

Scalability of Mobile Cloud Storage

<https://doi.org/10.3991/ijim.v15i21.25413>

Nur Syahela Hussien¹, Sarina Sulaiman²(✉), Abdulaziz Aborujilah¹,
Merlinda Wibowo³, Hussein Samma²

¹Universiti Kuala Lumpur Malaysian Institute of Information Technology,
Kuala Lumpur, Malaysia

²Universiti Teknologi Malaysia, Johor, Malaysia

³Institut Teknologi Telkom Purwokerto, Jawa Tengah, Indonesia
sarina@utm.my

Abstract—Today, there are high demands on Mobile Cloud Storage (MCS) services that need to manage the increasing number of works with stable performance. This situation brings a challenge for data management systems because when the number of works increased MCS needs to manage the data wisely to avoid latency occur. If latency occurs it will slow down the data performance and it should avoid that problem when using MCS. Moreover, MCS should provide users access to data faster and correctly. Hence, the research focuses on the scalability of mobile cloud data storage management, which is study the scalable on how deep the data folder itself that increase the number of works.

Keywords—mobile cloud storage, scalability, data management

1 Introduction

The main issues related to data management are the scalability of data and storage data [1], [2], [3]. In Mobile Cloud Storage (MCS) it is important to maintain the scalability even users access the data file with many cloud storage services at one time. Scalable and consistent data management is a challenge that has opposed database research and the public for more than twenty years. It deals with petabytes of data, serves online requests for stringent latency and availability requirements as well as causes the workloads. If this problem is not handled well, it will decrease the performance of CC and consequently the industry, which no longer uses this system.

MCS is important to maintain scalability even users access the data file with multiple cloud storage at one time. For instance, access Dropbox, Google Drive, Box, and One-Drive at the same time. The scalability needs to be able to maintain the performance of mobile Web pre-fetching. In this research, the scalable is on how deep the data folder itself increases the number of works. The deepness of the data folder is referred to as the data level. The highest data level is the deepness of the data folder. If users using the normal MCS, the scalability is worse because it takes a long time for users to access

the data. The purpose of this research is to enhance the scalability in handling several data in MCS environment.

2 Related work

The performance of scalability is measured based on response time [2], [4], [5]. To enhance service scalability and reduce the risk of performance degradation is by handling a larger number of data. Besides, fault tolerance mechanisms affect the system scalability and usability [4], [5], [6], [7], [8]. According to Rimal *et al.* [7], Microsoft Azure had an outage of about 22 hours in March 2008 that leads to critical issues if the downtime cannot be handled in Cloud Computing (CC) environment. Every year, thousands of websites need to face unpredicted downtime and hundreds of systems interruptions. Hence, the critical issue for CC is how to minimise such outages or failover to provide the scalability of the services. The issues of cloud scalability are investigated in this research and the performance measure is based on response time. Other researchers have used response time as a performance measure for cloud scalability [5], [6]. Due to concern in scalability, as well as performance and reliability arising from the management of multiple service queues and a large volume of service requests, centralised approaches are not suitable and architectures using service monitoring and management services are needed.

3 Material and methods

Besides MCS gives scalability, reliability, and efficiency in performance. Scalability is a key aspect of MCS [3], [9–11], which offers unlimited processing and storage capacity. It allows developers and IT operations to develop, deploy and run applications at the software level of CC, which easily grow in capacity, work fast and never fail. CC is reliable by enabling access to applications and documents anywhere in the world via the Internet. Therefore, users can easily access their applications anywhere. Moreover, CC provides efficient service by allowing an organisation to free up resources to focus on innovation and product development.

The questionnaire was distributed to 69 people but only 15 people were selected among them to measure the scalability of Mobile Cloud Storage (MOBICS), the latency per page ratio, precision, and recall. Only 15 users were selected from 69 respondents to test the scalability of MOBICS. The Nielson Norman group indicated that 5 users could uncover 85% of the major usability issues, and 15 users could find 100% [12] [13]. In addition, Ginny Redish stated that about 6 to 12 users needed to be involved in doing product testing and the testing worked on at the end of the process. Therefore, this research proposed to evaluate the scalability of MOBICS based on 15 users, the best number of respondents.

The scalability test was conducted by comparing different loads, which were data load level 1 to data load level 5 as shown in Figure 1. For example data level 1 consisted sort of data files, which were data 1, 2, 3, 4, and others. Then, data 1.1 in data level 2 was referred from data 1 from data level 1. Data 1.1.1 in data level 3 were referred from

data 1.1 in data level 2 and the other followed the same flow. The result of response time based on different data levels as depicted in Table 1. The scalable is how deep the data folder itself increase the number of works. When creating a scalability test, it is important to scale up in increments. The data must be scaled up in increments as each stage tests. Small loads ensure the system functions as it should be on a basic level. Medium loads test if the system can function at its expected level. High loads test if the system can cope with a high load. The scalability was measured based on the total number of responses. This is the measure of time needed for a response to complete (from request to last response). The response time was recorded using a screen recorder that could monitor the user's activity while using the MOBICS.

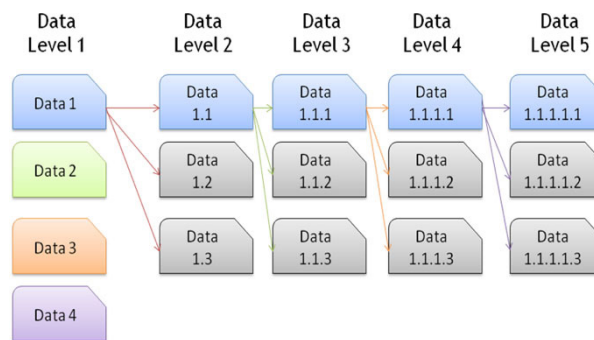


Fig. 1. The deepness of data level from level 1 until data level 5

The data levels are used as variables to measure the response time. Response time experiment result is the benchmark to measure the scalability [2], [14], [15], [16]. Data level is the number of data positions. This experiment involved 5 data load levels that were significant to measure the performance of MOBICS. Table 1 shows the total response time on data levels 1 to 5 by comparing them to without pre-fetching (W), with pre-fetching (P), and with DT rules. The most efficient one with a short response time is highlighted in bold. Based on Table 1, using pre-fetching was more efficient compared to without pre-fetching and DT. Data level 1 took 15.3 seconds to access data. However, when the data level was increased, the result showed using DT was more efficient than with or without the pre-fetching. As evident, the response time on data level 5 (the deepness data level) by using DT took only 22.2 seconds to access data compared to with or without the pre-fetching which took 60.2 and 27.2 seconds respectively. It concluded that the total response time using DT was faster compared with the pre-fetching only and without the pre-fetching even at the high data level (Data level 5) and the total response time using the pre-fetching only was better than without the pre-fetching at each of data level.

4 Results and discussion

Latency per page ratio, precision, recall, used as a benchmark in this experimental result as mention earlier. The equation of latency per page ratio is the ratio of the

latency that pre-fetching achieved to the latency with no pre-fetching. The latency per page is calculated by comparing the time between the browser initiation of an HTML page GET and the browser reception of the last byte of the last embedded image or object for that page. This metric represented the benefit perceived by the users, which will be better if it is low as its value [17]. Total response time is shown in Table 1 and it was recorded based on different data levels for 15 users. Five data levels were being used in this research to analyze the total response time with or without the pre-fetching. In addition, the total response time based on a DT algorithm was also recorded. Based on Table 1, it was proven that using pre-fetching was effective compared with traditional MCS (without the pre-fetching).

Table 1. Total response time on different data levels for 15 respondents

User	Total Time on Data Level 1 (seconds)			Total Time on Data Level 2 (seconds)			Total Time on Data Level 3 (seconds)			Total Time on Data Level 4 (seconds)			Total Time on Data Level 5 (seconds)		
	W	P	DT	W	P	DT	W	P	DT	W	P	DT	W	P	DT
1	32	13	15	15	10	14	21	17	16	26	21	18	60	27	22
2	26	16	16	20	14	15	23	15	15	25	22	13	58	25	23
3	16	15	16	25	15	17	25	15	15	26	21	15	61	28	22
4	28	14	15	22	11	15	26	17	16	27	20	16	65	27	22
5	17	17	17	22	16	17	23	17	17	25	18	17	66	27	22
6	27	17	18	26	17	16	30	15	13	26	19	15	63	27	22
7	30	16	18	27	13	18	31	10	12	27	17	15	60	28	21
8	11	15	18	30	19	16	23	11	13	25	17	15	59	26	22
9	17	13	17	29	21	14	26	13	12	27	19	13	58	29	23
10	16	16	16	29	17	15	25	17	12	28	23	16	60	28	21
11	19	17	17	22	19	15	27	15	15	24	17	17	60	27	22
12	23	15	17	26	16	15	22	10	13	26	16	16	59	27	22
13	22	16	18	27	14	13	26	11	14	25	16	16	58	25	24
14	27	18	16	21	17	12	28	15	13	27	17	16	59	28	22
15	26	12	15	25	16	14	24	15	12	25	15	15	58	29	23
Avg	22.7	15.3	16.5	24.1	15.5	15.2	25.2	14.2	13.9	25.9	18.53	15.6	60.2	27.2	22.2

Furthermore, Figure 2 depicts the latency per page ratio that was calculated to evaluate the latency on MCS. The results show the latency was even lower at a high data level. The latency per page ratio at data level 5 was 0.45 which was the smallest compared with the latency per page ratio at data level 1 which was 0.74. Therefore, using pre-fetching was effective in handling big data management commonly for multiple MCS.

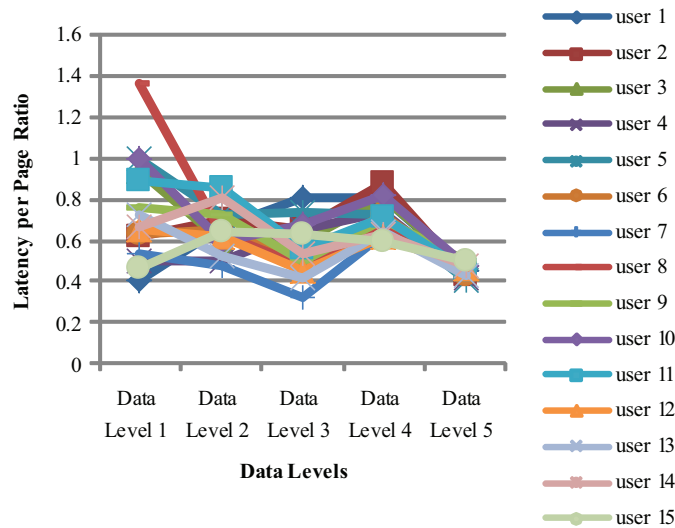


Fig. 2. Latency per page ratio on different data levels

In MCS evaluation recall and precision are greatly used to measure algorithm performance. Precision [18] and recall are frequently used to measure the effectiveness of performance of the pre-fetching [19]. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. While recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database.

Table 2 shows the result of precision and recall for the proposed MOBICS. The data were evaluated based on total the pre-fetch hit for 15 respondents to get the average value of precision and recall. The result proves the MOBICS provided high precision and recall with 86% and 76% respectively.

Table 2. Precision and recall for 15 respondents

	Pre-fetch Hits	Pre-fetch	User Request	Precision (Pc)	Recall (Rc)
User 1	11	15	11	0.73	1.00
User 2	6	7	20	0.86	0.80
User 3	19	21	16	0.90	0.94
User 4	16	18	19	0.88	0.58
User 5	20	23	20	0.87	0.95
User 6	5	6	22	0.83	0.68
User 7	14	15	18	0.93	0.72
User 8	11	12	8	0.92	0.75
User 9	11	13	28	0.85	0.68
User 10	16	19	23	0.84	0.70
User 11	15	19	27	0.79	0.74

(Continued)

Table 2. Precision and recall for 15 respondents (*continued*)

	Pre-fetch Hits	Pre-fetch	User Request	Precision (Pc)	Recall (Rc)
User 12	11	12	6	0.92	0.83
User 13	19	25	25	0.76	0.56
User 14	15	17	17	0.88	0.65
User 15	13	15	14	0.87	0.79
Average				0.86	0.76

5 Conclusion

The scalability test was successful as it required less total response time in accessing the data even at a high data level. The performance of pre-fetching was measured based on the latency per page ratio that was calculated to evaluate the latency on the MCS. The results showed the latency was low even at a high data level. Besides, precision and recall were used to measure the effectiveness of the performance of pre-fetching. The result proved that the MOBICS provided high precision and recall with 86% and 76% respectively. The scalability test had improved and proved MOBICS efficiency that took only 22.2 seconds to access data even at the high data level (data level 5) compared to with and without the pre-fetching which took 60.2 and 27.2 seconds respectively (refer to Table 1).

6 Acknowledgement

This work is supported by Universiti Teknologi Malaysia (UTM) UTM Encouragement Research Grant, Q.J130000.2651.18J64. The authors would like to express their deepest gratitude to UniKL MIIT for supporting this research and Ircache.net for their support in providing the datasets to ensure the success of this research.

7 References

- [1] L. Wang, R. Ranjan, R., J. Chen, B. Benatallah, "Cloud Computing Methodology, Systems, and Applications", CRC Press Tylor and Francis Group. International Standard Book Number: 978-1-4398-5641-3, 2012.
- [2] S. Sharma, "Expanded cloud plumes hiding Big Data ecosystem," *Futur. Gener. Comput. Syst.*, vol. 59, pp. 63–92, 2016. <https://doi.org/10.1016/j.future.2016.01.003>
- [3] S. A. El-Seoud, H. El-Sofany, M. Abdelfattah and R.F. Mohamed, "Big Data and Cloud Computing: Trends and Challenges", *International Journal Interactive Mobile Technologies (iJIM)*, vol. 11, pp. 34–52, 2017. <https://doi.org/10.3991/ijim.v11i2.6561>
- [4] I. A. T. Hashem, I. Yaqoob, N. Badrul Anuar, S. Mokhtar, A. Gani, and S. Ullah Khan, "The rise of 'Big Data' on cloud computing: Review and open research issues," *Inf. Syst.*, vol. 47, pp. 98–115, 2014. <https://doi.org/10.1016/j.is.2014.07.006>

- [5] M. M. Falatah, and O.A. Batarfi, “Cloud Scalability Considerations”, *International Journal of Computer Science and Engineering Survey*, vol. 5, no. 4, pp. 37–47, 2014. <https://doi.org/10.5121/ijcses.2014.5403>
- [6] W. Tsai, Y. Huang, Q. Shao, and X. Bai, “Data Partitioning and Redundancy Management for Robust Multi-Tenancy SaaS,” *Int. J. Softw. Informatics*, vol. 4, no. 4, pp. 437–471, 2010.
- [7] R. S. M. L. Patibandla, S. S. Kurra, and N. B. Mundukur, “A Study on Scalability of Services and Privacy Issues in Cloud Computing,” *Springer-Verlag Berlin Heidelb*, pp. 212–230, 2012. https://doi.org/10.1007/978-3-642-28073-3_19
- [8] N. S. Hussien, N.S., Sulaiman, S.M., Shamsuddin, “Usability testing on intelligent mobile web pre-fetching of cloud storage scheme”, *International Journal of Advances in Intelligent Informatics*, vol. 3, no. 3, pp. 134–145, 2017. <https://doi.org/10.26555/ijain.v3i3.129>
- [9] F. Almodarra and B. Qureshi, “Issues in Adopting Agile Development Principles for Mobile Cloud Computing Applications,” *Procedia Comput. Sci.*, vol. 52, pp. 1133–1140, 2015. <https://doi.org/10.1016/j.procs.2015.05.131>
- [10] I. Arpaci, “Understanding and predicting students’ intention to use mobile cloud storage services,” *Comput. Human Behav.*, vol. 58, pp. 150–157, 2016. <https://doi.org/10.1016/j.chb.2015.12.067>
- [11] J. Shamsi and M. Ali, “Data-Intensive Cloud Computing : Requirements, Expectations, Challenges , and Solutions,” vol. 11, pp. 281–310, 2013. <https://doi.org/10.1007/s10723-013-9255-6>
- [12] J. Nielsen, “10 Usability Heuristics for User Interface Design“ [Online]. Available from: <https://www.nngroup.com/articles/ten-usability-heuristics>, 1995. [retrieved: January, 2018].
- [13] Mirkowicz, Magdalena and Grodner, Grzegorz. “Jakob Nielsen’s Heuristics in Selected Elements of Interface Design of Selected Blogs”, *Social Communication*. vol. 4, pp. 30–51, 2018. <https://doi.org/10.2478/sc-2018-0013>
- [14] M. Maurer, I. Brandic, and R. Sakellariou, “Adaptive resource configuration for Cloud infrastructure management,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 2, pp. 472–487, 2013. <https://doi.org/10.1016/j.future.2012.07.004>
- [15] N. Fernando, S. W. Loke, and W. Rahayu, “Mobile cloud computing: A survey,” *Futur. Gener. Comput. Syst.*, vol. 29, no. 1, pp. 84–106, 2013. <https://doi.org/10.1016/j.future.2012.05.023>
- [16] N. S. Hussien, A. Aborujilah, S. Sulaiman, “Mobile intelligent web pre-fetching scheme for cloud computing services in industrial revolution 4.0”, *Test Engineering and Management*, vol. 81, no. 11–12, pp. 964–971, 2019.
- [17] J. Domènech, A. Pont, J. Sahuquillo, and J. Gil, “A user-focused evaluation of web prefetching algorithms”, *Computer Communications*, vol. 30, no. 10, 2213–2224, 2007. <https://doi.org/10.1016/j.comcom.2007.05.003>
- [18] Sv. Manikanthan, T. Padmapriya, A. Hussain and E. Thamizharasi, “Artificial Intelligence Techniques for Enhancing Smartphone Application Development on Mobile Computing”, *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 17, pp. 4–19, 2020. <https://doi.org/10.3991/ijim.v14i17.16569>
- [19] Tan, Mingjie & Shao, Peiji, “Prediction of Student Dropout in E-Learning Program Through the Use of Machine Learning Method”, *International Journal of Emerging Technologies in Learning (IJET)*, vol. 10, no. 1, pp. 11–17, 2015. <https://doi.org/10.3991/ijet.v10i1.4189>

8 Authors

Nur Syahela Hussien is a senior lecturer at Universiti Kuala Lumpur Malaysian Institute of Information Technology (UniKL MIIT) in Creative Multimedia Animation

Section. Her research interests include mobile cloud computing, soft computing, web caching and pre-fetching, mobile application development, game development, data mining and human computer interaction. Currently, she is a program coordinator for a diploma in Multimedia at UniKL MIIT. She can be reached at syahela@unikl.edu.my or nursyahela_90@yahoo.com.

Sarina Sulaiman is a senior lecturer at Universiti Teknologi Malaysia and researcher at UTM Big Data Centre, Ibnu Sina Institute for Scientific and Industrial Research, School of Computing. Her research interests include soft computing, web caching, web pre-fetching, web usage mining and social networking. She can be reached at sarina@utm.my.

Abdulaziz Aborujilah is a senior lecturer at Universiti Kuala Lumpur Malaysian Institute of Information Technology (UniKL MIIT).

Merlinda Wibowo is a lecturer at Department of Informatics, Faculty of Informatics, Institut Teknologi Telkom Purwokerto, Indonesia.

Hussein Samma is a senior lecturer at School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia.

Article submitted 2021-07-12. Resubmitted 2021-10-05. Final acceptance 2021-10-05. Final version published as submitted by the authors.