World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:9, No:11, 2015

NSBS: Design of a Network Storage Backup System

Xinyan Zhang, Zhipeng Tan, Shan Fan

***Abstract*—**The first layer of defense against data loss is the backup data. This paper implements an agent-based network backup system used the backup, server-storage and server-backup agent these tripartite construction, and the snapshot and hierarchical index are used in the NSBS. It realizes the control command and data flow separation, balances the system load, thereby improving efficiency of the system backup and recovery. The test results show the agent-based network backup system can effectively improve the task-based concurrency, reasonably allocate network bandwidth, the system backup performance loss costs smaller and improves data recovery efficiency by 20%.

***Keywords*—**Agent, network backup system, three architecture model, NSBS.

I. Introduction

WITH the development of international technology, society is increasingly dependent on computer systems, more and more attention has been paid on the protection of the data security. Statistics show that 51% of infected companies of 9.11 were unable to function fully within the following 2 years due to the difficulty of data recovery. The other 43% went bankrupt directly, and only 6% survived [1]. Data disaster backup [2] is facing numerous challenges: increasing data throughput, shrinking backup time window, multi-storage complexity and multi-platform interoperability. It is necessary to design a backup system [3]-[5] which tackles with these problems while providing the scalability capacity to meet future growth at the meantime. Network-based data backup becomes a promising solution.

Network backup [6], [7] technology is to back up the data hosted on various spots within the network [8]. It needs to fulfill central management and cross-network backup ability, using a single workstation and minimum human cost to manage the whole network, while supporting cross-platform backup [9], [10]. To realize this objective, multi-agent based methodology can be used to accomplish operation integration with different platform systems. A specific agent can be used at client side for a specific platform and data processing, while the server side processes unified requests from the client agents and manages them altogether. The client agent deals with specific interface while the server deals with a single interface for processing. When a new backup client is added, the server only needs to dispatch a new agent to the client system while keep using the same data processing logic to process them. This guarantees favorable system scalability [11].

In a network-based backup scenario, backup agent will be always in a passive listening state, it becomes a bottleneck as the backup server has numerous data to process. The agent needs satisfy the following requirements for solving this problem: (1) it can proactively send the backup or recovery commands, make the local scheduling for the backup and realize the backup automation; (2) Reduce server's resource consumption by balancing the backup server's pressure with the help of various strategy; (3) Ensure the consistency and integrity of data so as to efficiently serve uses for data backup.

Xinyan Zhang is with the Wuhan National Laboratory for Optoelectronics, School of Computer Science, Huazhong University of Science and Technology, Wuhan, China (e-mail: zhangxinyanhust@gmail.com).

Zhipeng Tan is with the Wuhan National Laboratory for Optoelectronics, Key Laboratory of Data Storage System, Ministry of Education, School of Computer Science, Huazhong University of Science and Technology, Wuhan, China (corresponding author, e-mail: zhipengtan@163.com).

II. Related work

Network-based data backup has become a hot technology ever since it brings network to data storage industry. There are several ways to realize data backup. Judging from backup mode, there can be physical and logical ones. Categorized by backup strategy, there can be full-scale backup, incremental or dispersion backup. They can be cold or hot backup according to whether the system can receive the user response and data update. An ideal backup system should provide multiple backup solutions [12], according to the importance of the data needed to backup, the time given to data backup, performance of object host and other factors, uses can choose the right backup strategy to protect the data at maximum and save storage space and reduce runtime impact meanwhile.

Researchers have done some significant study in the key points of storage backup technology, current studies include the following:

(1) Data mirroring technology. It is a process that creates more than one mirror imaging of the source data on different disks or its subsystems. The main mirroring system is source data and the standby is the replica of data backup. Mirroring is also called data replication technology. The mirror replicas can be constantly updated and replaced the mirror source when the latter is corrupted and lost in general. This take-over overhead is minimal so that business continuity can be ensured. Presumably [13] will double the storage space which is not efficient. Based on distance, the mirroring can be seen as either local mirroring or distant mirroring. Local mirroring happens within the same disk array, while distant mirroring spread on some different array, thus enjoying better disaster recovery.

(2) Snapshot [14], [15]. Snapshot is based on Point in Time (PIT) and aims at recording [16] and saving data mirroring [17] at one time. It has a very short backup window, it can keep and backup all data information at one time without affecting routine the normal I/O operations. There are three snapshot technologies involved: mirror separation, copy on write (COW) and write re-direction. Mirror separation

takes a full mirrored replica of the source just before the snapshot. When the snapshot time comes, it separates the replica with the source data and stops all the write operations to the source data, thus forming a snapshot volume. When the mirror volume finishes the data backup, it will be synchronized again and turn to be the next round of the snapshot mirrored volumes. As the operating time of mirror detachment snapshot technology is only the mirror detachment time, so it almost has no influence on normal operations. However, it takes a huge amount of space, and each scheduled snapshot time needs a full copy of the source volume. Copy on write (COW) is also called pointer based snapshot, it only creates a logic copy of source data volume and backups address pointer of all data blocks when the snapshot time comes. If an application has write operations in the source data block subsequently, the source data block will be copied to the snapshot area first, and update the snapshot pointer, then finish the write operation to the source eventually. Copy on write is a highly flexible technology, which takes minimal space and negligible performance impact. But as it does not actually create a full copy of the source, so it is incapable of recovering the source object after it is destroyed. Write re-direction technology, still take a logical copy of the source volume using snapshot pointers, but when a write operation comes, it will write the modified value of source data area to the new data sector, then modify the pointer set of system source data. Compared with the Copy on write technology, it reduces the process that copied the source data block to the snapshot area. It applies for the write-intensive applications, while COW is suited for the read-intensive applications. Both of them are logical snapshot technologies; they cannot recover the data when the source volume is destroyed compared with the mirror separation technology.

(3) Continuous Data Protect (CDP) [18], [19]. SNIA defines CDP as following: Under the premise of not affecting the main data operation, it is capable of continuous capturing and tracing of any source data modification, and also capable of getting back to the status of any previous point in time. CDP can provide the backup at block-level, file-level and application-level. There is a class of products called Near CDP; it merely takes high-frequency snapshot, tens or hundreds in an hour, for example. The true CDP product is to monitor all the I/O ops on the protected target. When monitor the writing I/O ops it first intercepts and holds the writing ops, backup the source data and then actually fulfills the write ops. This method monitors any change of all data, can backup status of source data at any point of time, it has a desirable RPO (Recovery Point Object).

(4) Tivoli Storage manager (TSM) [20] is an enterprise scale network-based storage management application that provides automatic storage management. It utilizes storage backup and offline storage data copy to efficiently protect business data. It can protect up to hundreds of different types of OS, including laptop and mainframe-like servers. It continuously runs centralized data servers of Storage Manager 365 hours 24 days without interruption service.

(5) NetVault [21] is a storage backup solution designed for mid/small scale enterprise customers. This solution includes automatic backup management software, field deployment and after sale technical support. It can satisfy both product and service needs.

(6) Data Protection Manager (DPM) [22] is application software that can optimize disk-based backup and recovery. Recovery looks just as easy as browsing shared folders or downloading copies from server.

Based on above analyses, the data backup should resolve the following problems: (1) Tremendous volumes of data ask to shrink the backup window, complexity of multi storage platform and capability to cross-platform operation. (2) Network backup system needs the agent based technology to fulfill the cross platform requirement and simplify the storage complexity between those systems. (3) It also needs to provide automatic backup, reduces the time window and customer visibility. (4) It should ensure the availability, consistency and completeness of the backup. As a user interface, backup agent should provide friendly UI while meet with above requirements.

III. Architecture of NSBS

NSBS is an Agent-based Network Storage Backup System, it uses tripartite architecture of backup servers, storage servers and backup agent servers, and it can place backup processing server and backup storage server on different servers. This greatly enhances the system flexibility and also separates the command processing and storage data bottlenecks. Backup server and storage server exist in the same LAN so it speeds up the updating ops. It also increases deployment flexibility as the backup agent and server can belong to different WAN network segment.

A. System Architecture

Agent-based network backup system can be described as Fig. 1. The system includes backup server, storage server and backup agent, which can be described as the following:

1. Backup Agent

Backup agent runs on host machine, works as the interface between backup storage system and the user to provide user backup data and other info query interface. It can also accommodate backup jobs at user's ad-hoc backup demand. In data backup process, backup agent receives user's job request, formats the command and sends it to the backup server, wait for the response (mainly for server's address and checksum). Using this info, backup agent formats the backup data to the storage server. The transmission can be detailed in 3 steps: (1) file attribute streaming; (2) file data streaming; (3) file hash checksum. Data recovery process is very similar to the backup, while the data flow is from server to backup agent. To provide backup and recovery parallelism, multi-threading and threading pool can be used. Other than some particular commands (logon, register, deletion, revoke and query), backup agent first formats

commands and sends them to the backup server. Backup server responds and then the backup agent will display the response to the user, in case of user query.

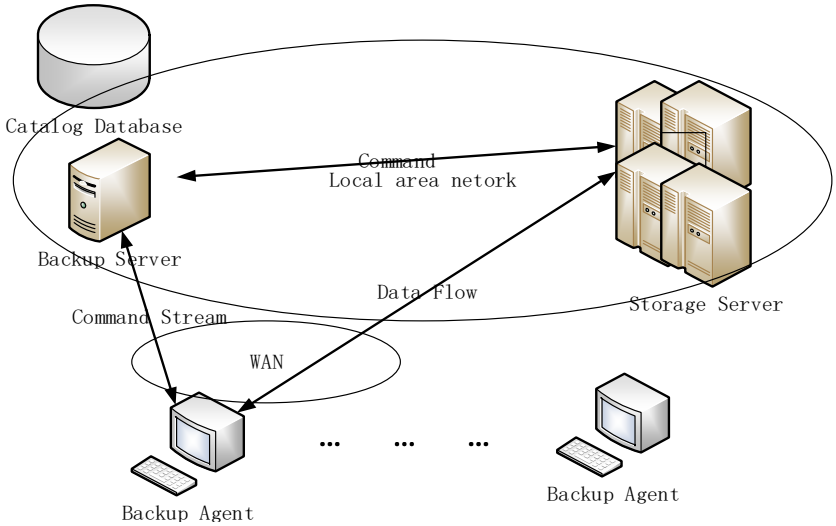


Fig. 1 System Architecture

2. Backup Server

Backup server is the central commander of the whole network backup system. It receives job requests from backup agent, coordinates those agents and distributes workloads. Backup server only interacts with backup agents and storage server via control stream. Backup and recovery data stream doesn't flow through backup server itself. This reduces the workload while enhances backup performance. Backup server will maintain a database, which contains all registered users in the backup system, and metadata of backup targets, versioning etc. for job scheduling and user query purpose.

3. Storage Server

Storage server is deployed at data DR center server, and mostly supported by RAID or tape. It is responsible for reading data from backup agent and write down to backup volume in some organized manner. During recovery, it uses corresponding searching technology to retrieve the data and sends back to agent to complete the recovery.

B. Backup Agent Design

Backup agent is the interface to user, which should provide a user-friendly interface. To ease the backup operation representing the user, both manual and automatic backup services are provided, the latter of which is implemented using job scheduling module. Job scheduling module uses multi-threading and threading pool to realize the parallelization of backup and recovery. Besides that, backup agent provides registration, login modules to serve as the portal for user login. Backup file deletion module makes it feasible for user to stop running backup and recovering jobs. Backup job query module provides user with detailed meta-data of the backup replicas. All of them together provide the distant disaster recovery and query service to users. Meanwhile, backup agent provides assistant service module for backup server. Multiplexing provides different channels for enquiry command stream and job control stream. Backup agent provides corresponding strategies, coordinating with backup server and provides consistent time synchronization in between. The backup agent module can be depicted as Fig. 2.

In the process of data backup, to reduce customer impact and ensure data consistency, backup agent needs to provide a certain level of technical support. At beginning of backup, a snapshot object is created for the data, and then being transferred. This makes sure that data is consistent. Meanwhile, the major part of backup process will not influence application's running. To double-check recovered file's data integrity, its hash value is checked against backup file. With above techniques, backup agent can guarantee data correctness and simple, reliable data backup service.

Backup agent separates data into meta-data stream, file data stream, integrity checking. Using snapshot technology, the consistency can be ensured. Using data info abstraction algorithm, the integrity can be ensured. Using job scheduling module, automatic backup can be provided. Under coordinated strategy with backup server, systemic optimization can be reached.

IV. Key Technologies in NSBS

A. File Data Organization

Backup agent is mainly used to retrieve local data to be stored and transmit in certain format to storage server. When storage server receives the backup data, it indexes the data based on file, job id, user profile etc. In process of file recovery, storage server index the file based on the target recovery file's info sent from backup server, and sends the file to backup agent. Backup agent creates the file based on file attributes and records the data into this file, and finally checks it against checksum. Thus, only when backup agent interacts with storage

server in some predefined format during transmission can it ensure data integrity.

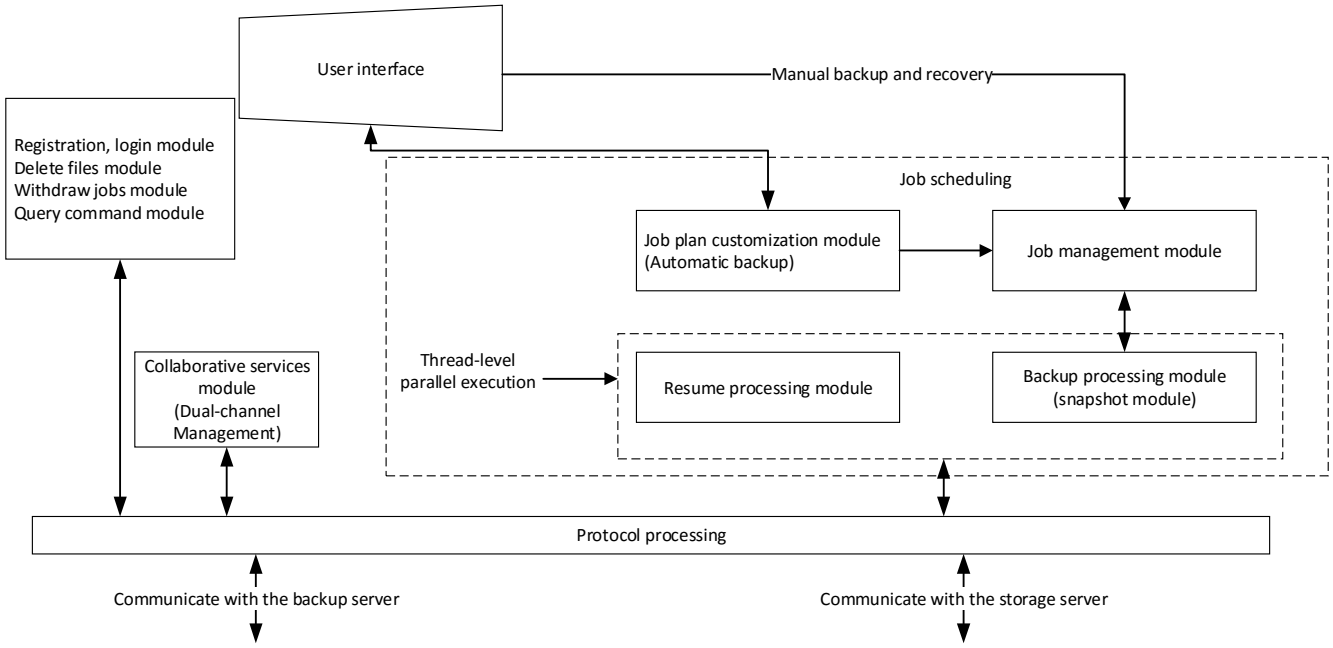


Fig. 2 Backup Agent Module

In backup process, the agent sends backup server the job request and gets corresponding storage server's address and then begins backup transmission.

Based on above analyses, the meta-data, data and data checksum is indispensible. To separate different info channels, the streams can be detailed as: (1) Data header, in format of <file indexing> <data stream identifier> <0>. (2) Data load. (3) Ending marker. The full process to send a file can be depicted as Fig. 3. When the file is getting too big, the data stream needs to be divided into several data packages. The indexing will mark each package corresponding to offset in file. This helps storage server to form indexing and reconstruction during recovery.

Data recovery is the reverse process of data backup. The storage server will first receive attribute stream, and react differently given file attributes. Backup agent creates the file based on file attributes. Ongoing file data is then written to this new file. Finally, checksum is calculated and compared to ensure data integrity. Similarly, each info package of recovery process also constitutes of data header, data and ending marker.

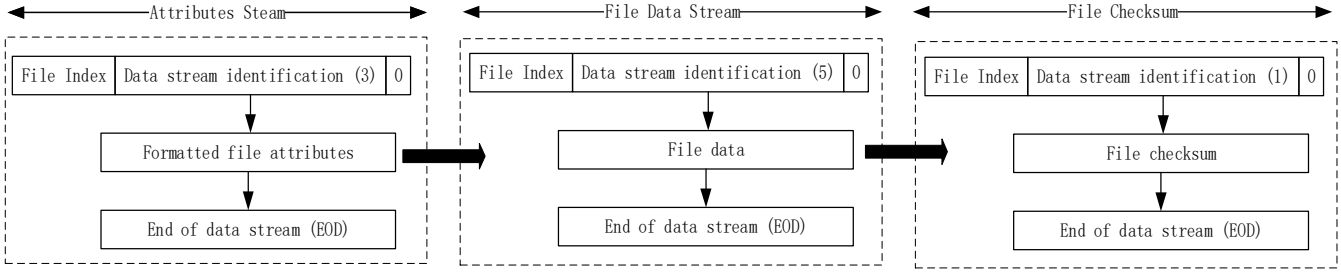


Fig. 3 File transmission

B. Data Correctness Insurance

Different backup files under the same backup target have certain correlations. Meanwhile, different data blocks under the same file are correlated. If the backup data becomes inconsistent with source data, the whole backup process becomes meaningless. Only by ensuring the correct correlation between backup data and consistency can the data be useful after recovery.

For the single file, the data can become corrupted between backup and recovery, which makes the recovered data incomplete or incorrect. To ensure the integrity, checksum can be performed to prove the usability of the recovered data.

System uses OpenSSL [23] technique to realize integrity checking. The data abstraction can be performed as: (1) abstraction algorithm initialization, pick algorithm H, offset file abstraction variable h. (2) Calculate the i^{th} data block h′, $h' = H(\text{datai})$. (3) Accumulate the i^{th} data block abstraction

value h, as $h = H(h \otimes h')$. If not complete, loop on step 2 to calculate the next one. Finally, return the abstraction value h. The algorithm can be summarized as Fig. 4.

The integrity checking can be detailed in 3 steps: divide data into data blocks, calculate abstraction value, and compare the value with source for consistency. During backup, file data is transmitted via network. Backup agent reads in data accordingly and put them into data package. When data package gets too big, it is divided into trunks. If the last trunk cannot meet the minimum requirement of required number of data bytes, it is padded with spaces. This process realizes the separation of original data. The recovery process is just the reverse of it. The whole process of transmission, calculation, and compare for integrity can be explained in Fig. 5.

The slow hashing algorithm described in Fig. 4 is used, backup agent can provide 4 summarization algorithms: MD5 (Message-digest algorithm 5), SHA1 (Secure Hash Algorithm), SHA256 and SHA512. Fig. 6 describes the hash calculation, updating hash sum process.

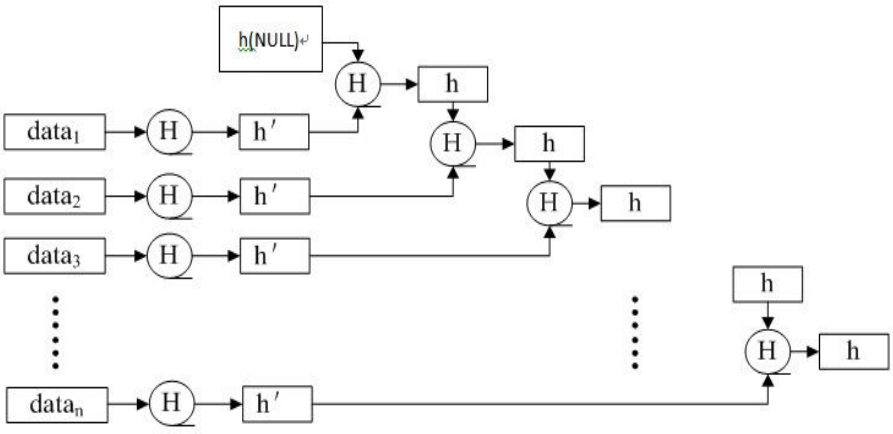


Fig. 4 Data abstraction calculation process

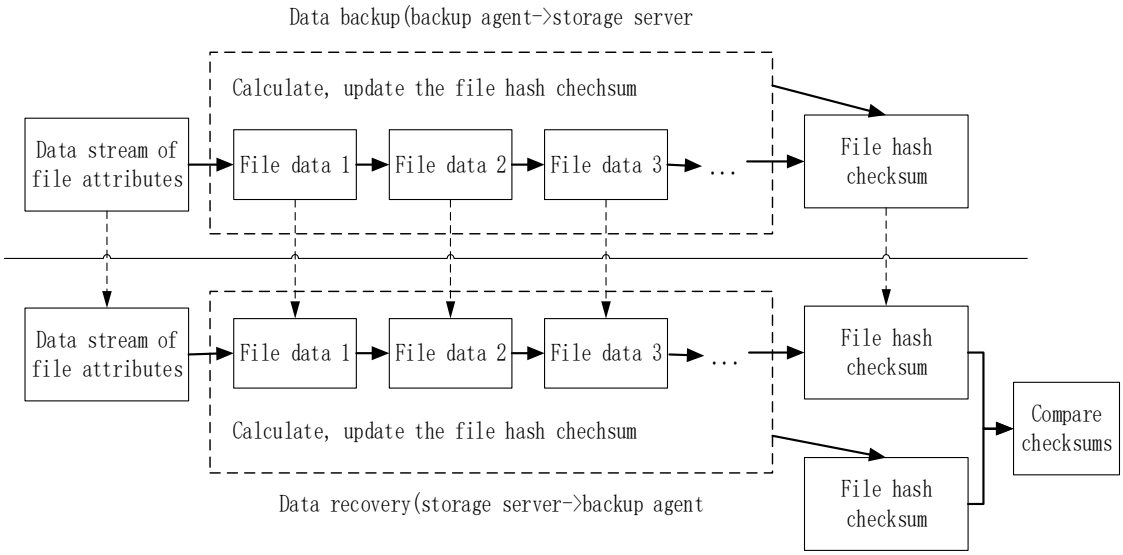


Fig. 5 The process of file integrity checking

Via comparing the hash between backup and recovery file, one can examine data completeness and ensure data integrity.

C. Coordination Service

From users' perspective, software agent needs to represent customer demand fully, providing intelligent backup service, as shown in previous Section B; as a part of the system, coordinating service for the whole system, offloads system workload to reach full-scale optimum.

1. Coordinating of Backup Servers

System adopt a three-party architecture, one for flexibility of system deployment, one for efficiently distributing command stream and data stream to avoid system performance bottleneck. The backup server and agent, though having different command streams, are correlated. This section explains how network dual-plex channeling works in details, and also how the corresponding agent privilege is resolved. The three-way architecture results in backup server having the

complete version while the backup agent holding the partial version. The following Section 1.1 also explains how the backup agent coordinates to ensure data consistency.

1.1 Network Dual-Channel Technology

Backup agent and server's command stream can be separated into two categories: one is enquiry command stream, and query-based operation such as backup data deletion and job revoking; to the backup system, the backup or recovery job has the highest priority to ensure accuracy and timing. This requires independent network channel support. Meanwhile, backup agent needs to support the inquiry during backup and recovery. In conclusion, the dual- channel technology splits the inquiry command streams and job control streams, reducing the influence between inquiry operation and job control, ensure that the job and query can be executed at the same time, enhancing system performance and efficiency.

As Fig. 7, the whole system adopts the strategy of splitting command stream and data stream. The dual- channel splits the streams and their time and space correlation, improving job processing efficiency.

Since network dual-channel introduces the different sockets for command interaction, the backup agent needs to correctly control the lifecycle of sockets. This requires a certain strategy to create and recycle network sockets to reduce resource consumption.

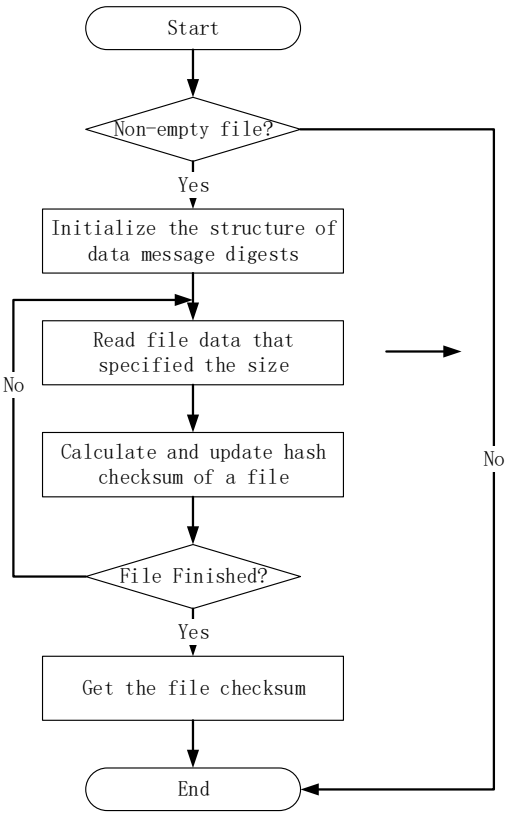


Fig. 6 Calculate, update hash sum

Backup agent's privilege control module is designed to effectively control the socket creation and destruction.

(a) Channel creation

For inquiry command flow, the system will utilize the socket created after log-in and initialize it for command inquiry channel; for job control flow, the system will automatically create one if the socket does not exist. A timer is kept after channel creation to effectively monitor the channel.

(b) Channel destruction

For inquiry command stream, the timer is always offset when the channel sends a command or receives one. When the timer expires, the channel will be closed. For job control flow, the timer will periodically check if the job queue is empty, in which case the channel will be closed too.

(c) Privilege control

When two channels are both closed, a heartbeat message is sent from backup agent to backup server to gain system allowed offline interval. The backup agent will be offline under such status.

(d) Come back online

When the backup agent is offline, the two-way channel will send an online request to backup server before it can function and proceed only when the resources are required. Meanwhile the backup agent will destroy all expired privilege control thread. If a channel is still in use while another channel is closed, the backup agent is still online. In such circumstance, the reopening of closed channel doesn't count as re-online.

1.2 Time Checking

The system time between backup agent and server is hard to be consistent. When the agent is performing incremental and differential backup, the agent needs to get the running time of last full size backup from the backup server, which will be served as the baseline to compare backup file's modification and creation time. If the file has been recreated or modified, this file needs backup. The backup time gained from database is the system time of backup server. This time is used by agent for adjustment.

As shown in Fig. 8, during time checking, the backup agent sends an Adjust Time Command to backup server at certain interval. The starting is marked as T_{start}; the backup server sends system time T_{get} to the agent; when the agent receives the time feedback, compare with local time. This differential time is as the under formula:

$$T_{different}=T_{get}-T_{start}-(T_{end}-T_{start})/2$$

For more accurate differential time, the first two time checking is dropped and the average of the rest is calculated.

To save backup server's resource consumption, the time checking is done by the agent, not by the server so as to reduce the workload of backup server and enhance system performance.

2. Coordinating between Backup Agents

As an interface between backup server and user, the backup server coordinates with server in exchange of backup server.

From the system's perspective, the backup agent coordinates with those servers to accomplish backup and recovery, enhancing the system performance and efficiency.

Besides the coordination with the server, the coordination among agents can greatly reduce the pressure among agents while at the same time improving performance. This introduces the concept of master-slavery backup agent. The main agent is the backup agent mentioned above. Only this master agent is visible to servers. The slavery agents form an IP-SAN storage grid with the master agent. The slavery agents get the distant backup service via master agent. The master agent is the proxy server for the LAN. As shown in Fig. 9, the slavery can map file system info to main agent via LAN protocol mapping. The master agent can get file info from slavery agent thus to protect those files on slaveries. Meanwhile, users can get continuous data protection service on demand on agents.

In order to achieve backup speed for the backup system, and to reduce the interaction time between them, the backup server should be deployed with storage server. Meanwhile, the main proxy sits with its slaveries in the same LAN. As the interface for backup servers, it accomplishes the task of backing the data to server. Besides, the policy setup and strategy constraints of the system effectively hide the slavery agent. This protects them from malicious attack and information leak.

The master agent realizes file granularity backup functionality, such as integration backup based on dates, weeks, months, years etc. to form a three-way backup architecture. The master-slavery mechanism can realize fine granularity backup, by using Continuous Data Protection (CDP) so as to achieve RPO and RTO goals, enhancing emergent response ability.

As shown in Fig. 10, the agent constructs CDP meta-data and CDP data and represents them in file format to the master agent for backup. Master agent can cache them locally, or directly send to servers for distant backup. The master agent needs a full strategy of indexing CDP metadata to manage the slavery CDP data.

The benefits of this master-slavery are: (1) Offload some system functionality; (2) Less resource management for servers (3) Master agent can provide certain level of caching to slaveries. When the WAN is not available, the master can save the data locally until the connection is erected again. Meanwhile, the system provides multi-granularity for data protection, and protects user data in real time.

V. EVALUATION

A. Analysis of Snapshot to Backup Performance

Snapshot brings certain effect to the system backup speed, but the overall performance of the backup system has increased greatly. The data backup process can be divided into five periods: T_p, T_{vi}, T_b, T_{vf}, T_u. T_p ($T_{prepair}$) is the preparation of the data backup, T_{vi}($T_{VSSinital}$) is the time of establishing data snapshots; T_b(T_{backup}) is the time for network data transmission; T_{vf}($T_{VSSfree}$) is the time of deleting the snapshot and other data structures at the end of the data backup; T_u (T_{update}) is the time of updating the index of system.

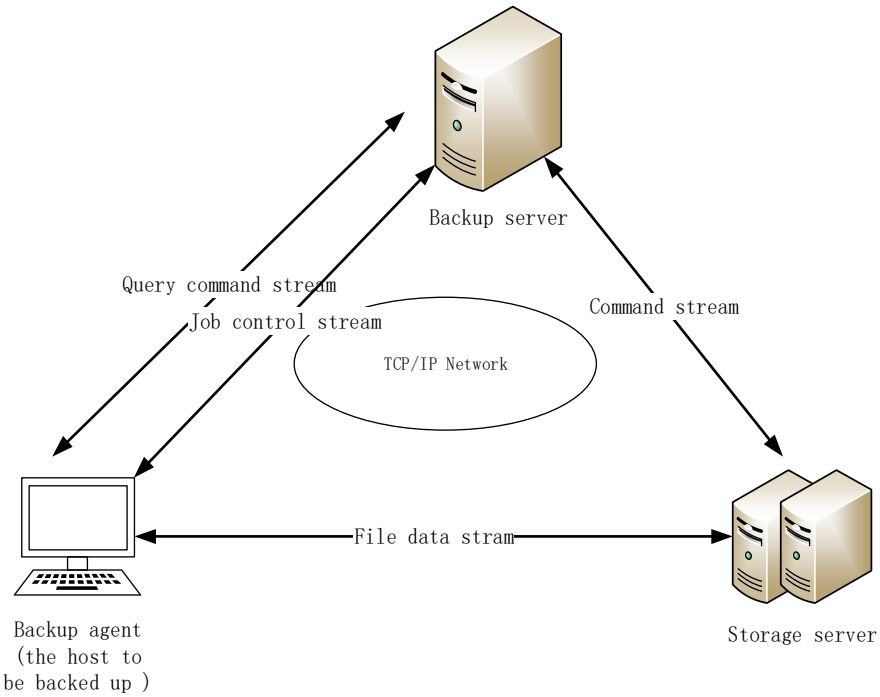


Fig. 7 Network dual- channel for splitting control flows

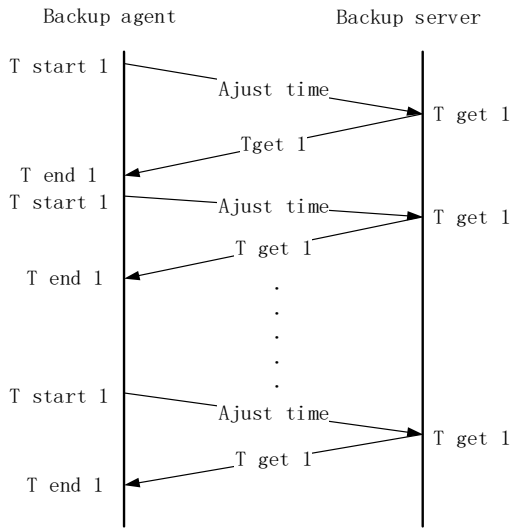


Fig. 8 Time checking process

As shown in Fig. 11, for one single file, the time proportion of four periods in the whole backup time. For one single backup file, the time of initialization (T_p) and update the index (T_u) are almost stable. With the backup file is increasing, the time percentage of T_p and T_u are decreasing, the influence on the backup time is smaller and smaller. As shown in Fig. 11 (a), when the file is small, (T_p) and (T_u) take up a larger proportion of the whole backup time, but they don't increase with the size of the file. As shown in Fig. 11 (b), with the increasing amount of data file, the backup time is mainly determined by the network transmission time, system initialization, index update and creation and destruction of snapshot have little influence to backup system.

If the file size is small, the snapshot technology has certain influence to the backup speed, but when the file size increases, time of creating and destroying does not increase, it is a stable value, snapshot's influence on backup speed is more and more little. At the same time, due to the snapshot, the backup system window is reduced to a fixed value; it is equal to advancing the performance of the backup system distinctly.

B. Effects of Classification Indexing to System Performance

We compare the hierarchical organization and grading index based on the user's data management method of NSBS to Bacula. Bacula is an open source backup software whose metadata is stored in the database, the real data stored in the storage server, files in the storage server no index, and read and write data in the form of tape library, each recovery is sequential reading and writing the disk, and no distinction of the volume of a storage device, multiple user data can be stored in the same storage volume, so one user's recovery operation need to traverse all the data on the storage volume. On the contrary, NSBS which uses data hierarchy organization and grading index management methods on the storage volume of the target user, which reduces the time to find recovery data, then it can improve the efficiency of system recovery.

The experiment selects jobs of different file size but they have the same number(100) of file, the jobs backup data from 4MB to 8Gb, every time the amount of data is twice the previous. From the Fig. 12, we can see that the backup speed of Bacula is a little more than NSBS, the reason is NSBS needs some time to create the index, but when the job size is 8GB, the speed gap of Bacula and NSBS become small, that is the time of create index is a small proportion of the whole time of backup when the jobs of backup data is large.

In obviously, the recovery time of two methods is increased gradually along with the increase of the amount of data. From Fig. 13, we can see the recovery speed of NSBS is higher than Bacula; the reason is the index helps NSBS to enhance the speed of recovery. The time of NSBS's data recovery can improve the efficiency of about 20%. From the two experiments we can draw a conclusion that the whole performance of NSBS is better than Bacula.

C. Comparison to NSBS and similar Backup system

Fig. 14 is the performance contrast test of backup and recovery for agent-based network backup system (NSBS) and several kinds of commercial backup systems. Test uses a single hard disk to backup and recover a single file of 600MB. The test result shows that the NSBS's performance of backup is not optimal because of the snapshot and index need to create before backup. But the influence will be smaller and smaller with the increase of amount of backup data, and the index and snapshot will give great help to data recovery. The performance of the whole backup system has greatly improved.

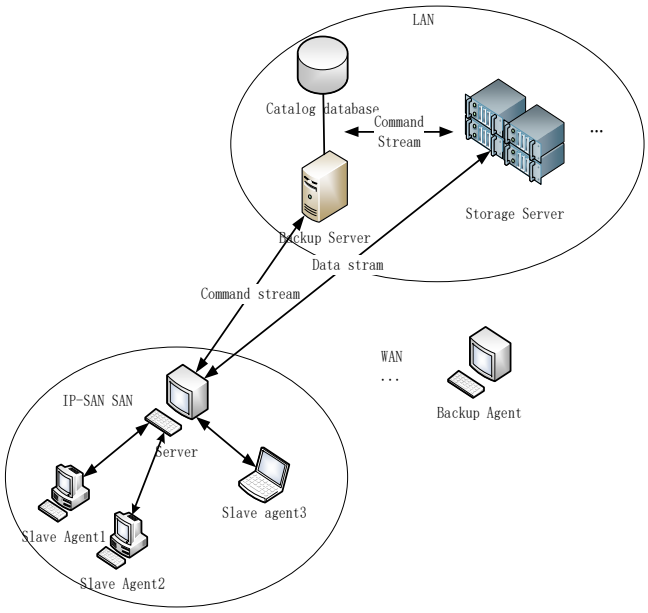


Fig. 9 The deployment design of master-slavery agents

VI. Conclusion

The paper advances an agent-based network data backup system-NSBS, it has three parts architecture of backup server, storage server and backup agent. We realize the snapshot and hierarchical index in the NSBS too. From the system performance test, the whole performance of NSBS is better than most the backup system. Now the NSBS has been used in

commerce in a certain range.

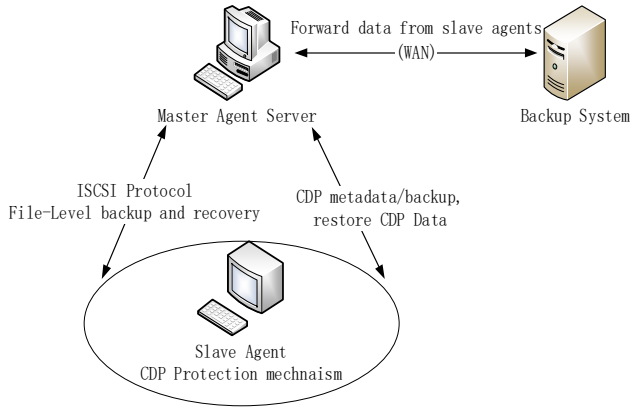
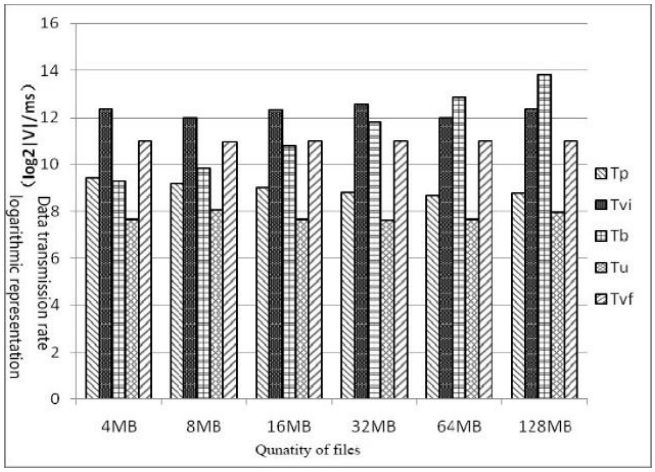
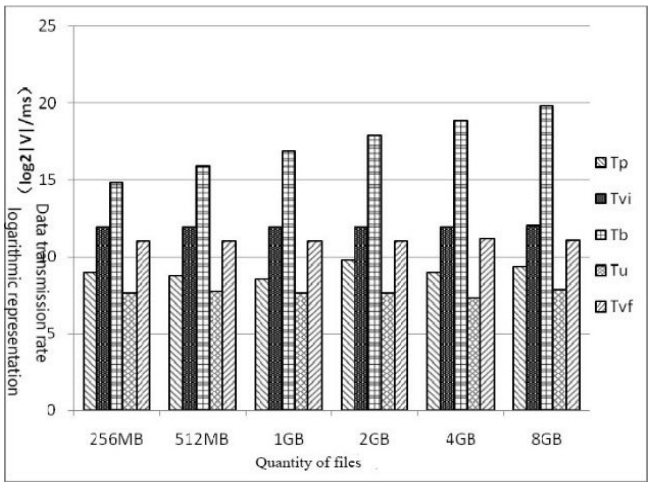


Fig. 10 The interaction between master and slavery agents



(a) A small amount of data backup



(b) A large amount of data backup

Fig. 11 Each time period comparison of Data backup

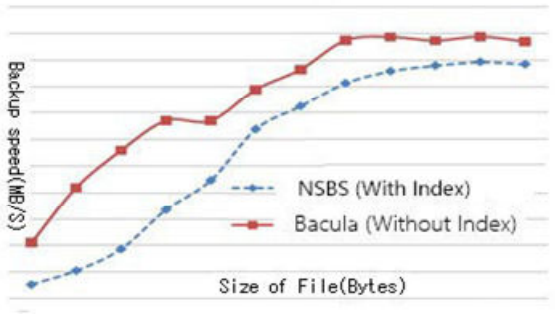


Fig. 12 Comparison data backup speed of NSBS and Bacula

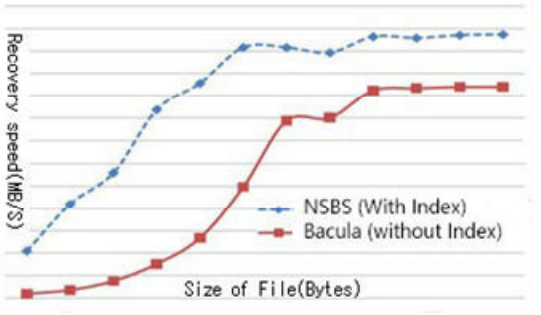


Fig. 13 Comparison data recovery speed of NSBS and Bacula

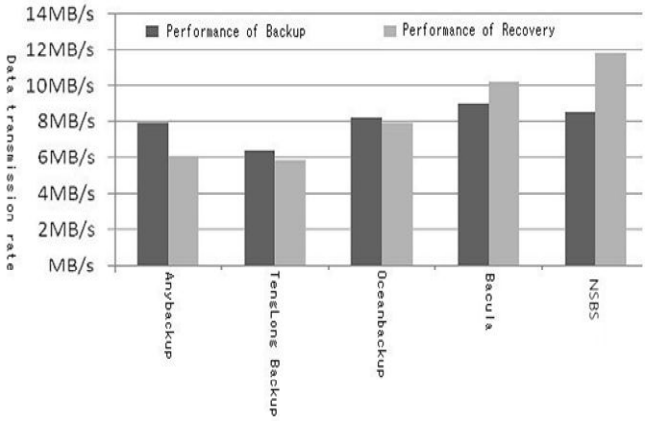


Fig. 14 Comparison of performance of backup and recovery for different backup system

ACKNOWLEDGMENT

This work is supported by 973 project 2011CB302301, the National Basic Research 973 Program of China under Grant by National University's Special Research Fee (C2009m052, 2011QN031, 2012QN099), Changjiang innovative group of Education of China No. IRT0725, is supported by Electronic Development Found of Information Industry Ministry.

REFERENCES

[1] Weng zongcheng, Li zhanhuai, Zhu Liping. Research on Key Technologies of remote disaster tolerant system. Computer processing, 2009, (1)147-150.
[2] G. Gonnet. Expected length of the longest probe sequence in hash code searching. Journal of the ACM, 28(2):289, 304, April 1981.

[3] C. Mohan, Kent Treiber. Algorithms for the Management of Remote Backup Data Bases for Disaster Recovery. in: Proc. Ninth International Conference on Data Engineering. Vienna, 1993.511~518.
[4] Richard P. King, Nagui Halim. Management of a Remote Backup Copy for Disaster Recovery. ACM Transactions on Database Systems, 1991,16(2):38~368.
[5] Jim Gray. What Next? A Dozen Information-Technology Research Goals. Journal of the ACM, 2003,50(1):41~57.
[6] Karl Larson. Improving Availability in VERITAS Environments. in: the 14th USENIX System Administration Conference, New Orleans: USENIX Association,2000.59~66.
[7] W. Curtis Preston. De-dupe and Disk Backups-What you need to know before taking the plunge. SearchStorage.com,2008.
[8] Hector Garcia-Molina, Christos A. Polyzois. Issues in disaster recovery. in: Proc. IEEE Compcon Sprint '90. Thirty-Fifth IEEE Computer Society International Conference. San Francisco,CA,1990.573~577.
[9] Data Domain. Disk-Based Backup and Recovery with Unprecedented Reliablity and High Performance at the Cost of Tape. LEGATO Networker DiskBackup Option & Data Domain DD200 Restore, 2004.1~16.
[10] Jon William Toigo. Disaster Recovery Planning: Strategies for Protecting Critical Information Assets. Second Edition. USA: Prentice Hall, 1999.12~14.
[11] Roselinda R. Schulman. Disaster Recovery Issues and Solutions. Hitachi Data Systems white paper WHP-2, 2004.2~6.
[12] Zhiwei Qu, Yan Chen, Zhenhua Zhang etc. Efficient Data Restoration for A Disk-based Network Backup System. in: Proc. Of the IEEE International Conference on Mechatronics, 2004:584~590.
[13] K.Keeton, C.Santos, D.Beyer, J.Chase, J.Wilkes. Designing for disasters. in: Proc. of 3rd Conference on File and Storage Technologies, San Francisco,CA,2004:59~62.
[14] L. Shrira, H. Xu. Thresher. An efficient storage manager for copy-on-write snapshots. in: Procedings of USENIX Annual Technical Conference. Boston, MA, USA, 2006.57~70.
[15] Kwangho Cha, Jin-Soo Kim and Seungryoul Maeng. snapPVFS: Snapshot-able Parallel Virtual File System. in: the 14th IEEE International Conference on Parallel and Distributed Systems. Victoria, Australia, 2008.221~228.
[16] Cunhua Qian, Syouji Nakamura and Toshio Nakagawa. Optimal Backup Policies for a Database System with Incremental Backup. Electronics and Communications in Japan, 2002, 85(4):1~9.
[17] Hugo Patterson, Stephen Manley, Mike Federwisch et al. SnapMirror: File System Based Asynchronous Mirroring for Disaster Recovery. in: Proceedings of the FAST 2002 Conference on File and Storage Technologies. Monterey, California: USENIX Association, 2002.81~91.
[18] Qing Yang, Weijun Xiao, Jin Ren. TRAP-Array: A Disk Array Architecture Providing Timely Recovery to Any Point-in-time. In Proceedings of ISCA: The 33rd Annual International Symposium on Computer Architecture, Boston, USA, 2006: 289~301.
[19] Jingning Liu, Tianming Yang, Zuoheng Li, Ke Zhou. TSPSCDP: A Time-Stamp Continuous Data Protection Approach Based on Pipeline Strategy. Frontier of Computer Science and Technology (FCST), Japan-China joint, 2008: 96~102.
[20] Barry Kadleck, Nigel Bentley, Michael Dann etc. IBM Tivoli Storage Manager Version 5.1: Technical Guide. International Business Machines Corporation, April 2002.
[21] Keiper C. NetApp SnapMirror Block Level Incremental Backup to Tape with NetVault Backup (J). Quest Software, 2012.Microsoft. Introduction to Microsoft System Center Data Protection Manager 2006 White Paper.
[22] Buchannan S, Hedblom R, Gomaa I. Microsoft System Center Data Protection Manager 2012 Sp1(M). Packt Publishing Ltd, 2013.
[23] Viega J, Messier M, Chandra P. Network Security with OpenSSL: Cryptography for Secure Communications (M). "O'Reilly Media, Inc.", 2002.