I.J. Information Engineering and Electronic Business, **2018, 4, 1-7**
Published Online July 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijieeb.2018.04.01

Map Reduce and Match Aggregate Pipeline Performance Analysis in Metadata Identification and Analysis for Document, Audio, Image, and Video

Mardhani Riasetiawan
Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences,
Universitas Gadjah Mada, Yogyakarta, Indonesia
Email: mardhani@ugm.ac.id

Received: 14 July 2017; Accepted: 22 September 2017; Published: 08 July 2018

Abstract—The study observes the metadata identification and analysis for Document, Audio, Image, and Videos. The process uses MapReduce and Match Aggregate Pipeline to identify, classify, and categories for identification purposes. The inputs are FITS array results and processed in form of XML. The works consist of the extraction process, identification and analysis, classification, and metadata information. The objective is establishing the file information based on volume, variety, veracity, and velocity criteria as part of task identification component in Self-Assignment Data Management. Testing is done for all file types with the number of files and the size of the file according to the grouping. The results show that there is a pattern where the match-aggregate-pipeline has a longer processing time than MapReduce on a small block size, shown in a block size of 64 Mb, 128 Mb, and 256 Mb. But once the block size is magnified the match-aggregate-pipeline has faster processing time at 1024 Mb and 2048 Mb. The results have a contribution in the metadata processing for large files can be done by arranging the block sizes in Match Aggregate Pipeline.

Index Terms—Metadata, Document, Audio, Image, Video, MapReduce, Match Aggregate Pipeline, FITS, Self-Assignment Data Management.

I. Introduction

Metadata can be defined as data about data or data that describes data. The metadata document contains information that describes the characteristics of a data, particularly its content, quality, condition, and manner of acquisition. Metadata is used one of them to document the resulting data product and answer the fundamental questions about who, what, when, where and for what data is created or prepared. Metadata plays an important role in the search mechanism and exchange of data.

Descriptive metadata identifies sources of information to facilitate resource discovery and selection. This data includes elements such as authors, titles, public year, subject or keyword headers and other information typically recorded in traditional categorization processes. In the library environment a bibliographic listing is made based on ISBD (International Standard Bibliographic Description) [1], AACR [2], classification charts such as DDC (Dewey Decimal Classification) [3], UDC (Universal Decimal Classification) [4], Library of Congress Classification [5], subject header lists that produce a representative document Document representation or document surrogate) standard that serves as a bibliographic listing.

Administrative Metadata provides information for the management of information resources, such as when and how it was created, file type, other technical data, and who owns it, and who is eligible to access it. Administrative metadata includes data concerning intellectual property rights and subsequent (rights management metadata), archiving and preservation metadata.

Structural metadata describes how a digital object is structured so that it can be combined into a single logical whole. Digital sources such as books, for example, consist of several chapters, and each chapter consists of pages each of which is a separate digital file. Structural metadata is needed to find out the relationship between physical files and pages, pages and chapters, and chapters with books as the final product. This then allows the software to display the contents of the book and instantly bring up the selected chapter (by click) by the user or navigate to other sections (pages) of the book. Multimedia objects consisting of audio and text components need to be synchronized, and for this, there must be structural metadata.

Metadata can be used to describe different types of data, from textual data, images, or reports that contain both. Metadata can also be entered directly in the data, such as HTML documents, existing metadata already embedded directly in the HTML data. This explains that metadata can be stored simultaneously with the data or information described or may be separate. Usually, metadata is stored separately with the data, this built metadata is directly related to the data created and any changes in the data must also be updated in the metadata. There is much that can be done with metadata. Metadata is built according to its purpose. The general purpose of metadata development is to provide an explanation that allows the search process data can be done.

This research focuses on job/task identification process using metadata for the purpose of setting resource allocation at Data Center. Metadata is analyzed and grouped with parameters based on service orientation in the form of large data services such as volume, variety, veracity, and velocity [6]. This identification process is carried out to provide opportunities in displaying service characteristics that are based on a specific task/jobs database in this case metadata, which is used is limited to several other Data Center management approaches. Task/jobs characteristic information is used in the task management component for resource allocation determination.

The study compares the MapReduce and Match Aggregate Pipeline in metadata identification and aggregation process for Document, Image, Audio, and Video. The process works based on metadata extraction has done by FITS [7]. The identification and classification have purposed of generateing volume, variety, velocity, and veracity classification. These parts use for supporting the identification of tasks in resource allocation process.

The paper consists of several sections. The Related Works explains the studies that build the research construction in metadata uses. The Research Works Section describes the research process based on Map Reduce and Match Aggregate Pipeline. Section IV explains the implementation of Match Aggregate Pipeline methods. The Results Section shows the performance and its analysis between methods. The conclusion and future works are the last section in this paper.

II. Related Works

The research by building workflow architecture on Data Center components by proposing enhanced scheduling algorithm [8]. Specifically designed specifically for managing data-intensive applications. Pandey conducts research by conducting a comprehensive technical mapping of scheduling and proposes an architecture that incorporates a data management component. Implementation is run on Functional Magnetic Resonance Imaging (fMRI) and Evolutionary Multi-objective Optimization Algorithm. Implementation is run by using a distributed Grid architecture, proposing several heuristic algorithms that take into account the time and cost incurred to transfer data. The research conducted a review and study on cloud computing approach for Data Center [9]. The main challenge that arises is on the issue of resource management. Research conducted by Teng combines the theory of scheduling by adjusting the hierarchy of the Data Center to address the different needs of cloud services. The research undertaken results in solving the problem of resource allocation at the user level. Teng proposed the theoretical algorithm game to predict the offer and auction pricing (capacity). By using Bayesian Learning, resource allocations can reach Nash Equilibrium even among non-cooperative users. Research conducted by Teng solves the problem of task scheduling problem at system-level Data Center with an on-line schedulability test which implemented in MapReduce. Of particular interest is the relationship between cluster utilization and map reduction ratios in MapReduce. Teng's research also completed an on-line evaluation model for probability tests that can show the overall system utilization. Teng's approach runs on MapReduce and cluster-specific environments to handle diverse services still require implementation in heterogeneous environments and real Data Centers. The proposed algorithm approach approximates the approach to service oriented and focuses on map reduction on MapReduce mechanism. The performance of the resources has not been analyzed in real terms since test-schedulability focuses on the scheduling mechanism in the mapping mechanism and reduces.

The research on a resource allocation strategy in virtual machines with a re-packaging approach for cloud-scale trade-offs, both horizontally and vertically [10]. This study analyzes the performance results of different VMS strategies applied. By using re-packaging approach combined with auto-scaling on virtual machines. This study has the optimal set identification, transition policy, reconfiguration cost, and decision making. The research also combines with vertical and horizontal elasticity for Virtual Management System. Re-packaging approach uses cost-benefit for implementation and determines the vertical and horizontal scale. This approach is suitable for resource management models that take into account the scalability aspects of Data Center services.

The research presents a load balancing approach for building adaptive and scalable load balancing used for server cluster metadata in cloud file systems [11]. Research introduces Cloud Cache as an adaptive and scalable load balancing mechanism. Inside the cloud cache, use adaptive diffusion and replication to collect load characteristics. Components communicate with distributed metadata management to build load balancing performance in an efficient way. Cloud Cache takes two steps: running adaptive cache diffusion and adaptive replication schemes.

The allocation of resources in a cloud computing environment can be presented based on preference [12]. Cloud computing is identical to Pay-as-usage, where users pay rent for several periods and pay according to user behavior. This study introduces a market-driven tender mechanism with the aim of identifying the allocation of user resources based on capacity. Capacity determines in payment based on buyer's service preferences. This research uses a system model consisting of resource allocation unit, auction subunit, and subunit payment. Test models are performed in CloudSim environment with multiple instances, single resource providers, and multiple

III. Research Works

Self-Assignment Data Management (SADM) is the process of identifying, analyzing and classifying metadata to generate classified metadata structures based on volume, variety, veracity, and velocity. SADM consists of metadata extraction, analysis, classification, and XML output metadata. Data sent or present in the Data Center will be extracted to be able to release metadata and content information from the data. The extract process will use the standard of FITS. FITS is used because it has a standard Dublin Core Metadata Standard extraction and has a complete method. Extracted results will be stored into storage in certain mechanisms that store the extracted metadata and the content in the database structure. Metadata will be identified and analyzed from the extraction process to be used as the basis for classification of metadata structure based on volume, velocity, variety, and veracity. The process of analysis and classification using match-aggregate-pipeline.

SADM has the components shown in figure 1, with components:

1. Metadata Extraction Process The process of extracting metadata to obtain items to be used in the classification process.
2. Metadata Identification and analysis Process through identification and analysis of metadata items obtained by using match process.
3. Metadata Classification and Storing The process of classifying the results of a match process and followed by an aggregate-pipeline process that results in a classification structure. Results and processes will be stored in unstructured databases.
4. Metadata Information. The results of self-assignment data management in the form of metadata-XML that will be used as information for task identification in the next process.

Metadata extraction process is done by several stages, namely collecting metadata structure from document group first, extraction process, verification of extraction result. The metadata extraction process in this study uses the selected FITS extraction tool because it has metadata items that are structured into info files, status files, identification, and metadata. Each structure has metadata items that are useful for the formation of metadata structures that are the object of this study. In the process of extraction of metadata on data or files is done by using FITS extraction tools. FITS performs the extraction process on the data or files that generate metadata structure information that is divided into 5 parts, namely File Identification, File Info, File Status, Metadata and Fits Execution Time. The metadata structure generated by FITS has two parameters: parameters and objects. Parameters are groups of metadata consisting of identification, info file, status file, and metadata. Object parameters contain metadata items of each parameter that may vary depending on the data or files.

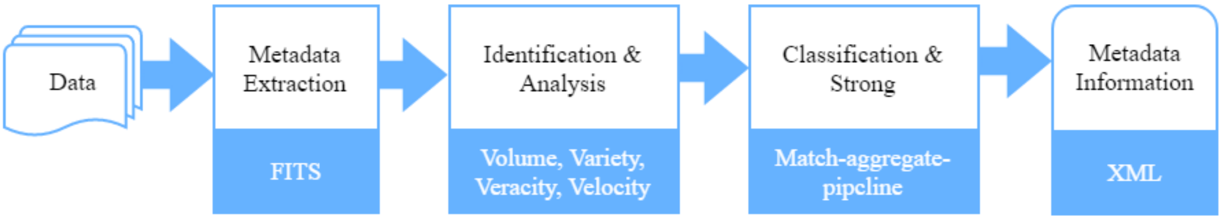


Fig.1. Extraction & Analysis process

The extraction process starts with FITS recognizing the data or files to be processed. This tool acts as a wrapper, requesting and managing the output of some other open source tools, namely DROID, Apache Tika, Jhove, Exiftool, NLNZ, FFIdent, and File Utility. The output of the tools is converted into XML form, then compared to each other and merged in the form of a single FITS XML file.

The workflow that occurs in FITS consists of several stages of the process. First, FITS reads the file then determines which tool to call. This process will affect the end result of the output. The next process, every tool, such as DROID, Apache Tika, Jhove, Exiftool, NLNZ, FFIdent, and File Utility, are called in parallel to perform the extraction process on the file. FITS will then convert all the extraction results from each tool into the FITS XML form. The last process, all XML files are merged into a single FITS XML and then converted into standard XML form. FITS converts the original output of any existing tools, then merged into a format called FITS XML

The classification of metadata into 4 parameters ie volume, variety, veracity and velocity compares between the match-aggregate-pipeline method and MapReduce. Testing is done by taking the metadata parameter object of extraction and metadata retrieval used as standard for sample test. Metadata retrieval is based on:

1. Compatibility of parameter objects including Volume, Variety, Velocity, Variety, and Value. It is based on the conformity of the definition of each parameter object with from Mckinsey (2010) [16].

2. The number of metadata parameter object occurrences of all uploaded files. Metadata objects that often appear in each file, into consideration to be the object of mapping parameters. With a note, the definition of the parameter object corresponds to one of the Big Data definition principles

The test is performed to evaluate the method used in the extraction process and the match-aggregate-pipeline process shown in Figure 2 compares the extraction methods using the JHOVE, DROID, NLNZ ME, ExifTool, FileUNity, and FFIdent methods. Compare the metadata classification between the match-aggregate-pipeline and MapReduce methods as shown figure 2.

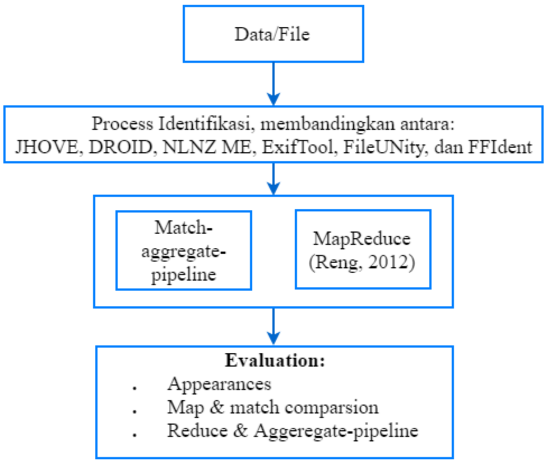

Data/File
Process Identifikasi, membandingkan antara:
JHOVE, DROID, NLNZ ME, ExifTool, FileUNity, dan FFIdent
Match-aggregate-pipeline
MapReduce (Reng, 2012)
Evaluation:
Appearances
Map & match comparsion
Reduce & Aggeregate-pipeline

Fig.2. Evaluation Process

IV. Match Aggrehate Pipeline Implementation

The identification and classification process uses a match-aggregate-pipeline approach implemented in the MongoDB Database in an unstructured format. The match process is a metadata parameter search of a standard XML FITS result that has been converted into an array, shown in figure 3. XML Standard results from FITS that have been converted into metadata arrays and entered into the database which then performed further analysis process with aggregate technique. This process performs the process of grouping metadata parameters present in the file, group and all the files into the parameters that belong to each file. The result of matching process is identification, file info, status file, output tool, statistic, and metadata group in categories document, image, audio, and video, shown in figure 4. The purpose of this stage is to know the mapping of metadata which can later be used to form XML Big Data as output with the base of the file group, as shown in figure 5. In the next stage the process of aggregating aggregation results from identification, info file, status file, output tool, statistic, and metadata into metadata structure based on document, image, audio, and video. Pipeline then grouped previously grouped files, converted into metadata structures based on volume, variety, veracity, and velocity.

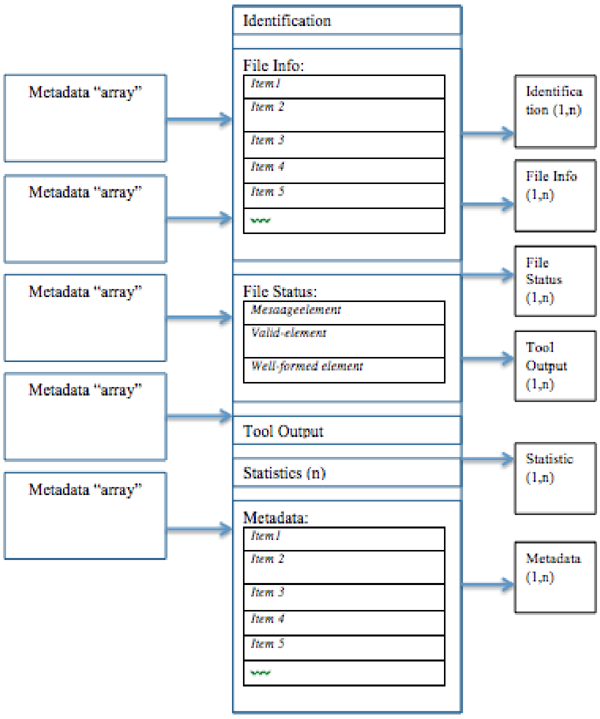


Fig.3. Metadata Extraction

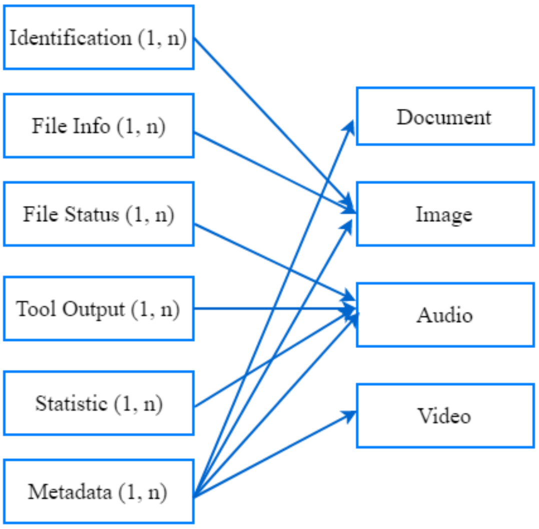


Fig.4. Metadata Match-Aggregate

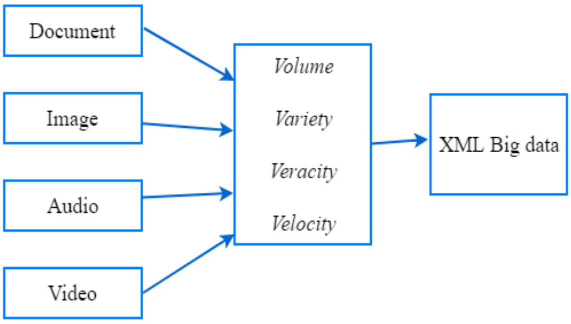


Fig.5. Metadata Pipeline

V. Results and Discussions

Evaluations were conducted on test results that focused on the conformity of parameter objects including volume, variety, velocity, and variety. The number of metadata parameter object occurrences of all uploaded files, shown in Table 1. Metadata objects that often appear in each file, into consideration to be the object of mapping parameters. There is a metadata parameter object with a low number of occurrences but remains a mapping paremeter. This happens because the object of the parameter becomes one component of the compiler of another parameter object. Thus, the two object parameters, although not meeting the standard number of the appearance of the object, can still be the parameter object for mapping to the 4V parameter.

Table 1. File Type and Numbers

	File Number	(%)
Document	324,567	43.17
Image	195,672	26.03
Audio	167,432	22.27
Video	64,094	8.53
Total	751,765	100.00

The metadata matching process is used to group metadata that can be mapped into categories based on volume, variety, veracity, and velocity. Table II shows the time required to run a match-aggregate-pipeline process compared to MapReduce by setting block size size of 64 Mb, 128Mb, 256Mb, 512Mb, 1024Mb, and 2048Mb. Testing is done for all types of files with junlah file and size of file size according to the grouping done. The results show that there is a pattern where the match-aggregate-pipeline has a longer processing time than MapReduce on a small block size, shown in block size of 64 Mb, 128 Mb and 256 Mb. But once the block size is magnified the match-aggregate-pipeline has faster processing time even though the resulting difference is not too large at 1024 Mb and 2048 Mb, shown in figure 6.

Figure 7 shows the time required to run a match-aggregate-pipeline process compared to MapReduce by setting block size size of 64 Mb, 128Mb, 256Mb, 512Mb, 1024Mb, and 2048Mb for Image types. Testing is done for all file types with the number of files and the size of the file according to the grouping done. The results show that there is a pattern where the match-aggregate-pipeline has a longer processing time than MapReduce on a small block size, shown in block size of 64 Mb, 128 Mb and 256 Mb. But once the block size is magnified the match-aggregate-pipeline has faster processing time even though the resulting difference is not very large, at 1024 Mb and 2048 Mb.

Table 2. Comparisons between Match Aggregate Pipeline and Map Reduce

Files	Block size	Match/map (ms)	
		Match-Aggregate-Pipeline	Map Reduce
Document 324567 (files) 31.51701795 TB	64Mb	96801	93876
	128Mb	92376	90386
	256Mb	88675	88034
	512Mb	87492	87563
	1024Mb	81654	81875
	2048MB	61365	64786
Image 195672 (files) 19.00069304 TB	64Mb	43810	44231
	128Mb	42116	42520
	256Mb	40924	41317
	512Mb	38967	39341
	1024Mb	37452	37812
	2048MB	34251	34580
Audio 167432 (files) 6.681556071 TB	64Mb	32897	33213
	128Mb	31867	32173
	256Mb	37943	38307
	512Mb	37521	37881
	1024Mb	36832	37186
	2048MB	35901	36246
Video 64094 (files) 2.557740783 TB	64Mb	16665	16825
	128Mb	16866	17028
	256Mb	16675	16835
	512Mb	16591	16750
	1024Mb	16354	16511
	20148Mb	16165	16320

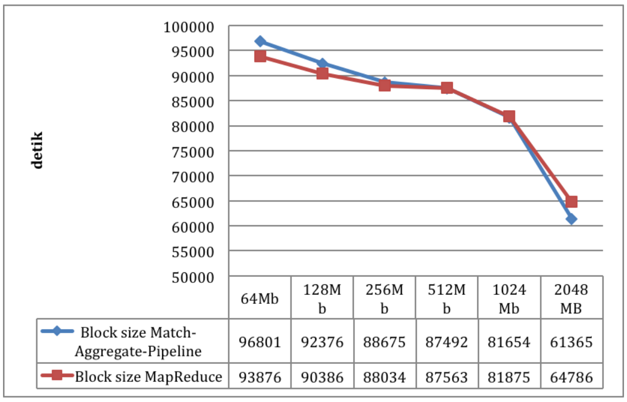
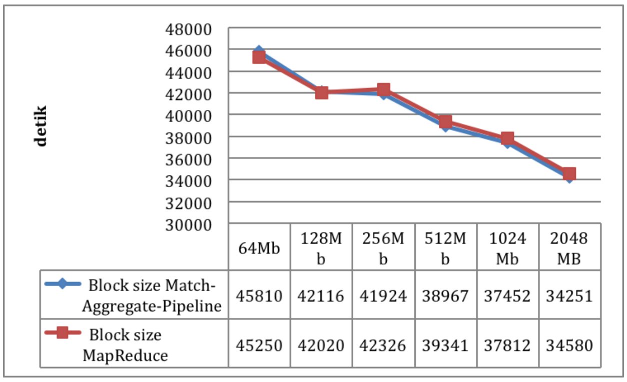


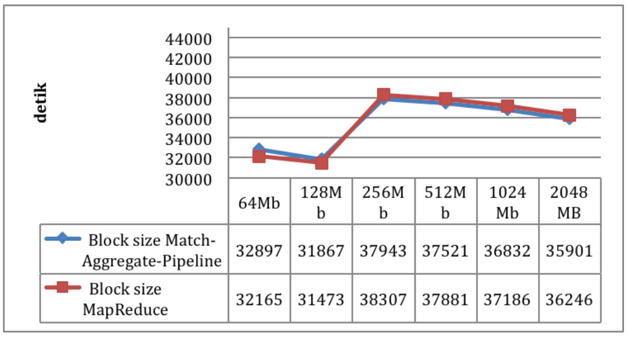
Fig.6. Match Aggregate Pipeline and Map Reduce Comparison in Documents



	64Mb	128Mb	256Mb	512Mb	1024 Mb	2048 MB
Block size Match-Aggregate-Pipeline	45810	42116	41924	38967	37452	34251
Block size MapReduce	45250	42020	42326	39341	37812	34580

Fig.7. Match Aggregate Pipeline and Map Reduce Comparison in Images

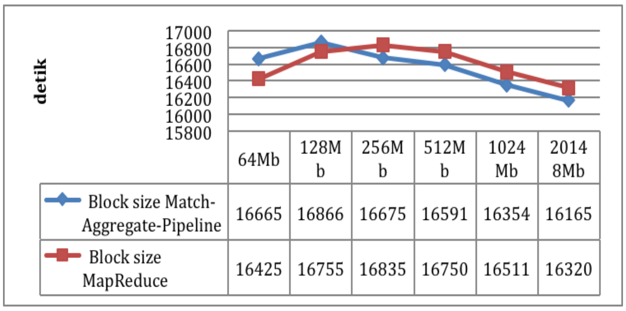
Figure 8 shows the time required to run a match-aggregate-pipeline process compared to MapReduce by setting the block size size of 64 Mb, 128Mb, 256Mb, 512Mb, 1024Mb, and 2048Mb for the Audio type. Testing is done for all types of files with junlah file and size of file size according to the grouping done. The results show that there is a pattern where the match-aggregate-pipeline has a longer processing time than MapReduce on a small block size. But once the block size is magnified the match-aggregate-pipeline has a faster processing time even though the resulting difference is not too large.



	64Mb	128Mb	256Mb	512Mb	1024 Mb	2048 MB
Block size Match-Aggregate-Pipeline	32897	31867	37943	37521	36832	35901
Block size MapReduce	32165	31473	38307	37881	37186	36246

Fig.8. Match Aggregate Pipeline and Map Reduce Comparison in Audios

Figure 9 shows the time required to run a match-aggregate-pipeline process compared to MapReduce by setting the block size size of 64 Mb, 128Mb, 256Mb, 512Mb, 1024Mb, and 2048Mb for Video types. Testing is done for all types of files with junlah file and size of file size according to the grouping done. The results show that there is a pattern where the match-aggregate-pipeline has a longer processing time than MapReduce on a small block size. But once the block size is magnified the match-aggregate-pipeline has a faster processing time even though the resulting difference is not too large



	64Mb	128Mb	256Mb	512Mb	1024 Mb	2014 8Mb
Block size Match-Aggregate-Pipeline	16665	16866	16675	16591	16354	16165
Block size MapReduce	16425	16755	16835	16750	16511	16320

Fig.9. Match Aggregate Pipeline and Map Reduce Comparison in Videos

VI. Conclusion

The development of jobs identification methods by utilizing metadata (Self-Assignment Data Management), which can be used by the Data Center workflow component to determine resource allocation. The identification process is done on task / jobs in the form of file submission with Document type, Image, Audio, and Video. SADM method consists of metadata extraction process with FITS, identification and classification with match-aggregate-pipeline, and generate metadata information in the form of XML.

The research succeeded in identifying tasks / jobs based on the metadata of file submission by using FITS, match-aggregate-pipeline, and generating XML containing information based on volume, variety, veracity and velocity parameters. The test was performed on metadata extraction method with JHOVE, DROID, NLNZ ME, ExifTool, FileUNity, and FFIdent obtained the result of evenly distributed metadata parameters for all extraction methods. The SADM method has better performance in processing data in the larger block size use Match Aggregate Pipeline compared to MapReduce.

The job/task identification process uses metadata for the purpose of setting resource allocation at the Data Center. Metadata is analyzed and grouped with parameters based on service orientation in the form of large data services such as volume, variety, veracity, and velocity (Manyinka et al., 2010). This identification process is carried out to provide opportunities in displaying service characteristics that are based on a specific task database in this case metadata, which is used is limited to several other Data Center management approaches. The tasks characteristic and information is used in the task management component for resource allocation determination.

ACKNOWLEDGMENT

The authors wish to thank Cloud and Grid Technology Working Group Universitas Gadjah Mada (cloud.wg,ugm.ac.id) for supporting and computing facilities.

REFERENCES

[1] ISBD(G): General International Standard Blibliographic Description, International Federation of Library Associations and Institutions, 2004.

[2] Jones, W., Ahronheim, J.R., Crawford, J., Cataloging the Web: Metadata, AACR, and MARC 21, ALCTS, July, 2000.

[3] Dewey Decimal Classification, OCLC, accessed at https://www.oclc.org.

[4] Universal Decimal Classification, UDC Consortium, accessed at www.udc.org.

[5] Library of Congress Classification, LOC, accessed at https://www.loc.gov/catdir/cpso/lcc.html.

[6] Manyinka, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H., 2011, Big Data: The Next Frontier for Innovation, Competition, and Productivity, *McKinsey Global Institute 2011 Report*, [Online], May 2011.

[7] FITS, File Information Tool Set, Harvard, accessed at https://projects.iq.harvard.edu/fits/home.

[8] Pandey, S., 2010, Scheduling and Management of Data Intenesive Application Workflows in Grid and Cloud Computing Environments, *Dissertation*, Department of Computer Science and Software Engineering, The University of Melbourne, Australia.

[9] Teng, F., 2012, Management Des Donnees Et Ordinnnancement Des Taches Sur Architectures Distributes, Desertation, Ecole Cenrale Paris Et Manufactures, Centrale Paris.

[10] Sedaghat, M., Rodriguez, F. H., Elmroth, E., 2013, *A Virtual Machine Re-packaging Approach to the Horizontal vs. Vertical Elasticity Trade-off for Cloud Autoscaling* of The 2013 ACM Cloud and Autonomic Computing Conference*.*

[11] Xu, Q., Arumugam, R. V., Yong, K. L., Wen, Y., Ong, Y. S., Xi., W., 2015, Adaptive and Scalable Load Balancing for Metadata Server Cluster in Cloud-scale File System, *Frontier Computer Science,* vol. 9, issue 6, pp.904-918.

[12] Kumar, N., Saxena, S., 2015, A Preference-based Resources Allocation In Cloud Computing Systems, *in 3rd International Conference on Recent Trends in Computing 2015*. Procedia Computer Science, vol 57, pp. 104-111.

Authors' Profiles

Mardhani Riasetiawan born at Surakarta, Indonesia in August 28, 1979. Doctor in Computer Science from Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada, Indonesia in 2017. Master of Engineering in Information Technology from Faculty of Engineering, Universitas Gadjah Mada, Indonesia in 2007. Bachelor in Accounting from Faculty of Economic and Business, Universitas Gadjah Mada, Indonesia in 2003.

Currently, works as researcher and lecturer in Department of Computer Science and Electronics, Faculty of Mathematics and Natural Sciences, Universitas Gadjah Mada. He also as Lead Researcher in Cloud and Grid Technology Working Group (cloud.wg.ugm.ac.id). Researcher in Big Data Working Group, Universitas Gadjah Mada, Indonesia.

How to cite this paper: Mardhani Riasetiawan," Map Reduce and Match Aggregate Pipeline Performance Analysis in Metadata Identification and Analysis for Document, Audio, Image, and Video", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.10, No.4, pp. 1-7, 2018. DOI: 10.5815/ijieeb.2018.04.01