# TECHNICAL WHITE PAPER FOR BLUE WHALE SAN FILE SYSTEM (BWFS)

Version: 4.5

BlueWhale蓝鲸

Tianjin Zhongke Bluewhale Information Technologies Co., Ltd.

**2012/4/25**

BlueWhale蓝鲸

# 1. About BWFS

## 1. Zhongke Bluewhale's Profile

Zhongke Bluewhale is a domestic leading supplier of new-generation network storage products and solutions. It is committed to helping users handle mass data storage, data access and management amid the stunning rapid growth of the IT industry, improving operating efficiency and cutting total costs of ownership, thus creating long-term value and stimulating growth for users. The BWStor cluster storage series, Zhongke Bluewhale's flagship, features an innovative structure and solid quality and has been used in areas such as media data editing and playing, scientific calculation, simulation, exploration data calculation and analysis, remote sensing information processing. It is designed to fulfill users' demand for high file concurrent access capability, flexible system scalability and reliability and has been highly acclaimed among users in national defense industry, aeronautics and astronautics industry, governmental scientific research institutes, television stations, telecom carriers and universities. Moreover, Bluewhale's products have attracted wide attention globally. In 2005, BWStor was highly valued by Gartner, a global leading IT research and consulting institute, in its first research report on the application of China's storage products.

## 2. About File Systems

A file system is a data structure and programming method to store and organize data. Generally, local file systems include NTFS, EXT3 and FAT, where users firstly establish directories, then set access authority, and put files under the directories. Such file systems are established on one or multiple disks, which might be physical disks in a server or LUN in FC SAN or IP SAN. Generally, a local file system can only be mounted and accessed by one server.

Despite that the above-mentioned systems can meet the fast data access need, they cannot support storage resource sharing and management as well as concurrent data processing. The tradition way to realize these features is applying a network file system protocol, such as NFS (Linux/UNIX) or CIFS (Windows), in a server to share the local file system with other servers, users or applications. This method is able to integrate storage spaces but unable to meet needs for fast mass data access and massive data input/output handling.

In addition, the increasingly urgent demand for the compatibility with heterogeneous multi-system platforms and high scalability of storage space and performance, especially in media, governments, scientific research, large engineering projects and data centers, has called for the combination of Windows and Linux and boosted the expansion of storage space and IO performance.

## 3. About BWFS

Blue Whale File System (BWFS) is a cluster file system developed by Zhongke Bluewhale for FC SAN/IP SAN. Designed to transform several FC or iSCSI disk arrays into a storage cluster that supports multi-server concurrent processing, it can provide high-performance, extendible file sharing services and support applications under multi-machine workflow and cluster environments.

BWFS can be used in video monitoring, digital media, exploration data analysis, remote sensor information processing, streaming media, scientific calculation and simulation and other information processing fields.

## 4. BWFS series products

Blue Whale cluster file system gateway BWFS series

Blue Whale cluster storage system BWStor BW series

http://www.bwstor.com.cn

## 2. System Structure

As shown in the following Figure 1, BWFS mainly comprises the Metadata Controller, or MDC, and Application Server, or AS.

MDC handles AS's requests for access to metadata and maintain file system's global namespace and file block mapping information. The core functions include:

1. Directory services: MDC provides directory services including establishing the hierarchical structure of directories and authorization verification, maintenance of metadata including directory files, file inodes, file properties, creating or deleting directories and files, obtaining or setting file properties and operating of metadata including authorization verification.
2. Layout services: including file layout distribution, mapping and deletion services, maintaining storage resources including data blocks and indirect address blocks.

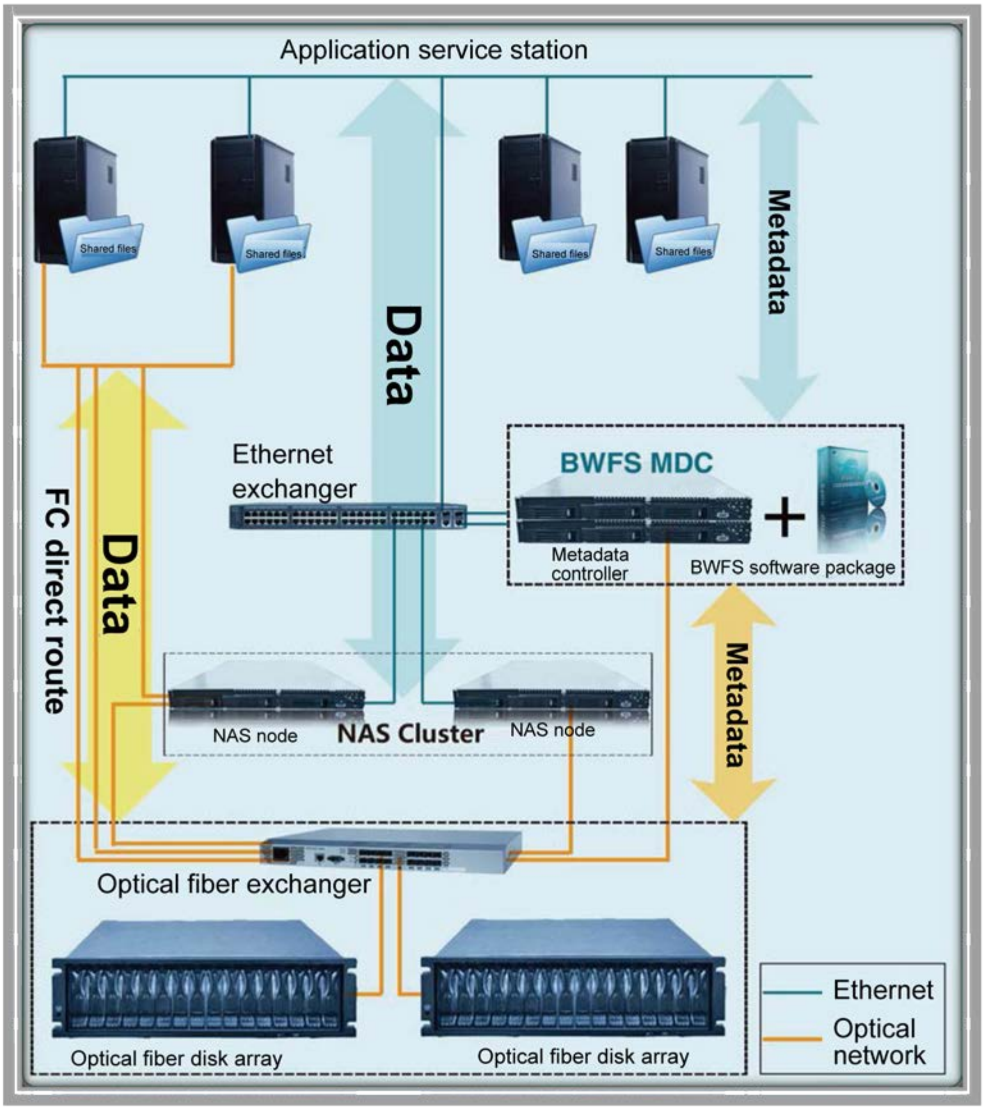


**Figure 1 -BWFS’s structure**

- BWFS is designed and employed a system that separates metadata with data storage. Metadata is stored in special storage devices separated from data storage, thus preventing conflicts between access to metadata and data. BWFS supports various different data storage devices and provide block device access interfaces.
- Users' application programs operate on AS and can access data in storage devices through file access interfaces compliant with posix semantics provided by BWFS. AS can access data storage devices concurrently using the out-of-band method, thus realizing high-performance data I/O access, accessing user data, submitting requests and accessing data without using the metadata server. During the entire access process, AS (client) first access the Metadata Controller to obtain the file's logic address, then to obtain the file's distribution and at last obtain the data at the physical address on the storage device.
- Currently, BWFS's highly efficient and reasonable system structure can fully meet the requirement for large space and high performance of large-scale storage systems. Moreover, it can provide additional features for the system according to application needs.

# 3. Core Technologies

## 1. Out-of-Band

BWFS adopts a structure that separates metadata and data services. BWFS provides the special metadata service module that manages metadata independently; clients can obtain metadata by visiting the metadata service module and access data through the Out-of-Band methodwithout using the Metadata service module..

Through the Out-of-Band method, users can access SAN storage devices directly, thus obtaining higher aggregate I/O performance and lower letency. The metadata service module can help clients access data concurrently and realize data sharing.

The separated system also brings the following benefits:

Given that data can be accessed without using the metadata controller, it eases the controller's burden and improves the system's metadata access scalability; as access to metadata is different from access to data, the separated system allows specific optimization;

The separation of data and metadatacan prevent interference caused by metadata random access and data sequential access on storage devices, thus improving the efficiency of storage devices.

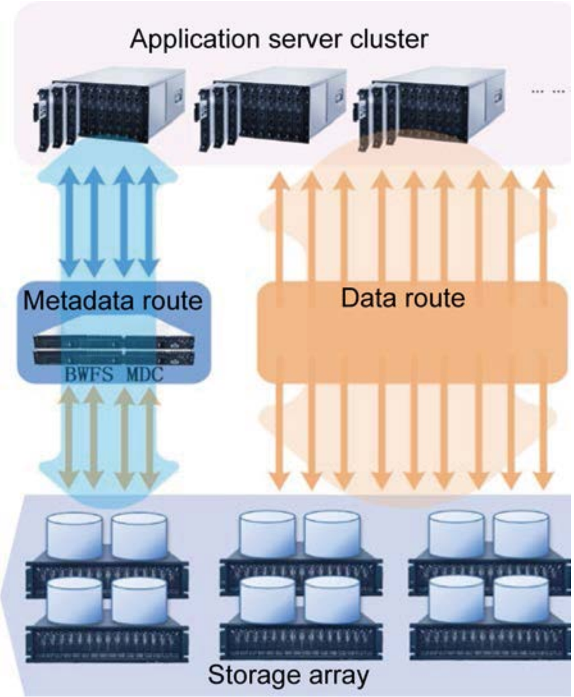


**Figure 2 -BWFS's application structure**

## 2. High-efficiency resource management

To construct a large, high-performance and high-scalability network storage system, BWFS is designed with a flexible and efficient storage resource management system. The metadata service module provides layout services including file layout distribution, mapping and deletion and maintains storage resources including data blocks and indirect address blocks. When writing files, the module can distribute data blocks for files; obtain the physical address of data when reading files; and release file layout and recover data blocks occupied when deleting files.

**BWFS adopts the resource distribution strategy that is based on Volume Groups. The hierarchy is as follows:**

a) Volume is the basic unit in terms of the storage scheduling strategy. A volume can only be attributed to one VG, which decides how resources are distributed;

b) Volume group, or VG, is higher than volume. VG may include multiple volumes and VG strategy distributes resources in its volumes;

c) The binding relationship between VGs and files or directories is on the top level of the system. After the binding relationship is determined, VG policy will distribute resources to files. Resources are distributed to files on the volume level. The binding relationship and properties of all the VGs decide how resources are distributed to files.

**Different VGs have different properties and strategies. For example, the system provides the following three strategies for distributing resources within VGs:**

a) File Level Fault Isolation: Different files whose data blocks are distributed on a VG are distributed on different volumes as mush as possible, while the same file is distributed on one volume as mush as possible.

b) Stripe: It is to distribute files to volumes in a VG through the means similar with RAID0, thus improving the concurrent IO access speed and IO bandwidth.

c) Fill: It is to fill VGs with files by means similar with the MAID mode. Only one Volume in one VG will provide resource distribution at a time. This strategy is applicable to write-once-read-many applications.

**In BWFS, metadata and data are stored in metadata volumes and data volumes respectively to prevent conflicts between access to metadata and data. The EXFS file system support the extent-based layout structure. Unlike the original three-level indirect address structure, each extent is a continuous segment of variable length and all the extents are organized using a B+ tree;**

a) Extent can be pre-allocated and distinguish data and holes distributed to it. It solves the problem that the three-level indirect address structure cannot distinguish holes from sparse file.

b) It supports larger files. The maximum size of files in the three-level indirect address structure is dependent on the length of the segment represented by the indirect address item, while that in the extent structure is determined by the maximum height of the B+ tree.

c) The distributor is more flexible in distributing extents of different sizes to different files, preventing wasting of resources or poor resource allocation caused by the original fixed-length allocation algorithm.

## 3. Client Layout Cache Technology

When receiving read/write requests, the client of the file system (file system driver) will firstly transform the deviation and length of file operations into read-write deviation and length on the corresponding disk locations. In BWFS, clients do not maintain these data mapping relations. The metadata controller is responsible for data mapping (Out-of-Band). Therefore, clients only need to operate GETBLOCK through RPC to obtain these corresponding relations. During the read-write of massive data, repeated RPC telecommunication is indispensable for obtaining the corresponding relation between logic and physical deviation and length. Network delays caused by the telecommunication will affect the performance of clients. Moreover, in the HA working environment, clients are unable to obtain corresponding relations from a metadata controller during HA switch, jamming read-write processes of applications (such as unsteady pictures in playing video when HA switches).

To solve this problem, BWFS provides metadata information cache on clients. While ensure consistency, it can minimize the information exchange with the metadata controller, thus reducing the network delay caused by RPC telecommunication and boosting the data handling capability.

BWFS's clients apply the layout cache with the rb_tree structure to replace the bcache mechanism with a data array structure. The layout cache system is designed with the following functions:

Obtaining files' extent modules from the metadata server (metadata's bcache) as the rb_tree node to search for, insert and delete extents.

Layout pre-fetching: Predicting the address of the next read action according to the read information and add it into the pre-fetching Linkedlist; creating a new asynchronous thread, accessing the segment to be read and obtaining extents through RPC telecommunication.
Using the LRU Linkedlist to control the use of layout cache memory: Given that the memory is limited, controlling the use of the cache can prevent overuse of layout cache's memory of the rb_tree. To do this, we will maintain a LRU Linkedlist to preset the total memory available for layout cache; the threshold value will be set based on certain algorithm. When the memory used by layout cache exceeds the threshold value, layout that has been leastly used recently will be replaced from the LRU Linkedlist based on the replacement algorithm.

## 4. Small File Optimization

According to research on file system burden, access to small files accounts for a large share of the aggregate in fields including web applications, scientific research, engineering development and personal applications, reaching 88%, 60%, 63% and 24% in education, scientific research, web applications and personal applications respectively. Therefore, the file system must boost its handling speed and capability for small file I/O requests.

In BWFS's existing structure, clients must send requests to the metadata controller to obtain metadata. The metadata controller will abstract access to metadata into interfaces similar with systems calls for clients' use.
Clients can use these interfaces to store and access metadata. Such interfaces fall in the RPC Procedures defined by the BWFS protocol. The metadata storage and access interface is stored inside the BWFS's metadata controller. The advantage of storing and accessing metadata by the Out-of-Band means is that all the access to metadata will be processed through the metadata controller, thus ensuring the consistency. The metadata cache in memory can help reduce the exchange between clients and the metadata server, improving the file system's performance.
When handling numerous small files, the file system has to deal with repeated telecommunication between clients and the metadata controller. Operations such as content searching, authorization checking, directory traversal, file creation, deletion and renaming will increase the metadata controller's burden and affect the concentration capability.
To better handle numerous small files, the BWFS is optimized as follows:

a) Using Compound RPC mechanism to merge RPC requests to ease the metadata controller's burden.
b) Using Delegation mechanism to reduce RPC, easing the metadata controller's burden; if the directory inode is cached by clients, before the delegation lease expires or is recalled, the system does not need to apply for the inode's latest attributes from the metadata controller or access the metadata volume for the latest attributes, thus reducing the metadata controller's pressure.
c) Using DAMV mechanism to enable clients to directly access metadata volumes, reducing the metadata controller's IO pressure.

The optimization has substantially improved BWFS system's aggregate performance of file opening, i.e., the aggregate performance of lookup when handling numerous small files; enhanced the aggregate performance of large directory READDIR; eased the metadata controller's pressure, improving the system's metadata loading capability and aggregate access handling performance. Moreover, directory modification operations (such as MKDIR, CREATE and REMOVE) will remain unaffected.

## 5. Storage Access Mode Management on Demand

The rapid and continuous development of storage technology has prompted various heterogeneous storage resources, such as FC, iSCSI, SATA and TAPE devices. These storage resources vary from each other in aspects including capacity, performance, availability, reliability and resource management.

Meanwhile, in a file system, different data has different characteristics. For example, metadata is largely different from data in operation characteristics and importance. Moreover, with the expansion of storage systems, more and more categories of applications with different features and requirements for storage resources have been created. For example, database services are generally service operations of the small-file read-write mode with a high requirement for the random read-write capability; streaming media services have a high requirement for the bandwidth, feature a large-block read-write mode and generally adopt large blocks.

BWFS manages heterogeneous storage resources using the multi-volume method and constructs a multi-volume strategy structure to provide suitable storage resources for specific applications and different data based on strategies, enhancing the system's overall performance.

a) Supporting separated storage of data and metadata

b) Supporting multiple data distribution strategies (File Level Fault Isolation, Stripe and Fill)

c) Supporting file-level data distribution storage

d) Supporting dynamic volume expansion

## 6. High Availability

With the wide use of the cluster system, systems have continued swelling with increasingly complex structures. As nodes increase in systems, node failures have risen sharply. Information technologies have been increasingly applied in various industries amid the rapid economic growth. However, application service interruption due to system failures has also caused painful losses. According to relevant statistics, service interruption will bring economic losses of up to over 1 million U.S. dollars each hour for service providers. Moreover, long-term service interruption will lower customer satisfactory, erode service providers' reputation and lead to losses of customers.

Major causes for system failures include hardware failures (such as CPU, memory and disk failures), software failures (operating system and application software failures) and environmental failures (improper operations and external environmental instability, including natural disasters).

Redundancy techniques, such as eliminating Single Point of Failure (SPOF), are mostly used to cope with hardware failures and improve the system's availability. Such techniques include power supply system redundancy, server redundancy, storage system redundancy and network system redundancy. Such redundancy systems are designed to provide backup for all classes of resources so that operations can continue when failures are detected.

BWFS's MDC adopts the Active/Passive mode to ensure its high availability. One node that is providing file system services is an Active node, while the standby node is a Passive node that will take over the file system services when the Active node fails.

BWFS's MDC HA system is designed with three major functions:

a) Redundancy function, eliminating the SPOF in the system;

b) Failure detection, i.e., detecting functions that have failed through systems including information transmission, supporting multiple failure diagnosis systems (such as disk examination, network examination and client poll);

c) Failure switch function, i.e., using redundancy resources to switch services after system failures are detected.

BWFS uses the Heartbeat mode to detect failures directly. Two nodes send messages to each other. Messages are sent through the network connecting mode to monitor operating of the server and services. BWFS provides the failure detecting and recovery functions between two metadata controllers (server network Heartbeat), metadata controller and disk array (disk Heartbeat) and metadata controller and client (client Heartbeat).

BWFS adopts the core management module FSMD to realize the HA function of the Active/Passive mode, such as the synchronization of HA cluster information; resource management, resource startup, shutoff and supervision; monitoring requests of other module reports; making decisions according to failures or requests; conducting searches and configuring HA's management interface; and monitoring and handling basic events of the file system. FSMD also supports non-HA modes, HA mode and modes working together with third-party HA (used together a third party's HA: when FSMD is started, the network Heartbeat will established between it and the third party's node. FSMD will provide the resource group management interface and the other HA software is responsible for starting and stopping the resource group. After receiving a failure report, FSMD will only record the event but will not handle the failure).

## 7. File-Level Mirror

The traditional reliable network storage solution is FC SAN + SAN FS, eliminating SPOF in a file storage system through a suitable configuration, including disk RAID, dual-controller, redundancy exchanger, double FC HBA cards, multi-path software and SAN FS MDC HA. However, given that data is stored in optical fiber disk arrays and that optical fiber disk

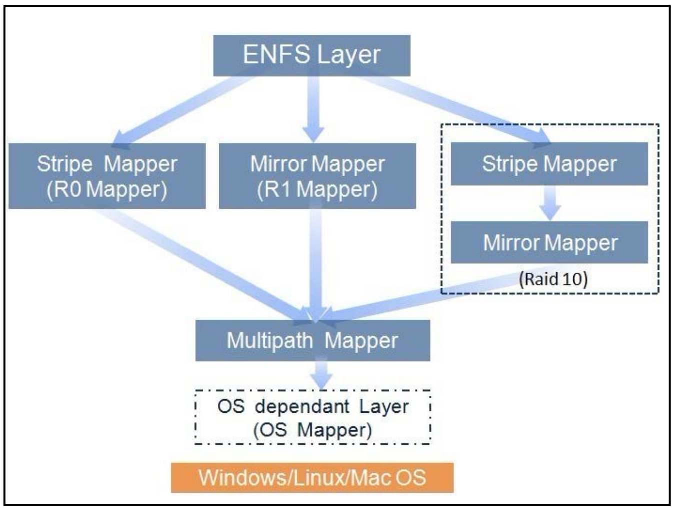
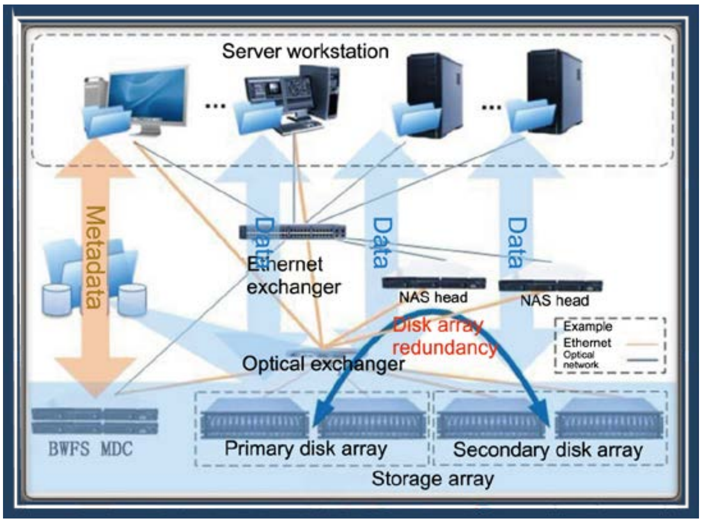


Figure 3- BVDM system



arrays feature the monolithic structure, when one optical fiber disk array fails, all the relevant front-end applications will be interrupted and the whole system will collapse. Although that double-controller optical fiber disk arrays are used in most cases and it is unlikely that the two controllers fails at the same time, the risk still exists. Therefore, BWFS is designed with the Blue Whale file-level mirror mechanism BWMirror to provide the disk array real-time disaster recovery function.

BWFS adopts the BWFS Virtual Device Mapping system (BVDM). As shown in the following Figure 3, it provides BWFS's clients with a high-performance and flexible basic IO structure that is irrelevant with the system's structure, and supports multiple paths and images based on the IO structure. Logically, BVDM is located between the file system and specific block device access interface. It is composed of multiple stackable Mappers to convert data IO operations into logic addresses in BWFS volumes, thus realizing the required block-level IO virtualization functions (including multi-path, stripe and RAID). By using the BVDM mechanism, BWFS can identify, configure and update multiple paths; construct and reconstruct Raid devices including RAID1 and RAID5; and bind stripe characteristics of RAID0 and RAID5 with the file system's distribution method.

BWFS uses the RAID1 module in the BVDM system to realize the file-level mirror image of data (BWMirror technology); By using relevant information through the file system level, creating the RAID1 mirror image on files and storing image files in different storage devices (disk arrays), BWFS is able to realize the real-time disaster recovery of the storage system. The storage system comprises multiple disk arrays and is managed by BWFS. Users can set part of the disk arrays as the primary array that provides major data services, set the others as the secondary array that will take over the primary array's work when it fails. When BWFS's disk array redundancy function is enabled, all the data in the client write-in system will be written into the primary and secondary arrays as mirror images. When either array fails, application programs will immediately switch to the other array. Therefore, the file system will continue operating when one disk array fails and operating services will remain unaffected. This realizes the full redundancy configuration of the storage system and completely eliminates the single point of failure, thus ensuring the reliability and availability of data in the storage system.

**BWMirror technology has the following features:**

a) Completely transparent to applications;
b) Data on disk arrays is synchronized continuously during write-in operations without file copy window and loss of data.
c) Mirror image files can be access during read operations and all the disk arrays are used.
d) When one disk array fails, operations will be transferred to the others, ensuring the continuity of services.
e) It supports heterogeneous disk array redundancy, thus saving costs for users.

## 8. Journal

After the file system collapses due to a power failure, the data and metadata in the system may become inconsistent (the system fails to update metadata accordingly with data), which will lead to a serious consequence. Generally, the file system will use the FSCK (file system check) program to scan the entire file system to keep the consistency between data and metadata and recover damaged data. With the file system expanding, FSCK becomes increasingly time-consuming, lowering the file system's availability. BWFS solves the problems through the log method.

BWFS will write revision record of files in logs synchronously. When the file system collapses, files can be recovered from the logs to the status before the collapse of the system, thus quickly recovering the file system.

BWFS provides three log modes for users to balance before the system's performance and safety.

# 4. System Features

## 1. Mass Storage

It supports files of up to 2PB, a maximum storage space of 64ZB and more than 1 billion directories and files:

a) Storage volume 64ZB

b) File quantity (directory and files) > 1,000,000,000

c) Quantity of files in each directory (directory and files) > 62,500,000

d) Maximum file size: 2PB

e) Quantity of LUN supported: 4093

## 2. Excellent Performance

a) Thanks to the Out-of-Band data transmission structure, the system is able to give full play to the bandwidth and superb performance of the SAN environment, breaking the bottleneck of the storage system's I/O bandwidth for concurrent access. Moreover, the direct data access has substantially improved the storage system's IOPS, meeting the high requirement for short delay.

b) BWFS's structure separates metadata volumes, log volumes and data volumes and is able to adopt different IO strategies, separated configurations and optimization.

c) The Windows platform provides the original driver for BWFS, which outperforms cross-platform file sharing systems that rely on third-party network sharing protocols.

d) The system's overall IO performance depends on the aggregate maximum performance of FC or iSCSI disk arrays. Therefore, it can be increased linearly with the increase of the FC or iSCSI disk arrays.

## 3. Supporting Application-Level Data Sharing Through Multiple Heterogeneous Platforms

a) BWFS allows users on Linux, Windows and Mac systems to share and access its data;

b) Mass data is stored uniformly and the global namespace is applied. Uniform interfaces are provided for applications to support file sharing and synergetic operations across various platforms.

c) A file-level sharing interface is provided for users as directories in Linux and logical drives in Windows;

d) Fully support POSIX or NFS semantics. Applications on NFS are fully compatible with BWFS. It allows cross-platform file access and operations.

## 4. Excellent Scalability (Horizontal,Vertical and Dynamic Expansion)

The storage system supports smooth expansion and seamless upgrade of the application platform. Users can purchase the suitable storage space according to their needs. The expansion includes work station node expansion and storage capacity expansion:

a) Work station node expansion: It is to increase the work station nodes. The expansion can be completed by connecting the networks of the work station nodes to the network of the BWStor storage system and completing basic configuration without closing down the system or discontinuing services. After the expansion, the application system's overall service capability will be improved to support the high-efficiency operation of front-end applications.

b) Storage capacity expansion: It includes horizontal and vertical expansion. Vertical expansion is to increase expansion cabinets, while horizontal expansion is to add secondary cabinets. The two expansion methods can expand the storage capacity without discontinuing current services or affecting application. As the storage capacity increases, the storage system's overall IO bandwidth and loading capability are enhanced linearly.

## 5. Enterprise-Level Data Reliability

a) Files and data are directly stored in disk arrays of BWFS. The dual-controller disk arrays, reputed as the most reliable system in the industry, ensure the absolute safety of data stored in BWFS.

b) With the file system log function, BWFS can automatically or manually check the file system's consistency on a regular basis, thus discovering and correcting file errors caused by system or power failures.

## 6. High Availability

a) Thanks to the two MDCs, BWFS has the fail-over function. When one MDC fails, the other one will take over the work and ensure the continuous operating of the whole file system;

b) BWFS can be constructed in the full-redundancy SAN environment, including the dual-controller disk array, double exchangers and dual-interface HBA card, thus realizing the full redundancy of the entire system. All the SPOFs are eliminated and any part of the system can be replaced without halting services, thus enabling the storage system to continue operating 7/24.

c) The online and storage expansion will not affect the operating of services.

## 7. User-friendly Interface and Management

a) BWFS is designed with a user-friendly controlling interface with multiple languages; administrators only need a few hours of training to grasp the operating procedures of the system; the grouped client management can largely facilitate the management of large-sized systems.

b) The system provides Web-based remote management tools, which will largely reduce the administrator's work, with functions including system diagnosis, event management, configuration file management and system failure e-mail warning;

## 8. Third-party Storage Supporting

The system supports a wide range of SAN storage devices and can perfectly integrate users' existing iSCSI device into a uniform namespace, helping users saving costs.