INFORMATICA, 2013, Vol. 24, No. 3, 381–394
© 2013 *Vilnius University*

Intelligent Reconfigurable Method of Cloud Computing Resources for Multimedia Data Delivery

Junho CHOI[1], Chang CHOI[1], Kangbin YIM[2], Jeongnyeo KIM[3], Pankoo KIM[1] *
[1]*Department of Computer Engineering, Chosun University*
309 Pilmun-daero, Dong-gu, Gwangju, Republic of Korea
[2]*Department of Information Security Engineering, Soonchunhyang University*
Asan, Chungnam, Republic of Korea
[3]*Mobile Security Research Team, Electronics and Telecommunications Research Institute*
218 Gajeong-ro, Yuseong-gu, Daejeon, Republic of Korea
e-mail: xdman@paran.com, endurancearua@gmail.com, yim@sch.ac.kr, jnkim@etri.re.kr, pkkim@chosun.ac.kr

Received: February 2012; accepted: February 2013

Abstract. While users increasingly use such large multimedia data, more people use the cloud computing technology. It is necessary to manage large data in an efficient way, and to consider transmission efficiency for multimedia data of different quality. To this end, an important thing is to ensure efficient distribution of important resources (CPU, network and storage) which constitute cloud computing, and variable distribution algorithms are required therefor. This study proposes a method of designing a scheme for applying MapReduce of the FP-Growth algorithm which is one of data mining methods based on the Hadoop platform at the stage of IaaS (Infrastructure As a Service) including CPU, networking and storages. The method is then for allocating resources with the scheme.
Key words: cloud computing, mapreduce, fp-growth algorithm, resource provisioning.

1. Introduction

Cloud computing is actively used for reducing computing environment costs and efficient use of IT infrastructure as many people use the Internet environment and user's request of multimedia data thus increases. Cloud computing is a more advanced technology for distributed processing, e.g., a thin client and grid computing, which is implemented by means of virtualization technology for servers and storages, and advanced network functionality. Cloud computing is expected to be a new paradigm to lead the next generation computing environment.

The tendency of contents services provided by global IT vendors is showing large scale multimedia data, e.g., movies, games, and music. Cloud computing is emerging as a

*Corresponding author.

key alternative for next generation digital contents services in terms of connection speed, service quality, and pricing which is a challenge for large scale multimedia services.

Issues involved in providing the multimedia data services include variable usage of CPU, management of storage usage, the long process of installation, high costs for facility investment, and continuous post-installation maintenance (Mladen, 2008). For addressing the issues, multimedia data service providers integrate physical resources by means of virtualization technology, and then provide services required by users. A cloud system should provide a user-friendly flexible execution environment to allow all users' applications to operate because resource request from users' applications changes over time. A mechanism is needed which ensures the performance of applications used by users. Therefore, it is important to balance resource usage with request thereof so as to flexibly support resource request changing over a given time.

In this study, we propose an intelligent cloud system model for analyzing user's resource usage to predict the amount of user's resource request in advance. The proposed method is a scheme for allocating resources for large scale multimedia data services on the basis of the Hadoop platform at the stage of infrastructure service (Infrastructure As a Service), e.g., CPU, network and storages. The proposed method is implemented by designing a scheme for applying MapReduce of the FP-Growth algorithm which is one of data mining methods, and predicting requested services and the amount of resources over time.

2. Related Studies

2.1. *Computing Technology for Large Multimedia Processing*

A distributed computing platform for large multimedia services is supported by the cluster management system which provides monitoring and scheduling for a distributed environment. The distributed computing platform is implemented by integrating distributed file system technology for storing large data, distributed data storage management technology for supporting searching and modification of large data, distributed parallel processing and analysis technology for supporting large data analysis (Chaiken, 2008). The distributed computing platform technology is a key technology of cloud computing together with the virtualization technology.

In the cloud computing environment, batch data processing for data processing and analysis takes several days or tens of days as the volume of Internet data continues to increase. For such large data processing, the MapReduce parallel processing model is commonly used as a standard. The MapReduce model is a parallel processing model proposed for quickly operating batch jobs for Internet services, and is applied to Google's various services (earth, news, analytics, search, indexing, etc.; Lammel, 2008). MapReduce is a model for processing ⟨key, value⟩-based data in parallel, and consists of two steps of carrying out the Map task on the basis of input data sources to create interim results, and carrying out the Reduce task by using the interim results as input to obtain final results, as shown in Fig. 1.

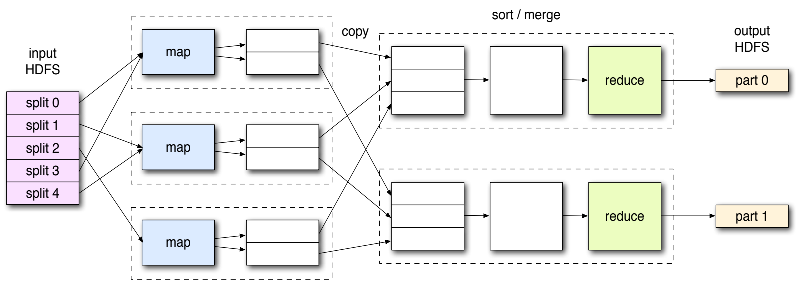

Fig. 1. Map split input data and reduce partitioned intermediate data.

The input data are divided into a plurality of data, for some of which the Map task is carried out in a plurality of nodes. Each of Map tasks stores the result of processing the input data allocated thereto in the local file system of each node. The Reduce task receives the interim results stored in the plurality of nodes for integrated processing to provide final results. Task distribution is implemented so that data can be processed in the node where they are placed if possible so as to minimize network traffic. To this end, data are divided in consideration of the status and location of data storage.

2.2. *FP-Growth Algorithm*

The Apriori algorithm is a representative association rule extraction technique for data mining, and finds frequent item sets for binary association rules. The Apriori algorithm finds association rules, using the rule that all of the subsets, not empty sets of the frequent item sets, are frequently found. However, since it is necessary to create candidate sets of the items in a transaction to find frequent item sets each time, it is required to scan related data a plurality of times. To address this issue, a technique to find complete frequent item sets with the Divide-and-Conquer without creating candidates is called the FP-Growth (Frequent Pattern Growth) algorithm (Li, 2008). For creating an FP-Tree in the FP-Growth algorithm, a header table is created, which is a top-down list of frequent 1-item sets of transaction data on the basis of frequent patterns. The header table is used to filter and line up the transaction data with only the items which meet the minimum support.

A FP-Tree is created by creating a rootnode whose node name is null and inserting the lined-up transaction data in the tree. In this case, each node saves the name and frequency of items. If an item with the same name as the node created by means of a previous transaction is added, the frequency of the existing node is increased instead of creating a new node. The item in the head table is connected to the node with the same name in the tree for the association rule extraction in the FP-Tree (Iko, 2003). For association rule extraction in the FP-Tree, the tree is searched from the lowest item to the root node in the header table, and a Conditional Pattern Base is created, which has a suffix of the node on the lower layer. The Conditional Pattern Base is used to create a Conditional Pattern Tree which meets the minimum support to extract an association rule.

3. Intelligent Reconfigurable Method of Cloud Computing Resources

In the cloud computing environment, it is necessary to be able to manage a number of nodes which consists a cluster for large multimedia data services and a key is efficient distribution of distributed file system resources (Papadimitriou, 2008). The file system which is proposed in this study and implements efficient resource distribution provides configuration and automatic management of a large cluster system. The file system aims to implement on-line efficiency and availability, and to improve the usage in a general data center for efficient distribution of resources.

3.1. *Data Collection and Pre-Processing Stage*

A collection is made of information including system specification, usage of resources, and task processing capacity so as to analyze the type of using nodes in the cloud computing environment at given time intervals to store the data in a log format. Nodes are selected on the basis of the collected data to allocate resources when users request them, so that the cloud system can provide appropriate resources to users. This study focuses on predicting stability and performance of resources depending on the type of using resources in each time zone. Such prediction contributes to dynamic configuration of services and enables efficient distributed processing of tasks. For efficient resource allocation, use is made of basic node information of log data, average use of resources by nodes, average processing time of nodes and average success rates of nodes as shown in Table 1.

For collecting the above data, a collection is made of cluster information for each node provided by the Hadoop file system, the use of resources by each node, and the use of resources of each system by means of a system resource monitoring tool. Modeling for the resources of each node is as follows: $\mathit{NodeRes} = \langle R, T, C, W, U \rangle$, wherein T represents time, R represents resource sets, C represents the number of resources, W represents the task carried out by resources and U represents the usage of workable resources.

Table 2 shows details of log files collected for analysis, which include stored information e.g., system specification, resource usage, and work capacity. The collected information goes through the pre-processing process to be applied to the parallel FP-Growth algorithm. The contents information of multimedia data is collected in the pre-processing process to be referred to in the resource allocation process of the Hadoop file system in

Table 1
Name node log data of Hadoop file system.

```
2012-01-21 11:25:03,453 INFO org.apache.hadoop.hdfs.server.namenode.FSNamesystem: Number of transactions: 0 Total time for transactions (ms): 0 Number of transactions batched in Syncs: 0 Number of syncs: 0 SyncTimes (ms): 0
2012-01-21 11:25:03,968 INFO org.apache.hadoop.hdfs.server.namenode.FSNamesystem: Roll FSImage from 117.16.23.173
2012-01-21 11:25:03,968 INFO org.apache.hadoop.hdfs.server.namenode.FSNamesystem: Number of transactions: 0 Total time for transactions (ms): 0 Number of transactions batched in Syncs: 0 Number of syncs: 1 SyncTimes (ms): 47
```

Table 2
Extracted information from log data.

Item	Properties
Node information	Node ID, CPU Info., Memory Info., HDD Info., Network Info.
Use of resource	Record Time, CPU Usage, Memory Usage, HDD Usage
Job information	Record Time, Job Type, Response Time, Job Status

Table 3
Collected data set through pre-processing.

Node ID	CPU	Memory	Network	Block size	Memory size	File buffer	Ramining (%)	Contents
Node 1	800	4 GB	100 MB	128 MB	100	131072	70.21	AVI
Node 2	800	2 GB	100 MB	64 MB	100	131072	80.34	AVI
Node 3	900	2 GB	100 MB	64 MB	100	131072	69.33	AVI
Node 4	800	2 GB	100 MB	128 MB	100	65536	81.21	MPEG
Node 5	900	2 GB	100 MB	128 MB	100	131072	71.98	MPEG
Node 6	900	2 GB	50 MB	128 MB	100	65536	54.67	MPEG
Node 7	800	2 GB	50 MB	128 MB	100	131072	89.12	AVI
Node 8	900	2 GB	50 MB	64 MB	100	65536	74.48	AVI
Node 9	900	2 GB	100 MB	64 MB	100	131072	61.32	AVI
Node 10	900	4 GB	100 MB	64 MB	150	65536	88.19	AVI

consideration of the type and format of the multimedia data, so that the contents information can be included in the data set of each node shown in Table 3.

3.2. *Designing and Applying Parallel FP-Growth Algorithm*

In this section, the parallel FP-Growth algorithm is designed in the steps shown in Fig. 2 for interworking the FP-Growth algorithm with MapReduce calculation for the purpose of distributed processing in the cloud computing environment.

Step 1: divides a log file or database to store it in each system in the cloud computing environment.

Step 2: carries out MapReduce calculation to find the support of all items of each data. This step searches for the item I and stores it in Frequent List. The item I is represented as dGi, and Mapper is created as $\langle key = d, value = 1\rangle$. Finishing the Mapper processing, key is created. The key set is represented as $S(key)$, and the pair $\langle key, S(key)\rangle$ for the key and the value thereof is transferred to the Reducer. Finishing the process, the Reducer outputs $\langle key = null, value = key + sum(S(key))\rangle$.

Step 3: divides the item $|I|$ stored in the group list of group G. This step is carried out by each slave system.

Step 4: inputs the data divided and created in step 1 to create processors independent of the group. Substituting ai corresponding to the group ID to find each item dGi, the data of $\langle key = g, value = Gi[1]Gi[n]\rangle$ is created. Calculating the Mapper, the value of $\langle Key, S(key)\rangle$ is transferred to the Reducer. The Reducer outputs the value of $\langle Key = null, value = key + sum(S(key))\rangle$ and collects the results in step 4 to get final results.

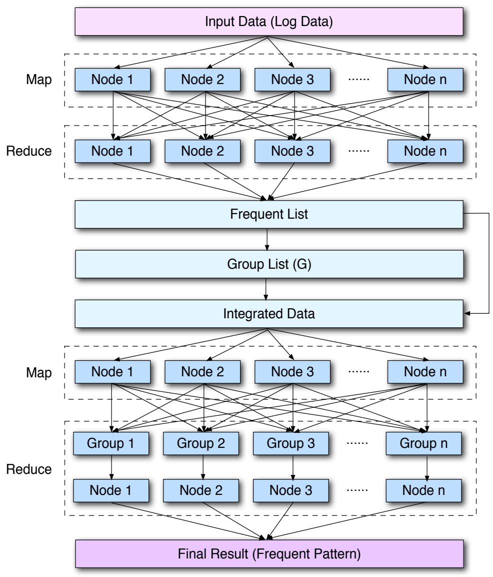

Fig. 2. Steps of applying MapReduce calculation of FP-Growth algorithm.

Table 4
The generated frequent patterns by parallel FP-Growth algorithm.

100, 900	2 GB, 900, 64 MB
AVI, 900	100, 2 GB, 900, 64 MB
100, AVI, 900	100 MB, 900, 64 MB
2 GB, AVI, 900	100, 100 MB, 900, 64 MB
100, 2 GB, AVI, 900	2 GB, 100 MB, 900, 64 MB
100 MB, AVI, 900	100, 2 GB, 100 MB, 900, 64 MB
100, 100 MB, AVI, 900	131072, 64 MB
2 GB, 100 MB, AVI, 900	900, 131072, 64 MB
100, 2 GB, 100 MB, AVI, 900	100, 900, 131072, 64 MB
2 GB, 900	AVI, 900, 131072, 64 MB
100, 2 GB, 900	100, AVI, 900, 131072, 64 MB
100 MB, 900	2 GB, AVI, 900, 131072, 64 MB
100, 100 MB, 900	100, 2 GB, AVI, 900, 131072, 64 MB
2 GB, 100 MB, 900	100 MB, AVI, 900, 131072, 64 MB
100, 2 GB, 100 MB, 900	100, 100 MB, AVI, 900, 131072, 64 MB
MPEG	2 GB, 100 MB, AVI, 900, 131072, 64 MB
900, MPEG	100, 2 GB, 100 MB, AVI, 900, 131072, 64 MB
100, 900, MPEG	2 GB, 900, 131072, 64 MB
2 GB, 900, MPEG	100, 2 GB, 900, 131072, 64 MB
100, 2 GB, 900, MPEG	100 MB, 900, 131072, 64 MB

Apply the minimum support 2 and the minimum support 4 to the data sets for each node created by means of the proposed algorithm to find frequent patterns shown in Table 4.

In Table 4, illustrated dominant frequent patterns are {100, 2 GB, 100 MB, 64 MB}, {100, 2 GB, 100 MB, 131 072}, {100, 100 MB, AVI, 131 072}, {100, 2 GB, 128 MB},

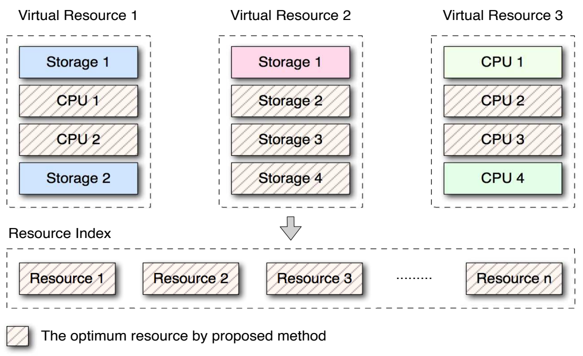

Fig. 3. An optimized resource distribution by the parallel FP-Growth algorithm.

{100, 2 GB, 100 MB}, and {100, 100 MB, AVI}. This implies that the above resource usage patterns are used in each slave system for processing large multimedia data by means of the Hadoop file system. Therefore, efficient resource distribution is achieved in setting the system in terms of system management as shown in Fig. 3 in consideration of resource usage when carrying out a specific task.

4. Test and Evaluation

4.1. *Test Environment*

In this study, a system used is a Hadoop platform to implement a cloud computing-based system. The Hadoop platform is developed by the Apache Project Group which is an open software project. For the purpose of test, the cloud computing environment consisting of 10 PCs as a node of a different nature and different performance was constituted. The OS of each node was constituted with Linux and Windows Server which supports the Hadoop platform.

4.2. *Testing Proposed Architecture*

Now, application of the proposed architecture is tested so as to evaluate the performance of resource extension to which the parallel FP-Growth algorithm proposed in Section 3 is applied, and the parallel FP-Growth algorithm is carried out to analyze the result. The proposed method of resource extension may be lowered in terms of performance depending on inefficient memories between the master node and the slave node, processes and network status. The test is carried out by comparing the performance of the system to which the proposed resource extension method is applied with the performance of the single processing system with respect to processing time to get performance results. The RandomWriter module was set to create input data of 70 GB, and to request processing 1 000 to 10 000 tasks to evaluate processing performance. The following equations (1) and

Table 5
The reduction rate of processing time.

Number of processed tasks	Single node processing time (s)	Processing time by applying proposed resource (s)	Reduction of processing time (%)
1000	108.33	93.13	14.03
2000	290.23	200.67	30.86
3000	481.11	310.12	35.54
4000	589.78	425.87	27.79
5000	764.39	561.36	26.56
6000	1011.71	753.34	25.54
7000	1234.59	928.01	24.83
8000	1382.81	1082.45	21.72
9000	1571.32	1184.21	24.64
10000	1791.14	1303.12	27.25

(2) are for calculating the sum of processing (P) for the time (PT: Processing Time) taken only for the processing in the node in the cloud environment used in this test. Here, the time is related to the time in a single node ($SNPT$ Single Node Processing Time) and the processing time (CPT: Cloud Processing Time) in the cloud computing environment to which the proposed algorithm was applied. Also, the average in Table 5 was obtained for the reduction rate of processing time each time 1000 tasks increased.

$$SNPT = \sum_{n=1}^{p} P_{SingleNode} T_n. \tag{1}$$

$$CPT = \sum_{n=1}^{p} P_{Cloud} T_n. \tag{2}$$

In this test, the number of tasks were increased from 1000 to 10 000 for processing to measure the average processing time and then to calculate the result on the basis of the number of tasks.

Figure 4 illustrates the result of measurement in the test. As the number of tasks carried out by applying the proposed method in the cloud computing environment increased, the effect of application was greater. The ratio was calculated as a ratio (PTR: Processing Time Ratio) of the single node processing time ($SNPT$: Single Node Processing Time) to the processing time (CPT: Cloud Processing Time) in the cloud computing environment to which the proposed algorithm was applied, so as to identify the relation with processing time.

$$PTR(\%) = \frac{SNPT}{CPT}. \tag{3}$$

The processing time in the method of resource allocation to which the proposed algorithm was applied in the cloud computing environment revealed the processing time efficiency of approximately 25.88% as compared to the single node processing time. It implies that it is possible to reflect the performance of each slave node to a maximum

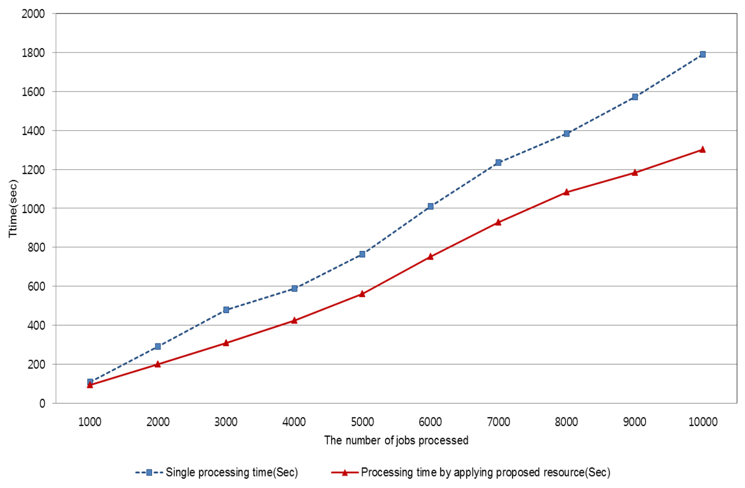

Fig. 4. A comparison of processing time through resource extension.

in the context with issues of increasing single node processing tasks and network transmission time. It is considered that the processing time by the resource allocation method proposed in the cloud computing environment is more efficient for processing tasks than a single node in that storage resources of cloud computing are used.

4.3. *Testing Application of Parallel FP-Growth Algorithm*

For this test, input data of 600 GB (files of 60 GB, respectively, 10 per node) were created with the RandomWriter module provided in Hadoop. An example sorter was used to sort the data. Test data Table 3 were obtained for 10 test nodes by changing resources of each node, the buffer and block size of the Hadoop file system, and memory allocation values.

Application of the association rules for using resources in the cloud environment in independent nodes aims at finding reliable items for options of each resource. Therefore, it is possible to set best resources for a specific task by adjusting frequent items of each resource frequently used by each node in cloud computing for Minimum Support. Performance is measured by changing settings of resources of each node selected through this process.

First, the test data in the node where the Hadoop file system running with the default settings has been installed were created to measure the loading process, memory usage, CPU usage and network usage with respect to resource usage of each node. Each node had the data used for applying the parallel FP-Growth algorithm as Hadoop file system information.

A comparison was made between the result of storage clustering with default settings in one virtualization server environment and the result of storage clustering in a distributed environment by applying a method of allocation to which the parallel FP-Growth algorithm was applied. The minimum support of related settings of each system was set to 2%,

Table 6
A comparison of transmission time after applying the parallel FP-Growth algorithm.

The number of times	Node 1	Node 2	Node 3
Once	5.250	5.067	5.417
Twice	5.350	5.167	5.333
3 times	5.150	5.300	5.467
4 times	5.283	5.183	5.450
5 times	5.117	5.317	5.183
6 times	5.383	5.050	5.183
7 times	5.167	5.167	5.067
8 times	5.250	5.233	5.067
9 times	5.192	5.242	5.108
10 times	5.308	5.125	5.125
Average transmission time	5.245	5.185	5.240

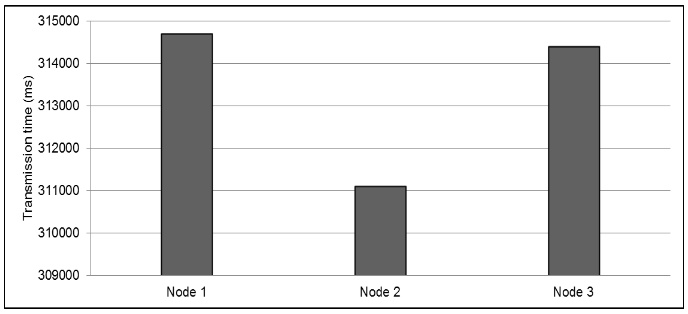

Fig. 5. Average transmission time after changing settings.

and the number of data input once to the map function was set to 10 000. As the size of data became greater, more time was taken for running the data in proportion to allow data of any size to be executed. Table 6 shows the time for completing transmission measured after applying the algorithm.

The average transmission time for the resource usage of Nodes 1, 2 and 3 is shown in Fig. 5 when the same test was carried out after changing the Hadoop environment with the settings by means of the parallel FP-Growth algorithm.

Figure 6 illustrates a graph about a comparison of results measured with default settings and the result measured with settings by means of the parallel FP-Growth algorithm. The usage of resources is calculated based on service performance (*SP*) including CPU (*Ci*), Memory (*Mi*) and Storage (*Si*). It can be expressed in Eq. (4) as below.

$$QoS[SP] = \sum_{n} SP \begin{bmatrix} Ci \\ Mi \\ Si \end{bmatrix}. \tag{4}$$

The change in the resource usage is within the average standard deviation, which is not significant.

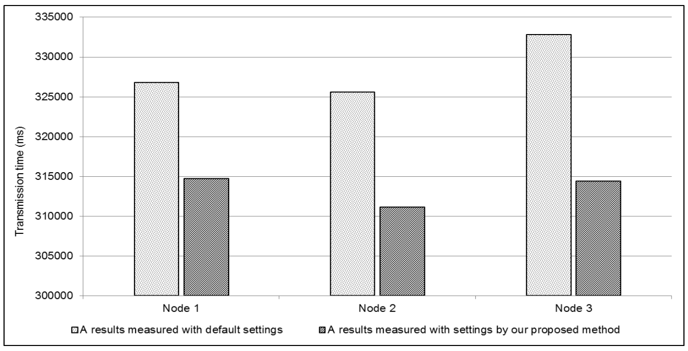

Fig. 6. A comparison of transmission time.

The node 2 which is finally on the first place among the nodes had a high level of information including CPU of 4 GHz, the memory of 4 GB, the hard disk of 300 GB, and the network card of 100 Mbps. Finally, it is seen the resource usage of the optimization selection node was stably kept the average approximately 23.2%. Change in the resource usage was within the average standard deviation which implies insignificant change. As a result, although Node 1 was rather on a low place when the recent data was used for analysis, Node 1 was on the highest place because of low node loads on the average and less changes.

5. Conclusion

This study proposed a method of providing large contents services by means of a method of managing contents for constituting an efficient cloud environment for large contents services in a cloud computing environment, and for implementing compatibility among the cloud computing platforms, and by means of distributed parallel processing. To this end, this study proposed a method of allocating resources for constituting an efficient cloud computing environment so as to provide high performance services for large contents on the basis of the Hadoop platform at IaaS (Infrastructure as a Service), e.g., CPU, networking and storages. The proposed method uses MapReduce programming and association rules used in detecting hidden patterns in data mining so as to implement high performance contents processing. The proposed method was applied to the Hadoop file system for cloud computing. As a result, the proposed method achieved improved performance and speed by more than 23.2% by applying the resource allocation method to which the parallel FP-Growth algorithm proposed in this study was applied, as compared to the environment in which resources were allocated with existing settings. This study demonstrated that the processing time can be reduced, which increases with continuously increasing data processing processes for data processing and analysis which is a challenge for high performance data processing in a cloud computing environment for large contents services.

Acknowledgements

This work was supported by the IT R&D program of MSIP/KCA [12-921-06-001, "Development of MTM-based Security Core Technology for Prevention of Information Leakage in Smart Devices"].

References

Azza, A., Kamil, B., Daniel J.A., Avi, S., Alex, R. (2009). HadoopDB: an architectural hybrid of mapreduce and dbms technologies for analytical workloads. *Proceedings of the VLDB Endowment: VLDB Endowment Hompage Archive*, 2(1), 922–933.

Chaiken, R., Jenkins, B., Larson, P.A., Ramsey, B., Shakib, D., Weaver, S., Zhou, J. (2008). Scope: easy and efficient parallel processing of massive data sets. *Proceedings of the VLDB Endowment: VLDB Endowment Hompage Archive* 1(2), 1265–1276.

Chu, C.T., Kim, S. K., Lin, Y.A., Yu, Y., Bradski, G., Ng, A. Y., Olukotun, K. (2007). Map-reduce for machine learning on multicore. *Neural Information Processing Systems*, 19, 281–288.

Dean, J., Ghemawat, S. (2010). MapReduce: a flexible data processing tool. *Communications of the ACM*, 53(1), 72–77.

Grossman, R.L., Gu, Y. (2008). Data mining using high performance clouds: experimental studies using sector and sphere in KDD. In: *KDD'08 Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 920–927.

Hung, C.Y., Ali, D., Ruey-Lung, H., Stott, P.D. (2007). Map-reduce-merge: simplified relational data processing on large clusters. In: *SIGMOD'07 Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, pp. 1029–1040.

Iko, P., Masaru, K. (2003). Parallel fp-growth on pc cluster. In: *PAKDD'03 Proceedings of the 7th Pacific–Asia Conference on Advances in Knowledge Discovery and Data Mining*, vol. 2637, pp. 467–473.

Isard, M., Budiu, M., Yu, Y., Birrell, A., Fetterly, D. (2007). Dryad: distributed data-parallel programs from sequential building blocks. In: *EuroSys'07 Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007*. ACM SIGOPS Operating Systems Review – EuroSys'07 Conference Proceedings 41(3), pp. 59–72.

Jeanna, M., Tal, G., Christofer, H., Jeff, W. (2009). Virtual machine contracts for datacenter and cloud computing environments. In: *ACDC '09 Proceedings of the 1st Workshop on Automated Control for Datacenters and Clouds*, pp. 25–30.

Jochen, L., Leidner, Gary, B. (2009). Building and Installing a Hadoop/MapReduce Cluster from Commodity Components technical report, pp. 1–15, `http://arxiv.org/ftp/arxiv/papers/0911/0911.5438.pdf`.

Kang, U., Tsourakakis, Faloutsos, C. (2009). PEGASUS: a peta-scale graph mining system implementation and observations. In: *Proceedings of the 9th IEEE International Conference on Data Mining*. IEEE Computer Society, Washington, pp. 229–238.

Karloff, H., Suri, S., Vassilvitskii, S. (2010). A model of computation for MapReduce. In: *SODA '10 Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 938–948.

Lamine, M.A., Nhien-An, L., Tahar, M.K. (2007). Distributed frequent itemsets mining in heterogeneous platforms. *Engineering, Computing and Archtecture*, 1(2).

Lammel, R. (2008). Google's MapReduce programming model – revisited. *Science of Computer Programming*, 70(1), pp. 1–30.

Li, H., Wang, Y., Zhang, D., Zhang, M., Chang, E.Y. (2008). PFP: parallel FP-growth for query recommendation. In: *RecSys '08 Proceedings of the 2008 ACM Conference on Recommender Systems*. ACM, New York, pp. 107–114.

Li, L., Eric L., Yimin, Z., Zhizhong, T. (2007). Optimization of frequent itemset mining on multiple-core processor. In: *VLDB '07 Proceedings of the 33rd International Conference on Very Large Data Bases*, pp. 1275–1285.

Luis, M.V., Luis, R., Juan, C., Maik, L. (2009). A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, 39(1), pp. 50–55.

Mladen, A.V. (2008). Cloud computing – issues, research and implementations. *Journal of Computing and Information Technology*, 16(4) pp. 235–246.

Papadimitriou, S., Sun, J. (2008). DisCo: distributed co-clustering with map-reduce: a case study towards peta byte-scale end-to-end mining. In: *Proceedings of the 8th IEEE International Conference on Data Mining*, IEEE Computer Society, Washington, pp. 512–521.

Pavlo, A., Rasin, A., Madden, S., Stonebraker, M., DeWitt, D., Paulson, E., Shrinivas, L., Abadi, D.J. (2009). A comparison of approaches to large scale data analysis. In: *SIGMOD '09: Proceedings of the 35th SIGMOD International Conference on Management of Data*, pp. 165–178.

Thusoo, A., Sarma, J.S., Jain, N., Shao, Z., Chakka, P., Anthony, S., Liu, H., Wyckoff, P., Murthy, R. (2009). Hive: a warehousing solution over a map-reduce framework. *Proceedings of the VLDB Endowment*, 2(2), 1626–1629.

J. Choi received a doctoral degree in the Department of Computer Engineering at Chosun University of Korea in 2004. Currently, he is working as a lecturer at the same university. His research interests include multimedia processing, semantic information processing, system security, ontology and semantic web.

C. Choi received a doctoral degree in the Department of Computer Engineering at Chosun University of Korea in 2012. Currently, He is working as a lecturer at the same university. His research interests include semantic information processing, semantic web, multimedia and system security.

K. Yim received his Ph.D. degree in the Department of Electronics Engineering from Ajou University, Suwon, Korea in 2001. He is an associate professor in the Department of Information Security Engineering, Soonchunhyang University and he is currently a visiting professor at Purdue University in 2011. His research interests include vulnerability assessment, malware analysis, access control mechanism, systems security and video surveillance system.

J. Kim received her MS degree and PhD in Computer Engineering from Chungnam National University, Republic of Korea, in 2000 and 2004, respectively. She studied at computer science from the University of California, Irvine, USA in 2005. Since 1988, she has been a principal member of engineering staff at the Electronics and Telecommunications Research Institute (ETRI), where she is currently working as a team leader of the Mobile Security Research Team. Her research interests include mobile security, secure operating system, network security and system security.

P. Kim received the BS degree in computer engineering from Chosun University of Korea and the MS and PhD degrees in computer engineering from Seoul National University of Korea in 1994. He is a full professor in the Department of Computer Engineering at Chosun University. His specific research interests include semantic web techniques, semantic information processing and retrieval, multimedia processing, semantic web and system security.

Skaičiuojamųjų išteklių debesų kompiuterijoje perkonfigūravimo metodas daugialypių duomenų tiekimui

Junho CHOI, Chang CHOI, Kangbin YIM, Jeongnyeo KIM, Pankoo KIM

Vis dažniau naudojami dideli daugialypiai duomenys ir vis daugiau žmonių naudoja debesų kompiuterijos technologijas. Būtina veiksmingai valdyti didelius duomenis ir turėti omenyje daugialypių duomenų perdavimo efektyvumą esant skirtingai kokybei. Yra svarbu užtikrinti veiksmingą debesų kompiuterijos išteklių (procesorių, tinklo ir saugyklų) paskirstymą, dėl to įvairūs paskirstymo algoritmai yra reikalingi. Šis tyrimas siūlo MapReduce taikymo schemos projektavimo metodą dažnų sekų paieškos augimo algoritmui, kuris yra vienas iš duomenų tyrybos metodų Hadoop platformoje infrastruktūros kaip paslaugos etape, įskaitant procesoriams, tinklams ir saugykloms. Pasiūlytas metodas skirtas ištekliams paskirstyti naudojant šią schemą.