http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (17): 3964-3969, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.3964.3969

A new Strategy of Resource Management for Cloud Computing

[1]Ping Guo, [1]Lijiang Ning, [1]Liping Su and [2]Lingling Bu
[1]College of Computer Science, Chongqing University, 400044, Chongqing, China
[2]Changan Ford Automobile Co., Ltd., 400010, Chongqing, China

Abstract: Resource management is one of core technology of cloud computing and it directly affects the availability, reliability and security of cloud platform. Hadoop is a rather mature cloud platform and it takes the Slot as the unit for resource management. In this study, we propose a fine-grained resource management strategy by taking CPU, memory, disk and network bandwidth as different kinds of recourse. By applying this strategy, the cloud platform can manage and schedule the resource in a more flexible way according to the actual requirements of the assigned task. Thereby, it will improve the utilization of the resource and optimize the overall performance of the cloud platform.

Key words: Cloud computing, resource management, fine-grained, hadoop

INTRODUCTION

Cloud computing is a new computing mode which enables people to attain dynamic scalable computational resources in internet service way (Chen and Zheng, 2009). In recent years, cloud computing has developed much rapidly in foreign and domestic countries. The major internet companies in the world supply better service to users by deploying their own cloud computing platform. Nowadays, the cloud computing platform divides into two kinds, one is commercial cloud computing platform such as EC2, Windows Azure, Google App Engine, Blue cloud, Force.com, HP Cloud System, Big cloud and so on. The other is open source cloud platform such as Hadoop, Eucalyptus, abiCloud, StormGrind, Vertebra, Apache Nuvem and so on. Each of them has its own advantage and characteristic.

The cloud computing platform processes huge available resources. Highly efficient resources management is not only one of core technology but also a challenge during the development of cloud computing and it directly affects the availability, reliability and security of cloud platform (Luo *et al*., 2011).

In this study, a fine-grained strategy of resource management for cloud computing based on hadoop is proposed. The article is divided into four sections except introduction. In section one, it introduces relevant background knowledge. In section two, a fine-grained strategy of resource management for cloud computing based on hadoop is proposed and the performance of system adopting proposed strategy is analyzed. In section three, the experiments are executed and the results of experiments are analyzed. In section four, we summarize our work and present a future outlook.

HADOOP AND ITS RESOURCE MANAGEMENT

The study of this study is based on Hadoop which is a rather mature and extensive cloud computing platform (White, 2012). In this section, we analysis resource management system of Hadoop and seek a reasonable breakthrough point to future study.

Hadoop: The Apache Hadoop project develops open-source software for reliable, scalable, distributed computing. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures. Apache Hadoop's MapReduce and HDFS components originally derived, respectively from Google's MapReduce and Google File System (GFS) papers (Ghemawat *et al*., 2003; Chang *et al*., 2008; Lammel, 2008).

Hadoop consists of the Hadoop Common package which provides filesystem and OS level abstractions, a MapReduce engine and the Hadoop Distributed File System (HDFS). Hadoop adopts master-slaves distributed structure and HDFS provide data for MapReduce just like Fig. 1.

Corresponding Author: Ping Guo, College of Computer Science, Chongqing University, 400044, Chongqing, China

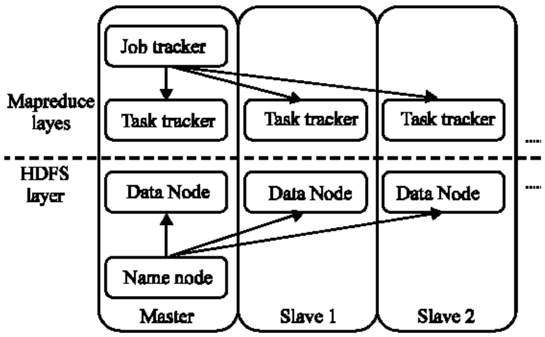

Fig. 1: Architecture of hadoop

A small Hadoop cluster includes a single master and multiple worker nodes. The master node consists of a JobTracker, TaskTracker, NameNode and DataNode. A slave or worker node acts as both a DataNode and TaskTracker, though it is possible to have data-only worker nodes and compute-only worker nodes. These are normally used only in nonstandard applications. In a larger cluster, the HDFS is managed through a dedicated NameNode server to host the file system index and a secondary NameNode that can generate snapshots of the namenode's memory structures, thus preventing file-system corruption and reducing loss of data. Similarly, a standalone JobTracker server can manage job scheduling. In clusters where the Hadoop MapReduce engine is deployed against an alternate file system, the NameNode, secondary NameNode and DataNode architecture of HDFS is replaced by the file-system-specific equivalent.

Hadoop's resource management: Hadoop's resource management system adopts centralized management model. All the relevant information of available resources is obtained by monitor and saved in NameNode. All he Tasks and available resources are managed uniformly by the central scheduler in the Jobtracker. Jobs submitted by users were added into job queue and be sorted according to the priority of the job. TaskTrackers communicate with Jobtracker periodically. At the same time they report the resources occupancy and be ready for accepting the instruction from Jobtracker (Sandholm and Lai, 2010; Thirumala and Reddy, 2012).

The slot is the basic unit of resources in hadoop. In fact, slot is a combination of various resources. Each TaskTracker processes a fixed number of slots and the number depends on the memory, CPU, the value of mapred.child.java.opts in hadoop.

Slot in hadoop is divided into Map Slot and Reduce Slot. Each TaskTracker allocates a slot configuration parameter according to its own memory, CPU and the parameter is non-modifiable after the hadoop start (Shih *et al.*, 2012; Wolf *et al.*, 2010). The central scheduler allocates a task to TaskTracker when it has a suitable unoccupied slot. Map Task can be executed only when it obtain the unoccupied Map Slot, as well as Reduce Task. If all of slots were occupied, the task had to wait in the task queue. In hadoop, a job submitted by users is divided in several independent tasks. Figure 2 show the job's assignment strategy.

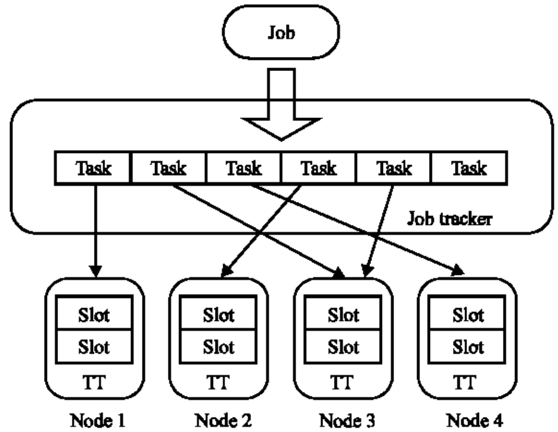

Fig. 2: Job's assignment strategy

Deficiencies in hadoop's resource management: There still some deficiencies Hadoop's resource management after we analyze its fundamentals and a large numbers experiments in hadoop. Deficiencies are mainly focuses on the following aspects:

- No. of Slots is non-dynamic
- Map Slot and Reduce Slot are not mixed-using
- One task only be executed in one slot
- Workload is not considered when TaskTracker apply tasks
- There are too many configuration parameters

FINE-GRAINED STRATEGY OF RESOURCE MANAGEMENT

Here, introduces a fine-grained strategy of resource management so as to avoid the preceding inconveniences mentioned in section one. By applying strategy, the cloud platform manages and schedules the resource in a more flexible way according to the actual requirements of the assigned task (Babu,2010). Thereby, it will improve the utilization of the resource and optimize the overall performance of the cloud platform.

System architecture: The job types under the fine-grained Strategy of Resource Management: The nodes of a Hadoop cluster make data interaction by interconnected network. The network bandwidth becomes a performance bottleneck. Identically, different jobs require unequal-sized memory and unequal-sized CPU. Therefore, jobs submitted by users can be divided into four types. They are compute-intensive, data-intensive, memory-intensive, bandwidth- intensive.

When selecting the node for a task, different types of jobs have different selection criteria. For example, it takes CPU idle rate as the first selection criteria for compute-intensive jobs. It takes memory idle rate as the first selection criteria for memory-intensive.

Types of resources: Hadoop's resources can be divided into three major types. They are CPU, memory, bandwidth. In order to describe expediently, the following definitions were given.

The CPU main frequency, the CPU cores and the CPU utilization rate were separately represented by S, H and U_C.

The memory size and the memory utilization rate separately represented by M and U_M.

The data receive rate, the data packet receive rate, the data send rate and the data packet send rate were separately represented by R_b, R_p, S_b and S_p.

The resources of node i is Wi. Wi = $\{S, H, U_C, M, U_M, R_b, R_p, S_b, S_p\}$.

The resources of the whole cluster were represented by W. W = $\{W_1, W_2, \ldots, W_n\}$

The required resources of a task are represented by R. It includes the types of jobs, CPU, memory and bandwidth and note as R = (T, C, M, N).

Resource monitoring: There are a number of nodes in cloud computing platform. The resource usage is time-varying. The bill of resource R shows in Table 1.

Strategy of resource allocation: It adopt strategy of task requirement driving. In other words, when there is an uncompleted task in job queue, the task is assigned to a TaskTracker by Namenode automatically. The Namenode use the strategy according to the bill of resource R. It can shorten response time with higher service efficiency.

The following strategies were considered when a TaskTracker was selected:

- Data locality. There are two kinds of locality that are sometimes distinguished: Temporal locality means that the program reuses the same data that it recently used. Spatial locality means that the program uses data close to recently accessed locations. It is of course possible for a program exhibit both types of locality at the same time. Good data locality is essential for good application performance
- Selecting node which possess the most remaining resources. For example, it selects a node whose UM is a minimum for a memory-intensive
- Selecting node which is best in history

The fine-grained strategy of resource management for cloud computing based on hadoop is proposed and its algorithms are given as Fig. 3 shows. The input is R = (T, C, M, N); W = $\{W_1, W_2, \ldots, W_n\}$ and output is an optimal node.

System performance analysis: Resource management affects the availability, reliability and security of cloud platform. An excellent strategy of resource management for cloud computing can improve utility of resource, shorten response time of jobs and improve the system throughput. Contrastive analysis of the two strategies is discussed from the following two aspects.

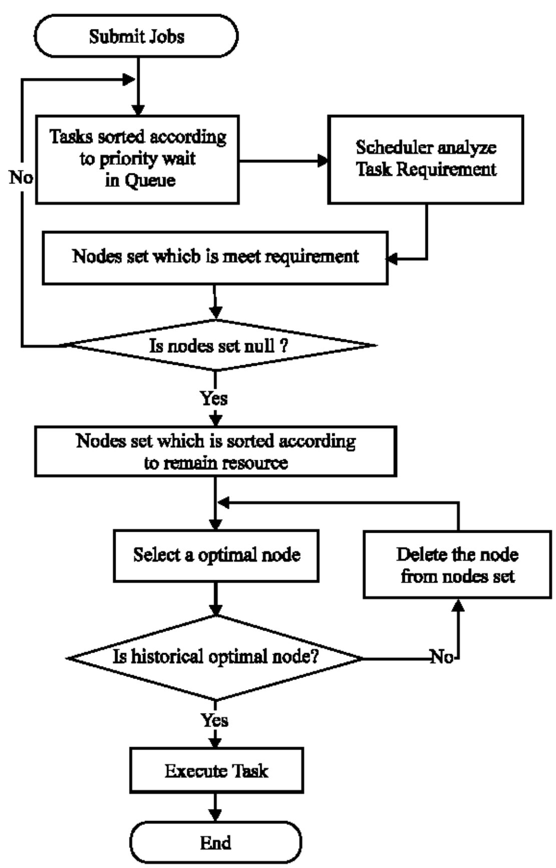

Fig. 3: Flow chart of resource management

First situation: the task requirement is less than Slot. They spend same time finishing jobs when there are not enough tasks. The tasks are being executed less than the fixed number of slot in unit time which waste resources. However, under the fine-grained strategy of resource management, more tasks can be executed in one node. It can improve utility of resource, shorten response time of jobs and improve the system throughput.

The second situation: the task requirement is not less than Slot. One task not can be accomplished only by using a slot the time growing and unoccupied slot couldn't be fully used. Under the fine-grained strategy of resource management, one task can be accomplished by making full of resources. As well, it can improve utility of resource, shorten response time of jobs and improve the system throughput.

EXPERIMENTAL RESULTS AND ANALYSIS

Here, Achieving high performance of system has been verified by experiments.

Experimental environment: The experimental environment includes 50 nodes, data block: 64 M, number of data replication: 3, CPU cores of each node: 4, memory size of each node: 4 G, bandwidth of network: 10 m sec^{-1}. scheduling strategy: FIFO and Slot: <CPU: 1, Mem: 512M>. Test cases show in Table 2.

Analysis of experiment results: The experiment results show the time for accomplishing various compute-intensive jobs under different workload. The jobs' size, or CPU tasks required is materialized by the value of α. The workload is materialized by the value of λ. The workload and λ is a negative correlation.

Figure 4-6 show the experiment results. The results show the fine-grained strategy can shorten accomplished time of jobs on heavy workload.

Table 2: Test cases

Groups	Tasks	Average of tasks	Jobs	Required CPU	Required time (sec)
1	1	1	20	α	10
2	2	2	11	α	10
3	3~20	10	7	α	10
4	21~60	50	5	α	10
5	61~150	100	3	α	10

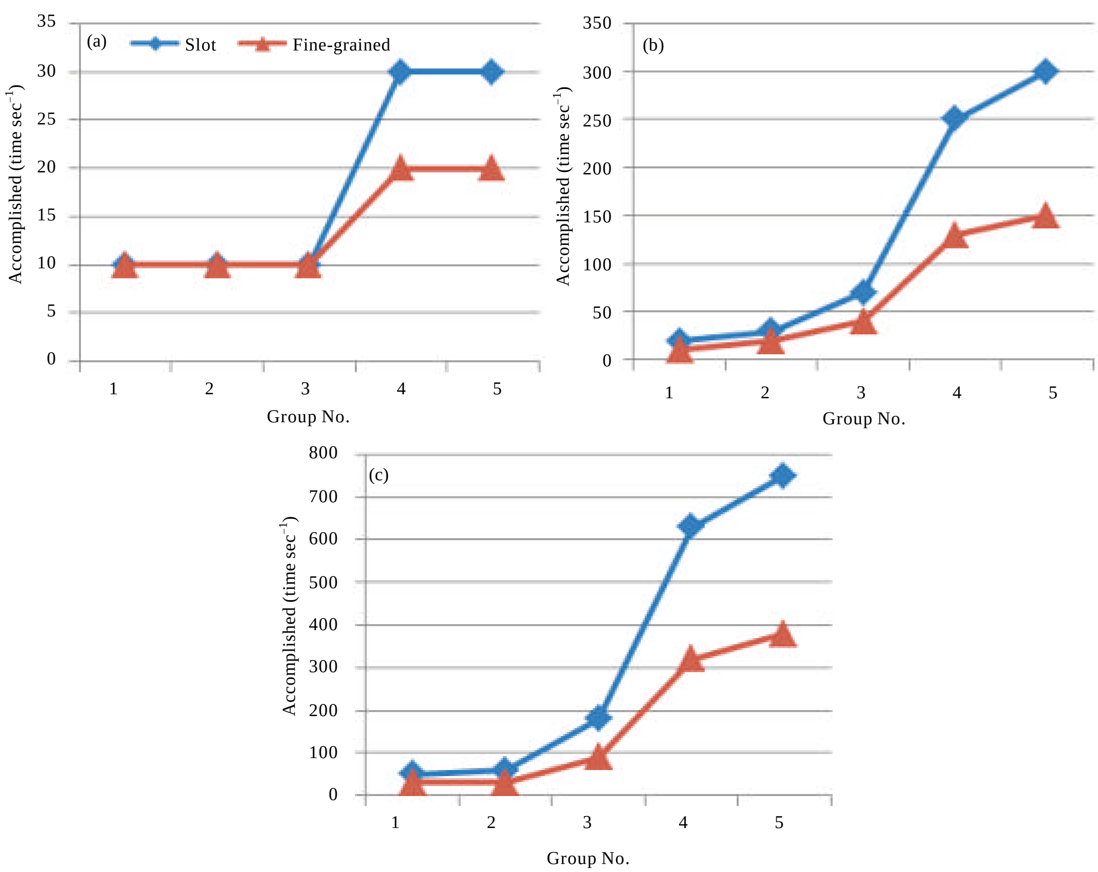

Fig. 4(a-c): Accomplished time (α = 0.5) (a) λ = 0.5, (b) λ = 0.05 and (c) λ = 0.02

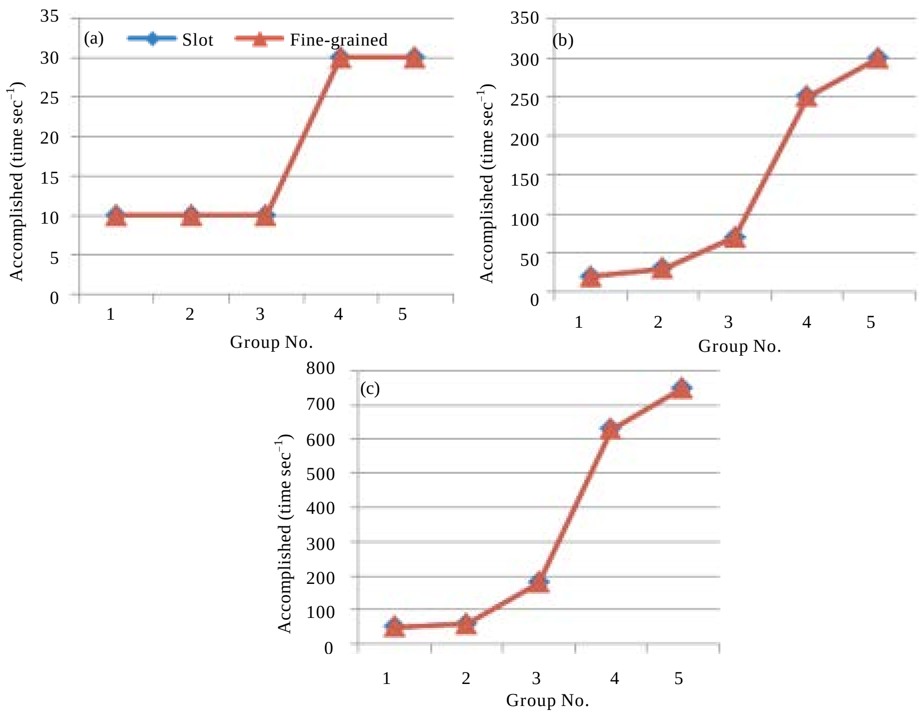

Fig. 5(a-c): Accomplished time (α = 1) (a) λ = 0.5, (b) λ = 0.05 and (c) λ = 0.02

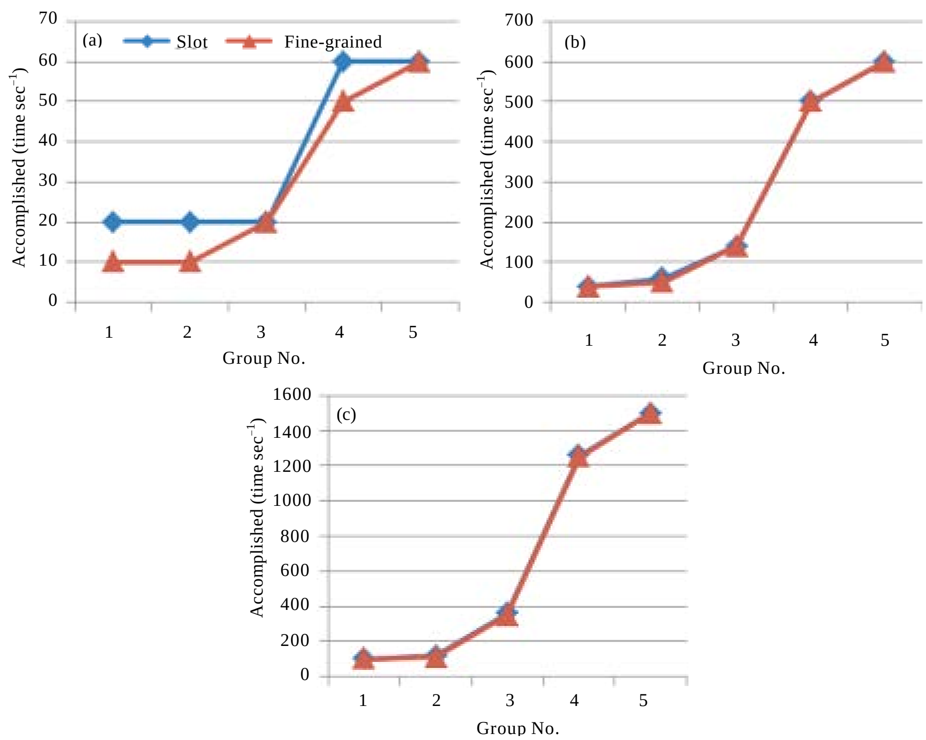

Fig. 6(a-c): Accomplished time (α = 2) (a) λ = 0.5, (b) λ = 0.05 and (c) λ = 0.02

SUMMARY

In the study, a fine-grained strategy of resource management for cloud computing based on hadoop is proposed. It can achieve high performance of system on heavy workload. It can improve utility of resource, shorten response time of jobs and improve the system throughput. In particular, it has more advantages to the little job.

REFERENCES

Babu, S., 2010. Towards automatic optimization of MapReduce programs. Proceedings of the 1st ACM symposium on Cloud Computing, Indianapolis, June 10-11, 2010, ACM, New York, pp: 137-142.

Chang, F., J. Dean, S. Ghemawat, W.C. Hsieh and D.A. Wallach *et al*., 2008. Bigtable: A distributed storage system for structured data. ACM Trans. Comput. Syst., Vol. 26, No. 2. 10.1145/1365815. 1365816

Chen, K. and W.M. Zheng, 2009. Cloud computing: System instances and current research. J. Software, 20: 1337-1348.

Ghemawat, S., H. Gobioff and S.T. Leung, 2003. The google file system. Proceedign of the 19th ACM Symposium on Operating Systems Principles, October 19-22, 2003, ACM, Lake George, NY., pp: 29-43.

Lammel, R., 2008. Google's MapReduce programming model-revisited. Sci. Comput. Program., 70: 1-30.

Luo, J.Z., J.H. Jin, A.B. Song and F. Dong, 2011. Cloud computing: Architecture and key technologies. J. Commun., 32: 3-21.

Thirumala, R.B.T. and L.S.S. Reddy, 2012. Survey on improved scheduling in hadoop mapreduce in cloud environments. Int. J. Comput. Appl., 34: 29-33.

Sandholm, T. and K. Lai, 2010. Dynamic proportional share scheduling in hadoop. Proceedings of the 15th International Workshop Job Scheduling Strategies for Parallel Processing, April 23, 2010, Atlanta, GA, USA., pp: 110-131.

Shih, H.Y., J.J. Huang and J.S. Leu, 2012. Dynamic Slot-based task scheduling based on node workload in a MapReduce computation model. Proceedings of the International Conference on Anti-Counterfeiting, Security and Identification, August 24-26, 2012, IEEE, National Taiwan University of Taipei, Taiwan, pp: 1-5.

White, T., 2012. Hadoop: The Definitive Guide. 3rd Edn., O'Reilly Media Inc., USA.

Wolf, J., D. Rajan, K. Hildrum, R. Khandekar and V. Kumar *et al*., 2010. Flex: A Slot allocation scheduling optimizer for mapreduce workloads. Proceedings of the ACM/IFIP/USENIX 11th International Middleware Conference, Bangalore, India, November 29- December 3, 2010, Springer, Berlin Heidelberg, pp: 1-20.