HIGH TECHNOLOGY LETTERS | Vol. 21 No. 1 | Mar. 2015 | pp. 63 ~ 70

doi:10.3772/j.issn.1006-6748.2015.01.009

A prediction mechanism of file access popularity and optimal strategy of replicas deployment in cloud computing①

Wang Ning (王　宁)[*,**], Yang Yang[*], Chen Yu[*], Mi Zhenqiang②[*], Meng Kun[***], Ji Qing[*]
([*] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, P. R. China)
([**] Department of Mechanical and Electrical Engineering, Yantai Engineering & Technology College, Yantai 264006, P. R. China)
([***] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, P. R. China)

Abstract

In cloud computing, the number of replicas and deployment strategy have extensive impacts on user's requirement and storage efficiency. Therefore, in this paper, a new definition of file access popularity according to users' preferences, and its prediction algorithm are provided to predict file access trend with historical data. Files are sorted by priority depending on their popularity. A mathematical model between file access popularity and the number of replicas is built so that the reliability is increased efficiently. Most importantly, we present an optimal strategy of dynamic replicas deployment based on the file access popularity strategy with the overall concern of nodes' performance and load condition. By this strategy, files with high priority will be deployed on nodes with better performance therefore higher quality of service is guaranteed. The strategy is realized in the Hadoop platform. Performance is compared with that of default strategy in Hadoop and CDRM strategy. The result shows that the proposed strategy can not only maintain the system load balance, but also supply better service performance, which is consistent with the theoretical analysis.

Key words: cloud computing, Hadoop, file access popularity, load balance, quality of service

0 Introduction

With the fast technology development of Web 2.0, the role of traditional Internet user has changed from data consumer to producer. This mode of data production, which is centered by users, leads to exponential increase in data volume. Cloud computing system management is facing austere challenge caused by this trend. We usually adopt the replica mechanism to improve the system reliability. As users interest is in different files, it's impropriate to set the same amount of replica for all files, which may waste the storage, and also influence the resource access efficiency and service cost. So it is necessary to back-up and deploy files reasonably, which has always been the active research area. Some scientific and commercial studies have accomplished prospective achievements. At present, distributed storage architectures are largely used to process the real-time big data by providing scalability and good storage performance. Some typical cloud storage providers such as Google, Amazon and Yahoo!, support scalable data management service and use parallel computing model like MapReduce to improve efficiency[1]. Widely applied distributed file systems include Google File System (GFS), Amazon Simple Storage Service (S3), which are used in commercial application, and Hadoop distributed File System (HDFS)[2], which is used for scientific research. The HDFS is an open source project. It copies the idea of GFS, and has a good scalability which makes it suitable for distributed data process on large scale.

Focusing on cloud storage system and file replica management strategy, Wei[3] raised the minimum replica quantity formula based on the probability of file failure and users' preferences. It was a dynamic replica management strategy. In Ref. [4], Li came up with a dynamic replica management strategy that provided better cost efficiency. This strategy enabled system to increase the amount of replicas. At the same time, storage cost was reduced and demand for reliability was met. The strategy mainly focused on data which was temperately used and had a low demand of reliability. In Ref. [5], Wu considered system load balance and storage usage. The authors brought up the replica preadjustment strategy based on the analysis of file popu-

① Supported by the National Natural Science Foundation of China (No. 61170209, 61272508, 61202432, 61370132, 61370092).
② To whom correspondence should be addressed. E-mail: mizq@ustb.edu.cn
Received on Dec. 16, 2013

larity. The disadvantage of this strategy was that it depended too much on the access mode of files. In Ref. [6], the authors suggested a fast self-adaption data block deployment algorithm. With the concerning of users' preferences and multi-dimensional QoS indicator, Sun[7] proposed a scheduling optimization algorithm of cloud resources to improve the performance of users' experience and resources accessibility. In Ref. [8], Li took the storage cost and data reliability into account, decreased the amount of replicas by tracing and detecting data blocks reliability positively. As a result, the storage cost was reduced. But this detection method would cause extra overhead. In our previous work, we ensured the number of different block replica, according to client's demands for each block, and gave the MCSB strategy, which can satisfy users' requirements and minimum storage cost. Meanwhile, to maintain the performance of service system, the system load was considered and an adaptive MCSB strategy[9] was presented. Ref. [10] raised a distributed self-adaption data replica algorithm, which acquired good effect in getting data position. Wu[11] presented the MingCloud system which could increase the data reliability and quality of service. The update, add and change operations of data in cloud nodes had a deep influence on system load. Concerning with that problem, Refs[12] and [13] put forward the load re-balance algorithm. Agarwala[14] advanced an integrated storage strategy which could select and place users' data automatically. Liao[15] proposed a dynamic replica deleted strategy based on the QoS perception. By using this strategy, demand for storage was reduced so that the maintenance cost was also reduced. In Ref. [16], the authors made a suggestion for a data deployment algorithm to supply high availability and reliability. This algorithm took the data weight distribution, load balance and the least cost into consideration.

Nevertheless, there are mainly two problems existing in the aforementioned researches: first, the researches on prediction mechanism of file access popularity are fewer and usually predict it through statistics methods, which lacks the more reasonable real-time mathematical model; second, facing to the replicas deployment of file, to improve the quality of service, the performance of node and the present network status are considered by most of work, but the effect on file access popularity for quality of service is underestimated. Aiming at solving the problems, satisfying different service requests and improving quality of service, this paper innovates in the aspects as below: (1) file access popularity prediction algorithm (FAPP) is raised by analyzing file access patterns, to predict the file popularity in the future. After studying the inner connection between file popularity and number of replicas, a new mathematical model is built. Upper bound and lower bound of replica amount are defined to control the service cost effectively so that quality of service is improved. (2) File priority classification algorithm (FPC) based on the file access popularity is suggested. Mathematical model for measuring the node performance is also built. (3) Optimal strategy of dynamic replicas deployment based on file access popularity (ODRDF) is presented by taking file priority, node performance and system load into account. Results of experiment show that the ODRDF strategy is effective, which is consistent with the theoretical analysis.

The rest of the paper is organized as follows: Section 1 suggests the file access popularity prediction algorithm; Section 2 builds the mathematical model of file access popularity; Section 3 raises the file priority clarify algorithm based on file access popularity and the optimal strategy of dynamic replicas deployment based on file access popularity (ODRDF); Section 4 takes experiments and analyses the results; Section 5 concludes the paper.

1 File access popularity prediction

1.1 Amount of file access analysis

Let us take the E-commerce for example. When a new kind of product starts to be offered for sale, because the advertisement and marketing method in the earlier stage are different, the relative file access popularity presents distinct law. It is concluded that the file access pattern possesses some statistical characteristics. This paper comes up with the definitions below for convenience.

Definition 1. File access popularity: the amount of file access to the *n*th file in a period of time, which can be indicated by hot_n.

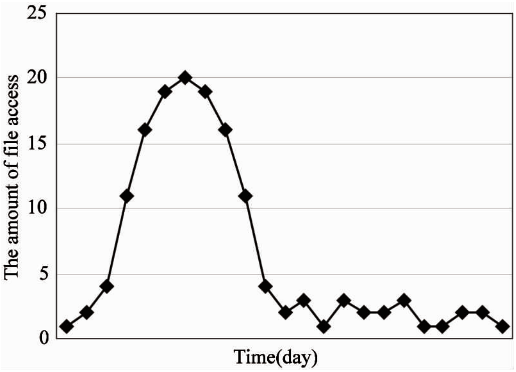
Definition 2. File access feature: the mathematical access feature presents in the period of time when the file is created and destroyed.

By researching on the file access pattern, one gets the result that file access amount usually exhibits a time feature of parabolic curve. The amount will get to the maximum in a period of time, but eventually it will recede. In this paper, It is called as Parabola Feature (PF). The file access feature can be classified into 4 types, which are shown in Fig. 1.

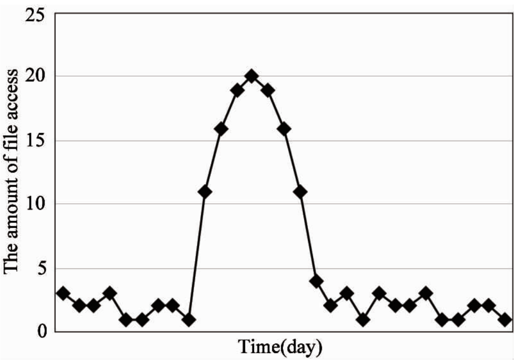
(1) Head-PF: Parabola Feature emerges right after the file is created and it appears only once.

(2) Middle-PF: Parabola Feature emerges during a period of time after the file is created, and it appears

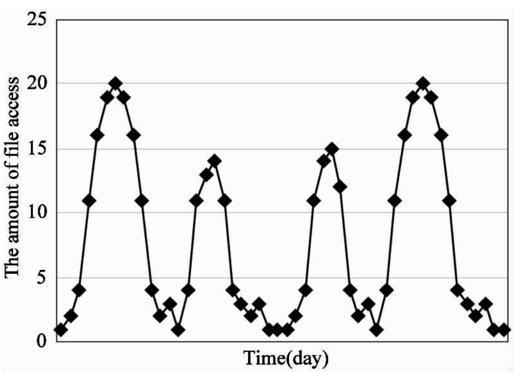
only once.



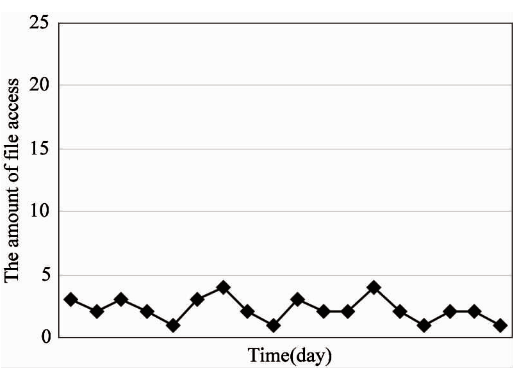
(a) Head-PF curve



(b) Middle-PF curve



(c) Random-PF curve



(d) Not-PF curve

Fig. 1 Parabola Feature of file access

(3) Random-PF: Parabola Feature emerges randomly and times it appears can not be determined.

(4) Not-PF: Amount of access always stays in a low level, and there are no apparent Parabola Features.

Facing to a new product, if merchant can do a lot of advertisements and effectively marketing methods ahead of release, the related file access amount will show the Head-PF feature. As time goes by, the access amount will decrease. The Middle-PF feature will emerge without the marketing in the earlier stage because people need some time to adopt this new commodity. The Parabola Feature will arise randomly if sales promotion and other operations are applied during the selling of the product, then the Random-PF turns up. But the Not-PF will appear if the marketing operations are applied wrongly or the quality of product is poor.

1.2 File access popularity prediction

By analyzing the file access trend, some mathematical features can be found. Among these features, the latest history information is most important for the trend prediction of file access. In this paper, we will realize the prediction of file access trend via the historical information. Dynamical optimization of file replicas is done based on the trend to satisfy service requests of different users and improve the quality of service.

Exponential Smoothing (ES) is a commonly used time series prediction method[17]. The ES supposes the trend of time series is stable and regular, so the postpone of time series is reasonable. The latest information is most important for the prediction, which is corresponding with the problem of file access popularity. In the paper, we will assign the larger weight for the latest information and the lower for earlier. The basic formula is shown in Eq. (1)[17]:

$$S_t = \alpha \times y_t + (1 - \alpha) \times S_{t-1} \qquad (1)$$

where S_t indicates the smooth value at time t; y_t is the actual value at time t; S_{t-1} indicates the smooth value at time $(t-1)$; α is the smooth constant, value range of which is $[0,1]$.

The Exponential Smoothing method can be classified by the times of smoothing[17]. One time smoothing is suitable for series which change little with time. Two times smoothing is another smoothing operation based on the one time smoothing which can be applied on the linear change time series. Three times smoothing is another smoothing based on the two times smoothing, which is designed to cope with the time series having quadratic curve feature. The result of experiment shows that the file access popularity has a feature of typical quadratic curve, so we use the three times smoothing for prediction.

A file access popularity prediction (FAPP) algorithm is given in this paper. Detailed description is as follows:

Firstly, the data of time series present a trend of quadratic curve feature, so the three times smoothing is applied to predict file access popularity. The formula is Eq. (2)[17]. In the formula, $S_t^{(3)}$ is the value after three smoothing operations.

$$\begin{cases} S_t^{(1)} = \alpha \times y_t + (1-\alpha) \times S_{t-1}^{(1)} \\ S_t^{(2)} = \alpha \times S_t^{(1)} + (1-\alpha) \times S_{t-1}^{(2)} \\ S_t^{(3)} = \alpha \times S_t^{(2)} + (1-\alpha) \times S_{t-1}^{(3)} \end{cases} \quad (2)$$

Secondly, a mathematical model of file access popularity prediction is built using the three times smoothing, as Eq. (3):

$$hot_n(d) = a + bd + cd^2 \quad (3)$$

where $hot_n(d)$ indicates the access popularity prediction of nth file during time $t+d$; a, b, and c are the prediction factors to be determined, which can be calculated by[17].

$$a = 3S_t^{(1)} - 3S_t^{(2)} + S_t^{(3)} \quad (4)$$

$$b = \frac{\alpha}{2(1-\alpha)^2}[(6-5\alpha)S_t^{(1)} - (10-8\alpha)S_t^{(2)} + (4-3\alpha)S_t^{(3)}] \quad (5)$$

$$c = \frac{\alpha^2}{2(1-\alpha)^2}[S_t^{(1)} - 2S_t^{(2)} + S_t^{(3)}] \quad (6)$$

Finally, we can calculate the three times smoothing prediction formula, through which file access prediction during time $t+d$ can be done. hot_n is the final result to be output.

2 Mathematical model of replica amount

Suppose there are $M(M>=3)$ nodes and N files in the cloud computing system. Each file will have 3 replicas by default; the total amount of replicas is $3N$. Let the amount of nth file's replicas be c_n ($1 \leqslant n \leqslant N$).

The principle of replicas presented is that the file access popularity is proportional to the file replicas amount. Then the mathematical model is presented in

$$\frac{A_{\max}}{C_{\max}} = \frac{hot_n}{c_n} \quad (7)$$

$A_{\max}$ is the threshold of file access popularity. When the amount of access to a file is larger than $A_{\max}$, it will have the maximum replica amount, indicated by $C_{\max}$, value of which is determined by $C_{\max} = kM$, ($0 < k \leqslant 1$). The hot_n ($1 \leqslant n \leqslant N$) is the access popularity of nth file during a period of time. The dynamic replicas amount can be calculated by

$$c_n = \lceil \frac{hot_n \times C_{\max}}{A_{\max}} \rceil \quad (8)$$

Among them, $1 \leqslant c_n \leqslant kM$, ($0 < k \leqslant 1$) and k is defined as the cost factor. The value of k can be determined by the service cost and expert knowledge. This factor will increase when the service cost is low and decrease otherwise. To effectively control the service cost, we bound the total amount of file replicas, which is denoted by S_c in

$$N \leqslant S_c = \sum_{n=1}^{N} c_n \leqslant N \times kM \quad (0 < k \leqslant 1) \quad (9)$$

3 Optimal strategy

3.1 File priority classification

By the analysis in Section 1.2, the file access popularity is predicted, based on which the file priority can be classified. This optimization will approve quality of service. File priority classification (FPC) is raised based on the file access popularity. It can be described like this:

Suppose there are N files, hot_n is the file access popularity of nth file in a period of time. In this paper, files priorities are classified into three levels, which are A, B, C from high to low. A_{set}, B_{set} and C_{set} are three sets, which are the highest priority set, the higher priority set and the lowest priority set respectively.

Firstly, when the amount of file access is larger than the average value A_{ave}, the file priority is B and the file belongs to B_{set}. In contrast, if the amount of file access is smaller than A_{ave}, its priority is C and it belongs to C_{set}. A_{ave} can be calculated by

$$A_{ave} = \frac{\sum_{n=1}^{N} hot_n}{N} \quad (10)$$

Secondly, define H ($1 \leqslant H \leqslant N$) to indicate the number of files in B_{set}. The threshold of file access popularity $A_{\max}$ ($A_{\max} \geqslant A_{ave}$) can be calculated in Eq. (11). The files with access popularity larger than $A_{\max}$ are thought to have the highest priority, then they belong to A_{set} and their priorities belong to A. The rest files in B_{set} will still stay in the original set.

$$A_{\max} = \frac{\sum_{n=1}^{H} hot_n}{H} \quad (1 \leqslant H \leqslant N) \quad (11)$$

Finally, by above steps the files in cloud are stored in A_{set}, B_{set} *and* C_{set} by the file access popularity respectively.

3.2 Optimal strategy

There are hundreds of nodes which are different from each other in performance in the cloud computing system. The nodes' performances may make great difference to the efficiency of improving storage and fetch. Here the optimal strategy of dynamic replicas deployment is presented based on file access popularity, focusing on nodes' performances and system load, to improve quality of service. The main idea is: the files in set A_{set} should not only be copied more, but also be deployed onto high performance nodes to obtain better user experience. Instead, the files in set C_{set} should be copied less and be deployed onto lower performance nodes because of less amount of access. In this way, storage of distributed file system will be efficiently used to solve users' service requirements.

Node's performance, denoted by p, is closely interrelated with factors of storage utilization, network I/O, CPU frequency and memory utilization. It can't be calculated simply because units of these factors are different. Units' normalization should be done and, here, we also import weight value. One typical node will be selected as a representative; the other nodes' values will be changed to the ratio to this node. These values are parameters after normalization. The normalized parameters can be indicated by S, I, U and E respectively. Their weight is $W = \{W_S, W_I, W_U, W_E\}$ to correctly measure the nodes' performance. W is determined based on actual situation because of accumulating big data, which bring severe challenges to the storage of data. On one side, big data have made node storage a key factor, because it determines the data volumes that can be stored, on the other side, CPU frequency, memory available and network I/O can directly affect the efficiency of data storage and fetch. In conclusion, larger value of p means higher performance. The node performance p can be calculated by

$$p = W_s \times (1 - S) + W_I \times I + W_U \times U + W_E \times (1 - E) \qquad (12)$$

$$W_s + W_I + W_U + W_E = 1 \qquad (13)$$

All nodes' performance values are sorted and put into a performance table in database to guarantee the load balance. When large amount of files need to be deployed, they will be put into the sorted nodes by files' priorities.

Suppose the node collection is $\boldsymbol{D} = \{D_1, D_2, \cdots, D_M\}$. All the nodes are sorted in descending order by their performance values, that is, $\boldsymbol{P} = \{p_1, p_2, \cdots, p_M\}$. The performance exponent of each node E_i will be decided by

$$E_i = \frac{p_i}{\sum_{i=1}^{M} p_i} \qquad (14)$$

To improve the quality of service the maximum amount of replicas stored in each node is determined by its performance. The values can be got

$$R_{\max i} = E_i \times \sum_{n=1}^{N} c_n \qquad (15)$$

In summary, detailed description of optimal strategy of dynamic replicas deployment based on file access popularity (ODRDF) is shown in Table 1.

Table 1 ODRDF strategy

ODRDF strategy
(1) Predict file access popularity using FAPP algorithm;
(2) The priorities of all files are classified into A, B and C, and all files are stored in set A_{set}, B_{set} and C_{set} using FPC algorithm;
(3) Determine the amount of file replicas
(a) Let the amount of file replicas in set A_{set} be kM;
(b) The amount of file replicas in set B_{set} and C_{set} can be calculated by formula (8);
(4) Judge the node performance
(a) Do units' normalization of all performance parameters;
(b) Calculate performance of each node by formula (12) and (13);
(c) Sort the nodes in descending order based on their performance;
(d) Calculate performance exponent E_i of each node using formula (14);
(e) Determine the maximum amount of replicas R_{maxi} in each node by formula (15);
(5) Replica optimal deployment
(a) Count the amount of all replicas in each node;
(b) If the total amount is not larger than R_{maxi}, deploy the files in set A_{set}, B_{set} and C_{set} into sorted nodes in sequence by their priorities;
(6) Output the result of optimal deployment.

4 Experiment

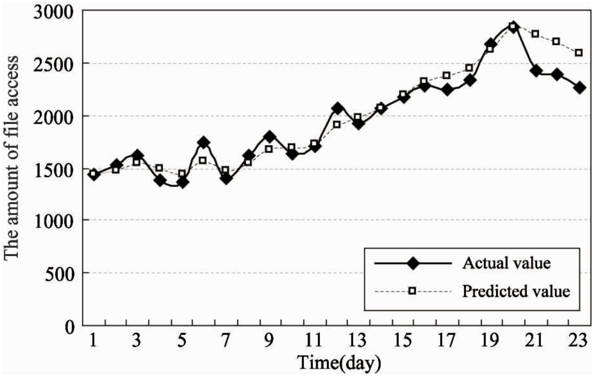
Our cloud platform consists of 20 nodes with different performance. Their configurations are shown in Table 2. Configuration of D_1 is considered as the representative and parameters of D_1 are set to 1. Parameters of other nodes are the result of ration to the corresponding value of D_1, so that they are normalized. Set the cost factor $k = 30\%$, which will make the maximum amount of file replicas 6. Let $W_S = W_I = W_U = W_M = 25\%$.

Table 2 The configuration of each node

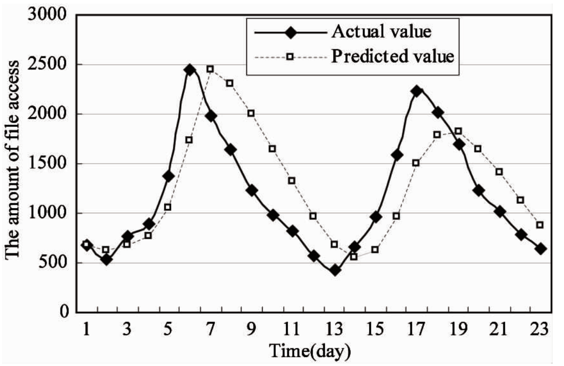
Node	D_1-D_4	D_5-D_8	D_9-D_{12}	D_{13}-D_{16}	D_{17}-D_{20}
CPU	Core 3.2GHz	Core 2.8GHz	Intel 2.5GHz	Intel 2.5GHz	Intel 2.2GHz
Memory	8G	4G	4G	4G	2G
Disk	1T	500G	500G	350G	300G

4.1 Rationality analysis

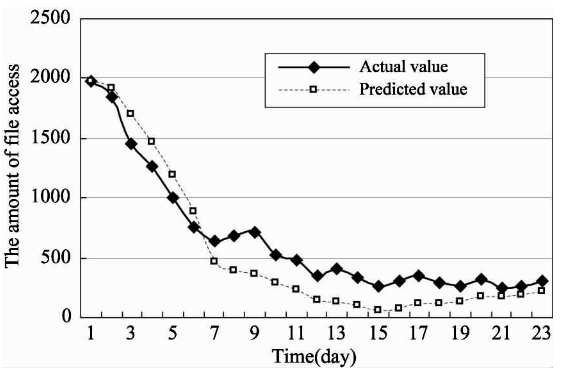
To guarantee the effectiveness of ODRDF, some experiments are conducted with the FAPP algorithm to verify its integrity and rationality. The data used are the files access records of Microsoft MSN Storage Server in 2008 and they are taken as the access popularity data. In the experiment, a prediction of file access in the next 24 days is made based on the data 24 days ago. And, comparisons between the predicted value and actual value are done. The result is shown in Fig. 2(a), 2(b) and 2(c). Fig. 2(a) is the condition that the popularity of files is smooth at the beginning and rises rapidly until reaching the maximum then declines. Fig. 2(b) describes the feature that two times peak appears and then stays steadily. Fig. 2(c) has no peak and declines at the beginning. From the experiments, the predicted results are fit with the actual trend, which verifies the rationality of the FAPP algorithm.



(a) Prediction of ascending series



(b) Prediction of multi-peak series



(c) Prediction of descending series

Fig. 2 Prediction of file access popularity

4.2 Service performance

10000 files for access popularity are predicted to be used to verify that the ODRDF strategy can improve the user experience of accessing to the high popularity files. Also load balance in the system is considered. According to data records, the total amount of access is 11840 in one specific day. The 10000 files are classified into set A_{set}, B_{set} and C_{set} by the FPC algorithm. Meanwhile, based on the result, the amount of file replicas is determined using methods mentioned in this paper. Comparisons are done in experiments with the typical strategies which are Hadoop default strategy and CDRM strategies[3] in system load and quality of service.

Replicas are determined and deployed to nodes in the cloud system using the 3 different strategies; detailed distribution is shown in Fig. 3, in which we can recognize that ODRDF can dynamically and effectively deploy replicas. Though performance of each node declines with the load, load balance is reached by deploying more replicas on higher performance node, which ensures the quality of service. This theory is also verified in service quality test. Although the CDRM strategy ignores the effect on the file access popularity for

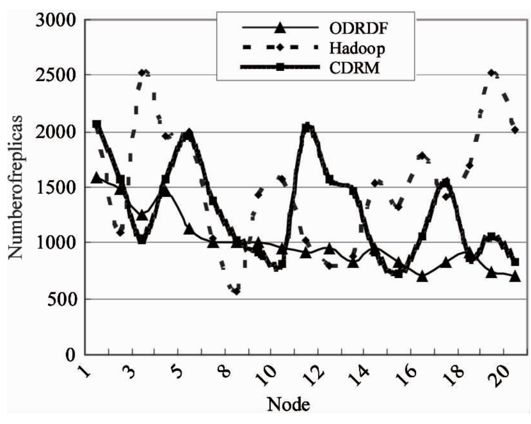


Fig. 3 System load

quality of service, it has considered network throughputs in system load. So the system load is better. In system load, ODRDF and CDRM are better than Hadoop default strategy. But ODRDF can effectively control the total number of replicas, the total amount of replicas is significantly lower than another two strategies. Notice that in Fig. 3, the horizontal axis follows the decreased order of nodes' performance.

Finally, the ability for dealing with the changes in service requests is measured. By detecting average response time, which is selected as a criterion, we get the result in Fig. 4. The output shows that ODRDF can satisfy the service requests of high access popularity using methods in which files are classified by different priorities and deploy onto nodes that are sorted by performance. When 10000 users access to files in A_{set} concurrently, the average response time is shorter than the two other strategies apparently, which, in detail, are 15.6% shorter than CDRM and 24.3% shorter than Hadoop default strategy. The time is still shorter when it comes to the files in B_{set}. But when accessing the files in C_{set}, the Hadoop default strategy, the replicas amount of 3 for each file, is better than other options, which is consistent with the theory analysis. But file access popularity in C_{set} is very small in the system, so too many replicas will cause waste of storage and enlarge of service cost.

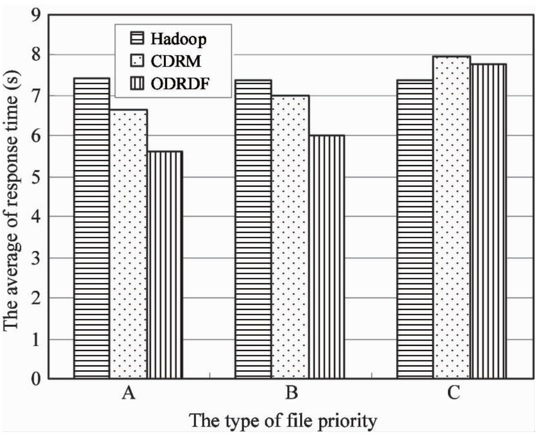


Fig. 4 Average response time of different strategies

5 Conclusion

In this work, aiming at improving service performance and satisfying service requests, file replicas amount and deployment are studied. Firstly, the file access popularity pattern is analyzed and a file access popularity prediction (FAPP) algorithm is put forward using exponential smoothing method. Also, files are classified into different types based on their access popularity. Secondly, the mathematical model is proposed as a foundation to determine replica amount by linking file access popularity with it. Thirdly, considering files priorities, nodes performance and system load, ODRDF is presented to deploy replicas dynamically by distributing high priority files onto nodes which have better performance to improve the quality of service. Finally, this strategy is realized in Hadoop platform and compared with two other typical strategies in the aspects of load balance and quality of service. Results certify the effectiveness of our strategy, which is consistent with theory analysis.

References

[1] Li J J, Cui J, Wang D, et al. Survey of MapReduce parallel programming model. *Acta Electronica Sinica*, 2011, 39(11):2636-2643

[2] Zhang D W, Sun F Q, Cheng X, et al. Research on Hadoop-based enterprise file cloud storage system. In: Proceedings of the 3rd International Conference on Awareness Science and Technology, Dalian, China, 2011. 434-437

[3] Wei Q S, Veeravalli B, Gong B Z, et al. CDRM: a cost-effective dynamic replication management scheme for cloud storage cluster. In: Proceedings of 2010 IEEE International Conference on Cluster Computing, Crete, Greece, 2010. 188-196

[4] Li W H, Yang Y, Yuan D. A novel cost-effective dynamic data replication strategy for reliability in cloud data centres. In: Proceedings of the Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing, Sydney, Australia, 2011. 496-502

[5] Wu S C, Chen G Z, Gao T G, et al. Replica pre-adjustment strategy based on trend analysis of file popularity within cloud environment. In: Proceedings of IEEE 12th International Conference on Computer and Information Technology, Chengdu, China, 2012. 219-223

[6] Li W H, Yang Y, Chen J J, et al. A cost-effective mechanism for cloud data reliability management based on proactive replica checking. In: Proceedings of the 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Ottawa, Canada, 2012. 564-571

[7] Bansal S, Sharma S, Trivedi I. A fast adaptive replication placement for multiple failures in distributed system. In: Proceedings of the 1st International Conference on Parallel, Distributed Computing Technologies and Applications, Tirunelveli, India, 2011. 495-502

[8] Sun D W, Chang G R, Li F Y, et al. Optimizing multi-dimensional QoS cloud resource scheduling by immune clonal with preference. *Acta Electronica Sinica*, 2011, 39(8): 1824-1831

[9] Wang N, Yang Y, Meng K et al. A user experience-based cost minimization strategy of storing data in cloud computing. *Acta Electronic Sinica*, 2014, 42(1): 20-27

[10] Abad C L, Lu Y, Campbell R H. DARE: adaptive data replication for efficient cluster scheduling. In: Proceedings of IEEE International Conference on Cluster Compu-

ting, Austin, USA, 2011. 159-168

[11] Wu J Y, Fu J Q, Ping L D, et al. Study on the P2P cloud storage system. *Acta Electronica Sinica*, 2011, 39 (5): 1100-1107

[12] Hsiao H C, Chung H Y, Shen H Y, et al. Load rebalancing for distributed file systems in clouds. *IEEE Transactions on Parallel and Distributed Systems*, 2013, 24 (5): 951-962

[13] Chung H Y, Chang C W, Hsiao H C, et al. The load rebalancing problem in distributed file systems. In: Proceedings of IEEE International Conference on Cluster Computing, Beijing, China, 2012. 117-125

[14] Agarwala S, Jadav D, Bathen L A. iCostale: adaptive cost optimization for storage clouds. In: Proceedings of IEEE 4th International Conference on Cloud Computing, Washington, DC, USA, 2011. 436-443

[15] Liao B, Yu J, Sun H, et al. A QoS-aware dynamic data replica deletion strategy for distributed storage systems under cloud computing environments. In: Proceedings of the 2nd International Conference on Cloud and Green Computing and the 2nd International Conference on Social Computing and the Its Applications, Xiangtan, China, 2012. 219-225

[16] Myint J, Naing T T. A data placement algorithm with binary weighted tree on PC cluster-based cloud storage system. In: Proceedings of 2011 International Conference on Cloud and Service Computing, Hong Kong, China, 2011. 315-320

[17] Zhu J P. Economic Forecasting and Decision-making. Xiamen: Xiamen University Press, 2012. 79-87

Wang Ning, born in 1978. She received her Ph. D degree from the University of Science and Technology Beijing in 2015. She received her B. S. degree from Shandong University of Technology in 2003, and the M. S. degree from Yunnan Normal University in 2006. Her current research interests include cloud computing and data mining.