PROCEEDINGS OF SCIENCE

Experience with Fabric Storage Area Network and HSM Software at the Tier1 INFN CNAF

First Author[1]**:** Pier Paolo Ricci *on behalf of the INFN CNAF Tier1 Staff*
INFN CNAF
Viale Berti Pichat 6/2, Bologna ITALY
E-mail: pierpaolo.ricci@cnaf.infn.it

Other Authors: Angelo Carbone, Antimo Dapice, Luca dell'Agnello, Giuseppe Lo Re, Barbara Martelli, Vladimir Sapunenko, Dejan Vitlacil
INFN CNAF
Viale Berti Pichat 6/2, Bologna ITALY
E-mail: angelo.carbone@cnaf.infn.it; antimo.dapice@cnaf.infn.it; luca.dellagnello@cnaf.infn.it; giuseppe.lore@cnaf.infn.it; barbara.martelli@cnaf.infn.it; vladimir.sapunenko@cnaf.infn.it; dejan.vitlacil@cnaf.infn.it

Abstract: This paper is a report from the INFN Tier1 (CNAF) about the storage solutions we have implemented over the last few years of activity. In particular we describe the current CASTOR v.2 installation at our site, the HSM (Hierarchical Storage Manager) software chosen as (low cost) tape storage archiving solution. Beside CASTOR, we also have in production a large GPFS cluster relying on a Storage Area Network (SAN) infrastructure to obtain a fast and disk-only solution for the users. In this paper, summarizing our experience with these two storage system solutions, we focus on the management and monitoring tools implemented and on the technical solutions needed to improve reliability on the whole system.

XI International Workshop on Advanced Computing and Analysis Techniques in Physics Research
Amsterdam, the Netherlands
23-27 April, 2007

[1] Speaker

PoS(ACAT)019

1. TIER1 CNAF Storage resources description

1.1 Storage resources.

The TIER1 INFN CNAF disk storage resources have grown to a PetaByte in the last few years of activity in accordance with the LHC and the other HEP resources requests. The back-end infrastructure has been used for years of activity and now it represents a stable and well tested model. The SAN implementation of this model gives a very flexible and modular method for managing storage, with the best possible quality/price ratio, due to a yearly tender method of buying new hardware. Currently we have roughly 1PByte of disk storage on-line over a single SAN fabric, which is formed by storage boxes from different vendors. The disk storage is therefore composed by:

9 Infortrends A16F-R1211-M2	**56TB**
1 SUN STK Bladestore	**32TB**
4 IBM FastT900 (DS 4500)	**200TB**
5 SUN STK Flexline FLX680	**290TB**
3 DELL EMC CX3-80	**400TB**

The capacity reported should be considered as Raw space, that is the theoretical number calculated by multiplication of the single Hard Disk capacity and the number of these disks in the storage boxes. Experience shows that net space (the real capacity, available for the end user) is about a 15-25% less, due to RAID and the filesystems overhead, which strongly depends on the vendor specific characteristics of the storage boxes. In general we find that RAID5 of 2-4TB (depending on the size of the disks) give the best compromise between performance, reliability and net space. As filesystem we are mainly using XFS and GPFS as well as ext3 partitions. Most of the disks are based on ATA technology to reduce the cost. This choice has came as the result of the tenders, since the better performance or higher Mean Time Between Failures (MTBF) of Fibre Channel (FC) disks that could by granted using this technology, cannot compensate the much higher cost. So we currently have FC disks in production only for very specific applications (such as Oracle Databases) for which the performance and MTBF of ATA disk are not high enough. The SAN infrastructure is based on Brocade switches supplied by the tenders for the disk storage. Two Fabric Director Switches (one 24000 with 128 2GB/s ports and one 48000 with 4GB/s ports extendable to 256 ports) represent the core of the SAN, while two SilkWorm 3900 (total of 64 ports) are connected as peripheral switches. It's very likely that this sort of "star" connection (central core switches with lower cost satellite switches) will be maintained in the future expansions since it has proved its stability. The access to the storage is provided by dedicated server machines (diskservers) with Scientific Linux as Operating System. Currently we have in production a total of ~90 diskserver with redundant Qlogic HBA connections to the SAN and Gigabit connection to the LAN. The standard diskserver is a single Unit rack mountable biprocessor with 4Gbyte of RAM and redundant power supply other that mirrored (RAID1) hot swappable system disks. The Qlogic SANsurfer[1] tool is used as the standard Fibre Channel Path-Failover configuration tool to assure connections redundancy. Also in some cases (i.e. EMC hardware) the vendor specific Path-failover has been installed in order to assure an higher reliability or specific features (in case of EMC the PowerPath tool add a real

load balancing algorithm that proved to be somewhat useful). The general idea is to provide a near No-single-point-of-failure (NSPF) system, in order to guarantee access to the storage also in case of programmed shutdown (i.e. firmware upgrades) or in case of failures of single elements of the system (Qlogic HBA connection, Fibre Channel switch or part of one Director Switch, controller of the disk storage box). In Fig. 1 we report a schema of a standard diskservers storage connection and implementation. In this schema the single CX3-80 array with eight 4Gb/s outputs over the 2 redundant raid controllers (Storage Processor A and B) is accessed by 12 dedicated diskservers. The whole 110TB net space is sliced in 2TB partitions (LUN) which are seen as different scsi devices by the operating system. The SAN zoning and the storage box LUN masking is made in such a way that only the 12 diskserver could access to these LUNs, which is the simplest and most effective way of accessing the data. Since all the 12 diskserver could access to all the LUNs is possible to implement a parallel filesystem (GPFS as described in the dedicated paragraph) to improve performance and especially reliability. In fact is possible to configure the GPFS diskservers in the Network Shared Disk (NSD) [2] configuration and in this way failure on one of the diskserver machines will be completely trasparent to the client machines. So the SAN gives two main advantage in term of flexibility:

1) we can dynamically vary the diskserver/disk-storage assignment (adding diskservers, or changing "how much storage" is accessed by a specific diskserver)

2) we can use as diskpool clustered filesystem like GPFS

1.2 Tape resources

The tape library Sun Storagetek L5500 has been in production for the last 3 years so far. It's partitioned with two different types of media/drive, the LTO-2 drives with dedicated 2000 slots and the 9940B drives with dedicated 3500 slots. Both of the two media have a capacity of 200GByte uncompressed with a throughput of 25-30MB/s. The total tape space is roughly 1PByte uncompressed, which is usually the real space since the data archived in our tape facility from the LHC experiments are usually already compressed in archives or they are essentially binaries with a low possibility to be substantially reduced. The total number of LTO-2 drives is 6 and we also have 7 9940B drives which could guarantee a total bandwidth of 300-400 MB/s for the disk-to-tape streams and vice versa. The single tape drive is accessed using a dedicated server (tapeserver) for obtaining the best performance and reliability level. The standard tapeserver is a single unit (1U) rack mountable machine with Intel Xeon dual processor and 2Gbyte of RAM with no hardware redundancy. The reason for a low hardware redundancy is that a single failure to one server hardware is a rare event compared to failure rate of the drives themselves and, in any case, the unavailability of one server will reduce the total bandwidth but it cannot block the overall data flow to the tape back-end. The tapeservers are connected to the local LAN using Gigabit Ethernet connections, and to the drives using point-to-point 2Gb/s single Fibre Channel link. The tape software HSM system implemented in our site is CASTOR v.2 [3] developed by CERN (Conseil Européen pour la Recherche Nucléaire) which is better described in the dedicate paragraph. The dedicated servers for the central services of CASTOR

have better redundancy respect to the standard tapeserver, they are equipped with 2 hot swap power supplies, RAID 1 (mirroring) hot swappable system disks in addition to more memory and higher CPU speed.

1.3 SAN System and Monitoring

The SAN in production in our centre is here represented in Fig. 2. As described before the star topology will be probably maintained in the future since two central fabric "core" switches and satellite lower-cost standard switches will be enough for our implementation of GPFS and the no-single-point-of-failure approach. Nevertheless, lots of switches mean an increase on the level of complication in administration and monitoring of the whole system. The first choice to reduce this level of complication was the decision to buy Fibre Channel switch of one specific vendor to guarantee the best compatibility. Since from first SAN tender we obtained FC switches based on Brocade technology we decide to maintain this approach and stay with Brocade hardware, which makes it possible to create a single logical SAN and to use a single GUI as point of access to the management console. The Brocade switches have also a good price/port ratio in addition to a wide choice of products. The GUI management console of the whole SAN system is the standard Brocade WebTool. In our implementation all the switches are configured in a single logical SAN so all the information of zoning (subset of diskserver and storage disks devices) and other configuration are shared across all the switches. The WebTool permits to modify these parameters from any switch in the SAN and it also gives important real time performance analysis tool and fault detection system.

The other tool that is used to manage and monitor the SAN is the Brocade Fabric Manager Software. This software must be licensed and runs on a dedicated Windows machine to provide a 24h supervision of the SAN with a historical data archiving system. A screenshot of the Software is reported in Fig. 3. In the screenshot a graphical outline of the switch interconnections is displayed (on the left) which is automatically created by the Fabric Manager Software and on the right an example of the Performance Monitor Analysis on one-day scale is reported. The software is used to provide historical traffic analysis on day to day basis and to recognize possible bottlenecks or channels what need optimization. Also an automatic e-mail notification system has been configured on the Fabric Manager to alert SAN administrators in case of hardware failures of the system elements or in case of unexpected events (like down of Fibre Channel links). It's also possible to configure planned events (i.e. reboot of the switches or firmware upgrades) in scheduled sequences in order to optimize downtime and to provide a single central point-of-view from which the whole maintenance operation is controlled.

2. CASTOR and Monitoring System

2.1 CASTOR software

CASTOR stands for the CERN Advanced STORage manager and is a hierarchical storage management (HSM) system developed at CERN and used to store physics production files and user files.

In CASTOR, data is copied from user resource to CASTOR front-end (disk servers) and then subsequently copied to back-end (tape server). Buffering data permits to the system to optimize the disk to tape copy (migration process). Currently at CNAF we have 45 disk servers as front-end and each of them has about five or six filesystems. Typical size is between 1.5 to 2 TB each filesystem. Both XFS and EXT3 are used. Disk servers are connected to the SAN using full redundancy on FC 2Gb/s (on latest machines 4Gb/s) connections (dual controller HW and Qlogic SANsurfer Path Failover SW or in other cases vendor specific software). For CASTOR back-end we use 13 tape servers (each one with dedicated tape drive) which are connected via 2Gbit/s FC interface to 6 LTO2 and 7 9940B tape drives which are sited inside STK L5500 silos. The silos in question is partitioned with 2 form-factor slots. For LTO2 tapes there are 2000 slots and the other 3500 slots are for 9940B tapes (5500 slots in total). Both the technologies use 200GB cartridges, so the total capacity is roughly 1.1 PB non compressed.

In a same way data is obtained from tape and copied to disk (recall process). Files can be stored, listed, retrieved and accessed in CASTOR using command line tools or applications built on top of the different data transfer protocols like RFIO (Remote File IO), ROOT libraries, GridFTP and XROOTD the latest not used at our Tier1. CASTOR provides a UNIX like directory hierarchy of file names. The directories are always rooted /castor/cnaf.infn.it . The CASTOR namespace can be viewed and manipulated only through CASTOR client commands and library calls and it is provided by Name Server.

At CNAF, currently, we are running 2.1.1-9 CASTOR version. The Core services run on machines with SCSI disks, hardware raid 1 and redundant power supplies. We have distributed CASTOR Core Services, basically on three different machines (Scheduler, Stager and Name Server) and on a fourth machine we run Distributed Logging Facility (DLF) for centralizing log messages and accounting information. CASTOR is driven by a database-centric architecture. A great number of components are stateless and the code is interfaced with Oracle relational database. So, since the Stager Logic uses a dedicated Stager Catalog and Name Server uses Name Server Catalog, we have 2 more machines for the Oracle databases. Name server Oracle database runs on Oracle 9.2 while Stager Catalog runs on Oracle 10.2 platform.

At CNAF, CASTOR is used by four LHC experiments (ALICE, CMS, ATLAS, LHCb) and by eight others (LVD, ARGO, VIRGO, AMS, PAMELA, MAGIC, BABAR, CDF), and the ones which have chosen to use CASTOR with tape back-end, have diskpool and tapepool. In this case, diskpool has cache function and all obsolete files are candidates for Garbage Collector (a CASTOR component) which protects disks free space. Even if CASTOR is principally used for management of tape back-end some experiments use CASTOR as pure disk, without tape migration. CASTOR uses LSF scheduler [4] (version 6.1 at CNAF) to determine the best candidate resource (filesystem) for a castor job. The decision depends on the load of disk servers. LSF is distributed via NSF and it is exported by the LSF Master node. LSF slots are dedicated to each diskserver (from 30 to 450) and can be modified in case of need. SRM (v.1) [5] interface to CASTOR, for grid users, is provided by three end-points. One is dedicated only for tape service class while other two are dedicated for disk-only service class. Two of these SRM endpoints are DNS balanced.

2.2 Monitoring system

The monitoring and notification system based on Nagios [6] has been used for the last few years at the INFN–CNAF with good results. Nagios is a well known, widely used and reliable tool designed to run under Linux and is able to perform monitoring of network services and host resources. Plugins has been developed for very specific usage in addition to the set of Nagios native plugins in order to monitor the CASTOR software central daemons and the tape/disk usage. Also the relevant variables are graphically displayed using RRD [7] and an e-mail notification system is used for managing the alarms. Otherwise the Nagios system is somewhat limited (it's not based on a database software backend) and need to be customized for every new use. So we decided to install a new software as the main monitoring and alarm system in parallel to the old Nagios software. The monitoring solution we choose is the LEMON (LHC Era MONitoring) [8] software with Oracle backend, a tool developed at CERN within the project named ELFms (Extremely Large Fabric). A client/server monitoring system thought as a monitoring solution for a distributed systems, LEMON has modular architecture, it is the CERN suggested monitoring tool, and it provides a strong integration with CASTOR.

The first act is measuring and obtaining needed information by Monitoring Sensor (MS) on client's hosts. LEMON has numerous metrics, information coming from a resource, which can be an device, software, daemon or whatever we want, due to fact that user can define an new metric as well as sampling period. All the taken samples of data from any kind of information that should be monitored (e.g hardware, software, database, etc.) are stored in Local Monitoring Repository. The main reason for this procedure is to prevent loss of data during the transmission to Monitoring Repository (MR). The Monitoring Sensor Agent (MSA) is a daemon which transfers data from monitored node to Monitoring Repository using either UDP (User Datagram Protocol) or TCP (Transmission Control Protocol). After the measuring process is over, data collection is done by Monitoring Repository, the server application. This application receives data from all monitored nodes and stores them either to a flat file or to an Oracle based back-end, because LEMON offers two different types of installation. Once the information are stored in chosen back-end user can view them using LEMON CLI (LEMON Command Line Interface) or, the more user friendly, LEMON Round Robin Database Framework (LRF). In this, last, case a python application takes data from Monitoring Repository and creates RRD files from which are created graphs which are finally visualized via HTTP using PHP and Apache as presentation level. LEMON is a complete monitoring tool and it's been designed to cover all performance information. Even if LEMON is provided by alarming and event handling tool, our decision was to go on with Nagios as a more mature notification product and very useful for some other types of monitoring already mentioned.

Lemon sensor is running on all storage machines in a way that we have basic performances information of each and every machine. As the principal LRF's point of force is aggregation of some desired informations into graphs, once Virtual Cluster is defined by users, we've decided to create several VCs on different bases. All CASTOR diskservers are grouped into Virtual Clusters, by experiment, and all together are inside one greater CASTOR Disk VC. Other VC's are grouped by function (tapeserers, FTS/SRM/LFC... machines) or by technology they use (GPFS, XROOTD ...).

3. GPFS

GPFS is a general purpose distributed file system developed by IBM [9]. It provides file system services to parallel and serial applications. GPFS allows parallel applications to simultaneously access the same files in a concurrent way, ensuring the global coherence, or even different files, from any node which has the GPFS file system mounted.

GPFS is particularly appropriate in an environment where the aggregate I/O peak exceeds the capability of a single file system server.

Some of the GPFS performance advantages are accomplished by:

1) Striping data across multiple disks that are attached to multiple nodes.

2) An efficient client side caching mechanism that includes read-ahead and write-behind techniques.

3) Utilizing a configurable block size, a feature that is especially important with some of the newer disk technologies where very large block sizes are paramount to aggregate storage performance.

4) Using block-level locking techniques that are based on a very sophisticated token management system that provides data consistency while allowing multiple application nodes to have concurrent access to the files.

Deployment of the GPFS software was done in the line with the operating system installation using Quattor toolkit [10]. To do so, we did custom repackaging of the originally distributed packages adding pre-compiled compatibility layer modules for selected versions of the kernel.

Our production GPFS cluster consists of 22 disk server which are serving 20 file systems and about 700 clients. File system sizes vary from 4 to 48 TB, each file system is being served by 2 or 4 disk servers. Total storage space under GPFS in our installation is about 230TB. The GPFS version adopted was 3.1.0-10. Every SAN partition served by one Primary and one Backup NSD server, both of them having direct Fibre Channel access to the disks. Nodes not directly attached to the disks have remote data access over the local area network (Gigabit Ethernet) to the NSD servers.

In our tests (see next chapter) we realized a GPFS file system using 24 disk servers, aggregating all the SAN partitions available on each disk server in one global file system.

The file system which was used for the tests and all corresponding disk servers have been added to the production GPFS cluster, in the way to provide “direct” (or POSIX) access to the file system from all worker nodes. Apart from the other solutions employed in our tests, from the point of view of the applications, there is no need of compiling the client code with ad-hoc libraries.

4. GPFS and CASTOR test

4.1 Test layout

The test layout consists on 24 disk-servers integrated into the Storage Area Network (SAN) via Fibre Channel (FC) links, connected to the computing nodes, via Gigabit LAN (2x10 Gb/s trunked up-link), with an Extreme Black Diamond 10000 switch server.

As disk-storage hardware we used 2 EMC CX3-80 SAN systems, with a total of 260 TB of raw-disk space, assembled by aggregating nearly 520 500 GB SATA disks. Each EMC system comprised two storage processors (with 4 GB of RAM each), connected to a Brocade 48000 Fibre Channel fabric director provided with 8 links, for a total theoretical bandwidth of 32 Gb/s.

The 24 Dell 1950 disk-servers were equipped with Xeon 1.66 GHz dual-core bi-processors with 4MB L2 Cache, 4 GB of RAM, two Qlogic 2460 FC Host Bus Adapter (HBA) and 1 Gigabit Ethernet link. The Operating System (OS) installed was the Scientific Linux CERN (SLC) [4.1] vesion 4.4, with a 2.6.9-42.0.8.EL.cernsmp kernel operating at 64 bits.

For these specific tests a fraction of the production farm was used. It consisted of 280 rack mountable 1-Unit bi-processor worker nodes hosted in 8 racks corresponding to a total of 1100 possible simultaneously running processes. Each rack had a 2x1 Gb/s trunked up-link to the Extreme Black Diamond network switch and each computing node was connected with a Gigabit trunked up-link to the rack switch. A sketch of the test-bed is depicted in Fig. 4.

4.2 Sequential write/read local test from the disk-servers

The first test consist in measuring the local throughput by accessing the SAN devices directly from the disk-servers. The file-system used on the devices was XFS which offers good performance on medium/big size files and an high reliability. We realized sequential write and read operations by means of the dd standard Linux utility using a block size of 1MB and a file size of 12 GB concurrently from an increasing number of disk-servers. This measurement was used to evaluate the maximum achievable read and write throughput for the given hardware setup. Fig 5 shows the sequential write and read throughput as a function of concurrent processes. The number of processes was varied from 8 to 75 balancing over the different XFS file-systems and over the different disk-servers. The maximum write throughput was reached at 1.1-1.2 GB/s , while the maximum read throughput was roughly 1.5 MB/s. After the test with the XFS filesystem a single GPFS “big” filesystem was created over the available space (more than 110TB) and the same dd tests were repeated over this filesystem. The results proved that GPFS works using parallel I/O so the maximum bandwidth (plateau at about 1.2 GB/s when writing and 1.5 when reading) was reached with a very limited number of thread (one dd process for each diskserver is enought). Also the maximum bandwidth was identical to the one obtained with the XFS local filesystem and this proves that the GPFS layer doesn’t introduce considerable overhead over the disk access and in our case the controller array limit is the real “natural” bottleneck.

4.3 Sequential read/write remote tests from the Farm nodes using CASTOR and GPFS

The sequential remote writes and reads operations were performed using the dd application as job submitted in the Farm nodes production batch system. In the CASTOR case, i.e a non-POSIX system, the dd application is realized by means of a C-language code compiled using the appropriate client libraries. The test consisted of writing 1000 different files of 5GB in block of 64kB concurrently by 1000 processes running on 1000 worker nodes, ensuring a

synchronized start of the processes. In the same way the sequential read test was successively performed by reading from 1000 processes the files previously written. The Fig. 6 shows the read and write throughput as a function of time for GPFS and CASTOR data access platform. For the writes GPFS was about 30% faster than CASTOR achieving a rate of 1.3 GB/s. For the read operations CASTOR and GPFS showed similar performance with GPFS touched the highest rate of 1.5 GB/s. CASTOR showed a significant number of failures, which is evident in the CASTOR-read plot where the integral of the CASTOR curve is clearly smaller than the GPFS one, mainly due to CASTOR ver.2 queue timeouts.

5. Conclusion

In this paper we briefly summarized the storage resources in production in our Tier1. The main two systems we use to store the data at CNAF (GPFS as a disk-only pool and CASTOR as an HSM system with a tape library backend) were compared in some test of sequential access from the production farm. The main idea of the test was to obtain preliminary information about the maximum performance and the general reliability of the two different systems during heavy "stressful" access from a production environment. Both the systems performed well but the GPFS show the best performance (30% faster during write operations) and the best reliability since only a 2% of jobs failure rate was detected versus a minimum 10% of jobs failed during CASTOR access. These preliminary results prove that both the systems could be efficiently used in production despite their advantages and disadvantages (i.e. castor has the tape backend while GPFS is a disk-only system but GPFS is a cluster so the performance and reliability of the service are higher). The reason to maintain in production both the systems is that the different policy on the use of the storage implemented by the High Energy Physics experiments in their computing models could find their best fulfilment in one specific system.

References

[1] see www.qlogic.com "SANsurfer Application user guide" for more information

[2] IBM Global Service Deployment Guide " IBM GPFS for Linux: Concepts, Planning, and Installation guide"

[3] http://castor.web.cern.ch/castor/ fom more documentation about the CASTOR software and RFIO protocol

[4] http://www.platform.com/Products/Platform.LSF.Family/Platform.LSF/

[5] http://sdm.lbl.gov/srm-wg/

[6] Nagios home page, http://www.nagios.org.

[7] RRD home page, http://people.ee.ethz.ch/~oetiker/webtools/rrdtool

[8] http://lemon.web.cern.ch/lemon/docs.shtml

[9] http://publib.boulder.ibm.com/epubs/pdf/bl1ins11.pdf

[10] http://quattor.web.cern.ch/quattor/

[11] http://linux.web.cern.ch/linux/scientific4/

Illustrations:

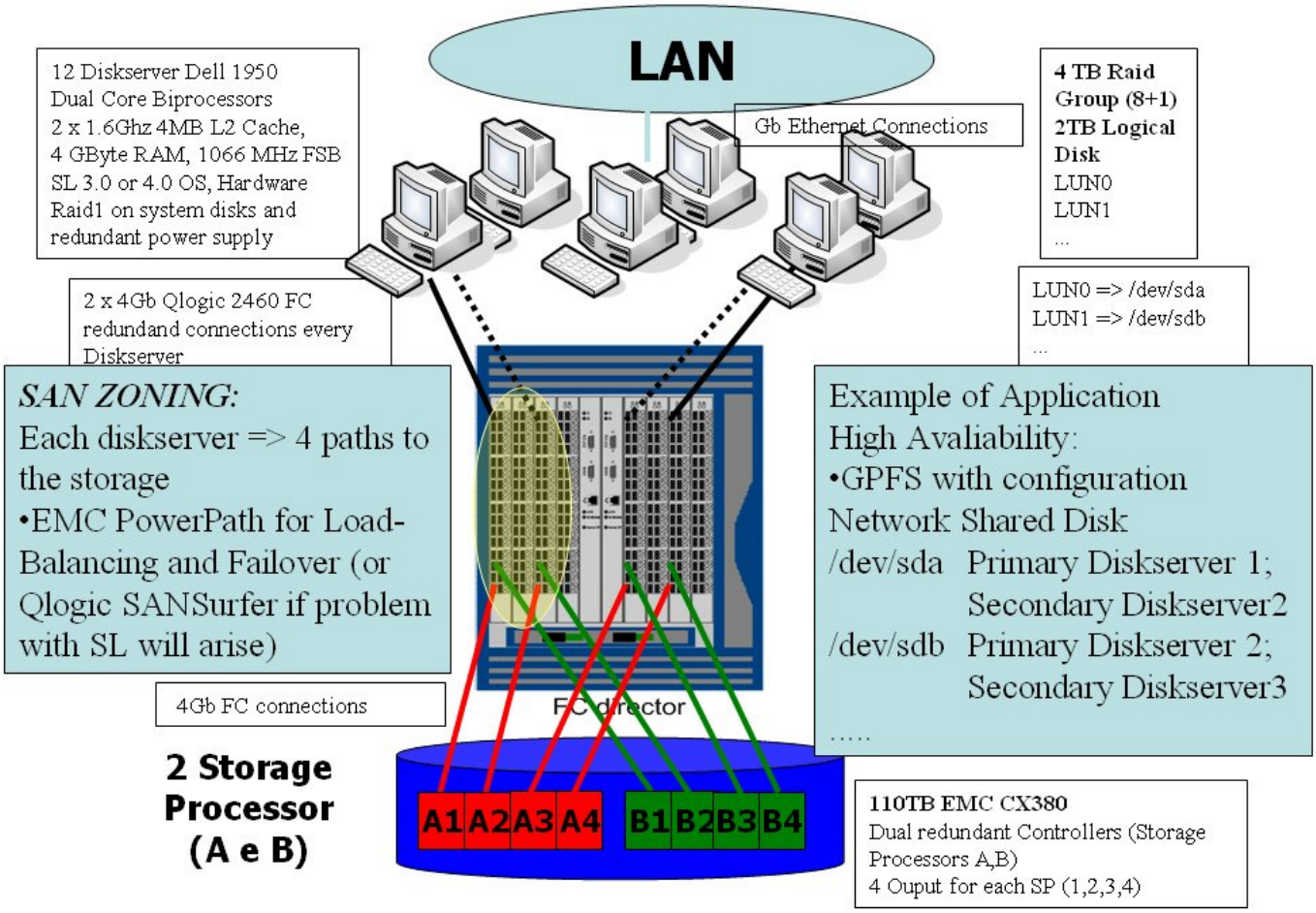

Fig.1: Schema of a standard diskservers-to-storage connection and implementation.

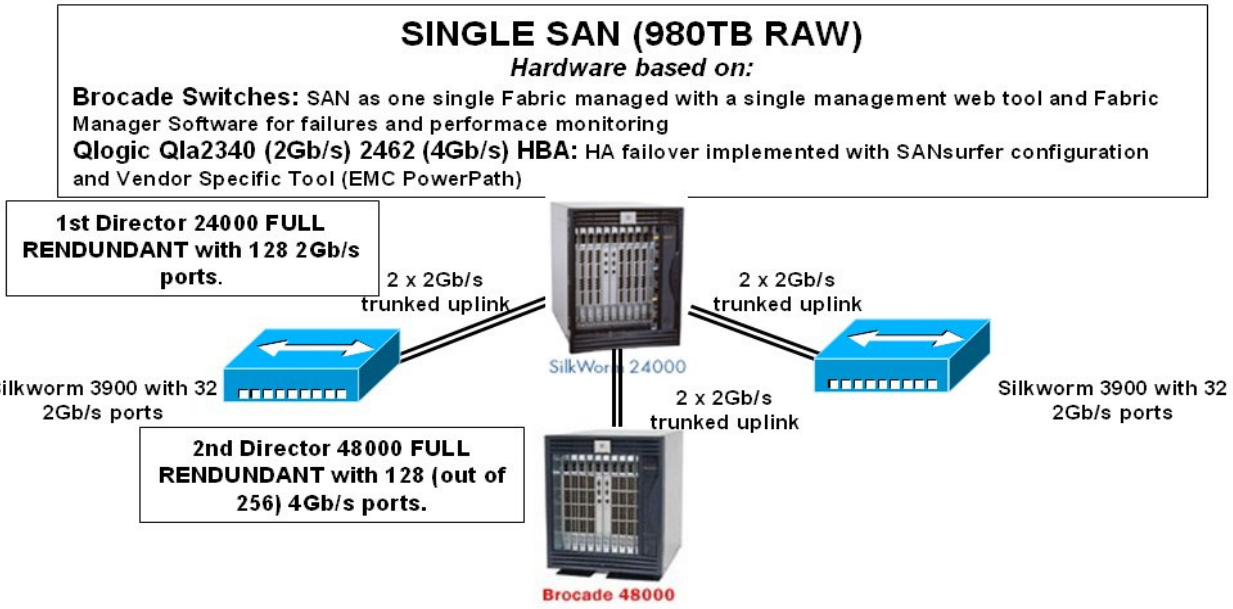

Fig.2: Single SAN in production with the star topology connections between switches

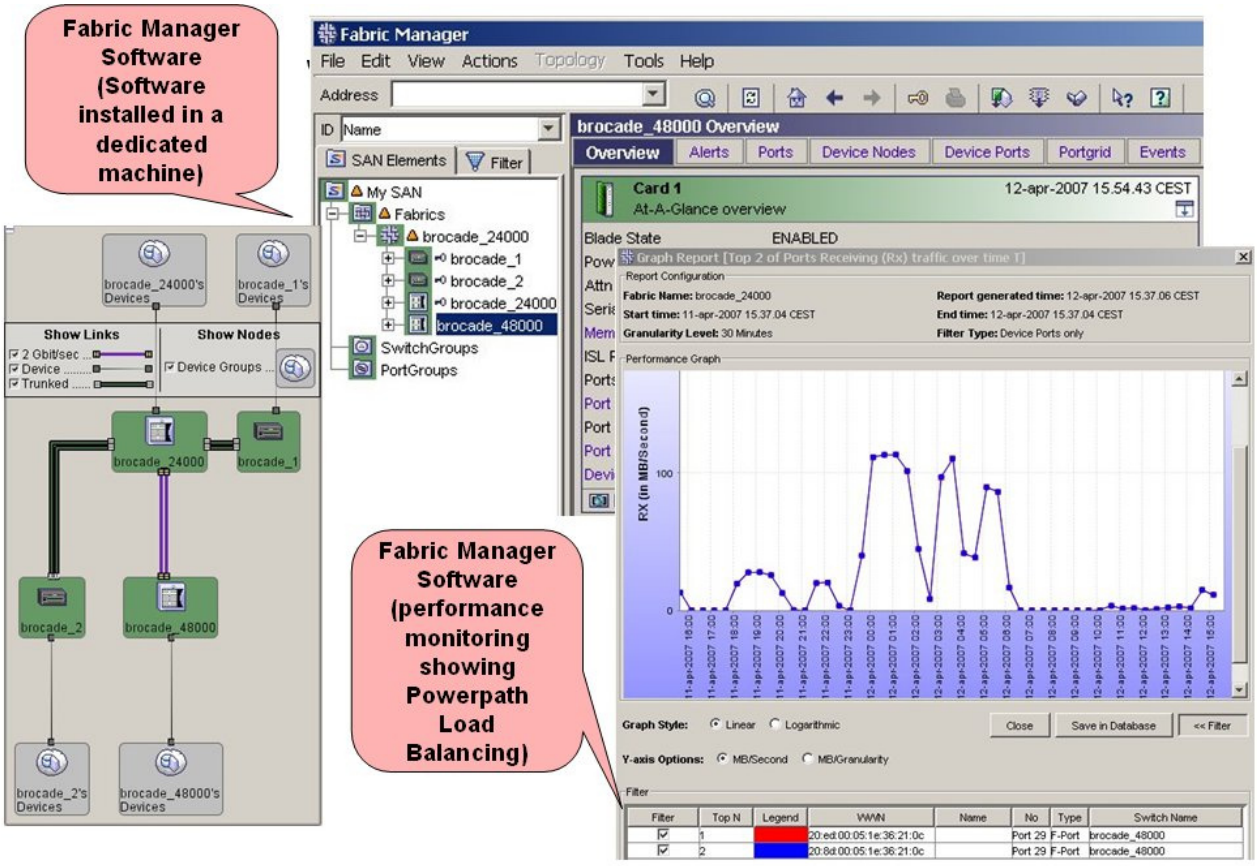

Fig.3: Screenshots of the Fabric manager Software

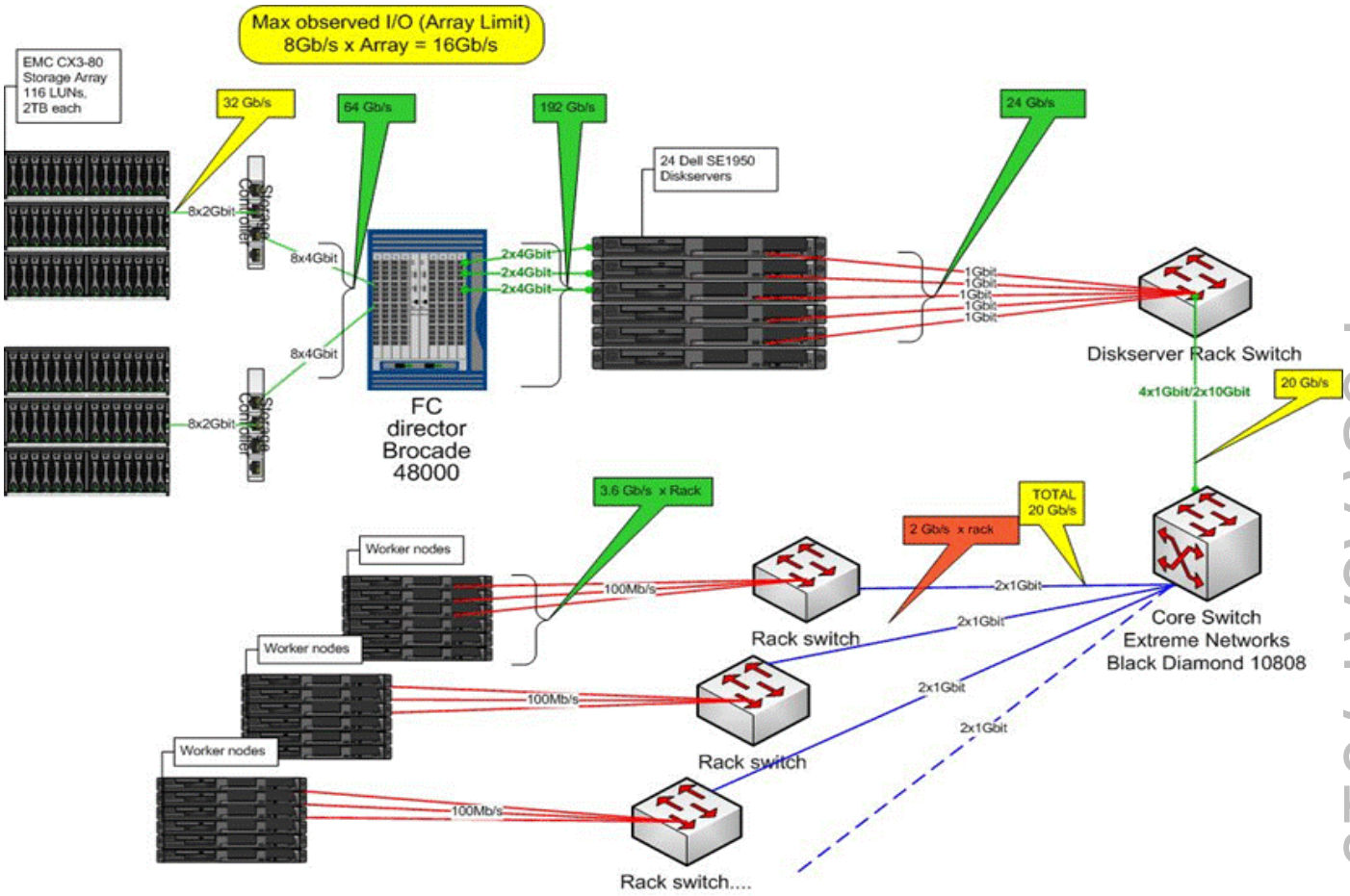

Fig.4: Schema of the testbed connections

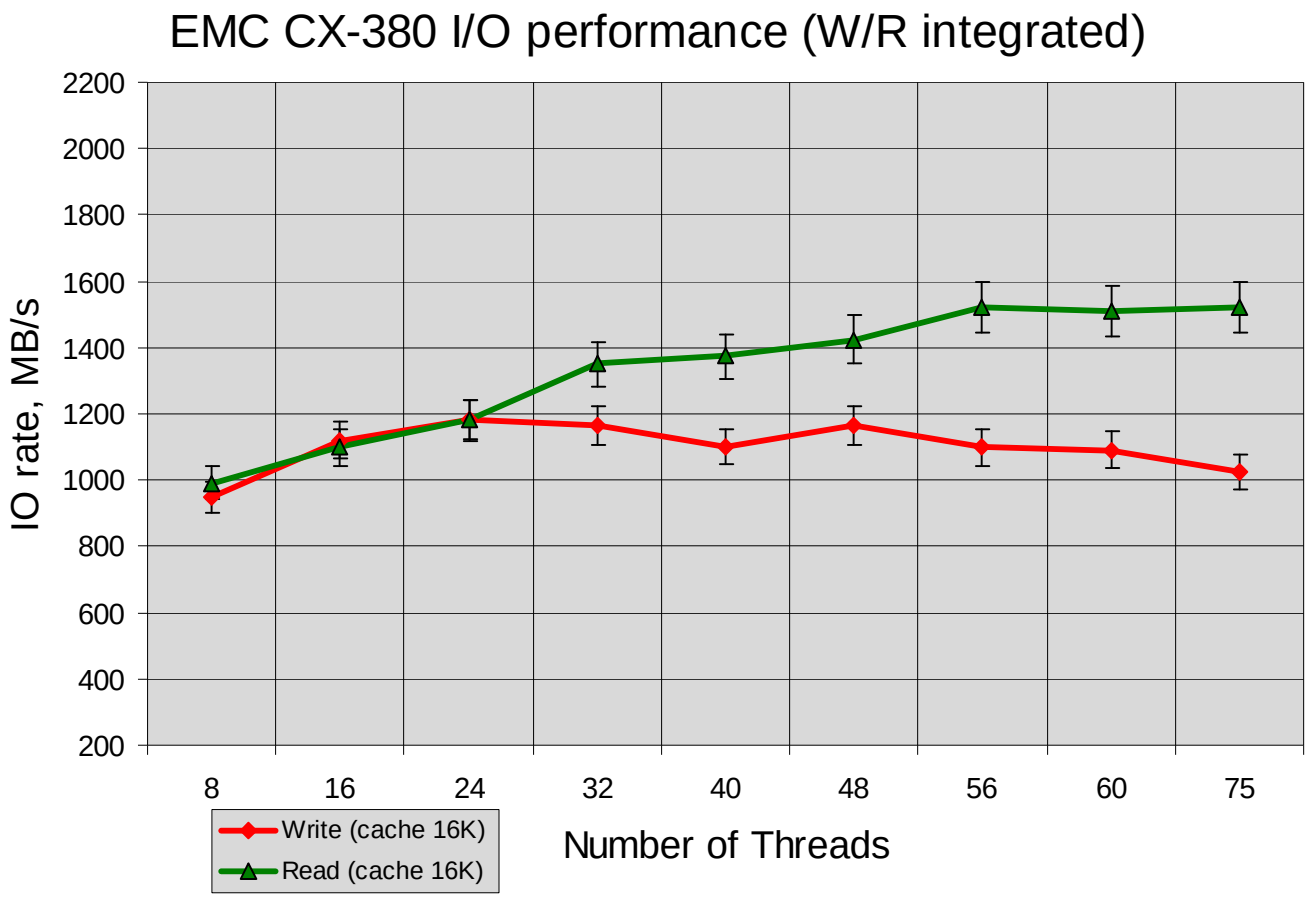

Fig.5: Result of the Local test using the “dd” utility running over xfs filesystems.

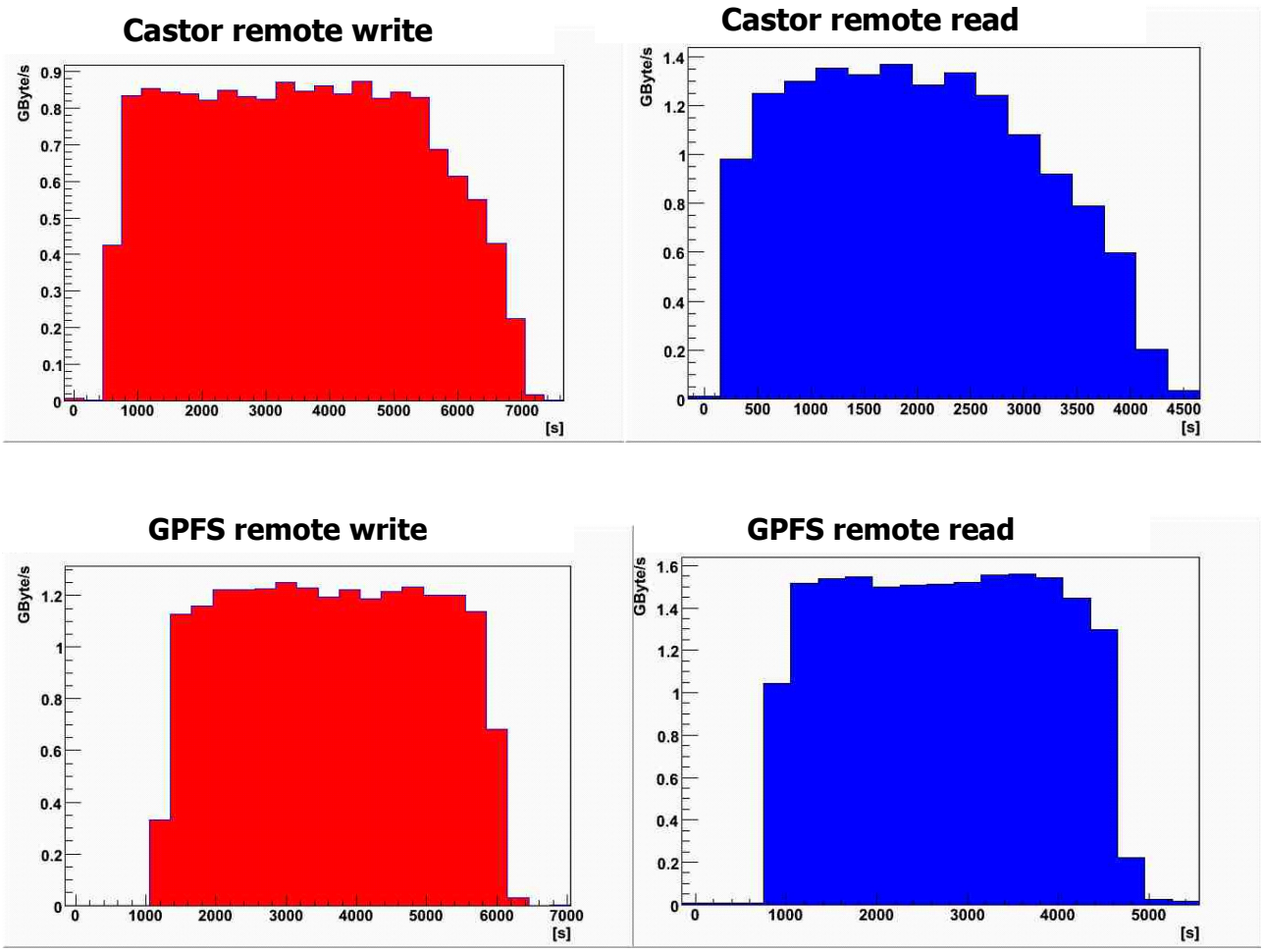

Fig.6: Remote test results. The two upper graphs show the result obtained with the Castor diskpool, the lower graphs show the result with the GPFS cluster filesystem.