Designing real-time research data portal of JTI POLINEMA

G Budiprasetyo*, Y Yunhasnawa, M Mentari and D C Pratama
Dept. of Informatics Engineering, State Polytechnic of Malang, Malang, Indonesia

*gunawan.budi@polinema.ac.id

Abstract. Research culture in Department of Information Technology Polinema is always developed continuously every time. In addition, many research titles that are critical of environmental problems are beginning to be seen. The Information Technology Department has an outstanding focus on issues, namely smart systems which include smart education, smart living, smart healthy, smart city and smart tourism. Research resources in the Information Technology Department Polinema are competing to realize that research focus. For this reason, efforts are needed to increase the availability of facilities that are not only material. The fast response can be solved by providing Real-Time Research Data Portal. The contribution of this paper is to design of middleware on web portals that have high scalability capabilities for providing large data facilities and fast access. Real-time big data processing is the path that will be described in the architecture that will be created and adapted to the needs of the Department of Information Technology Polinema. The results of the latency evaluation test indicated that RabbitMQ allowed us to enforce concurrency, thereby reducing overall RTC by up to 50% compared to traditional HTTP.

1. Introduction

Research in Indonesia has gradually increased over time. The Indonesian government also provides a lot of support in the form of annual grants which provide research opportunities for Indonesian researchers. Researchers, namely lecturers in the industrial era 4.0 and disruptive innovation who open the need to collaborate and be open in sharing research data each other [1]. This results in the need for direction and governance in an effort to increase research. Entering the industrial era 4.0, namely the era of digitalization, there are main and very important areas in its implementation, namely the field of Information Technology (IT). One of the industries that uses IT support, especially the Internet of Things (IoT) tool in carrying out all existing activities [2]. And it turns out that real time processing supported by IoT and big data also improves services better and has an impact on increasing certain collaborations.

Back to the discussion of research in Indonesia, especially in higher education institutions like State Polytechnic of Malang (Polinema), especially the Information Technology Department (JTI), apparently needs the same amount of money in fulfilling research facilities. Real time data access for each research group, such as in Julian and Surya [3] for the Computer Network, Architecture and Data Security group research which explains about industry in droves using integrated into a single integrated network and also IoT which requires assurance [4], data security whose reality must be accessed in real time; research group computer vision requires real time data access as well as Pan et al [5] on real time surgery has performed accurate detection and tracking of corneal contours; a research group of artificial intelligence according to Ionescu et al [6] which carried out route tracking with low power consumption for robots in real time; Information Systems research group look at research Cheung et al [7] on simulation of
real time operating performance in information technology (IT) equipment of data centers; and research group multimedia real time access games used in various applications from then until now [8]. The needs and work of various kinds of group research will be better if they are collected into a centralized source at JTI-Polinema in the form of a real time web portal according to the needs in this digitalization era. JTI-Polinema does not yet have a web portal regarding real time research data. Therefore, this study aims to design a Real-time Research Data Portal JTI POLINEMA. Flexible communication systems and models are needed, because the application is naturally separated in layers and geographically. An application that has flexible communication with technology is needed to address real time data access. The data portal designed based on Hadoop using the RabbitMQ message broker to handle large amounts of data access because access is done in real-time.

2. Related work

Data portals are becoming a trend to meet current data collaboration needs. Many countries and institutions have created data portals like Koppe and Schafer [9] on the German Marine Research Data Portal which provides “one-stop-shop” condition. In addition, technology concentration is also carried out to improve the performance, efficiency and productivity of an organization using a Web-based portal [10]. Application of data storage and data processing in real time [11] on an integrated platform that builds Hadoop and Spark cluster environments that have the principle of reading data interactively can cause bottlenecks so that data cannot be accessed and uploaded quickly on the data portal. Therefore, real-time data portals require additional technology that can solve these problems. Such as processing data streams in parallel and distributed environments [12] handled by the framework as parallel processing center and message broker [13] to obtain smooth processing, fault tolerance and efficient. Message broker systems commonly used in distributed systems that perform parallel and real time processing are Kafka, RabbitMQ, ActiveMQ, etc. A study on the comparison of two types of message brokers [14] namely RabbitMQ and Kafka proved that in the basic setup consisting of one node file, single channel, single partition and unreplicated RabbitMQ performance outperformed ActiveMQ. In addition, a literature study regarding RabbitMQ and distributed system. The developer implemented RabbitMQ's clustering capabilities to improve reliability [15]. Exchange messages in bulk, getting the best results from Advanced Message Queuing Protocol (AMQP) [16]. The performance of a single queue drops significantly when the queue is mirrored across the cluster for fault tolerance (at least 10% -20% better than full mirrored) [17]. Challenge to achieve the real-time capability using Map Reduce which open source and provides off-the-shelf high scalability that can significantly shorten the processing time for big data [18]. RabbitMQ Faster generates a message from the broker (when the client receives the message) [19]. In terms of latency, both systems (Kafka and RabbitMQ) were able to provide low latency results (i.e. an average / median of about 10 ms) [20]. Technologies for processing parallel data streams and highlighting problems and experiences, scale well [21]. When a large number of users send requests to the web application at the same time, the results will be more stable if using RabbitMQ as a message-oriented middleware rather than the REST API communication method [22]. RabbitMQ as a load balancer can share the workload evenly, thereby reducing the latency time of the Naive Bayes Classification by 30.3% [23]. A distributed system on a real-time research data portal requires basic support in the form of a collection of software tools like Apache Hadoop [11].

Research Data portal carries a complex data handling architecture in the form of web observatories (WO) so that researchers can be involved with each other in the research data that has been carried out or that has been produced [24]. During its development [2], data access is needed in real-time in the era of digitalization. This has an impact on the need for quick and precise decision making. Real-time data or message access so that the infrastructure is made first so that it can interact with each other on a distributed system [25]. This can be handled using a message broker like RabbitMQ. RabbitMQ is a message broker that can control the sending and receiving of messages [15]. Rabbit MQ is a message queuing system used in middleware for distributed systems that implements Advanced Message Queuing Protocol (AMQP) [17].
3. JTI-Polinema real time research data portal design

For real-time research data designing portal JTI-Polinema, a prototype is needed to prove the latency of data access using a message broker.

3.1. Hadoop with traditional HTTP

The real-time data portal access traditionally shown in Figure 1 can be accessed by users from the front-end side via browse and will be forwarded as soon as the HTTP request is received. Then the request continued on the back-end REST access with Spring (support Java-based applications). Rest response will be given after Spring instruction is performed. Because the data is estimated to be accessed in real-time which allows for complex data transactions and operations, middleware is needed to handle it. After going through the middleware layer, the client requests large data will be continued on HDFS to be divided into fixed nodes.

![Figure 1. Technology without message broker.](image)

3.2. Design real-time research data portal of JTI POLINEMA

JTI Polinema's real-time research data portal prototype design is shown in Figure 2. However, it is realized that there are not only many transaction data problems that occur in real-time data access, but also the possibility of bottlenecks. Therefore, it is necessary to add AMQP Message Broker, especially RabbitMQ. RabbitMQ will manage the sending of requests from the client, then it will be handled with middleware on each RabbitMQ that has been installed so that it is expected to create speed of sending requests and also get uncaught in the middle of sending data process in HDFS.

![Figure 2. Technology with message broker.](image)
4. Research and discussion
To determine the performance evaluation on the design prototype made, a performance evaluation (PE) test was carried out on five (5) Different Sizes, namely: 1 KB, 10 KB, 100 KB, 1000 KB, and 10000 KB and also Four (4) Different Tasks: Store Only, MD5, Base 64, Word Count.

4.1. Performance evaluation (HTTP) over performance evaluation (HTTP and RabbitMQ)
The performance evaluation of the four types of tasks and five types of sizes for HTTP shown in table 1 and shown in table 2 for HTTP with RabbitMQ. Only HTTP has the fastest handling results for 1KB files, task store only type, which is 8268 micro sec. And the highest is the longest time that is on the data size 10000KB with the type of task Base64 Encode is 384996,25 micro sec. For HTTP with RabbitMQ the fastest handling results for 1KB files, task storage type only, which is 8191 micro seconds. And the highest is the longest time, namely the data measuring 10000KB with the type of task Base64 Encode of 192595,125 micro seconds.

Table 1. Performance evaluation (HTTP).

| File Size Each | Request Completion Time (Micro Sec.) | Average |
|----------------|-------------------------------------|---------|
| 1 KB           | Request Completion Time              |         |
| 10 KB          | 8268                                | 10092   | 10049   | 7603    | 9003    |
| 100 KB         | 8588                                | 11394   | 13856   | 9455,25 | 10823,3125 |
| 1000 KB        | 11610,75                            | 26896   | 19001   | 12343   | 17462,6875 |
| 10000 KB       | 16273,75                            | 27175,75| 42387,25| 79142   | 41244,6875 |

Table 2. Performance evaluation (HTTP and RabbitMQ).

| File Size Each | Request Completion Time (Micro Sec.) | Average |
|----------------|-------------------------------------|---------|
| 1 KB           | Request Completion Time              |         |
| 10 KB          | 8191                                | 10111   | 10111,5 | 7387,25 | 4571,5 |
| 100 KB         | 8370                                | 11797   | 13701   | 9481,2625 | 5495,15625 |
| 1000 KB        | 5860,375                            | 13540   | 9550,5  | 6251,5  | 8800,59375 |
| 10000 KB       | 8204,875                            | 13645,875| 21243,625| 39645   | 20684,8438 |

4.2. Overall latency
Overall shown in Figure 3 HTTP performance coupled with RabbitMQ shows much better latency than HTTP only. This value can be seen in the block on the far right of the chart where the yellow block is far below the blue block (almost half size). This means that the HTTP latency coupled with RabbitMQ is almost 50% with only HTTP.

Figure 3. Chart overall latency.
5. Conclusion
Each figure should have a brief caption describing it and, if necessary, a key to interpret the various lines and symbols on the figure. We proposed a prototype of Hadoop based dataset repository for the Polinemas’s Department of Information Technology. In our proposed architecture, we incorporated RabbitMQ as a message broker to reduce the overall amount of Re-quest Completion Time (RCT). We observed the capability of RabbitMQ enable us to implement concurrency, thus reducing the overall RTC up to 50% compared to traditional HTTP.

References
[1] Fourianalistyawati E, Uswatunnisa A, Arif M M, Paulus S A, Soebandhi S, Reni A and Kurniasih N 2018 Research Development in Indonesia: Ethics Committee in Open Science and Collaboration Era JPhCS 1114 1 012069
[2] Nagy J, Oláh J, Erdei E, Máté D and Popp J 2018 The role and impact of Industry 4.0 and the internet of things on the business strategy of the value chain—the case of Hungary Sustainability 10 10 3491
[3] Jang-Jaccard J and Nepal S 2014 A survey of emerging threats in cybersecurity Journal of Computer and System Sciences 80 5 973-993
[4] Porambage P, Okwuibe J, Liyanage M, Ylianttila M and Taleb T 2018 Survey on multi-access edge computing for internet of things realization IEEE Communications Surveys & Tutorials 20 4 2961-2991
[5] Pan J, Liu W, Ge P, Li F, Shi W, Jia L and Qin H 2020 Real-time segmentation and tracking of excised corneal contour by deep neural networks for DALK surgical navigation Computer Methods and Programs in Biomedicine 197 105679
[6] Ionescu L M, Mazare A, Lita A I and Serban G 2015 Fully integrated artificial intelligence solution for real time route tracking 38th International Spring Seminar on Electronics Technology (ISSE) pp 536-540 IEEE
[7] Cheung H, Wang S, Zhuang C and Gu J 2018 A simplified power consumption model of information technology (IT) equipment in data centers for energy system real-time dynamic simulation Applied energy 222 329-342
[8] Hillman M F and Subriadi A P 2019 40 Years Journey of Function Point Analysis: Against Real-time and Multimedia Applications Procedia Computer Science 161 266-274
[9] Koppe R and Schäfer A 2015 Enabling central access to marine data: Data portal German marine research OCEANS 2015-Genova pp 1-5
[10] Rajkumar S, Abraham S E and Santhi V 2017 Web based portal using a biometric interface and Android application to assist non-government organizations International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT) 898-902 IEEE
[11] Yang C T, Chen C J, Tsan Y T, Liu P Y, Chan Y W and Chan W C 2019 An implementation of real-time air quality and influenza-like illness data storage and processing platform Computers in Human Behavior 100 266-274
[12] Singh R K and Verma H K 2020 Redis Based Messaging Queue and Cache Enabled Parallel Processing Social Media Analytics Framework Journal of King Saud University-Computer and Information Sciences
[13] Nahri M, Boulmakoul A, Karim L and Lbath A 2018 IoV distributed architecture for real-time traffic data analytics Procedia computer science 130 480-487
[14] Dobbelaeere P and Esmaili K S 2017 Kafka versus RabbitMQ: A comparative study of two industry reference publish/subscribe implementations: Industry Paper Proceedings of the 11th ACM international conference on distributed and event-based systems 227-238
[15] Jones B, Luxenberg S, McGrath D, Trampert P and Weldon J 2011 RabbitMQ performance and scalability analysis, project on CS 4284
[16] Fernandes J L, Lopes I C, Rodrigues J J and Ullah S 2013 Performance evaluation of RESTful web services and AMQP protocol 2013 Fifth International Conference on Ubiquitous and
Future Networks (ICUFN) 810-815 IEEE

[17] Rostanski M, Grochla K and Seman A 2014 Evaluation of highly available and fault-tolerant middleware clustered architectures using RabbitMQ 2014 federated conference on computer science and information systems 879-884 IEEE

[18] Liu X, Iftikhar N and Xie X 2014 Survey of real-time processing systems for big data Proceedings of the 18th International Database Engineering & Applications Symposium 356-361

[19] Ionescu V M 2015 The analysis of the performance of RabbitMQ and ActiveMQ 2015 14th RoEduNet International Conference-Networking in Education and Research (RoEduNet NER) 132-137 IEEE

[20] Dobbelaere P and Sheyk Esmaiili K 2019 Industry Paper: Kafka versus RabbitMQ A comparative study of two industry reference publish/subscribe implementations

[21] Tennant M, Stahl F, Rana O and Gomes J B 2017 Scalable real-time classification of data streams with concept drift Future Generation Computer Systems 75 187-199

[22] Hong X J, Yang H S and Kim Y H 2018 Performance analysis of RESTful API and RabbitMQ for microservice web application International Conference on Information and Communication Technology Convergence (ICTC) 257-259 IEEE

[23] Nugroho A W and Kusumawardhani S 2020 Distributed Classifier for SDGs Topics in OnlineNews using RabbitMQ Message Broker Journal of Physics: Conference Series. 2nd 2019 ICERA

[24] Tiropanis T, Hall W, Shadbolt N, De Roure D, Contractor N and Hendler J 2013 The web science observatory IEEE Intelligent Systems 28 2 100-104

[25] Eugster P T, Felber P A, Guerraoui R and Kermarrec A M 2003 The many faces of publish/subscribe ACM computing surveys (CSUR) 35 2 114-131