ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

A Survey on Fine-Tuning MapReduce Slot Configuration for Hadoop Clusters

P. Ramarajpandiyan[1], R. Dharmaraj[2]

PG Scholar, Department of CSE, Srividya College of Engineering & Technology, Virudhunagar, India [1]

Associate Professor, Department of CSE, Srividya College of Engineering & Technology, Virudhunagar, India [2]

Abstract: The MapReduce is an open source Hadoop framework implemented for processing and producing distributed large Terabyte data on large clusters. Its primary duty is to minimize the completion time of large sets of MapReduce jobs. Hadoop Cluster only has predefined fixed slot configuration for cluster lifetime. This fixed slot configuration may produce long completion time (Makespan) and low system resource utilization. Our proposed scheme is to allocate resources dynamically to MapReduce tasks. It can be done by following slot ratio configuration between map and reduce tasks, by updating the workload information of recently completed tasks. Many scheduling methodologies are discussed that aim to improve completion time goal.

Keywords: MapReduce, Makespan, Workload, Dynamic Slot Allocation.

I. INTRODUCTION

Cloud is an emerging technology that provides various storage and compute services to its consumers. MapReduce is a programming model, especially for processing, distributed and parallel big data processing. Hadoop MapReduce framework is mostly used for writing big data applications. It is implemented in a number of cloud providing companies such as Amazon, Hortonworks, Facebook, Yahoo, and so on.

A classic Hadoop cluster has a single name node and multiple data nodes. The name node, which is configured with job tracker, is responsible for job scheduling and job execution co-ordination. Each data node configured with task tracker, which manages MapReduce slots. Hadoop has a static slot configuration, which means a fixed number of map slots and reduce slots which are only used for processing map reduce tasks. Map tasks can run by map slots, and reduce tasks can run in reduce slots. This static slot configuration may lead to poor performance and low resource utilization.

Apache Hadoop components are responsible for running large data sets. Main Hadoop parallel processing components are Hadoop Distributed File System (HDFS), Hadoop YARN, and Hadoop MapReduce.

We propose dynamic slot configuration, which dynamically allocates slots for map and reduce tasks. Our aim is to modify name node functionality, which means to increase additional responsibility for monitoring workload information, dynamic slot assignment, and scheduling. Also, we need to modify the task tracker slot allocation policy to dynamically allocate tasks to MapReduce tasks without any slot specification. We can make use of map task slots (map slots) to reduce slots and vice versa. The main idea behind dynamic slot configuration is to avoid idle slot in the MapReduce slots.

The job tracker estimates the current workloads in each task tracker using workload monitoring component. The slot assigner component decides the optimum slot for assigning tasks. The schedulers are used to schedule the tasks in the data nodes.

The task tracker sends the status report to the job tracker for every 3 minutes. Failure tasks are assigned to the next nodes based on this status report. The job tracker is always monitoring the task execution and slot assignment.

The resources are allocated to map and reduce tasks by job tracker based on different job schedulers and resource allocation policies. Various schedulers are used that include FIFO, capacity, SLO, task schedulers, fair scheduler.

These schedulers follow different resource allocation strategies that include the Longest Approximation Time to End, delay, resource aware, deadline constraint, epoch based, moldable, malleable, fair4s job scheduling to improve MapReduce completion time and Hadoop performance.

The purpose of this study is to analyze the various slot configurations, advantages, and disadvantages of all schedulers and also different resource allocation policies in MapReduce.

The rest of the paper is ordered as follows. In section 2, we described related work. In section 3, we focused an evaluation methodology. In section 4 and 5, it is clear that how the research works is examined. In section 6, observation of research questions is answered. In section 7, we conclude.

II. RELATED WORK

In literature, there was research study on performance optimization of Hadoop MapReduce jobs. An essential way for upgrading the performance of a MapReduce job is dynamic slot configuration and job scheduling. J. Polo et al. [2] calculated the map and reduce task completion time

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

dynamically and update it every minute during job execution. Task scheduling policy was based on the priority of each job. Priority was estimated based on the concurrent allocation of jobs. The dynamic scheduler is pre-emptive. It affects resource allocation of low priority jobs. J. Wolf et al. [3] implemented flexible scheduling allocation scheme with Hadoop fair scheduler. A primary concern is to optimize scheduling theory metrics, response time, makespan, stretch, and Service Level Agreement. They proposed penalty function for measurement of job completion time, epoch scheduling for partitioning time, moldable scheduling for job parallelization, and malleable scheduling for different interval parallelization.

J. Dean et al. 2008 [1] discussed MapReduce programming model. The MapReduce model performs operations using the map and reduces functions. Map function gets input from user documents. It generates intermediate key/value for reducing function. It further processes intermediate key/value pairs and provide output key/value pairs. At an entry level, MapReduce programming model provided the best data processing results. Currently, it needs to process the large volume of data. So it provides some consequences while processing and generating data sets. It takes much execution time for task initialization, task coordination, and task scheduling. Parallel data processing may lead to inefficient task execution and low resource utilization.

Verma et al. [5] proposed deadline aware scheduler, called SLO scheduler. The SLO scheduler takes decisions of job ordering and slot allocation. This scheduler's primary duty is to maximize the utility function by implementing the Earliest Deadline First algorithms. It measures how many numbers of slots required for scheduling the slots dynamically with a particular job deadline. B. Sharma et al. [7] proposed a global resource manager for the job tracker and a local resource manager for the task tracker. A global resource manager function is to manage each MapReduce task. It processes resource needs and resource assignments for each task. A local resource manager's duty is to identify each task. It examines resource usage and task completion time of the task. It deals with detecting bottlenecks with resources and resource contention.

Apache Hadoop released next generation MapReduce, called YARN [8]. It replaces MRv1 fixed slot configuration. YARN deals with CPU cores and memory requirements. It splits the job tracker into two components; they are resource managements and job scheduling. MapReduce tasks assignment is based on CPU cores and memory requirement of each task. YARN users simply update their MRv1 by installing mrv2 compatibility API and recompile the MRv1 application. J. Wang et al. [9] proposed fair slot setting for dynamically allocate available slots to particular tasks. They used FRESH for static and dynamic slot configuration. The static slot configuration slots are allocated before cluster launch based on previous task execution records. It uses deduct workload function to update current workloads of running jobs in the cluster. The fair scheduler was proposed to achieve fairness metric. The dynamic slot assignment slots are allocated during task execution. It used Johnson indices to represent the level of fairness.

S. Tang et al. [11] proposed three techniques to improve MapReduce performance. They categorized utilized slot into the busy slot and idle slot respectively. The primary concern is to increase the number of the busy slots and decrease number of idle slots. Dynamic Hadoop Slot Allocation observes idle map and reduce slots. DHSA allocated the task to the unallocated map slots for overflowed reduce slots. Speculative Execution Performance Balancing provides performance upgrade for a batch of jobs. It gives the highest priority to failed tasks and next level priority to pending tasks. The slot prescheduling improves the performance of slot utilization with data locality without any negative effects on fairness metric. A.U. Patil et al. [13] discussed scheduling algorithms in the MapReduce environment. They analyzed schedulers and scheduling policies. The default FIFO scheduler follows the First in First Out queue for schedules the job. A single job is divided into a small number of chunks called tasks. The FIFO queue allocates tasks to free slots presented in the task tracker. The fair scheduler provides the fair share of resources to cluster users. Capacity scheduler estimates the number of users sharing cluster resources and focus fair allocation of resources to users. The primary concern is to maximize the throughput and utilization of entire cluster. Scheduling policies include the Longest Approximation Time to End, Deadline constraint, delay scheduling, resource aware, and Fair4s scheduling.

Z. Liu focused [14] partition skew problem. Data skewness causes the problem in execution time for larger and smaller tasks. Commonly this can be raised while partitions are unevenly distributed by the hash function. They proposed a new architecture called DREAMS. It predicts partition size and estimates reduce task performance metrics like CPU and memory impacts. Reduce phase performance model also detects the relationship between partition size and task execution time. After completion of reduce task performance estimation, DREAMS allocates resources to tasks.

Y. Yao et al. [15] proposed Tunable knob for reducing the Makespan of MapReduce (TUMM) for dynamic slot configuration. They modified the job tracker functionality by adding additional components. Main components are workload monitor and slot assigner. The workload monitor collects information about running and completed workloads. The slot assigner finds the best slot for dynamically assigning MapReduce tasks in the task tracker. They used FIFO schedulers for both static and dynamic slot configuration. They also introduced slot configuration for homogeneous and heterogeneous clusters. For the heterogeneous environment, H_TUMM slot assignment algorithm was implemented. Authors used work count, histogram rating, classification, inverted index, and grep jobs for experimental results.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

III. EVALUATION METHODOLOGY

The survey focused on various schedulers and scheduling policies for allocating resources for tasks. Dynamic slot allocation was proposed by many researchers. Recently TUMM [15] proposed for dynamic slot configuration in homogeneous Hadoop clusters and H_TUMM proposed for heterogeneous Hadoop clusters. But they used FIFO schedulers for both concepts. We analyzed more about schedulers and its disadvantages in this survey. A literature study has been made on the different schedulers and scheduling policies that focuses dynamic slot configuration and resource allocation are shown in table 1.

AUTHOR & YEAR	TITLE	METHODOLOGY	DISADVANTAGES
J. Dean et al., 2008	MapReduce: simplified data processing on large clusters	MapReduce provides distribution and automatic parallelization of large data sets in high-performance large clusters. MapReduce model gets a set of input key/value pairs and produces a reduced output key/value pairs. Two functions are used in the MapReduce library. map(String key, String value) reduce(String key, Iterator values) In MapReduce, a master is responsible for allocating jobs to map and reduce function. Master follows some data structures to represent map/reduce state such as idle, in-progress, completed.	It is not suitable for processing a large number of short online transactions. Intermediate key generation needs to interact to all the mapper process. A large amount of execution time is used in task initialization, co-ordination, and task monitoring. Misconfiguration of parallel processing parameter may lead to inefficient execution time and low resource utilization.
J. Polo et al., 2010	Performance-driven task co-scheduling for MapReduce environments	Multiple applications need same resources to complete their workloads. The workloads may be different such as simple, almost interactive, executions, complex. Task scheduler duty is to select a task from multiple jobs. It predicts currently running job performance of concurrent MapReduce tasks. Using this strategy application can meet their performance without wasting physical resources **Proposed Concepts:** Job performance estimation Task scheduling **Components:** scheduling policy Task Scheduler	The dynamic scheduler is pre-emptive scheduling. It interrupts one job execution in order to allocate resources to high priority jobs. So low priority jobs execution can be affected by higher priority jobs. It is an issue in reduce phase.
J. Wolf et al. 2010	Flex: A slot allocation scheduling optimizer for MapReduce workloads	FIFO scheduling in MapReduce causes job starvation. Hadoop Fair scheduler (HFS) implemented for achieving a degree of fairness. The goal is to optimize scheduling metrics such as completion time length, response time, stretch, and Service Level Agreements. FLEX is an add-on module integrated with HFS. **Proposed Concepts:** Penalty Functions Epoch-Based Scheduling Speedup Functions	Low remaining workload estimation due to the use of extrapolation techniques. Improvement is needed for accurate and dynamic estimation. Some low-quality metric specification in FLEX scheme. So much improvement is needed for some metric dependent schemes.

DOI 10.17148/IJARCCE.2015.41081

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

		Moldable and Malleable Scheduling	
A. Verma et al. 2011	ARIA: Automatic resource inference and allocation for MapReduce environments	MapReduce does not have any job schedulers for deadline and amount of resources meet SLO (Service Level Objective). Share MapReduce clusters need the ability to control resource management and allocations for better performance. **Proposed Concepts:** Completion Time Estimation SLO-based Performance Model **Proposed Components:** job profile Map Reduce performance model SLO-scheduler	The resource under utilization may lead to capacity degradation. So more number of tasks will be waiting. The SLO-scheduler must keep more number of resources.
B. Sharma et al. 2012	Mrorchestrator: A fine-grained resource orchestration framework for MapReduce clusters	MRorchestrator dynamically predicts bottlenecks in resources and reconfigure the slots with needed resources. The job tracker contains a global resource manager, which is responsible for allocating on-demand, fine-grained resources to the tasks. A local resource manger is present in the task tracker. It sends the resource request to the global resource manager. **Proposed Components:** Global resource manager components include a Contention Detector and Performance Balancer. Local resource manager components include a resource profiler, and an estimator.	It does not focus all computing resources. It only uses the CPU, memory, and I/O resources. The Mrorchestrator configured in single node Hadoop cluster. Its optimized results are based on small environment.
V. Kumar Vavilapalli et al. 2013	Apache Hadoop YARN: Yet another resource negotiator	MRv1 has fixed slot configuration without any resource management policies. The Proposed model needs to solve the problem in tight coupling of a programming model to resource management and centralized control flow for endless scalability. YARN decouples the model from resource management infrastructure and also represents many scheduling methods to per-application components. **Proposed Components:** Resource Manager Application Master Node Manager	YARN log aggregation component increases pressure on Name Node of Hadoop distributed file system while processing complex jobs.
J. Wang et al. 2014	FRESH: Fair and Efficient Slot Configuration and Scheduling for Hadoop Clusters	Processing batch jobs, the default Hadoop setting has low resource utilization and long completion time. It Proposes FRESH for optimized slot setting, dynamic slot configuration, and slot assignment. They measured Johnson indices to	Current job running Pool minimum share is not met; Fair scheduler can get jobs from other pool by pre-emption, which may kill pool dependent tasks while moving to the other pools. So fairness property is not achieved.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

		represent the level of fairness. **Proposed Components:** Fair scheduler Static slot configuration with FRESH Dynamic slot configuration with FRESH	In static slot configuration, FRESH preconfigured the MapReduce slots before execution. It leads to maximum of makespan.
S. Tang et al. 2014	Dynamic MR: A Dynamic Slot Allocation Optimization Framework for MapReduce Clusters	Slot-based MapReduce provides poor performance because of scheduling and resource allocation policies. So there is need to optimize the scheduling scheme and resource allocation policies. **Proposed Concepts:** Dynamic Hadoop Slot Allocation Speculative Execution Performance Balancing Slot PreScheduling **Algorithms:** PI-DHFS task assignment algorithm PD-DHFS task assignment algorithm	Slot PreScheduling improves only the data locality with no effect of fairness but needs more resources (memory, slots) for implementing cloud environment. Running jobs only have a chance of slot pre-scheduling and other jobs must be waiting for prescheduled slots. It shows the performance increase in only single node Hadoop cluster. It does not focus on deadline and budget.
A. Bansal et al. 2014	Healthcare Data Analysis using Dynamic Slot Allocation in Hadoop	Hadoop MRv1 is the slot-based system. It suffers from poor performance due to unoptimized resource allocation. Large scale healthcare systems data should be properly analyzed and computed. Due to static slot configuration a number of medical data processing units is waiting for another data completion time (Map and reduce). So they propose alternate slot configuration called dynamic slot configuration, which dynamically allocate slots based on idle slots and provide healthcare data in various forms.	It does not improve completion time goal. Because dynamic slot configuration without any scheduling and execution methodologies. Makespan of healthcare system MapReduce tasks is not well optimized. The system does not mention what type of dataset needed for processing health care.
A.U.Patil et al. 2015	Recent Job Scheduling Algorithms in Hadoop Cluster Environments: A Survey	Hadoop has default FIFO scheduler; Jobs are scheduled First in First out manner. Numerous schedulers are available for Hadoop MapReduce performance upgrade. More schedulers are not working well with small jobs. Propose Fair4S scheduling with extended functionalities for large and small jobs with efficient fairness without starvation. **Scheduling algorithms:** Default FIFO Scheduler Fair Scheduler Capacity Scheduler **Scheduler Improvements:** The Longest Approximate Time to End	LATE is functioning with speculative execution. It leads to unreliability of jobs and aforementioned bugs. These bugs are unchangeable so that tasks are not well performed. The fair scheduler has two issues. The First one is head-of-line scheduling and other one is sticky slots. Deadline schedulers only allocate jobs with a minimum number of map and reduce tasks for available jobs. The different deadline needs other jobs help. So there is always a need to check deadlines.

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

		Delay Scheduling Deadline Constraint Scheduler Resource Aware Scheduling Fair4s Job Scheduling	
Z. Liu et al. 2015	DREAMS: Dynamic Resource Allocation for MapReduce with Data Skew	Hadoop schedulers affected from partitioning skew, Map tasks unevenly distributed with reduce tasks. Proposed DREAMS provides run-time partitioning skew. It balances intermediate data for reducer tasks. It should eliminate partitioning overhead. **Proposed Concepts:** Predicting Partition Size Reduce Phase Performance Model Scheduling Algorithm **Proposed Components:** Partition Size Monitor Partition Size Predictor Task Duration Estimator Resource Allocator Fine-grained Container Scheduler	It improves job performance only by solving intermediate and reducer phase data skew. The main issue of RPC protocol is increased by scheduling cost for context switching. RPC does not give flexibility in hardware architecture. Data skew may occur while partitioning mapper phase.
Y. Yao et al. 2015	Self-Adjusting Slot Configurations for Homogeneous and Heterogeneous Hadoop Clusters	Hadoop MRV1 has fixed a number of MapReduce slots. Due to static slot configurations, the completion time of MapReduce is too high and it leads to poor performance. It must implement dynamic slot allocation to reduce completion time. **Proposed Concepts:** TUMM- Tunable knob for reducing the Makespan of MapReduce H_TUMM: Heterogeneous TUMM. **Proposed Components:** Workload Monitor Slot Assigner	FIFO scheduling policy only provides minimum completion time, but simple workloads need to be waiting during complex workloads running. FIFO scheduler does not give better performance in shared Hadoop clusters with the large set of users and different jobs.

Table 1 literature study

IV. SEARCH PROCESS

The manual search process is done for reviewing the conference and journal papers. This manual searching gives various papers related to dynamic slot configuration and scheduling concepts from 2008. The sequential and random search processes are done manually. Research references are collected from various sources like search engines, staff members, and web links. Search engines like Google, Bing provide papers randomly. Transaction papers like IEEE provide papers sequentially.

V. RESEARCH METHOD

This study is examined to be an evaluation over the dynamic slot configuration and scheduling techniques for MapReduce cluster. The questions are always given new innovative research ideas and clarity about research.

Research questions play a vital role for identifying concepts, and issues in the survey. The questions related to our study are given below.

A. Which is optimal slot configuration method either static or dynamic?
B. Why authors prefer FIFO schedulers for task assignment?
C. What is the reason to use different schedulers for slot assignment?
D. Why authors prefer single node Hadoop cluster for experimental results?
E. What is the Reason for using independent and dependent pools in slot allocation?

VI. OBSERVATION

A. Which is optimal slot configuration method either static or dynamic?

The dynamic slot configuration is always optimal because the static slot configuration assigns the task to MapReduce slots before the cluster launch. So the number of idle slots may increase due to the completion of map slots, and also chances of occurring overloaded reduce slots. Surely it affects completion time of the task. Unlike static slot

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 10, October 2015

configuration, dynamic slot configuration allocates slots during task execution time. It reduces the number of idle slots and increases busy slots

B. Why authors prefer FIFO schedulers for task assignment?
The FIFO scheduler is the default Hadoop scheduler implemented in MapReduce applications. Some authors still prefer FIFO scheduler for their research, especially Y. Yao et al. [15]. There are two Common reasons are to select default first in first out schedulers. Firstly, N numbers of jobs are waiting for acquiring resources. Secondly, all jobs can get resources without any starvation.

C. What is the reason to use different schedulers for slot assignment?
Schedulers are classified based on the performance metric, deadline aware, fairness metric, delay, resource aware, and fair4s scheduling. Each scheduler has configured with one of the metric specified above. Based on research preference different schedulers are used.

D. Why most people prefer single node Hadoop cluster for performance optimization?
Apache Hadoop installation comes with single node Hadoop cluster and multi-node Hadoop cluster. Mostly researcher configures single node cluster because it is easy to install with low cost and easy analysis of performance results. Hadoop multi-node cluster configuration needs more amount of hardware and network facilities. Multi node configuration is possible only to create a cloud environment. This is the reason for configuring single node cluster.

E. What is the Reason for using independent and dependent pools in slot allocation?
A task can be allocated with in the pool and across the pool. The fair scheduler allocates the same amount of resource to active tasks. Sometimes the task within the pool needs resource across the pool. This is the main reason for dividing pools into independent and dependent. The dependent pool slots can dynamically allocate across the pool. But independent pool slots only allocates within the pool dynamically.

VII. CONCLUSION

Dynamic slot configuration is one of the important factors while processing a large data set with MapReduce paradigm. It optimizes the performance of MapReduce framework. Each job can be scheduled using any one of the scheduling policies by the job tracker. The task managers which are present in the task tracker allocate slots to jobs. From the examined paper, it is concluded to prefer a dynamic slot allocation strategy that includes active jobs workload estimation, optimal slot assignment, and scheduling policy.

REFERENCES

[1] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters", in Communications of the ACM, vol. 51, 2008.
[2] J. Polo, D. Carrera, Y. Becerra et al., "Performance-driven task co-scheduling for MapReduce environments", in NOMS'10, 2010.
[3] J. Wolf, D. Rajan, K. Hildrum, R. Khandekar, V. Kumar, S. Parekh, K.-L. Wu, and A. Balmin, "Flex: A slot allocation scheduling optimizer for MapReduce workloads", in Middleware 2010, ser. Lecture Notes in Computer Science, I. Gupta and C. Mascolo, Eds. Springer Berlin / Heidelberg, vol. 6452. pp. 1, 2010.
[4] Apache Hadoop. Reference link: http://hadoop.apache.org.
[5] A. Verma, L. Cherkasova, and R. H. Campbell, "ARIA: Automatic Resource inference and allocation for MapReduce environments", in International Conference on Autonomic Computing, 2011.
[6] Apache Hadoop YARN (yet another resource negotiator) Reference Link: https://hadoop.apache.org/docs/current/hadoop-yarn/hadoop-yarn-site/YARN.html.
[7] B. Sharma, R. Prabhakar, S.-H. Lim et al., "Mrorchestrator: A fine-grained resource orchestration framework for MapReduce clusters", in CLOUD'12, 2012.
[8] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth et al., "Apache Hadoop yarn: Yet another resource negotiator", in Proceedings of the 4th annual Symposium on Cloud Computing. ACM, 2013.
[9] Jiayin Wang, Yi Yao, Ying Mao, Bo Sheng, N. Mi, "FRESH: Fair and Efficient Slot Configuration and Scheduling for Hadoop Clusters", IEEE 7th International Conference on Cloud Computing, DOI: 10.1109/CLOUD.2014.106. Anchorage, AK, pp 761, June 2014.
[10] Capacity scheduler Reference Link: https://hadoop.apache.org/docs/r1.2.1/capacity_scheduler.html
[11] S. Tang, B. Lee, and B. He," Dynamic MR: A Dynamic Slot Allocation Optimization Framework for MapReduce Clusters", IEEE Transactions on Cloud Computing, vol. 2, issue. 3, September 2014.
[12] A. Bansal, A. Deshpande, P. Ghare, S. Dhikale, B. Bodkhe, " Healthcare Data Analysis using Dynamic Slot Allocation in Hadoop", International Journal of Recent Technology and Engineering Vol. 3 Issue 5, November 2014.
[13] A.U.Patil, T.I Bagban , A.P.Pande, "Recent Job Scheduling Algorithms in Hadoop Cluster Environments: A Survey", International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, no. 2, February 2015.
[14] Z. Li, Q. Zhang, M. F. Zhani, R. Boutaba, Y. Liu, Z. Gong et al., "DREAMS: Dynamic Resource Allocation for MapReduce with Data Skew", Integrated Network Management (IM), DOI 10.1109/INM.2015.7140272, 2015 IFIP/IEEE International Symposium on May 2015.
[15] Y. Yao, J. Wang, B. Sheng, C. Tan and N. Mi, "Self-Adjusting Slot Configurations for Homogeneous and Heterogeneous Hadoop Clusters", IEEE Transactions on Cloud Computing, Vol. PP, September 2015.