it shows that cloud data center energy consumption accounted for 2% of total US power grid, and will be doubled every five years. By Google engineer's study (Duffy, 2011), the rate of CPU utilization is less than 50%, and a large number of nodes are in a low load state which is near 75% of their span time. Since the prices of server are constantly going down, undoubtedly the energy costs will increase its proportion in the total costs for the cloud data center in the future. Therefore, it is critical issue to minimize energy consumption for the data centers meanwhile satisfy service requirements of customers.

In addition, the characteristic of the web-based big data applications are also known to vary dynamically with time. Here, web application is defined as the submission of new job is done dynamically from the web interface. Due to unexpected external happenings for the coming applications, how to provide proper resources takes a considerable role for fair resource allocation. Dynamic resource allocation technologies (Li et al., 2012; Warneke & Kao, 2011; Zhang, Li, & Wu, 2014) are proposed to manage online big data application with changing workloads.

Taking into account the energy consumed, for cloud server provider, it is not in their best interests to execute the services as fast as they can in order to minimize the makespan, and they are obligated to support service by user's specified SLA (e.g. jobs' deadline). Therefore, in order to save energy while meet user SLA, we design an energy-aware scheduling model and algorithm, furthermore, turning on/off nodes strategy is adopted to realize dynamically node management in this paper.

1.1. Our contribution

In this paper, we address dynamic resource management and scheduling problem for online big data applications in a shared and multi-tenant large cluster. We model node resources scheduling and

* Corresponding author.
E-mail address: chunlin74@tom.com (C. Li).

https://doi.org/10.1016/j.cie.2018.02.006
Received 21 March 2017; Received in revised form 11 December 2017; Accepted 6 February 2018
Available online 08 February 2018
0360-8352/

assume that each application is allocated a certain fraction of resources. By adopting a combination of online measurement monitor, our techniques can dynamically update the node resource, based on (i) its SLA (deadline) requests, (ii) the node performance on dealing with specific Mapreduce (Apache Hadoop, 2016) task. The main goal of our techniques is to allocate and schedule resources for big data applications in an efficient energy way while meeting users' SLA. Specifically, this paper owns two specific contributions as the followings.

(1) In the energy-aware fair scheduling framework with dynamic node managment, there are two ways to save energy: one is energy-aware resources scheduling; the other is turning off the idle node which has been kept for a specified duration.
(2) We model the energy-aware MapReduce jobs scheduling as multi-dimensional knapsack problem (MKP), the energy-aware greedy algorithm (EAGA) is proposed for fine-grained task placement to minimize the energy consumption while meet user SLA.

1.2. Related work

A few researchers have addressed the issues of saving energy and meeting users' SLA. The related work is summarized from three aspects: *dynamic resource management techniques*, *MapReduce scheduling on job execution* and *energy saving in data centers.*

1.2.1. Dynamic resource management techniques

The objective of dynamic resource management is to scale up/down cluster capacity with changing workload. Due to high startup costs, it is an existed challenge which has resulted in many strategies, including predictive, reactive and reactive-predictive hybrid approaches. Chandra, Gong, and Shenoy (2003) addressed dynamic resource allocation problem for web applications running on shared data centers and used system architecture with prediction and resource allocation techniques. Wu, Hwang, Yuan, and Zheng (2010) present a new adaptive hybrid method (AHModel) for load prediction guided by trained confidence windows in a distributed Grid environment. Dinda and O'Hallaron (2000) compared MA, AR-MA, ARIMA, and ARFIMA (due to self-similarity) linear prediction model and showed AR was sufficient for load prediction on a single CPU host. Prevost, Nagothu, Kelley, and Jamshidi (2011) characterized the ability of neural network and autoregressive linear prediction algorithms to forecast loads in cloud data center applications. Song, Baek, Hong, and Jang (2005) employed fuzzy regression analysis in the short-term load forecasting problem with good accuracy for power systems. These papers used predictive method to estimate the request rate. Le and Pang (2013) proposed an augmented discrete event control and a max-throughput-min-energy reactive scheduling framework. Nathuji, Kansal, and Ghaffarkhah (2010) used online feedback to capture performance interference interactions and build a multi-input multi-output (MIMO) model to perform closed loop resource management. These two researches adopted reactive policy to deal with resource allocation problem. Chen, Arlitt, Marwah, and Gandhi (2011) used hybrid resource provisioning IT resources: predictive control provisions the base workload at coarse time scales (e.g. Hours or days)) and reactive control handles any excess demand at finer time scales (e.g., minutes). Heilig, Lalla-Ruiz, and Voß (2016) proposed an efficient Biased Random-Key Genetic Algorithm (BRKGA) for solving the CRMP based on a real-time cloud brokerage mechanism that supports consumers to select an appropriate set of cloud resources from multiple cloud providers. In this work, we employed turning on/off node strategy to deal with dynamic resource allocation for online big data applications.

1.2.2. MapReduce scheduling on job execution

Tang, Lee, and He (2014) presented Dynamic Hadoop Slot Allocation (DHSA) technique by slot-based model to argue MapReduce system suffering from poor performance. Verma, Cherkasova, and Campbell (2011) proposed a MapReduce job scheduler that allocated the resources to production jobs by profiling a job that runs routinely and uses its profile to estimate request resources for meeting the deadline. Chen, Liu, and Xiao (2014) developed maximum cost performance (MCP) for dealing with the straggler problem, which improves the effectiveness of speculative execution significantly. Yao, Wang, Sheng, Lin, and Mi (2014) proposed HaSTE scheduler, which can effectively decrease the makespan of MapReduce jobs in YARN considering the information of requested resources, capacities and dependency between tasks. Nightingale, Chen, and Flinn (2006) showed that speculator substantially improved the performance of existing distributed file systems. Zaharia, Konwinski, Joseph, Katz, and Stoica (2010) discussed speculative execution problem in heterogeneous MapReduce environments and designed a robust scheduling algorithm, Longest Approximate Time to End (LATE), which uses estimated finish times to speculatively execute the tasks that hurt the response time. Pastorelli, Barbuzzi, Carray, Dell'Amico, and Michiardi (2013) proposed a size-based technique to schedule jobs by evaluating online job size without wasting resources. However, these papers evaluated execution performance of jobs, without considering energy efficiency as their targets. While these algorithms focus on performance, the scope of our work is saving energy. He, Cai, Deng, Meng, and Wang (2016) designed a queuing-oriented optimizing scheduling algorithm (QTJS) to improve the resource utilization in Cloud. Mashayekhy, Nejad, Grosu, Zhang, and Shi (2015) presented two heuristic algorithms (EMRSA-I and EMRSA-II) which assigns the map/reduce tasks with suitable slots by profiling technology in order to minimize energy while meeting users SLA. But the paper only focused a single MapReduce job scheduling for saving energy. Our research pays attention to multiple online big data jobs and addresses the energy-aware fair scheduling problem.

In Nita, Pop, Voicu, Dobre, and Xhafa (2015) proposed a multi-objective scheduling algorithm (named MOMTH) of many independent MapReduce tasks under the constraints of deadline and budget. Compared with MOMTH, the proposed approach focuses on task assignment to energy efficiency nodes centralized in order to turn off idle node for saving energy. Wang and Shi (2014) discussed the problem of scheduling a batch of MapReduce jobs as a workflow with budget and deadline constraints. Our work pays attention to online energy-aware MapReduce jobs scheduling and nodes management on Hadoop YARN cluster.

1.2.3. Energy saving in data centers

Until now, the strategies for energy saving in data centers can be divided into three categories: power on/off nodes, virtual machine and DVFS (Dynamic voltage and frequency scaling) policy. Leverich and Kozyrakis (2010) exploited the replication of MapReduce framework and presented covering set (CS) strategy to allow scale-down of operational frequency by changing the data placement policy in a Distributed File System (DFS). Lang and Patel (2010) proposed All-In Strategy (AIS) on energy management in the MapReduce cluster that the cluster's nodes state were transitioned between "all-nodes-on" and "all-nodes-off" considering utilization of resources. These works primarily exploited data layout in these two extreme techniques, whereas our work focuses on fine-grained resources management of node to save energy by scheduling right accurate number nodes to MapReduce jobs. Gandhi, Harchol-Balter, and Raghunathan (2012) introduced dynamic capacity management policy (named AutoScale) that added or removed servers as needed in Multi-Tier data center caused by the unpredictable, time-changeable current load while meeting response time SLAs. Their focus was in a multi-tier data center. Our work considers a shared and multi-tents large cluster. Cardosa, Singh, Pucha, and Chandra (2012) proposed spatio-temporal tradeoff VM placement policy on batch and online processing scenarios, further dynamically changed cluster size to continuously improve energy efficiency in homogeneous MapReduce cluster. Hsu, Slagter, Chen, and Chung (2014) proposed an energy-aware task consolidation (ETC) technique in order to minimize energy