Research on Normalized Network Information Storage Method Based on Deep Reinforcement Learning

Qiang Wang[1](✉) and Lai-feng Tang[2]

[1] School of Intelligence Technology, Geely University, Chengdu 641402, China
[2] Xinjiang Institute of Technology, Aksu 843100, China

Abstract. This paper makes a deep research on the standardized network information storage method, applies deep reinforcement learning to the standardized network information storage, and proposes a standardized network information storage method based on deep reinforcement learning. Each file to be archived is firstly divided into non-overlapping semantic fragment data blocks according to its information content. Each data block will encrypt its content through hash function, and obtain a signature as its identifier for data archiving. The archived data is divided into task data, resource data and document data. The rack-aware data placement strategy was developed based on the deep reinforcement learning algorithm to improve data reliability, availability and network bandwidth utilization, and the cloud storage model was designed to normalize the network information storage of classified archived data. Experiments show that this method has a high average disk read and write speed and can meet the requirements.

Keywords: Deep reinforcement learning · Normalized network · Information storage · Hash function · Archived data

1 Introduction

Electronic information processing technology and the rapid expansion of digital information resources continue to develop, the continuous development of storage technology and change has been quietly to [1]. Jim Gray, a Turing Prize winner, has pointed out that in an online environment, the amount of data generated every 18 months is roughly equal to the sum of all previous data [2]. For enterprises, data and electronic information have become an important resource for their survival, and play a key role in promoting the sustainable development of enterprises. However, information storage systems will still be inaccessible due to natural disasters or man-made damage, or even face the risk of loss of data. If the disaster causes the loss or damage of important information, the enterprise will not be able to carry out normal business operations, the enterprise and customers will suffer serious economic losses [3]. Global data centres have a disaster probability of approximately 0.2 per cent per year, meaning that 2 out of 1000 data centres have experienced severe data disasters and data loss. After the 911 terrorist attacks, many

W. Fu et al. (Eds.): ICMTEL 2021, LNICST 387, pp. 579–589, 2021.
https://doi.org/10.1007/978-3-030-82562-1_56

enterprises have lost a lot of critical business data because of the lack of disaster-tolerant application, which makes them unable to run their applications normally. According to IDC, 55% of US companies that suffered a system disaster in the 1990s failed immediately because they could no longer operate, 29% went bust within two years, and only about 16% remained open. Studies show that if a company's data and application systems go down for an hour, the company loses $150,000 to $6.45 million. Based on this background, the normalized network information storage method is deeply studied, and a new normalized network information storage method based on DRL is proposed. The innovation of the research method is that deep reinforcement learning goes deep into the standardized network information storage method to meet the use needs to a certain extent.

2 Design of Standardized Network Information Storage Method Based on Deep Reinforcement Learning

2.1 Data Archiving

Each file to be archived is first cut up into discrete chunks of semantic fragment data based on its information content, each chunk encrypts its content through a hash function, and calculates the signature as its identifier [4]. The metadata information for these chunk

Table 1. Main function interface and function description of file segmentation.

Serial number	Interface APIs	Function	Description
1	TagRetrieval	Tags list tagsist = tagRetrieval (File* f, out Data* buo)	Find metadata information for a file
2	FileDriver	Chunk* cp = fileDriver (File* f, Tagsist* t1)	Breaks files into semantic fragments based on their metadata information, and returns a list of pointers to those fragments
3	Check_chunk	Objects op = check chunk (chunk* cp)	Check if semantic fragments should be further divided into objects before being stored
4	ChunkDriver	Object* op = chwikDriver (Chunk*cp)	Divides semantic fragments into data objects as actual units of storage and returns a list of pointers to those objects

fragments is stored in the MDS, which compares the identifiers of the chunk fragments to determine whether there is currently a duplicate data fragment, and the non-duplicate chunk fragments are encapsulated into a fixed-length data object [5]. Sharding of files based on semantic- related information about the files, using a unified file sharding interface to provide standard interfaces for callers [6]. The main function interfaces and functional specifications for file sharding are shown in Table 1.

2.2 Data Typing

The archived data is divided into task-based data, resource-based data and document-based data [7], as shown in Table 2.

Table 2. Data breakdown details.

Serial number	Definition	Give an example	Characteristic
1	Task-based data	Facebook uses Hadoop for log analysis and recommendation systems; Baidu uses Hadoop for log storage and statistics, web page data analysis and mining, etc.; Yahoo! Hadoop will be applied to their own anti- spam system and web search queries	In addition to storage space, task-based data also require nodes to have strong computing power, especially for some large data analysis and processing applications, node computing power is often the performance bottleneck of task execution. The owner of task-based data is usually some enterprise user
2	Resource data	Users will be some commonly used software, hot movies, music, pictures and other data files uploaded to the cloud, through file sharing other users can also browse or download. Or the user uploads the data just for backup purposes. This data is usually stored as a resource to the user, and the operations performed on it are mostly read and write, and generally do not involve large or complex computations	This kind of data storage usually requires data nodes to have better storage space, and safe and reliable, the computing power of the node is not high. In addition, this kind of application is sensitive to data transmission speed and usually requires data nodes to have smaller data access latency and larger data transmission bandwidth

(*continued*)

Table 2. (*continued*)

Serial number	Definition	Give an example	Characteristic
3	Documental data	Various office files, drawings, program codes and other data related to users' work, study or life	These data are usually private data, confidentiality requirements are relatively high. Security and reliability of data storage are the most important considerations for users

Task-based data is often the source data for some applications, and is stored by the user to provide more valuable data or statistics that the user cares about. Data generated based on applications such as data backup or information sharing is usually resource-based. Documental data refers mainly to some working documents or materials of users.

2.3 Data Copy Placement

Making rack-aware data placement strategies based on deep reinforcement learning algorithms to improve data reliability, availability, and network bandwidth utilization [8]. The first copy of the data block is placed on one node of the local rack, the second copy is placed on another node of the same rack, and the third copy is placed on a node of a different rack. This strategy reduces the transmission of data between racks and shortens the completion time of write operations. Generally, the probability of rack failure is much lower than that of node failure, so the strategy does not affect the security and reliability of the data [9]. At the same time, this strategy reduces the network bandwidth of data transmission because the data copies are only placed on two different racks. However, when using this data placement strategy, the data blocks can not be placed evenly on the various data nodes. Among them, 1/3 of the data blocks exist in a node, 2/3 of the data blocks in a rack, the remaining data blocks are evenly distributed in other racks. This strategy improves the efficiency of write operation without reducing the performance and reliability of data write.

The Chunk module is grouped based on the data queue of the current large-scale network library information data set. If the differentiation vector coefficient k is less than 1 in the current data storage space structure, the current data node is a directional structure. At this time, the structure is set as the key transmission channel and depends on the routing control, then the quantization formula is as follows:

$$K(l) = \frac{M_n(l)}{h_m} \tag{1}$$

In the formula, $K(l)$ is the actual index weight of the current data storage data network transmission channel L; h_m represents the current actual load rate of data storage; $M_n(l)$ is the actual expected load of the current transmission channel; To reflect the quality of the current data transmission channel, the higher the expected value, the better the quality.

According to the actual physical definition of $AVE(l)$, the transmission channel of current data structure may be blocked in some cases. If this part of transmission channel structure is treated as node algebra, its weight will increase. Therefore, the optimization function of distributed data structure is proposed:

$$F_{HJG} = R_j^i \times D_{lj}^i \times K(l) \tag{2}$$

In the formula, R_j^i represents the effective path distribution rate between the current network transmission channel node i and node j, and D_{lj}^i represents the actual flow element in the current data structure matrix.

2.4 Network Information Storage

Design a cloud storage model to normalize network information storage for classified archived data [10]. The designed cloud storage model can be divided into four main parts by structure: user access layer, data service layer, data management layer and data storage layer, as shown in Fig. 1.

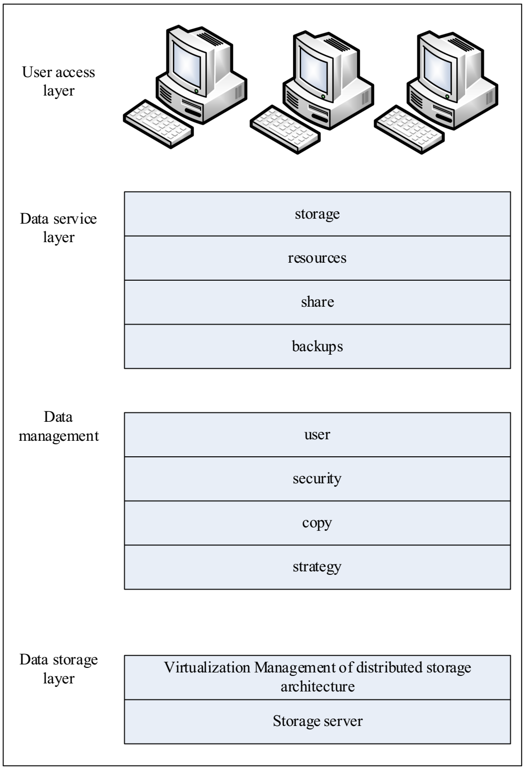

Fig. 1. Cloud storage model designed

The data storage layer is mainly responsible for the actual data storage management of cloud storage model. An excellent cloud storage model can provide users with different types of storage services, and all kinds of data will be uniformly stored in the cloud storage model, forming a huge data storage pool. Traditional data organization based on a single data storage server is not suitable for data storage under network conditions, and it has great limitations in network throughput and storage capacity. However, the data organization based on network architecture needs a huge number of nodes and complex coding algorithm to ensure the reliability of its data. In contrast, the data organization based on multi-storage servers can better meet the needs of online data storage services. When the number of users is large, the distributed data center can provide better data storage services for users [11]. The data storage layer of cloud storage model can connect different types of storage devices to form an organic whole, and realize the centralized and unified management of mass storage data. At the same time, the cloud storage model can monitor the running state of storage devices and dynamically expand storage capacity.

As the name implies, data management is mainly responsible for cloud storage model data management, including data copy strategy, data security access management and other operations. In a cloud storage model architecture, the data management layer provides a common management interface to the upper data services layer [12, 13]. Through user identity management, security access management, data copy management and data distribution policy management, the bottom layer of storage management can be seamlessly connected with the upper layer of storage application, and a large number of heterogeneous storage devices can work together to provide cloud storage users with better storage and management services.

The data service layer is responsible for service matching of user's storage task requests [14, 15]. The data service layer is a flexible part of the cloud storage model, which can provide different business application interfaces and services directly according to the user storage tasks [16, 17]. For example, data storage services, storage space rental services, multi-user data sharing services and data backup services.

The user access layer is the interface between the storage model and the user. Through the User Access Layer, any cloud storage service user can login to the cloud storage platform and use cloud storage service in any place using different network terminal devices according to the standard application interface provided by the cloud storage model [18].

According to the structure and characteristics of cloud storage model, the model can be divided into physical device virtualization, storage node virtualization and storage area network virtualization. By layering the virtualized storage resources, the cloud storage model greatly reduces the complexity of storage management, and makes the model more flexible and scalable. The cloud storage model only needs to add all storage devices to the storage resource pool to create virtual volumes for virtualization of storage resources. The cloud storage model is managed in a centralized manner and does not need to focus on the state of physical devices in the data center.

Using deduplication technology, cloud storage model can optimize the utilization of storage resources, and can save storage space by removing duplicate files or data blocks from storage model; on the other hand, data de-duplication technology can also help cloud storage model reduce data transfer, improve network throughput of storage model, and reduce resource consumption and network resource utilization cost.

3 Experimental Results

3.1 Experimental Design

In order to verify the performance of the designed normalized network information storage method based on deep reinforcement learning, a performance verification experiment was carried out on Hadoop-1.0.0. In the experimental environment, three racks (racks A, B and C) shall be set up, under each rack shall be placed five machine nodes, plus a name node, which is composed of 16 PCs, and the network topology thereof shall be shown in Fig. 2.

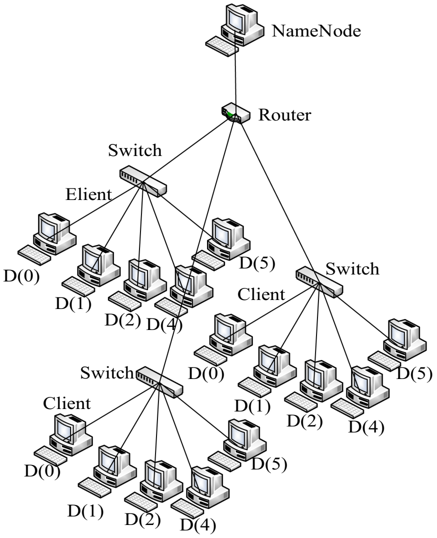

Fig. 2. Experimental network topology.

Among them, the first node D[0] under rack A is the client node, which does not store data, namely there are 14 data nodes in the experimental network cluster. Network communication between nodes uses the TCP/IP protocol, each node has 100M of Ethernet bandwidth. In order to accord with the heterogeneity of cluster environment in cloud storage service, 16 PCs in the cluster are configured to form a heterogeneous experimental environment, which makes the experimental results more authentic. The main configurations of the 17 PCs are shown in Table 3.

The operating system for all machines is Windows xp sp3, and VMware 7.0 is installed on all the machine nodes to emulate the Linux environment. The version of Linux in the virtual machine uses Ubuntu 10.04.

The average disk read and write speeds of the normalized network information storage method based on deep reinforcement learning were tested by HD _ Speed disk

Table 3. Main configurations of 17 PCs.

Group	Cluster node	Processor	Memory	Hard disk
	NameNode	Intel core i5 2.4 GHZ	4 GB	7200 RPM 500 GB
Rack A	Client	AMD P320 2.1 GHZ	2 GB	5400 RPM 320 GB
	D[1]	AMD Athlon64 X2 2.7 GHZ	2 GB	7200 RPM 500 GB
	D[2]	Intel E5200 2.5 GHZ	4 GB	7200 RPM 320 GB
	D[3]	Intel dual core 2 GHZ	1 GB	5400 RPM 320 GB
	D[4]	Intel dual core 1.8 GHZ	2 GB	5400 RPM 320 GB
Rack B	D[0]	Intel dual core 2 GHZ	2 GB	5400 RPM 320 GB
	D[1]	Intel dual core 2.16 GHZ	3 GB	5400 RPM 500 GB
	D[2]	ntel core i5 2.4 GHZ	4 GB	7200 RPM 500 GB
	D[3]	AMD Athlon64 X2 2.7 GHZ	2 GB	7200 RPM 500 GB
	D[4]	AMDLlano APU A6 3.4 GHZ	1 GB	7200 RPM 320 GB
Rack C	D[0]	Intel dual core 1.6 GHZ	1 GB	5400 RPM 320 GB
	D[1]	Intel dual core 2.16 GHZ	2 GB	5400 RPM 320 GB
	D[2]	Intel core i5 2.4 GHZ	4 GB	7200 RPM 500GB
	D[3]	Intel dual core 2.4 GHZ	2 GB	5400 RPM 320 GB
	D[4]	Intel dual core 2 GHZ	1 GB	5400 RPM 320 GB

performance testing tool. The standardized network information storage method based on deep reinforcement learning is proposed as follows:

Step1: Measure the stored information according to the current data information transmission channel, and set the output efficiency value D_j^i of each major source end of the transmission channel to the extreme of the current network traffic demand;

Step2: Update the current network transmission channel;

Step3: Import the updated network transmission channel into the substitute value, and bring in and calculate the current flow at the calculation end;

Step4: Update the source end to resolve the structure rate;

Step5: Standardize the network information storage method according to the structure rate.

3.2 Analysis of Results

When the amount of storage data is small, the experimental data of average disk read and write speed at each node of normalized network information storage method based on deep reinforcement learning is shown in Table 4.

When the storage volume is medium, the experimental data of the average disk read and write speed of each node of the normalized network information storage method based on deep reinforcement learning are shown in Table 5.

Table 4. Experimental data of average disk read and write speed at each node.

Group	Node	Average reading speed (MB/s)	Average write speed (MB/s)
Rack A	D[1]	50.2	40.3
	D[2]	55.6	42.3
	D[3]	54.5	46.1
	D[4]	56.3	41.0
Rack B	D[0]	54.2	46.8
	D[1]	59.3	45.6
	D[2]	54.2	44.5
	D[3]	57.2	49.5
	D[4]	54.0	44.5
Rack C	D[0]	57.2	41.3
	D[1]	58.0	45.3
	D[2]	57.6	48.2
	D[3]	59.3	49.3
	D[4]	57.0	44.2

Table 5. Experimental data of average disk read and write speed at each node.

Group	Node	Average reading speed (MB/s)	Average write speed (MB/s)
Rack A	D[1]	40.2	32.6
	D[2]	42.3	31.5
	D[3]	44.2	37.2
	D[4]	47.2	36.3
Rack B	D[0]	41.3	34.2
	D[1]	45.6	38.0
	D[2]	44.2	36.9
	D[3]	47.0	35.2
	D[4]	42.3	38.0
Rack C	D[0]	45.0	31.2
	D[1]	44.2	35.0
	D[2]	48.9	35.2
	D[3]	45.1	33.2
	D[4]	44.2	30.2

When there is a large amount of storage data, the experimental data of average disk read and write speed at each node of normalized network information storage method based on deep reinforcement learning are shown in Table 6.

Table 6. Experimental data of average disk read and write speed at each node.

Group	Node	Average reading speed (MB/s)	Average write speed (MB/s)
Rack A	D[1]	34.3	20.6
	D[2]	36.9	26.9
	D[3]	35.2	29.6
	D[4]	35.0	29.6
Rack B	D[0]	35.4	28.6
	D[1]	32.6	28.4
	D[2]	31.5	24.5
	D[3]	38.8	28.2
	D[4]	36.4	29.3
Rack C	D[0]	31.5	24.25
	D[1]	37.2	24.6
	D[2]	36.3	29.6
	D[3]	34.2	24.5
	D[4]	38.9	29.9

Experimental results show that the designed normalized network information storage method based on deep reinforcement learning has higher average read and write speed and can meet the use requirements.

4 Conclusions

With the continuous increase in the scale of information, the world created 1234EB of total information in 2011, of which 234EB data is closely related to personal lives of multimedia electronic information, personal information storage data has accounted for 70% of the total size of the digital world. According to IDC, worldwide data storage is growing at an annual rate of 58%. The normalized network information storage method based on deep reinforcement learning realizes high average disk read and write speed, which is of great significance to the normalized storage of network information.

References

1. Blancquaert, L., Everaert, I., Missinne, M., et al.: Effects of histidine and β-alanine supplementation on human muscle carnosine storage. Med. Sci. Sports Exerc. **49**(3), 602–609 (2018)

2. Chen, S., Wang, W., Xu, W., et al.: Plant diversity enhances productivity and soil carbon storage. Proc. Natl. Acad. U S A **115**(16), 4027–4032 (2018)
3. Wang, J., Zhang, W., Li, B., et al.: Effects of Fe Modified Na 2 WO 4 Additive on the Hydrogen Storage Properties of MgH 2. J. Wuhan Univ. Technol.-Mater. Sci. Ed. **34**(5),1030–1036 (2019)
4. Wu, J., Lu, C., Xu, X., et al.: Preparation of Cordierite-mullite Ceramics for Solar Thermal Storage **34**(5), 1062–1070 (2019)
5. Xu, X., Song, J., Wu, J., et al.: Preparation and thermal shock resistance of mullite and corundum Co-bonded SiC ceramics for solar thermal storage. **35**(1), 16–25 (2020)
6. Gentil, D.F.D.O., Ferreira, S.A.D.N., Rebouças, E.R.: Germination of Psidium friedrichsthalianum (O. Berg) Nied. seeds under different temperature and storage conditions. J. Seed Sci. **40**(3), 246–252 (2018)
7. Carvalho, M.L.M.D., Lopes, C.A., Ribeiro, A.M.P., et al.: Could packing and pelleting keep the quality of tobacco seeds during storage? J. Seed **40**(3), 296–303 (2018)
8. Liu, S., Bai, W., Zeng, N., et al.: A fast fractal based compression for MRI images. IEEE Access **7**(99), 62412–62420 (2019)
9. Pero, M., Askari, G., Skra, T., et al.: The change in the color of heat treated vacuum packed broccoli stem and floret during storage: effects of process conditions and modeling by ANN. J. Ence Food Agric. **98**(11), 4151–4159 (2018)
10. Guofang, X., Xinhua, W., Kaihong, W., et al.: Effects of 1-methylcyclopropene on texture properties of Rabbiteye blueberry during long-term storage and simulated transportation. Food. Technol **38**(2), 188–192 (2018)
11. Liu, S., Li, Z., Zhang, Y., et al.: Introduction of key problems in long-distance learning and training. Mobile Networks Appl. **24**(1), 1–4 (2019)
12. Fu, W., Liu, S., Srivastava, G.: Optimization of big data scheduling in social networks. Entropy **21**(9), 902 (2019)
13. Jung, J.T., Lee, J.K., Choi, Y.S., et al.: Effect of rice bran and wheat fibers on microbiological and physicochemical properties of fermented sausages during ripening and storage. Korean. J. Food, Anim. Resour. **38**(2), 302–314 (2018)
14. Ronghui, Y., Shuwei, Y., Yi, X., et al.: NetGO: Improving large-scale protein function prediction with massive network information. Nucleic Acids Res. **1**, 1–9 (2018)
15. Xue, K., Zhou, H., Meng, W., et al.: A Lightweight and Secure group key based handover authentication protocol for the software-defined space information network. IEEE Trans. Wireless Commun. **6**(99), 10–15 (2020)
16. Bai, L., De Cola, T., Yu, Q., et al.: Optimal dynamic multi-resource management in earth observation oriented space information networks. IEEE Wirel. Commun. **26**(2), 8–9 (2019)
17. Guo, S., Zhang, B., Yang, T., et al.: Multitask convolutional neural network with information fusion for bearing fault diagnosis and localization. IEEE Trans. Industr. Electron. **67**(9), 8005–8015 (2020)
18. Ghada, J., Rahim, K., et al.: An adaptive duty-cycle mechanism for energy efficient wireless sensor networks, based on information centric networking design. Wireless Networks **26**(2), 791–805 (2020)