Scientific Journal of Impact Factor (SJIF): 4.72

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 4, Issue 11, November -2017

A survey on "Resource management for Big data platforms"

Study on : Survey of papers which study
Resource Management techniques for Big data paltforms

Sharayu D Mirasdar

Department of Computer Engineering, BVUCOE Pune

Abstract —*This paper contains of two Survey papers which has a focus on 'Resource management for Big Data platforms'. Both the papers focus on what are new Resource Management frameworks that can be applied to Big Data platforms. Fundamental design issue for big data processing systems in the Cloud has been studies and addressed.*

Keywords- *Resource Management, Big data, Cloud, Hadoop*

A SURVEY ON "RESOURCE MANAGEMENT FOR BIG DATA PLATFORMS"

"A New Efficient Resource Management Framework for Iterative MapReduce Processing in Large-Scale Data Analysis "[IEICE TRANS. INF. & SYST., VOL.E100–D, NO.4 APRIL 2017 Seungtae HONG†a), Kyongseok PARK††b), Chae-Deok LIM†c), Nonmembers, and Jae-Woo CHANG†††d), Member]

To analyze large-scale data efficiently, studies on Hadoop, one of the most popular MapReduce frameworks, have been actively done. Meanwhile, most of the large-scale data analysis applications, e.g., data clustering, are required to do the same map and reduce functions repeatedly. However, Hadoop cannot provide an optimal performance for iterative MapReduce jobs because it derives a result by doing one phase of map and reduce functions.
The existing iterative processing frameworks have the following drawbacks.

- First, because the existing frameworks are implemented based on the version v0.20 of Hadoop, they do not support the efficient resource management of cluster.
- Second, because the existing frameworks do not consider the characteristic of iterative applications, they do not provide an invariant data caching mechanism efficiently.
- Thirdly, since the existing frameworks do not consider the entire resources of cluster, the tasks for processing MapReduce job can be skewed towards particular nodes.
- Finally, they do not provide a stop condition check mechanism for preventing unnecessary computations.

This paper has proposed a new solution to solve this issue. They propose a new efficient resource management framework for iterative MapReduce processing in large-scale data analysis. This paper has suggested four steps for Iterative MapReduce as below:

1. Design an iterative job state-machine for managing the iterative MapReduce jobs.
2. An invariant data caching mechanism for reducing the I/O costs of data accesses.
3. An iterative resource management technique for efficiently managing the resources of a Hadoop cluster.
4. A stop condition checks mechanism for preventing unnecessary computation.

Finally, the proposed framework performance is compared with existing frameworks and shown the superiority of the new solution.

FUTURE WORK

From performance analyses done in this paper, it was shown that the new framework outperforms the existing works. In case of the page rank and the descendant query applications, the proposed framework shows 7.9 to 8.3 times better performance than the existing Hadoop, by using the reduce input cache. In case of the k-Means application, the proposed framework shows 7.8 times better performance by using the map input cache. As a result, iterative data processing framework mentioned in this paper, is suitable for iterative applications because it can provide the efficient job and resource scheduling with both our invariant data caching and stop check mechanisms. As future research, the authors plan to study on an indexing technique that can automatically detect invariant data in their caching mechanism.

'"Resource Management in Big Data Processing "
Shanjiang Tang, Bingsheng He, Haikun Liu and Bu-Sung Lee

Resource management is a fundamental design issue for big data processing systems in the cloud. Different resource allocation policies can have significantly different impacts on performance and fairness. In this paper a study on the economic fairness for large-scale resource management on the cloud according to some desirable properties including sharing incentive, truthfulness, resource-as-you-pay fairness, and pareto efficiency. Both single-resource and multi-resource management are studied for cloud computing.
In many application domains such as social networks and Bio-informatics, the data is being gathered at unprecedented scale. Efficient processing for "big data" analysis poses new challenges for almost all aspects of state-of-the-art data processing and management systems. For example, there are a few challenges as follows:
(i) the data can be arbitrarily complex structures (e.g., graph data) and cannot be efficiently stored in relational database;
(ii) the data access of large-scale data processing are frequent and complex, resulting in inefficient
disk I/O accesses or network communications; and (iii) last but not least, to tackle a variety of unpredictable failure problems in the distributed environment, data processing system must have a fault tolerance mechanism to recovery the task computation automatically.

Cloud computing has emerged as an appealing paradigm for big data processing over the Internet due to its cost effectiveness and powerful computational capacity. Current Infrastructure-as-a-Service(IaaS) clouds allow tenants to acquire and release resource in the form of virtual machines (VMs) on a pay-as-you-go basis.

TYPES OF RESOURCE MANAGEMENT

Resource management is a general and fundamental issue in computing systems. In this section, we present the resource management for typical resources including CPU, memory, storage and network.

CPU AND MEMORY RESOURCE MANAGEMENT

Current supercomputers and data centers (e.g., Amazon EC2) generally consist of thousands of computing machines. At any time, there are tens of thousands of users running their high-performance computing applications (e.g., MapReduce , MPI, Spark) on it. The efficient resource management of the computing resources such as cpu, memory is non-trivial for high performance and fairness. Typically, the resource management includes resource discovery, resource scheduling, resource allocation and resource monitoring. Resource discovery identifies the suitable computing resources in which machines that match the user's request. Resource scheduling selects the best resource from the matched computing resources. It actually identifies the resource where the machines are to be created to provision the resources. Resource allocation allocates the selected resource to the job or task of user's request. Actually, it means the job submission to the selected cloud resource. After the submission of the job, the resource is monitored. There are a number of resource management tools available for supercomputing. For example, SLURM is a highly scalable resource manager widely used in supercomputers. For data-intensive computing in data center, YARN and Mesos are two popular resource management systems.

STORAGE RESOURCE MANAGEMENT

Storage resource management (SRM) is a proactive approach to optimizing the efficiency and speed with which available drive space is utilized in a storage area network, which is a dedicated high-speed network (or subnetwork) that interconnects and presents shared pools of storage devices to multiple servers. The SRM software can help a storage administrator automate data backup, data recovery and SAN performance analysis. It can also help the administrator with configuration management and performance monitoring, forecast future storage needs more accurately and understand where and how to use tiered storage, storage pools and thin provisioning. Network Resource Management Managing and allocating the network flows to different applications/users is a non-trivial work. Particularly, Software-defined networking (SDN) is nowadays a popular approach
that allows network administrators to manage network services through abstraction of lower-level functionality. This is done by decoupling the system that makes decisions about where traffic is sent (the control plane) from the underlying systems that forward traffic to the selected destination (the data plane).

There are some other high-level as well as application-specific systems that are built-on top of previous data computing systems to form an ecosystem for a variety of applications. For example, for Hadoop, Apache Pig and Hive are both SQL-like systems that are running on it to support analytical data querying processing. HBase is a NoSql database system built on top of Hadoop system. Apache Giraph is an iterative graph pro- cessing system running on Hadoop. Similarly, for Spark, Shark and Spark SQL are two analytical data query system built on Spark, and Graphx is a graph processing system for graph applications. We have also witnessed some other data processing sys- tems/platforms that are running on currently emerging computing devices such as GPUs.

Systems	Computation Model	In-memory computation	Resource Management Type				Resource Fairness	
			CPU	Memory	Storage	Network	Single	Multiple
Hadoop	MapReduce	no	yes	yes	no	no	yes	yes
Dryad	Dryad	no	yes	no	no	no	yes	no
Pregel	Pregel	no	yes	no	no	no	yes	no
Storm	Storm	no	yes	no	no	no	yes	no
Spark	RDD	yes	yes	yes	no	no	yes	yes

Table 1: Comparison of representative big data processing systems

In this section of the paper, authors have discussed the importance of resource management for big data processing and surveyed a number of existing representative large-scale data processing systems. One of the classic issues for resource management is fairness. The chapter reviewed 34 the memory less fair resource allocation policies for existing systems and showed their unsuitability for cloud computing by presenting three problems. A new Long-Term Resource Allocation (LTRF) policy was then proposed to address these problems, and we provably and experimentally validate the merits of the proposed policy. This chapter next focused on the resource management for virtual machines on the cloud, considering VM migration and consolidation in the cloud environment. Finally, there are many open problems that need more research efforts in this field.

REFERENCES

[1] Namdeo, Jyoti, and NaveenkumarJayakumar. "Predicting Students Performance Using Data Mining Technique with Rough Set Theory Concepts." International Journal 2.2 (2014).

[1]. Giraph. In http://giraph.apache.org/.

[2]. Hadoop. In http://hadoop.apache.org/.

[3]. Loan agreement. In http://en.wikipedia.org/wiki/Loan agreement.

[4]. Max-min fairness (wikipedia). In http://en.wikipedia.org/wiki/Max-min fairness.

[5]. Software-defined networking. In https://en.wikipedia.org/wiki/Software-definednetworking.

[6]. Storage resource management. In https://en.wikipedia.org/wiki/Storage Resource Management.

[7]. Storm. In http://storm-project.net/.

[8]. Tez. In https://tez.apache.org/.

[9]. S. Abrishami, M. Naghibzadeh, and D. H. Epema. Cost-driven scheduling of grid workflows using partial critical paths. Parallel and Distributed Systems, IEEE Transactions on, 23(8):1400–1414, 2012.

[10]. Amazon. http://aws.amazon.com/solutions/case-studies/.

[11]. V. Anuradha and D. Sumathi. A survey on resource allocation strategies in cloud computing. In Information Communication and Embedded Systems (ICICES), 2014 International Conference on, pages 1–7. IEEE, 2014.

[12]. M. Armbrust, R. S. Xin, C. Lian, Y. Huai, D. Liu, J. K. Bradley, X. Meng, T. 36 REFERENCES Kaftan, M. J. Franklin, A. Ghodsi, et al. Spark sql: Relational data processing in spark. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pages 1383–1394. ACM, 2015.

[13]. AutoScaling. http://aws.amazon.com/autoscaling.

[14]. P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield. Xen and the art of virtualization. In Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles, SOSP '03, pages 164–177, New York, NY, USA, 2003. ACM.

[15]. A. Beloglazov and R. Buyya. Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. Parallel and Distributed Systems, IEEE Transactions on, 24(7):1366– 1379, 2013.

[16]. A. Beloglazov, R. Buyya, Y. C. Lee, A. Zomaya, et al. A taxonomy and survey of energy-efficient data centers and cloud computing systems. Advances in computers, 82(2):47-111,2011.

[17]. A. A. Bhattacharya, D. Culler, E. Friedman, A. Ghodsi, S. Shenker, and I. Stoica. Hierarchical scheduling for diverse datacenter workloads. In Proceedings of the 4th Annual Symposium on Cloud Computing, SOCC '13, pages 4:1–4:15, New York, NY, USA, 2013. ACM.

[18]. E.-K. Byun, Y.-S. Kee, J.-S. Kim, and S. Maeng. Cost optimized provisioning of elastic resources for application workflows. Future Generation Computer Systems, 27(8):1011–1026, 2011.

[19]. L. Cherkasova, D. Gupta, and A. Vahdat. Comparison of the three cpu 37 REFERENCES schedulers in xen. SIGMETRICS Performance Evaluation Review, 35(2):42–51, 2007.

[20]. M. Chowdhury and I. Stoica. Coflow: An application layer abstraction for cluster networking. In ACM Hotnets, 2012.

[21]. C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In Proceedings of the 2nd con- ference on Symposium on Networked Systems Design & Implementation-Volume 2, pages 273–286. USENIX Association, 2005.

[22]. A. Corradi, M. Fanelli, and L. Foschini. Vm consolidation: A real case based on openstack cloud. Future Generation Computer Systems, 32:118–127, 2014.

[23]. J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. Commun. ACM, 51(1):107–113, Jan. 2008.

[24]. C. Delimitrou and C. Kozyrakis. Quasar: Resource-efficient and qos-aware cluster management. ACM SIGPLAN Notices, 49(4):127–144, 2014.

[25]. A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queue- ing algorithm. In Symposium Proceedings on Communications Architectures &Amp; Protocols, SIGCOMM '89, pages 1–12, New York, NY, USA, 1989. ACM. [26]. M. Drozdowski. Scheduling for Parallel Processing. Springer Publishing Company, Incorporated, 1st edition, 2009.

[27]. A. Faraz, L. Seyong, T. Mithuna, and V. T. N. Puma: Purdue mapreduce bench- marks suite. Technical Report EECS-2012-Oct, School of Electrical and Computer Engineering, Purdue University, Oct 2012.

[28]. H. M. Fard, R. Prodan, and T. Fahringer. A truthful dynamic workflow scheduling mechanism for commercial multicloud environments. Parallel and Distributed Sys- tems, IEEE Transactions on, 24(6):1203–1212, 2013.

[29]. L. George. HBase: the definitive guide. " O'Reilly Media, Inc.", 2011. 38 REFERENCES

[30]. A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica. Dom- inant resource fairness: Fair allocation of multiple resource types. In Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation, NSDI'11, pages 24–24, Berkeley, CA, USA, 2011. USENIX Association.

[31]. A. Ghodsi, M. Zaharia, S. Shenker, and I. Stoica. Choosy: Max-min fair sharing for datacenter jobs with constraints. In Proceedings of the 8th ACM European Confer- ence on Computer Systems, EuroSys '13, pages 365–378, New York, NY, USA, 2013. ACM.

[32]. D. Gmach, J. Rolia, and L. Cherkasova. Selling t-shirts and time shares in the cloud. In Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgrid 2012), CCGRID '12, pages 539–546, Washington, DC, USA, 2012. IEEE Computer Society.

[33]. J. E. Gonzalez, R. S. Xin, A. Dave, D. Crankshaw, M. J. Franklin, and I. Stoica.Graphx: Graph processing in a distributed dataflow framework. In Proceedings of OSDI, pages 599–613, 2014.

[34]. Google. https://cloud.google.com/customers/.

[35]. R. Grandl, G. Ananthanarayanan, S. Kandula, S. Rao, and A. Akella. Multiresource packing for cluster schedulers. In Proceedings of the 2014 ACM Conference on SIG- COMM, SIGCOMM '14, pages 455–466, New York, NY, USA, 2014. ACM

