ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, Vol.19, No.10, October 2008, pp.2517−2526
DOI: 10.3724/SP.J.1001.2008.02517

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

# 面向复杂网络存储系统的元胞自动机动力学分析方法*

陈进才[1,2+], 何 平[1], 葛雄资[1]

[1](华中科技大学 计算机科学与技术学院,湖北 武汉 430074)
[2](武汉光电国家实验室 信息存储研究部,湖北 武汉 430074)

# Dynamics Analysis Method of Cellular Automata for Complex Networking Storage System

CHEN Jin-Cai[1,2+], HE Ping[1], GE Xiong-Zi[1]

[1](College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)
[2](Division of Data Storage System, Wuhan National Laboratory for Optoelectronics, Wuhan 430074, China)
+ Corresponding author: E-mail: heping.hust@yahoo.com.cn

**Chen JC, He P, Ge XZ. Dynamics analysis method of cellular automata for complex networking storage system.** ***Journal of Software*****, 2008,19(10):2517−2526.** http://www.jos.org.cn/1000-9825/19/2517.htm

**Abstract**: Some inherent dynamics rules are concealed in large-scale network storage system on account of the complexity of data transmission behaviors. This paper studies object-based storage system and proposes two storage cellular automata models called SNCA and OSDCA from macro and micro aspect respectively, and these models can be used to analyze behaviors and rules of complex and dynamic network storage by capturing the intelligent and initiative properties of storage object. In the model SNCA, the lifetime attribute of storage object is used to analyze the data flow rules in storage network to ascertain the congestion degree from macro aspect based on special lattice network topology architecture, and simulation results show that data object flow has global relativity with the phase transition of data flow. In the model OSDCA, data migration and replication mechanism are combined to analyze hotspot migration rule based on the load distribution condition among storage nodes, and simulation results show that data distribution in OBS system has characters of some self-organization.

**Key words**: object-based storage; cellular automata; network topology; data migration; data replication

**摘 要**: 大规模网络存储系统中复杂的数据传输行为隐藏着一定的动力学规律性.针对基于对象的大规模网络存储系统,结合存储对象的智能性和主动性特征,分别在宏观与微观两个层次上提出了用于复杂网络存储动态行为规律分析的存储元胞自动机模型 SNCA 和 OSDCA.在 SNCA 模型中,对网格拓扑结构的存储网络,结合存储对象的生命周期属性,可在宏观上分析网络存储系统的数据流动规律,确定存储网络拥塞程度,仿真结果揭示数据对象流动和存储网络中的相变具有全局相关性;在 OSDCA 模型中,综合热点数据的迁移和复制机制,在微观上分析 I/O 负载动态分布特性和存储热点迁移规律,仿真结果表明对象存储系统中的数据分布具有一定的自组织特性.

**关键词**: 对象存储系统;元胞自动机;网络拓扑;数据迁移;数据复制
**中图法分类号**: TP393 **文献标识码**: A

* Supported by the National Basic Research Program of China under Grant No.2004CB318201 (国家重点基础研究发展计划(973)); the National Natural Science Foundation of China under Grant No.60773189 (国家自然科学基金)

Received 2007-01-19; Accepted 2007-04-28

## 1 Introduction

The typical characteristics of network storage system such as the complex change of time, space and discrete storage events intricate, the topology of network architecture and the complicated service rules, and dynamical I/O requests require special analysis way. Cellular Automata (CA) model used by physicists, mathematicians, computer scientists, and biologists to simulate complex phenomena is proved to be an effective way to study complex discrete space-time systems. Therefore, CA can also be utilized to simulate the internal complex behaviors and evolvement rules of a complex storage system. Furthermore, it can also provide theoretic support for constructing complex storage system based on the next generation network.

Compared with traditional modeling methods, CA can be used to simulate the interaction among the cells by several simple rules, and the eventual evolution results can demonstrate some complex behaviors. Recently, many researches have been done on traffic network by CA, but it seems that no research has focused on the complex storage system analysis by CA. Toru Ohira[1] studied the critical flow behavior in the packet switching network by CA, such as the issue of phase transition in the network. In accordance with the topology and route of packet switch system architecture, Chen[2] studied the influence produced by the tiny inconsistence of the topology structure, and certificated that difference stimulates some perpetual nodes which tend into congestion phase more easily.

In this paper, we synthesize the models brought by Toru Ohira and Chen, and propose efficient CA models for object-based storage systems (OBS)[3]. On one hand we study the storage network cellular automata (SNCA) model from macro aspect to describe the characteristics of object-based storage network, while on the other hand we build an OSD (object-based storage device) cellular automata (OSDCA) model which demonstrates the intelligence of OSD, and self-organization and self-adapt characteristics in OBS by looking into the inside of the storage system from the micro aspect.

The remainder of the paper is structured as follows. In Section 2, the topology control problem is addressed, a detailed description of a CA model SNCA is developed, and some simulation results are presented. In Section 3, we propose an algorithm combined with migration and replication to eliminate load unbalance based on a CA model OSDCA and analyze the simulation results. Finally, our concluding remarks and future work are presented in Section 4.

## 2 SNCA: A Storage Network Cellular Automata Model

The complex network topology, which brings about the phenomenon of data transporting traffic in a large-scale storage system, has commanded much attention recently. To improve the performance of storage system, it is significant to reduce the time of transporting data in the perplexing storage network by all means.

Many researchers tried to solve the network traffic problem[4−6], and some referred to CA. In Toru Ohira[1] and Chen's[2] papers, they both treat the remainder packets in the buffer equally, just like that the inner node forwards the packet in the first of its queue, puts it at the end of the queue of the selected node. However different applications have distinct request requirements, for example, the online video service need be responded as quickly as possible, while the backup event does not care about the waiting time like others[4]. In the OBS system, we can greatly benefit from the intelligent objects, especially for their multiple and intelligent attributes which describe the characters of the data. With a special attribute added by the system, it contributes greatly to the intelligence of data object. Since the QoS agreement of data object describes the tolerance degree to the storage network delay, system can take measures to fit for these different requests by the intelligent storage objects. In this section we describe a model called SNCA using lifetime attribute which demonstrates the time the data object can exist in storage network to solve the network traffic in the OBS system. From the result, we can observe that the system abandon less packages by considering the lifetime of every object.

### 2.1 Model definition

OBS storage network basically contains the client nodes, OSD nodes (storage nodes), metadata servers, and routers. Whereas the chief responsibility of metadata servers is to maintain the metadata of which the proportion in the whole data resources is very small, is during dealing with access request, their primary work is to verify the status of clients, then send the credentials and metadata to clients. After that, clients can access OSD nodes directly due to the intelligent data objects. Therefore we ignore the metadata servers in the SNCA model.

As the number of these three kinds of nodes is large, the topology architecture among the nodes appears so complicated. If there are $N$ nodes placed on the OBS storage network, with $X_1$ standing for the set of client nodes, $X_2$ standing for the set of OSD nodes and $X_3$ standing for the set of router nodes, the topology space can be described as

$$\Omega=(X_1,X_2,X_3,L),\ |X_1|+|X_2|+|X_3|=N \tag{1}$$

$$L\subseteq(X_1+X_2+X_3)\times(X_1+X_2+X_3) \tag{2}$$

Therefore the number of topology will total up to $2^{N(N-1)/2}$. In this paper, we choose a simple model SNCA of perfection topology architecture to describe the storage network as bidimensional lattice space. It is with $N$ nodes (client nodes and OSD nodes) on each side and $N^2$ nodes as a whole. The status of the inner nodes (routers) is defined as a limited set $S\subseteq\{0,1\}^N$, of which the state 0 means this router contains data objects, while state 1 means no object stays in the router.

The two-dimension cellular automata space can be described as

$$\Pi=\{ie_1+je_2:i,j\in Z,0\le i,j\le N\} \tag{3}$$

The relation of neighbors can be described as[2]

$$F(X)=\{X+e_1,X-e_1,X+e_2,X-e_2\},\ X\in\Pi \tag{4}$$

Transition rules:

(1) Data objects are generated and destroyed on the boundary nodes of the lattice (on squares in Fig.1). A node on the boundary generates a data object according to the Poisson stream with arrival rate of $p$ and sends it to a destination node selected randomly from the nodes on the boundary.

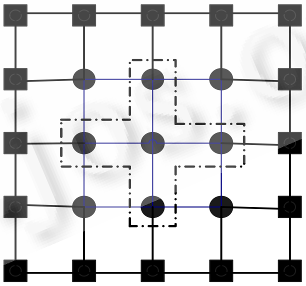


Fig.1 The model of storage network architecture with Von Neumann neighborhood[7]

(2) Inner nodes (routers) only forward data objects received from neighbor nodes. There is a buffer with the size of $L$ in each inner node, which stores the remainder data objects that need to wait for forwarding the next time. Each inner node chooses the shortest lifetime data object in the buffer and forwards it to the neighbor to ensure better performance. When data object arrives at the destination node, it will be destroyed to balance the total number of data objects in lattice.

(3) The way of choosing a neighbor follows the shortest distance principle[8], choosing the one closest to the destination. If the chosen neighbors are more than one, it selects one randomly. If no neighbors, data object will be abandoned because of unable to be transmitted.

(4) The time step updates one by one.

### 2.2 Simulation and results

MATLAB was used as the tool for simulation. During the simulation, some complex and diverse characters of real storage network were neglected. To make use of average field theory, we regard the dynamics system as a simple Jackson-open network, in which inner cells are service nodes resembling as the Jackson network. According to Jackson theorem and Little formula[9], if the network is only composed of the nodes with single service and a fixed rate, the communication intensity is

$$u_i=p_i/\mu_i \tag{5}$$

In which $\mu_i$ is the velocity of transmitting data objects.

The utilization ratio of node $i$ is described by the following equation

$$\rho_i=P(Q_i\geq 1)=u_i \tag{6}$$

Inner node $i$ is abstracted as a service queue $M/M/1$ approximately. As the utilization ratio is $\rho_i$, the average number of tasks in this queue is

$$\rho_i = L_i = \frac{\rho_i^2}{1-\rho_i} \tag{7}$$

In this lattice network, the total number of data objects can be defined as the whole objects waiting for being forwarded in the buffer of the inner nodes. We assume the number of data objects in every buffer is $q_i$, and as a result of Eq.(7), the total number of data objects in the lattice is

$$q=\sum_{i=1}^{(N-2)^2}\frac{\rho_i^2}{1-\rho_i}=\sum_{i=1}^{(N-2)^2} q_i \tag{8}$$

The average number of data objects in each buffer is

$$\overline{q}=q/(N-2)^2 \tag{9}$$

The first experiment was designed to discover the relationship between the critical rates of phase transition and data objects' arrival rate $p$. The system size is varied as $N$=25, 50, 100, and the results are drawn for the readers' convenience. As we can observe in Fig.2, if the arrival rate $p$ is less than the critical rate $p_{\mathrm{c}}$, the system holds a high capacity for data transportation, while $p$ comes greater than $p_{\mathrm{c}}$, the system gets into congestion phase in a hurry. However no matter how we change the nodes number, the critical rate is almost the same. Hence we can draw a conclusion that the interaction between the cells has global dynamic characteristics.

To examine the effect of data objects' attributes for the shift of phase transition point[5,6], the ratio of abandoning data objects as a function $u$ is shown in Fig.3. In the case when the arrival $p$ is low, the network is not too busy and the influence seems unobvious. Once the network comes into the traffic congestion, the node forwarding the objects in the waiting queue without considering their lifetime will cause the increase of abandoned data objects. While in SNCA model, the attribute lifetime helps the node to choose which one to transfer in prior. From the results (Fig.3) of our second experiment it can be observed that compared with switching data object randomly, giving priority to the short lifetime data objects can reduce the ratio of abandoned data objects. As $p$ increases, the performance can be improved by more than 30%.

In conclusion, to realize the high performance storage system, we can utilize the attributes of data object to optimize the performance of storage network. The two experiments results show that data object flow has global relativity with the phase transition.

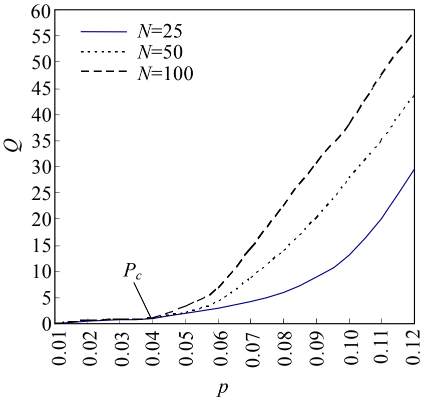


Fig.2 The number of data objects in the network with different arrival rate $p$

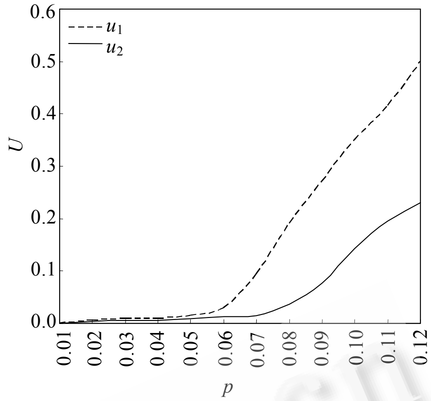


Fig.3 The ratio of abandoning data objects ($u_2$ uses the attribute lifetime while $u_1$ not)

## 3 OSDCA (object-based device cellular automata) Model

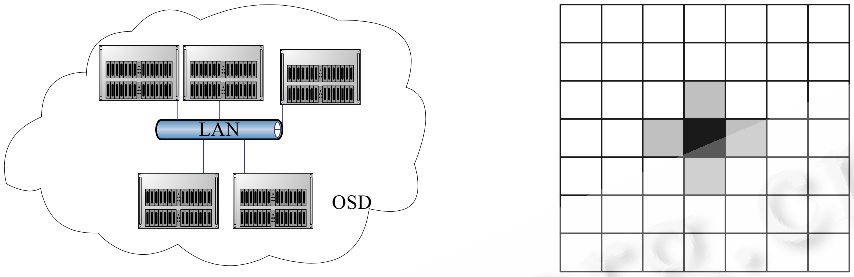
The access behavior seems as a complexity in a massive storage system. As the data stored in a special physical space, different requests play distinct impacts on the storage system. It cannot avoid that at times the requests are centralized on some special data excessively, and perhaps most of the requests will last for a long time. If the access frequency goes beyond the tolerance of the special physical area, it will slow down the access speed obviously, and in the end, the system may crash down. Such data access behavior brings about the emergence of hot data in most large-scale storage systems. Former researchers contributed greatly on how to achieve load balance in a storage system; however, most of them merely focus on data migration[10] or data replication[11] algorithm. Dr Qin[12] has accomplished some research on the combination of the two mechanisms in an OBS system, whereas his work was centralized on the analysis of the complex transfer phenomenon of data and I/O command by some complicated mathematic metric methods. We proposed a model called OSDCA using several simple transition rules based on CA to control the access traffic problem in a massive OBS system.

In OBS system, for each data object contains characteristics such as self-organization, they can shift the data they manage to other physical area dynamically[3]. During the migration or the replication, the data object does not need to communicate with the metadata service, since the OSDs can manage the data object by themselves. The clients send out their request without knowing what has happened inside the storage system, and finally they greatly benefit from the data transformation. Nevertheless the dynamical action of adjusting the distribution state of the data object is very complex, in addition, data migration on the one hand can reduce the workload of one node remarkably, and on the other hand the system can accelerate by data replication in that the mirrors can promote the I/O parallelism. Never can we neglect that the migration will cause the target nodes overloading[11], though data migration often reduces load unbalance effectively. To maintain the load balance in OBS system, the combination of such two mechanisms is necessary. The nodes have many choices to migrate or replicate data each time, hence we need to recognize the intricate phenomenon and implement an adaptive way to avoid load imbalance in a large-scale OBS system.

In the OSDCA model, we use a bidimensional CA to simulate the behavior about how these data objects alter the physical data layout to achieve load balance. In this model we just take consideration of read request for it is the primary activity in a storage system as proved in practice. From the simulation results we found that the data objects can initiatively adapt various access behaviors by using some specific mechanism.

### 3.1 Model definition

The storage system considered in this model consists of nodes placed as a two-dimension lattice. In Fig.4(a), it shows that the storage space of OBS system mainly consists of numerous OSDs. To simplify the problem, we consider it as a square with $N^2$ nodes (storage space) (Fig.4(b)). Each data objects control some specific physical zone, like different nodes or part of some nodes.



(a) (b)

Fig.4 Model of OSD storage space

We use Von.Neumann model[12] like the model SNCA mentioned above. The neighborhood is a mapping from the lattice set as Eq.(4) (Fig.4(b)). We assume the four nearby neighbors as the nearest storage space, for the purpose of reducing the time cost by using data migration and replication.

**Algorithm 1**. Data self-organization in an OBS system

```
while the storage system is alive do
    for each cell do
        if the node is overloaded do
            look up the set of the neighborhood nodes' state
                if the set adapt to migrate do
                    Data migration to the target node
                else if the set adapt to replicate do
                    Data replication to the target node
                end if
                end if
        end if
    end for
end while
```

The state of a site represents the data access frequency. Although the size of each node cannot be altered, each data object can manage part of various nodes. That is to say, the node (storage space) may belong to more than one data objects. Consequently, not withstanding a specific data object maintains a high access frequency, we found the nodes did not appear as hot spots by the algorithm. A pseudocode of the algorithm can be seen in Algorithm 1.

In the proposed algorithm, each time the system will check all the storage objects to find all the overloaded nodes, and then all these nodes make decision where to migrate or where to replicate by the set of their neighbor's state.

### 3.2 Simulation and results

We also use MATLAB as the tool for simulation. We quantify the traffic congestion of the storage system by the hotspot rank of the entire storage space. The physical storage space can be described as a large lattice space, and each lattice has a special access frequency rank.

To strongly express the massive storage system, we assume the value of $N$ is 50.

We assume there are five access ranks, from 0 to 4 which increase gradually. These 5 ranks represent 5 states of the nodes. The five states are described as the following metrics:

State 0: Access frequency value between 0 and 1;

State 1: Access frequency value between 1 and 2;

State 2: Access frequency value between 2 and 3;

State 3: Access frequency value between 3 and 4;

State 4: Access frequency value greater than 4.

State 0 is the lowest load, state 2 is the critical spot of low load and hot spot, and state 4 is the threshold level which demonstrates the extreme overloading in this node. If the state of the node is more than 2, the node needs to move some data to the near node.

According to the following rules, the system can adjust the data distribution to fit for different access behaviors.

1. The clients make request to the data of the lattice space randomly with a probability $p$. Each cell has the same chance to be accessed next time. Every time step if a request is sent to a cell, the access frequency value of that node will be added. When the rank is more than 4, new request will be plug into the waiting queue.

2. When the rank of a node is more than 2, it will look up the set of its neighbors' state to decide whether to migrate or to replicate. As we assume that each node has five possible states and four neighbors, 625 rules need to be defined for nodes' behavior. Only the set with 0 and 1 is taken into consideration for only the nodes in state 0 or state 1 contribute to data transfer. If the set contains state 0, the node will choose one with state 0 to migrate data. As a result, the access frequency value of this node will decrease by $\lambda$ and one of the target nodes will increase by $\lambda$. While state 1 and other three states expect state 0 to comprise the neighbors state set, the node will replicate part of its data to one of the target nodes whose states are 1. We assume that the access frequency value of this node will decrease by $\lambda/2$, and one of the target nodes will increase by $\lambda/2$. Unfortunately if the set contains neither 0 nor 1, the overloading state will continue.

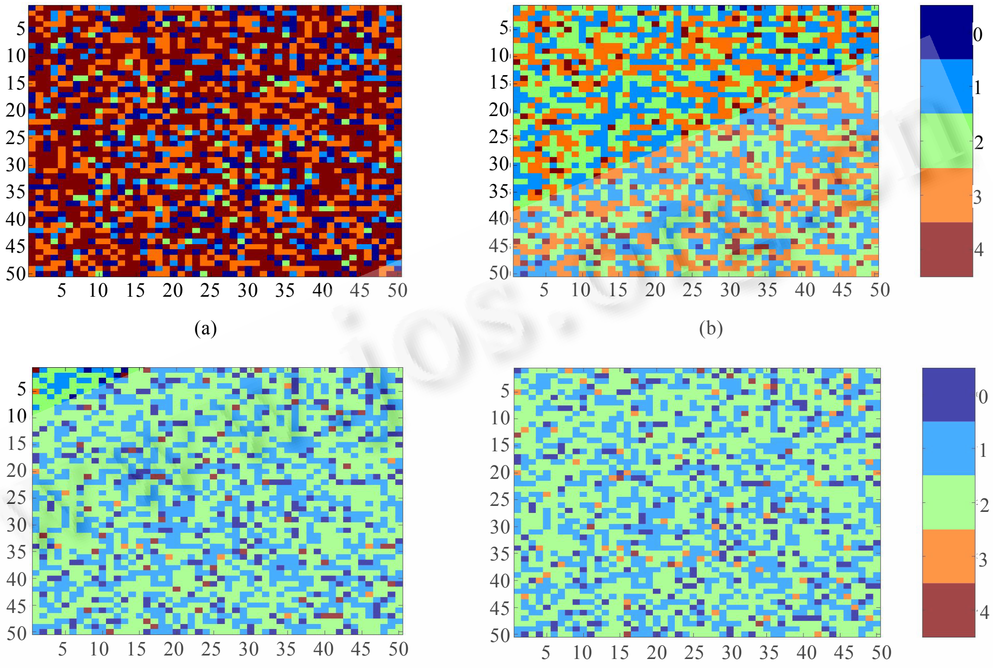
3. The whole system responses the request with a speed $\mu$ randomly, which leads to the decrease of the access frequency value of the nodes.

4. The time step updates one by one.

In our first experiment, $\mu$ is assumed to be a little greater than $p$. Under this circumstance, the system has spare capacity to handle the remaining request on the nodes. At the beginning, the state of the system is initialized as there are plenty of hotspots, and the distribution is 1200 nodes in state 4, 600 nodes in state 3, 100 nodes in state 2, 200 nodes in state 1, and 400 nodes in state 0.

Now we turn our intention to the comparison of the simulation result from Fig.5. Figure 5(a) shows the initial state of the system with so many overloading nodes. When time steps update, the amount of hotspots decreases gradually (Fig.5(b)). At last, as observed in Fig.5(c), almost all the nodes stand on the states 0, 1, and 2. Still it can not be avoided that some nodes will be accessed extremely in the process of execution, and such event does not occur on the same node continuously in most case (Fig.5(c) and Fig.5(d)) by using the transition rules.

By changing $p$ and $\mu$, we observe the step number, the system needed to get to the right balance state, increases as $p$ increases and decreases as $\mu$ increases. Compared with the real system, the system restoration consumes more time when the access frequency is higher. In the worst case, the system is subject to extreme heavy load constantly, no matter how the system makes use of the migration or the replication mechanism, the situation will not change. Therefore the processing speed ($\mu$) of the system plays a significant role in the load balance course. Yet in a real storage system, we cannot prevent the worst access behaviors from occurring at times, and actually overloading appears frequently in a large-scale storage system. The way to adapt these challenges is to cut down the proportion of hot objects with less time cost, hence the system has enough capability to fit for the occurrence of the next worse access behavior.



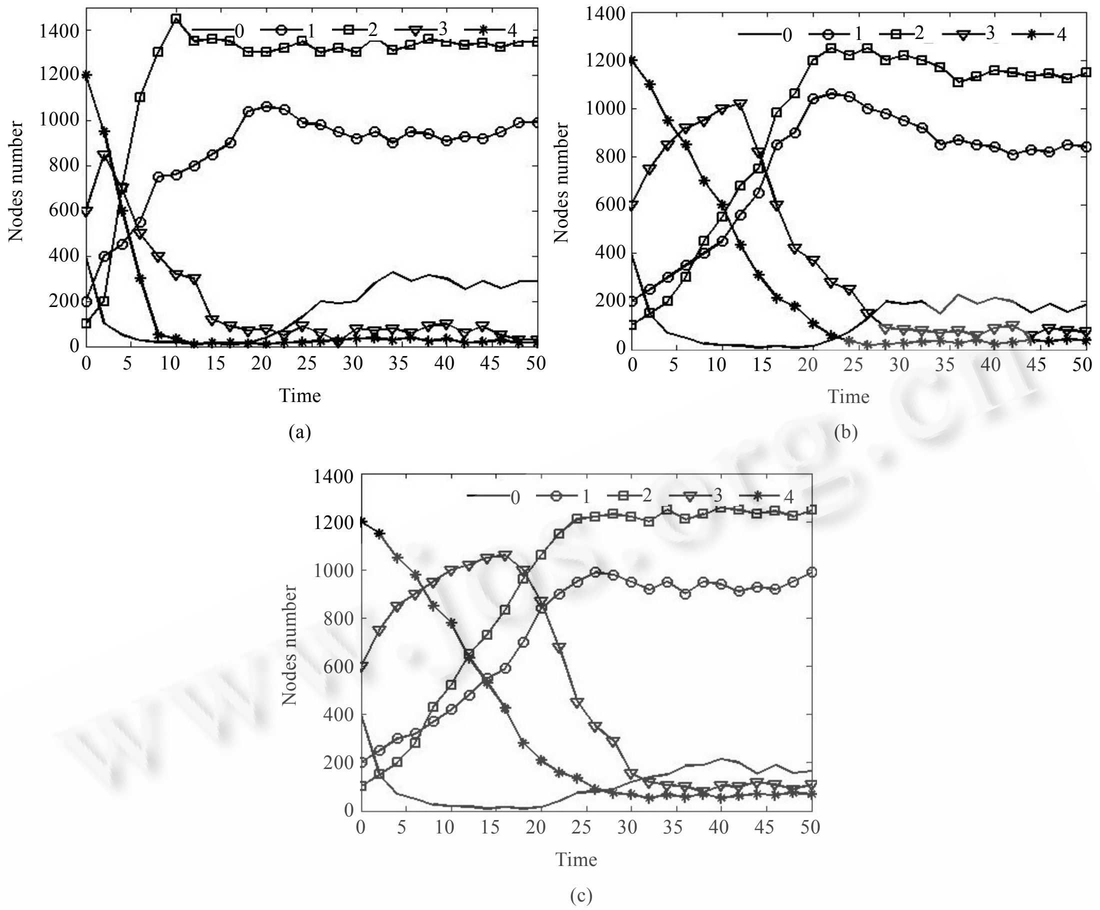
(a) (b)

(c) (d)

Fig.5 Execution steps of the simulator

Our second experiment is used to compare the algorithms which only use migration or replication respectively with our algorithm. Figure 6 demonstrates the comparison of only using migration mechanism or replication mechanism and combining the two methods as our algorithm. As shown in Fig.6(a), the time steps the system needs to arrive at the load balance state is about 15. After that time, about ninety percent of the nodes are in state 1 or state 0. We can also observe from Fig.6(b) and Fig.6(c) that about 25 time steps are needed by using migration lonely and about 30 time steps are required by only using replication. Though data migration makes use of free storage resources fully, the clients can access different nodes for the same recourses simultaneously, as a result of data replication. Therefore combining the advantages of the two mechanisms together expedites the disappearance of overloading nodes.

Compared with the conventional massive storage systems, storage devices in OBS system can be aware of the access behaviors. By using some mechanisms such as the algorithms proposed above, the system can achieve adaptability to external behaviors like load unbalance. In conclusion, with the intelligence of OSD, data distribution in large scale OBS system can be self-organized.



(a)

(b)

(c)

Fig.6 Time steps cost with the three algorithms

## 4 Conclusion and Future Work

In this paper we studied OBS system from macro and micro aspects by using CA. From the macro aspect we propose a SNCA model, which uses an attribute of the object called lifetime to distribute the object in the storage network. The simulation results demonstrate that when the switches forward the object with considering the lifetime, different application requests will be served correspondingly. From the micro aspect we propose an algorithm combining with data migration and replication to solve overloading problem in the storage system, from the results of our experiments the simple rules of the OSDCA model greatly improve the retrieval speed.

In the analysis of the intrinsic characteristic those two models and the measure used to improve the performance of the storage system are based on the intelligence of the OBS system. The data object has its own attributes which management can benefit for handing, and the OSD devices can manage their inner objects without communicate with the metadata servers, thus the storage data layout can be self-organized.

Actually the object can possess various attributes, and each attribute could have one or more impact on the system, besides there are various kinds of objects, in our future work we will study how the other useful attributes contribute to the improvement of our model. Since the prevention of data traffic congestion has proved to be complicated, we intend to do further research on our algorithm based on CA to make full use of the intelligence of OSDs.

## References:

[1] Ohira T, Sawatari R. Phase transition in a computer network traffic model. Physics Review E, 1998,58(1):193−195.

[2] Chen MK, Li X. Behavior of a packet-switched lattice network model with inactive sites. Chinese Journal of Computers, 2005, 28(7):1130−1137 (in Chinese with English abstract).

[3] Mesnier M, Ganger GR, Riedel E. Object-Based storage. Communications Magazine, 2003,41(8):84−90.

[4] Chen MK, He T, Li X. A mean-field theory of cellular automata model for distributed packet networks. LNCS3090, Springer-Verlag, 2005,1(7):1130−1137.

[5] Yuan J, Ren Y, Shan XM. Investigation of a cellular automaton model for computer network. Acta Physica Sinica, 2000,49(3): 398−402 (in Chinese with English abstract).

[6] Yuan J, Ren Y, Liu F, Shan XM. Phase transition and collective correlation behavior in the complex computer network. Acta Physica Sinica, 2001,50(7):1221−1225 (in Chinese with English abstract).

[7] Chopard B, Luthi P, Masselot A. Cellular automata and lattice boltzmann techniques: An approach to model and simulate complex systems. Advances in Complex System, 2002,32(1):80−107.

[8] Fuks H, Lawniczak AT. Performance of data networks with random links. Mathematics and Computers in Simulation, 1999,51(2): 101−107.

[9] Lin C. The Performance Ion of Computer Network and Computer System. Beijing: Tsinghua University Press, 2001 (in Chinese).

[10] Dasgupta K, Ghosal S, Jain R, Sharma U, Verma A. QoSMig: Adaptive rate-controlled migration of bulk data in storage systems. In: Proc. of the 21st Int'l Conf. on Data Engineering. IEEE, 2005. 816−827. http://icde2005.is.tsukuba.ac.jp/

[11] Fujiwara T, Miyazaki J, Uemura S. Data migration for a widely distributed storage system using autonomous disks. In: Proc. of the Data Engineering Workshops, 21st Int'l Conf. Tokyo: IEEE, 2005. 1267−1267. http://icde2005.is.tsukuba.ac.jp/

[12] Qin LJ, Feng D. An adaptive load balancing algorithm in object-based storage systems. In: Anon, ed. Proc. of the Int'l Conf. on Machine Learning and Cybernetics. Guangzhou: IEEE, 2006. 297−302. http://www.icmlc.org/2006/welcome.htm

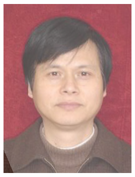
## 附中文参考文献:

[2] 陈茂科,李星.不完全活动的分组交换格点网络模型的行为.计算机学报,2005,28(7):1130−1137.

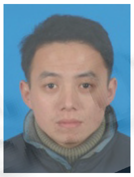
[5] 袁坚,任勇,山秀明.一种计算机网络的元胞自动机模型及分析.物理学报,2000,49(3):398−402.

[6] 袁坚,任勇,刘锋,山秀明.复杂计算机网络中的相变和整体关联行为.物理学报,2001,30(7):1221−1225.

[9] 林闯.计算机网络和计算机系统的性能评价.北京:清华大学出版社,2001.75−88.



**CHEN Jin-Cai** was born in 1960. He is an associate professor at the College of Computer Science and Technology, Huazhong University of Science and Technology (HUST). His research areas are computer networking, networking storage and embedded system.



**GE Xiong-Zi** was born in 1982. He is a Ph.D. candidate at the Department of Computer Science and Engineering, HUST. His current research areas are network storage, mobile storage and wireless networking.

**HE Ping** was born in 1983. She is a master candidate at the Department of Computer Science and Engineering, HUST. Her current research area is network storage.