High Availability Services of Client in Large-scale Cluster System

Zhongmin Zhang*, Yin Tian, Lei Li and Yufang Wang

College of Information and Communication, National University of Defense Technology, Xi’an, Shaanxi, 710106, China

1*Corresponding author’s e-mail: zon_8986@126.com

Abstract. Aiming at the disadvantage of high availability based on node redundant, a high availability scheme based on storage devices redundant was researched and designed in large-scale cluster system with network storage devices by working on system kernel. The advantages of low-cost, low-complexity of maintenance and high-utilization of resources were received. Besides, an asynchronous prototype system of high availability based on log was implemented. Test proved that anticipated goals in system performance, data consistency and availability were achieved.

1. Introduction

Currently, information processing has been depended on distributed cluster system more and more. With the scale of the system is gradually enlarged, single point of failure (SPOF) has increasingly become serious problems. Because running time of task in distributed cluster system may be so long, it will take enormous loss if task is interrupted due to SPOF. System fault tolerant can be provided by technology of High Availability (HA) [1], and system reliability can be increased too.

There are some troubles below when HA is transplanted into cluster system directly [2]: (1) Workload and complexity of service deployment are increased, it is required to deploy availability system for every service. (2) Complexity of system management is increased. Workloads that manage every availability service system independently are unimaginable in the system that hundreds of services were provided. (3) Utilization of system resources is low, it will take certain degree resource waste that can not share resources among different availability service systems.

Hereat, improved method of high availability services of client is given in this article. When system is running normally, services on local disk run, and service synchronization with background redundant equipment is being worked with strategy set beforehand simultaneously; service can be switched to backup in background equipment which could provide service continuously when SPOF happens.

2. Data consistency

Data consistency (recoverability) is a key factor to measure HA system. If service data that had been synchronized can not keep consistency status, that is, service backup is in a state of inconsistency, the service on the host where SPOF happened is irrecoverable.

Definition Timestamp $\tau^p(t)$ and $\tau^s(t)$ represent image, at the time of $t$, belong to production center and backup center respectively. If it exists $t \leq t'$, at the time of $t'$, make $\tau^p(t') = \tau^s(t)$, we can
assume that production center and backup center satisfied Data Consistency. Phase shift between production center and backup center known as master-slave phase shift (marked for $V$), $V = t - t_{\text{max}}$, $t_{\text{max}} = \max \{ t | t < \tau \land \tau \leq \tau (\tau) \}$.

Data consistency must be ensured in HA systems. If the production center is not available for some trouble, data need recovering or services need restarting from backup center, and master-slave phase shift can be used to evaluate data volume lost in time of form.

If system or application program can be restarted from backup data, the data can be considered consistency data. For example, data contained a file system, on which fsck can be run normally, we can think that is consistency data. If and only if the data contains all the updated cut-off at some point and time after that there are no any new updates arrived, we can believe that data is consistent. For instant, the recently created file is likely to disappear after a file system restarted.

Since the existence of the operating system cache, part of the write operations of I/O operations may not be immediately written to disk, but stored in the cache. Only under the condition that submission of written requests of the upper application were suspended and the system cache was emptied, data consistency of the production center and backup center can be ensured in application-level.

### 3. The fundamental architecture of HA systems

HA usually based on node redundancy [3], the practice is equipped with another single node for the source node to form the service redundancy, the source node to provide services is called master node, the increased node to provide service redundancy is called slave node, together constitute HA system of the node. The master node provides services in normal mode, the slave node will take over services once the master node failed. Shortcomings of this pattern are: need to equip with slave node for each service node to form HA systems of the node; thus, a large number of additional resources will be equipped in a large cluster system, while management and maintenance of these resources have greatly increased the complexity; resource utilization is not high for reason that normal mode only the master node in the provision of services. Therefore, we propose to design HA based on storage device redundancy, the fundamental architecture shown in Figure 1.

![Figure 1. HA architecture of client based on storage device.](image)

The architecture, each node has at least two storage devices, local disk and ND (network disk), together constitute HA system of the service. In normal mode, local disk called master device and ND called slave device, the master device running the service in the node, at the same time synchronizing service data to the slave device with strategy set beforehand, once the master device failed, computing resources and ND storage resource of the node would be bound, and boot the system backup from ND to constitute a new service environment, replacing the service environment of source node to provide services continuously to achieve HA service.

Compared to high availability based on node redundancy, HA based on redundant storage devices only need redundant storage devices in the node, in this system it is a virtual pool of network storage devices, only a large capacity storage device needs to be added in the whole system. Increased storage
devices can be centralized maintenance and management, which will not significantly increase the system complexity.

4. Achieve HA services of client based on data migration
When services are provided normally by client system, service data running on local disk is backed up to ND in file-level mode with designated strategy by data migration technology [4]. The advantages are that the normal operation of local services will not be affected by the backup process, and consistent feature of service data backup is ensured. Once client system is at a single point of failure, system service backup can be started from ND. However, backup with continuous time points is impossible in this way due to the discrete backup time; therefore, service data backup can not reflect the latest state before failure of the client and does not have real-time nature, only a state of some time before failure can be recovered.

5. Achieve HA services of client based on synchronous I/O request
In file-level backup mode, for a large file, even if only changed by one I/O operation, the overall total should also re-file backup, system overhead is relatively large. Such as email services, all mail inbox is usually stored in a large file, which will be changed while each received a new mail. In the implementation of file-level backup of the case, the large file will be re-filed backup while each received a new mail. In fact, it is not necessary that the old mails received will not be changed and only new incoming mails should be backed up. So, file-level backup is not applicable to the large files that have frequent I/O operations. Therefore, it is necessary to study HA services based on data block size [5]. If only a single data block of the file storage is changed by an I/O operation, back up the data block rather than all data in the file area is required, and system overhead will be greatly reduced accordingly.

According to I/O request completion time, remote replication mechanism based on I/O requests can be divided into synchronous mode and asynchronous mode.

5.1. Synchronous I/O HA mode
Synchronous I/O copying refers to replicate a fully synchronized backup from local data to a remote storage server, each I/O request accomplished locally will not return complete information to the upper until it is finished in remote ND too and confirmation achieved, then the occupied system resources can be released. HA system structure based on synchronous I/O shown in Figure 2.

![Figure 2. HA system architecture of synchronous I/O replication mode.](image)

Device layer in the system are constituted by a local physical disk storage device and ND (network disk), a logical device virtual disk is presented upward by client agent, the upper application or system service can only interact with the virtual disk. Specific process is designed to: an I/O write request is also sent to ND when it is sent to Local Disk device from the upper, complete information will not be
returned to the upper until both devices are finished with the I/O request. Thus, consistency of data in both devices can be ensured. In other words, for each write I/O request, only when both devices are successfully completed, complete information can be returned upward; if one device failed, another device in the treatment is regardless of the outcome, failed information will be returned upward in both devices. This ensures that all I/O requests are processed with atomic pairs.

Sync and Recover between the two underlying devices in figure mean:

At the time of the initial construction of HA system, the node running the service in the physical device “Local Disk”, the service data on the local disk need synchronizing to the ND after network equipment ND added, this process is known as the initial build process of HA system, during which node suspended services. Synchronization rate should be maximizing to complete system construction as soon as possible; a kernel-mode synchronization thread is responsible for handling of the entire build process.

The node resumed providing services outward after system construction had been completed, while the bottom two devices officially entered the HA phase. Whenever I/O requests come, they will be sent to the bottom both devices by client agent. Only both devices returned complete information, client agent will return confirmation to the upper that the I/O request has been completed successfully by lower device. Once a device failed, fail information of the I/O request procession will be returned to the upper. Service data consistency of both devices is ensured by the atomic of I/O requests processing.

Once SPOF occurs, computing resources and ND storage resources will be bound immediately and system backup will be started by ND to form a new environment to provide services continuously. The recovery thread will be activated to recover services data to local disk when the node is idle [6]. Services will be suspended during the recovery, besides, just as the initial construction requirements of HA system, completing recovery process as soon as possible is requested to the synchronization rate. Services will be restarted from local disk after recovery completed to restore normal operation mode of the system and enter the normal flow of HA.

The advantage of synchronous I/O HA mode is simplicity of the system and strong protection of data consistency, in other words, the backup data is always available, bootable, and non-destructive recovery can be achieved. However, delay that receive and send I/O request packet (IRQ) at ND-side will be inevitably increased when network communication is inefficient, thus at Disk-side it takes less time to process I/O request while a lot of time to wait for returned information from ND-side, resulting in obviously delay, or even applications and services can not be responded normally in the system. Therefore, the mode which sacrifices system performance the price to receive in exchange for high availability, is suitable for the environment where exists low network latency and peak writing rate is not more than the network bandwidth.

5.2. Asynchronous I/O HA mode

Asynchronous I/O copying refers to that IRQ is copied by the agent after every I/O request arrived, each I/O request is processed independently at both sides, Disk-side returned to the upper directly while ND-side returned to the agent at the end of completion. The basic structure of asynchronous mode and synchronous mode are the same, and each constituent function is also similar, the difference is an increased synchronization logging mechanism between Disk and ND. In normal operation mode, successful confirmation is returned to upper layer as long as I/O processing is completed at Disk-side, regardless of the results of ND. The IRQ will be finished by the agent if it is completed successfully at remote ND, otherwise, it will be retried by some mechanism until the remote completed successfully. Thus, remote I/O is synchronized backstage, the local system performance is affected little and requirements to network bandwidth are also small, but the problem of inconsistent data will be brought in this mode.

5.2.1. The reasons for data inconsistency. After I/O write requests arrived, the request will be completed firstly as electrical characteristics at Disk-side, the corresponding data will be changed and the confirmation information will be returned to the upper; while the request may not reach remote ND
or has failed at this point due to uncertainty characteristic of the network, thus the data on both devices are inconsistent at this point, assuming that Disk failed and that the data was user data at this time, when the accessed data was redirected to ND-side by certain mechanism, it will be found that the data read is inconsistent with the data last written if the data was reread by the upper application. That is, for some reason (such as network outages or other anomalies), the same data block in the local and the remote did not been updated simultaneously, resulting in system failure of high availability. Relative to the synchronous mode, system performance is enhanced while the reliability of HA is reduced in asynchronous HA mode.

At present stage, delay of the network side always exists compared to physical disk accessing speed, therefore, the problem of inconsistent data will exist surely in the basic mode of asynchronous HA, and it is difficult to ensure HA of system. In order to obtain high system performance while assuring data consistency, we designed an asynchronous HA mode based on log.

5.2.2. Asynchronous HA mode based on log. Asynchronous I/O HA mode based on log is that journal function of read and write operations is increased in the basic asynchronous mode to track implementation of read and write operations in the local and the remote. Results of I/O operation at both sides, which recorded in the log, can be obtained by checking the log periodically; the I/O operation will be re-executed and logged if failure information was returned from one side. Log mechanism enables local and remote data consistency.

There are two types of I/O in asynchronous I/O HA mode based on log: one for synchronous I/O, is responsible for data synchronization between two devices; one for normal I/O, generated by the system services.

Service full synchronization was implemented from Disk to ND at the time of HA system initial build. During the process, local system can continue to provide services outwards without suspending. All log records were marked as desync (desynchronous) state initially; the relative log record will be marked as sync state correspondingly as soon as one synchronization finished.

During the implementation of synchronizing I/O, when a normal I/O which will write Disk arrived, check the log firstly, and then send the I/O directly to the Disk to be carried out if the data block is desync, otherwise, carry out the I/O operation at both devices and write the log simultaneously. During the implementation of synchronizing I/O, when a normal I/O which will read Disk arrived, it can be sent directly to the Disk to be carried out because I/O synchronization was always completed firstly at Disk-side when HA system was created initially and the Disk was sync state.

The normal service and I/O synchronization are parallel in process of initial construction of asynchronous HA system, synchronization rate cannot be maximized as synchronous HA mode in order not to result in oversized impact to the normal service. Synchronization rate should be reduced to support resource requests of the normal service if the current system service operation takes too many resources; otherwise, synchronization rate should be increased appropriately to complete HA system initial construction as soon as possible if there are many idle resources in the current system.

In normal I/O mode, each I/O write request will perform a number of write operations: write Disk-side firstly and write the return information to log after complete information was returned successfully; write the return information to log after complete information was returned from remote ND-side successfully; check log periodically by the special kernel-mode thread and take appropriate action according to the log records.

When each I/O read request is carried out, locating log firstly and checking whether the data block to read has been synchronized at both sides, read operation will be carried out at Disk-side if both the state is sync, otherwise read operation will be carried out at sync side and synchronization process will be implemented at another side simultaneously.

Once SPOF occurs, computing resources and ND storage resources will be bound immediately and system backup will be started by ND to form a new environment to provide services continuously [7]. The recovery thread will be activated to recover services data to local disk simultaneously. Services will be provided outwards continuously during the recovery, during this period the processing method to
normal I/O and synchronization I/O is the same as what HA system initial construction, but the roles of both sides of device are exchanged. In this case, system performance will be lower than normal mode because services are running on remote storage resources. Just as the initial construction requirements of HA system, completing recovery process as soon as possible is requested to the synchronization rate. Services will be restarted from local disk after recovery completed to restore normal operation mode of the system and enter the normal flow of HA.

5.3. HA mode selection criteria
As mentioned above, synchronous I/O HA mode sacrifices system performance the price to receive in exchange for high availability, asynchronous I/O HA mode improves system performance while reducing the reliability of HA. Therefore, the selection criteria of HA mode is that attention from these two indicators. Specifically, whether to allow the loss of committed transactions and to allow the amount of the loss, available network bandwidth, network latency, synchronized data quantity, etc. are all factors to be considered. However, no matter what kind of mode, the average bandwidth of the network should be able to meet I/O rate of application program.

Considering the advantages and disadvantages of HA implementation scheme above, in order to meet requirements of system performance, an asynchronous HA prototype system based on log is designed and implemented. As a log file, bitmap mechanism records the data synchronization situation, the kernel-mode thread of I/O synchronization and recovery, bitmap operation, etc. are designed to achieve the functional modules of HA system.

6. Summary
Several implementation schemes of equipment redundancy to achieve HA are presented and analyzed in this paper: based on data migration, based on kernel-mode I/O request processing synchronized and asynchronized, etc., the conclusion that system I/O performance and HA are inversely proportional to each other is drew. Besides, the balance between system I/O performance and HA is found by the log mechanism, on this basis, prototype system based on kernel-mode I/O request processing asynchronized scheme is implemented, in which a higher system I/O performance is achieved and data consistency is ensured simultaneously.

References
[1] Erfan Zamanian, Xiangyao Yu, Michael Stonebraker. Rethinking database high availability with RDMA networks. Proceedings of the VLDB Endowment, Vol. 12, July 2019, pp. 1637-1650.
[2] Hertong Song, Chikchai “box” Learngsuksum, Raja Nassar. High Availability on Jetstream: Practices and Lessons Learned. Proceedings of the 9th Workshop on Scientific Cloud Computing, June 2018, pp. 1-7.
[3] Guo Hui, Zhou Jingli, Li Yue. Design of A Dual-computer Cluster System and Availability Evaluation. Proceedings of the 2017 IEEE International Conference on Networking, Sensing & Control, March 2017, pp. 187-195.
[4] Laila Bouhouch, Mostapha Zbakh, Claude Tadonki. Data Migration: Cloudsim Extension. Proceedings of the 2019 3rd International Conference on Big Data Research, November 2019, pp. 177-181.
[5] Jaemyung Kim, Kenneth Salem, Khuzaima Kaudjee. Database high availability using SHADOW systems. Proceedings of the Sixth ACM Symposium on Cloud Computing, August 2015, pp. 209–221.
[6] Joao A. Silva, Tiago M. Vale, Ricardo J. Dias. Supporting Multiple Data Replication Models in Distributed Transactional Memory. Proceedings of the 2015 International Conference on Distributed Computing and Networking, January 2015, pp. 1-10.
[7] Zhang Zhongmin, Song Ping, Xu Lu. Research and Implementation of Service Online Migration System, Computer Science, Vol.34 No.12, pp. 111-113, December 2007.