The Grid Enabled Mass Storage System (GEMSS): the Storage and Data management system used at the INFN Tier1 at CNAF.

Pier Paolo Ricci\textsuperscript{1,4}, Daniele Bonacorsi\textsuperscript{2}, Alessandro Cavalli\textsuperscript{1}, Luca Dell’Agnello\textsuperscript{1}, Daniele Gregori\textsuperscript{1}, Andrea Prosperini\textsuperscript{1}, Lorenzo Rinaldi\textsuperscript{1}, Vladimir Sapunenko\textsuperscript{1} and Vincenzo Vagnoni\textsuperscript{1}

\textsuperscript{1} INFN CNAF Viale Berti Pichat 6/2 40127 Bologna, Italy.
\textsuperscript{2} University of Bologna, Physics Department Via Irnerio 46 40127 Bologna, Italy.
\textsuperscript{3} INFN Sezione di Bologna, via Irnerio 46, 40126 Bologna, Italy.

E-mail: pierpaolo.ricci@cnaf.infn.it

Abstract. The storage system currently used in production at the INFN Tier1 at CNAF is the result of several years of case studies, software development and tests. This solution, called the Grid Enabled Mass Storage System (GEMSS), is based on a custom integration between a fast and reliable parallel filesystem (the IBM General Parallel File System, GPFS), with a complete integrated tape backend based on the Tivoli Storage Manager (TSM), which provides Hierarchical Storage Management (HSM) capabilities, and the Grid Storage Resource Manager (StoRM), providing access to grid users through a standard SRM interface. Since the start of the Large Hadron Collider (LHC) operation, all LHC experiments have been using GEMSS at CNAF for both disk data access and long-term archival on tape media. Moreover, during last year, GEMSS has become the standard solution for all other experiments hosted at CNAF, allowing the definitive consolidation of the data storage layer. Our choice has proved to be very successful during the last two years of production with continuous enhancements, accurate monitoring and effective customizations according to the end-user requests. In this paper a description of the system is reported, addressing recent developments and giving an overview of the administration and monitoring tools. We also discuss the solutions adopted in order to grant the maximum availability of the service and the latest optimization features within the data access process. Finally, we summarize the main results obtained during these last years of activity from the perspective of some of the end-users, showing the reliability and the high performances that can be achieved using GEMSS.

1. Introduction
High Energy Physics (HEP) experiments have been for many years amongst the largest consumers of computing and storage resources in the world. Therefore several studies of data handling and computing solutions have recently arisen and have been used in production in various computing centers. Data management and access systems are hot-topics in the HEP world, and constitute one of the most demanding tasks of HEP computing. The resources needed by the LHC experiments notably include large data storage sizes with high performance demands. Moreover, very reliable and high performance disk-based data access systems are required, as well as an efficient Mass Storage Systems (MSS) allowing archival of several PB of data per year on tape facilities, that should remain available over a time scale of several years.

The INFN-CNAF computing center hosts the Italian World-wide LHC Computing Grid (WLCG) Tier1 site, the largest Italian computing facility employed in the LHC distributed computing infrastructure [1]. Since 2008 we have started our tests and the consequent production activities.
focused on the integration of our IBM General Parallel File System (GPFS) [2] disk storage infrastructure with the IBM Tivoli Storage Manager (TSM) [3], aiming at realizing a full Hierarchical Storage Management (HSM) system, that we named GEMSS. GEMSS uses StoRM as Grid Storage Resource Manager (SRM) [4], as an interface to the WLCG world and it is a complete solution distributed under GSL license with standard rpm packages. The GEMSS system is currently used as the Tier1 storage solution for all the four LHC experiments and other HEP experiments, like BaBar (SLAC) and CDF (Fermilab), the astroparticle physics experiments VIRGO and ARGO, the AMS, GLAST and PAMELA satellites, the MAGIC telescope and others.

In terms of resources the INFN-CNAF is currently using a model where all our hardware storage systems (disk and tape drives) are accessed through Storage Area Network (SAN) [5] switches and Linux servers running Scientific Linux as operating system and equipped with redundant HBA (Fibre Channel Host Bus Adapter). This has demonstrated so far to be a robust, stable and very flexible approach.

At the moment a total of 8.4 PetaByte (PB) net used disk space (available to the end user) is managed by GEMSS, and this number is going to increase to 11.2 PB during summer 2012. The whole storage area is composed by the following storage hardware devices:

- 7 Data Direct DDN systems S2A 9950 for a total of 7 PB (the disks are SATA technology with a 2 TB capacity) served by about 40 disk servers with 10 Gb/s Ethernet connection to the LAN network;
- 7 EMC^2 CX3-80 + 1 EMC^2 CX4-960 for a total of 1.4 PB (SATA technology, 1 TB disk) served by about 90 disk servers with 1 Gb/s Ethernet connection to the LAN network.

In addition 3 Fujitsu Eternus DX400 S2 for a total of 2.8 PB (SATA technology, 3 TB disk) are under installation. An Oracle SUN SL8500 tape library with a total of uncompressed 14 PB tape space is also used in production with a number of 20 Oracle T10KB drives (100MB/s of bandwidth and 9000 1TB tape cartridges) and 10 Oracle T10KC drives (200MB/s of bandwidth and 1000 5TB tape cartridges). The connections to the tape drives use a subset of the Fibre Channel SAN that is referred to as Tape Area Network (TAN).

The storage resources are also accessible from the WAN by means of GridFTP servers with the SRM control. In the LAN environment the total number of Worker Nodes (computing resources) take advantage of the POSIX compliant GPFS client as they access the shared filesystem as it were local to the nodes. Therefore roughly 9000 CPU cores corresponding to a computing power of about 120.000 HS-06 are currently directly accessing via POSIX the GEMSS filesystems. This number will increase to 13000 cores, with an estimated computing power of 150.000 HS-06, during spring 2012.

2. Introduction to GEMSS concept and development

GEMSS is a full HSM integration of GPFS, TSM and StoRM, and its primary advantages are the high level of reliability that is obtained and the minimum management effort that is needed for daily maintenance. At the moment two staff members are employed for managing more than 8 PB of net disk space partitioned in several GPFS clusters and 9 PB of tape space (out to a maximum capability of the library amounting to 14 PB).

The original idea behind GEMSS was formulated in 2006, when we were searching for a performing and robust solution which could provide the best service reliability with minimal efforts. At that time first positive experiences were made at CNAF using different types of parallel filesystems such as GPFS and Lustre [6]. The CNAF storage team concluded the study in early 2007, when a series of stress tests aiming at the comparison among CASTOR [7], GPFS and Lustre as well as other solutions, i.e. dCache [8] and xrootd [9]. As an outcome of those tests, GPFS qualified itself as the

---

5 The HS-06 or HEP-SPEC06 is the HEP-wide benchmark for measuring CPU performance that has been developed by the HEPiX Benchmarking Working Group. For details http://w3.hepix.org/benchmarks/doku.php
most promising solution for our purposes and in our environment both in terms of easiness of management and high throughput performances [10].

While GPFS was being selected as the best choice, the implementation of the StoRM Grid interface was under development at INFN-CNAF. StoRM [11] is a so-called Grid Storage Resource Manager (SRM) that implements the SRM interface version 2.2 and it is designed to support guaranteed space reservation and direct access using native POSIX I/O calls to the storage. The main feature of StoRM is the possibility to take advantage of parallel file systems like GPFS and Lustre. Using GPFS and the StoRM interface to access the disk area from the WLCG grid, the first implementation of the so-called D1T0 Storage Class (disk1-tape0) [12] was realized at CNAF. In the WLCG model the D1T0 storage class means that the data is resident uniquely on disk and there is no guaranteed copy on tape.

Following the D1T0 StoRM/GPFS implementation two LHC experiments (LHCb and ATLAS) started to use this system at the Tier1 with satisfactory results since November 2007 [13]. The subsequent step was the development of a complete solution for implementing the remaining two storage classes of the WLCG model:

D1T1, where data files are permanently stored on disk and on tape as backup;
D0T1 where data files are primarily stored on tape, and the disk area is considered only as a temporary buffer managed by the system as a disk cache (also called staging area). Usually oldest files already migrated to tape are automatically removed from the staging area when disk space is reclaimed for new files.

It became clear that a real HSM solution based on a tape extension of the StoRM/GPFS system was needed. During 2008 INFN-CNAF launched a project to realize a complete Grid-enabled HSM solution based on StoRM, GPFS and the Tivoli Storage Manager (TSM) software. In this implementation, the GPFS specific features (available since version 3.2 and described in the next section) were basically combined with TSM and StoRM and StoRM itself was extended to include the SRM methods required to manage data on tape. In addition, a specific customization was added in order to provide the interface needed for the interaction between GPFS and TSM. In the late spring 2008 the first implementation of a D1T1 prototype was released and the LHCb experiment started to use it in production. The promising results achieved encouraged us to pursue the remaining development in order to realize in October 2009, the implementation of the D0T1 Storage Class. Since then, the full system was named the Grid Enabled Mass Storage System (GEMSS). In the same month CMS, one of the biggest LHC experiments at our Tier1, was migrated to GEMSS and performed a validation through a complete round of tests of the most relevant workflows, including a repetition of the STEP’09 tests [14]. The positive results demonstrated the robustness of the solution and at the end of 2010 all the LHC experiments in addition to other non-LHC ones started to use GEMSS in production. Once all the Tier1 datasets had been migrated the former production system based on CASTOR was decommissioned.

The GEMSS main elements, as mentioned before, are the GPFS filesystem, the TSM system for space management of the tape area and StoRM for accessing the system with the grid tools. GPFS is a clustered filesystem which provides a scalable POSIX file access. The GPFS clients (i.e. the farm Worker Nodes in our case) do not need a direct connection to the storage backend thought the SAN, but they access the data using the GPFS Network Shared Disk (NSD) mechanism [15]. The NSD diskservers have redundant Fibre Channel connections to the SAN and provide GPFS transparent access to the clients, as they are directly responsible for the I/O operation on the disks. In addition, GPFS works as a real cluster, so the NSD diskservers can provide the same level of service also in case of hardware/software failures of some components. At present a number of roughly 130 diskservers is used in production for accessing the whole disk space area, subdivided into different GPFS clusters. A total of seven clusters is used, one cluster for each of LHC experiments, one dedicated to SuperB/BaBar and two shared between no-LHC users. A further GPFS cluster comprising only clients is used for the farm Worker Nodes. The use of intensive parallel I/O over all the diskservers is useful for optimizing the performance and for a proper distribution of the load over the different hardware storage boxes.
The main elements of the TSM system are the master TSM server and some HSM nodes which are directly responsible for moving the data to and from the tape drives and run the TSM Storage Agents [16]. The TSM server is the core component, it relies on a database to store the metadata information and it also provides the space management services to the HSM nodes. The TSM server stores all the information on a dedicated shared disk volume, and a cold stand-by machine is ready for replacing the main server in case of hardware or software failure, in order to provide a fast recovery of the service. The TSM Storage Agents enable LAN-free data movements on the HSM nodes, using the dedicated TAN Fibre Channel connections to communicate with the drives, and this greatly improves the performances avoiding traffic congestion on the LAN when moving data between the disk and tape media. In our setup a total number of 13 HSM nodes are largely enough for providing all the data movements with optimal performance. In order to avoid contention and for redundancy purposes, a minimum of two HSM nodes is dedicated to each GPFS cluster. Since the GPFS NSD diskservers are separated from the HSM nodes, it is possible to interrupt the tape access service for maintenance while keeping the disk service online: this is very useful in case of tape library failures or scheduled upgrades of the software.

The GPFS and TSM interaction is the main component of the GEMSS system and therefore represents the essence of the software. The data flow mechanism between the two will be described in order to fully understand the implementation schema.

3. GPFS and TSM interaction: how GEMSS works

3.1. GPFS advanced features
GPFS provides advanced features that are the keys for its integration with TSM. These notably include the Information Lifecycle Management (ILM) subsystem and the implementation of the XDSM Data Management API (DMAPI) [17] interface. With the utilization of ILM, the filesystem can be partitioned into a number of so-called GPFS “storage pools”. ILM uses a SQL-like policy language to implement file placement on a specific storage pool and data migration rules from one pool to another according to some user-defined criteria.

Starting from GPFS version 3.2, the concept of “external storage pool” was introduced into ILM. External storage pools allow the implementation of a policy-driven migration/recall system to a tape storage backend such as TSM. In this case, TSM takes on the role of manager for an external storage pool [18] and could be seen as a tape extension of GPFS. Interfacing GPFS with specific implementations of tape back-ends does not come natively, but must be realized by means of specific external pool script elements that are components of the GEMSS system. The GPFS ILM policy engine automatically scans the filesystem metadata, it builds candidate lists for migration using the ILM scan operation of the filesystem, it builds lists of files whose path names and other attributes match the given search criteria. These lists are then transferred to external scripts or programs for user-defined processing. In our case, the external programs takes these lists as input and then move the files to and from tape by means of TSM HSM client commands. In order to perform fast file system scans, GPFS is provided with optimized metadata structures. The scans can also be parallelized across many nodes in the GPFS cluster for better performance and scalability.

Another important element that is used in the GEMSS system is the XDSM DMAP interface. The DMAP interface provides the possibility to use a set of file system extensions that are the keys used to implement the HSM system. Amongst the most important features we mention:

- **DMAPI events.** They provide notification of file system activities to a Data Management application.
- **DMAPI managed regions.** Files can be divided into distinct portions known as managed regions. The notification of an event to the Data Management application upon their access can be produced on a per region basis and this allows for finer granularity of event notification.
- **DMAPI extended attributes.** They are extensions of the usual UNIX attributes, and can be associated with each file system inode in order to provide unique information for the file. The unique
information can be used in implementation and in our case they are used for interaction between the different GEMSS components.

Both these elements, ILM and DMAPI, play a center role in the GPFS/TSM interaction as described in the next subsections.

3.2. Pre-migration and migration: from disk to tape
In the TSM terminology, “pre-migration” stands for the action of copying a file to tape, but keeping the original copy also on disk. With “migration” instead, one refers to the action of copying the file to tape and removing the content of the file from disk (but not the inode structure), or simply removing the content of the file from disk if the file had been previously pre-migrated to tape. When the content of a file is removed from disk, a so-called “stub” file is kept, i.e. the inode remains on disk and can be listed as a normal file. During these operations some DMAPI extended attributes are added by TSM to the stub file, containing a sequence to be used as a unique key for identifying the file in the TSM database. DMAPI managed regions are set with specific events, namely READ, WRITE and TRUNCATE. When the file is in the migrated state, all the three events are set on the file, such that any of these three operations would trigger a signal, then intercepted by the appropriate DMAPI application listening for these signals. When the file is in a pre-migrated state instead, just WRITE and TRUNCATE events are set, as e.g. the DMAPI application must be able to forbid the operation or to invalidate the copy on tape in case the file is updated on disk.

GEMSS continuously starts filesystem scans (with a user-defined maximum frequency) with ad-hoc ILM policies in order to find file candidates that need to be pre-migrated to tape. The system tries to pre-migrate files to tape “as soon as is possible” (this a basic requirement for data custodial in WLCG data centers). The number of pre-migration threads (for each file system) running on each HSM node can be configured and must be chosen according to the available resources and desired degree of parallelism. When the file system reaches a configurable certain level of occupancy, garbage collection is started and the contents of already pre-migrated files are removed from disk, leaving in place just empty stub files.

3.3. Selective and transparent file recalls: from tape to disk
The data access is clearly immediate when files are only pre-migrated since the files are already totally resident on the disk area. But in case of accessing a migrated file only the stub file is present on disk and the real data needs to be recovered from tape. Therefore an optimal way to recall files from tape to disk has to be defined. In TSM, data files can be recalled from tape to disk using two distinct recall methods:

• **Selective recalls.** The user (or a specific service on his behalf) asks for a file to be recalled from tape prior to the first access. This typically happens in the WLCG world, where before submitting a job to a working node (i.e. making the first file access) the user asks for a file to be recalled from tape using SRM commands (StoRM). Only when all the needed files have been recalled the access or transfer (using GridFTP for WAN access) is performed.

• **Transparent recalls.** The file is accessed by a read operation (usually from user jobs) without a distinction between pre-migrated files (still on disk) or migrated ones (only the stub file is on disk). If the file is not on disk, the read operation triggers via DMAPI the recall of the file from tape. When the recall is finished and the file is accessible on disk, the control is given back to the user’s process.

The recall operation needs optimized access to data stored on tape and this is of primary importance for realizing a high performance HSM system. The tape recalls should be collected and grouped together in a specific time lapse and requests should be ordered (i.e. sorted) according to the files distribution on tapes to minimize the number of mount/dismount operation in the tape library and unnecessary tape “seek” operation. The standard TSM HSM behavior consists in recalling files as soon as they are requested by users, following the same order of the requests. As the user has no knowledge of where the files are stored, and in particular how the files are ordered within a tape, such
a procedure ends up in a very inefficient usage of the tape resources. To overcome this limitation, GEMSS implements its own aggregation and reordering of tape recalls before submitting them to TSM, as described in the next section.

4. GEMSS optimization, management and monitoring

4.1. GEMSS recall system latest enhancement

The main new improvement in the GEMSS development is the use of the DMAPI GEMSS Server. The DMAPI GEMSS Server role is to intercept all the read events using GPFS DMAPI directly during filesystem access operations. In Figure 1 a scheme of the present GEMSS components is shown. Following the scheme, the GEMSS Recall block is contacted directly when recall requests are made through StoRM, or when recalls are submitted via command line, thus triggering selective recalls. Otherwise, the user application running on Worked Nodes could trigger transparent recalls simply by accessing the files, that are intercepted by a GEMSS DMAPI application and internally submitted to the GEMSS system as selective recalls. In practice, the DMAPI GEMSS server is used to transform a transparent recall into a selective recall and for waiting the actual recall to be performed. For each transparent recall, the DMAPI server spawns a thread which waits submits the recall, waits until the recall is over, and then replies the client to go ahead if the recall was successful, or returns an error message if it failed. In the previous versions of GEMSS [19] a “preload C-library” on the client side was used in order to allow for transparent tape recalls. The preload library overwrote the libc “open” call, submitting recalls to GEMSS if needed, and then returning the ordinary open call. With the adoption of the DMAPI GEMSS server, the preload library is not necessary anymore and from the client perspective the access to files on tape is fully POSIX.

![Figure 1. Scheme of the GEMSS components.](image)

The selective tape-ordered recall system of GEMSS is composed of few main processes that are responsible for queuing the recalls and starting the recall streams. The interaction of the basic processes is sketched in Figure 2. The GEMSS system manages a FIFO recall queue, fetches files
from the queue and then builds sorted lists with optimal file ordering. Finally the actual recalls from TSM to the GPFS file system are performed using a dedicated stream for each tape, as shown in the figure with four streams as an example.

![Figure 2](https://example.com/figure2.png)

**Figure 2.** Interaction of the basic GEMSS processes for the selective tape-ordered recall operation.

The `yamssReorderRecall` process takes available entries from the recall queue and fetches them in chunks of configurable width. It then sorts the file lists according to tape ordering and puts ordered lists into a shared directory, i.e. a system directory in the GPFS file system itself. The `yamssReorderRecall` process also fetches file names to be recalled directly querying a dedicated StoRM service, in case of recalls initiated via SRM calls. The recall queue can also be populated by means of a client command, `yamssEnqueueRecall`, or by the GEMSS DMAPI server in case of transparent recalls. If a file list for a given tape already exists and is waiting for the recall stream to start, `yamssReorderRecall` adds new files to this existing list in the correct order. This process is scheduled by the `yamssMonitor` and wakes up every time that file recalls are submitted. The `yamssMonitor` runs as a service on all HSM nodes and is the supervisor of the reorder and recall phases. It discovers managed GPFS file systems on HSM nodes, reads the configuration file for each file system (stored in a dedicated system directory of the GPFS file system itself) and triggers the needed actions like starting other processes. It loops continuously in the background and spawns all needed `yamssProcessRecall` processes according to the configuration of each mounted file system. It also auto-detects changes in the configuration files for each file system and reconfigures actions to be taken accordingly. The `yamssProcessRecall` starts one recall stream for a given tape and sequentially executes all the recalls taking as input one ordered file list produced by `yamssReorderRecall`. More than one recall process per node can be started according to how many parallel recalls are desired. In the present implementation it chooses first amongst the available file lists that with the largest number of files to be recalled, unless a file list with one request older than a configurable time threshold exists. The `yamssProcessRecall` also manages error conditions in case of failures, re-inserting into the queue.
the failed file recall for a configurable number of times, in order to retry the operation and achieve some fault tolerance due to temporary issues.

The interaction of the processes described above is the software engine of the GEMSS selective recall mechanism, properly one of the most interesting part of the system, and they are running on all the TSM HSM nodes using the shared GPFS dedicated “system” directory area for information sharing.

4.2. GEMSS administration and monitoring
A set of administrative command lines for the daily operation of the system has been developed. They are used for monitoring, stopping and starting migrations and recalls in case of interventions, querying the current status of the system and produce reports, and other operations of common use. GEMSS comes as a RPM package for easy installation and software upgrades. Different log files are produced, containing information on every single operation done by the system. Just as example the log file containing the information on the tape recalls performed by the system, for each recalled file reports the following information:

- final status of the operation (REC OK or REC FAILED);
- relevant time stamps (recall request submission, start of recall stream on the given tape, end of recall stream, file creation time);
- complete GPFS file path;
- file size;
- total size of all files in the recall stream of the given tape;
- HSM node from which the recall was done and tape label hosting the file.

All these information can be collected and analyzed in order to produce cumulative statistics and reports. For example, using such information, the plots shown in Figure 3 are obtained which show the cumulative data traffic from disk to tape and from tape to disk for the last about 1000 days (starting from May 2012).

![Figure 3](image)

**Figure 3.** Cumulative traffic from disk to tape (a) and from tape to disk (b). The contributions of the various experiments are shown in different colours. A total of about 11 PB have been moved from disk to tape by GEMSS so far. The total size from tape to disk amounts to about 1.6 PB.

In addition to the administrative commands, the GEMSS components (i.e. StoRM, GPFS and TSM) have been integrated with two monitoring tools used at the INFN-CNAF Tier1: Nagios and Lemon.

Nagios [20] is a monitoring tool for hosts and services. Nagios has a wide set test scripts (plugins) and can be fully customized with new ones. The monitoring daemon runs intermittent checks on hosts and services using plugins and the status information are collected by a central server which takes actions according to the configuration. Nagios advanced features have been exploited in order to
control the state of GEMSS hosts and services. Therefore a specific set of plugins has been developed for monitoring the GPFS clusters, the TSM server and HSM nodes and the health of the StoRM service and nodes. In some critical cases Nagios executes corrective actions like the restart of the service or the removal of the failed node from the cluster, sending at the same time an e-mail notification to the system administrators.

Lemon [21] (LHC Era Monitoring) is a monitoring tool developed at CERN, available for Linux, providing plots on the current status and on historical information. New custom Lemon sensors have been written for the GEMSS system, in particular reporting GPFS utilization and Fibre Channel traffic to the tape drives. In addition the tape cartridge activity in the library (i.e. mount and dismount of tapes on drives) is collected and shown in plots. Lemon extracts also useful information from the GEMSS log files and aggregates them in specific plots.

In Figure 4 two screenshots of the Nagios (left) and Lemon (right) web interface are reported. In the Lemon plot it is possible to observe, for a specific TSM HSM node (named “tsm-hsm-4”) and the LHCb GPFS cluster the throughput of the GPFS to TSM data migration and the number of file successfully migrated.

**Figure 4.** Screenshots of the Nagios (a) and Lemon (b) web interfaces.

5. Experience with GEMSS from selected experiments

In this section we describe the experience of three of the most representative experiments which use GEMSS in production at CNAF.

5.1. ATLAS experiment

ATLAS has adopted GEMSS at the INFN-CNAF Tier1 commencing 2010 [22]. The ATLAS storage capacity at the INFN Tier1 currently consists of 2.7 PB of disk and 3.6 PB of tape space. According to the ATLAS computing model [23], the data placed on tape at the Tier1s tape are the experimental (real) RAW data produced by the ATLAS experiment at CERN and simulated HITS data. These data are distributed among the ATLAS Tier1s and the CNAF Tier1 receives a 10% fraction. An additional 10% of data having an analysis-suitable format is dynamically placed on the Tier1 disks, according to the data usage and popularity.

Between January 2011 and April 2012, the INFN-CNAF Tier1 received an average throughput of 84 MB/s of ATLAS data, recording a peak of 1800 MB/s and a sustained throughput during one good
day of 800 MB/s. The data transfer efficiency, evaluated as the fraction of succeeded transfers, has been 94%, similarly to the other ATLAS Tier1s. Figure 5 shows the ATLAS data transfer throughput to the INFN Tier1 tape over a week, when a peak of 600 MB/s was reached. The tape inbound traffic consists mostly of data exported from the Tier-0 and transfer of reconstructed data between Tier1 and Tier2 sites.

![Figure 5. ATLAS data throughput to INFN-CNAF Tier1 tape system.](image)

ATLASS data stored on tape are recalled mainly during reprocessing campaigns, and occasionally for specific detector studies. During data reprocessing campaigns, in few days, several tens or hundreds of TB of data need to be processed, and an efficient tape management system is needed to cope with the requirements. During a reprocessing job workflow, a bulk of data of the order of some TB is recalled on the disk buffer before the job execution. In this phase the data traffic between the TSM and GPFS can reach about 500 MB/s. In addition to the reprocessing activities, the system is also receiving data from the Tier0. During the 2011 data reprocessing campaigns, about 200 TB of data have been reprocessed at INFN-CNAF Tier1, and GEMSS showed good performance and stability.

In summary, GEMSS fulfilled all the ATLAS requirements at the INFN-CNAF Tier1, and allowed an efficient execution of the ATLAS computing activities.

### 5.2. CMS experiment

In cooperation with the CNAF storage team available, the CMS contacts at CNAF performed tests aiming to commission the use of GEMSS for the CMS experiment. The CMS VO at CNAF had historically been the main Castor user on-site, and over years the CMS support people have been giving constant support to the storage experts, so this came as a natural extension of the collaboration. CMS performed tests which are briefly outlined in the following paragraphs.

CMS switched operations to the new system in October 2009, when new data now started to land on GEMSS, and old data were fully moved from Castor to TSM. A wide plethora of preparatory tests were performed, all extremely successful in terms of transfer rates achieved and quality of transfers. Specific tests to reproduce the standard CMS workflows were performed, in particular using the PhEDEx [24] dataset replication system and the “LoadTest” infrastructure [25]. A set of tests were
aimed at demonstrating that the system could handle typical CMS rates. Another set of tests aimed at demonstrating that the CMS workflows efficiency was not impacted by the change of storage system or basic site configuration: these were done using the CMS "Job Robot" infrastructure. Finally, set of tests were performed with real CMS jobs on the CNAF farm, and were aimed to demonstrate that the disk storage of the GEMSS infrastructure could serve data at an acceptable rate.

The results obtained were very positive. Some tests on TSM using local access (i.e. not through the StoRM layer) were initially performed: both recalls from tape (with peak rates as high as almost 600 MB/s) and migration to tape (with peak rates of the order of 100 MB/s) were successfully tested. Local access to data on TSM from the batch farm nodes was recorded to be as speed as 400 MB/s. With the LoadTest we achieved up to 300 MB/s in export to several dozens of sites, sustained for several hours. With the Job Robot infrastructure we tested a comparison between GEMSS and Castor, and we observed no decrease in the overall success rates of the test jobs. With the tests done with real CMS jobs, we achieved ~1.2 GB/s served from the storage to the nodes, with a ~100% success rate. During these tests, and in parallel to ongoing production activities, a global migration of all data from the old CASTOR system to the new TSM-based system was performed: roughly 1 PB of data in total were copied from Castor to TSM in roughly 1 month with no service interruption, and with CMS operations continuing unaffected on the Castor set-up throughout the copy process.

As from this set of testing experiences, CMS gave a very positive feedback on the new system, and agreed to migrate over to it. The GEMSS system went in production for CMS in late 2009, without major changes with respect to the layout used for the tests (the main change was the upgrade of StoRM to include checksum verification and improved authorization). As from the experience with GEMSS so far, CMS gives a very positive feedback on the new system adopted at CNAF: a very good stability for transfers and data accesses has been observed so far. The CNAF imports from all Tiers has recently scaled up to the level of ~50 TB/week on average (sustained since years). The CNAF exports to all Tiers has reached ~1.5 PB over last year as showed in the Figure 6. An overall good stability of the storage system is observed, and the system seems to be fully adequate to fulfill the challenges posed so far by the CMS data handling needs at a Tier1 level.
5.3. LHCb experiment

According to its computing model [26], LHCb performs a “near-online” distributed reconstruction of raw data. Data acquired by the detector are uploaded to CERN and then immediately distributed to the Tier1s. Raw data (RAW) and reconstructed data (SDST) are stored on tape at each Tier1. As a third category, data files used for analysis also go to tape storage for long term archival (ARCHIVE). It is important to keep for a very long time datasets that were used to produce published physics results. At present, a total of about 0.75 PB of tape space is in use (1 TB tape cartridges):

- RAW: 168 tapes (about 1.7 PB);
- SDST: 175 tapes (about 1.7 PB);
- ARCHIVE: 409 tapes (about 4.1 PB).

The tape storage part of the LHCb GEMSS setup is implemented with a limited set of resources. There are 2 dedicated HSM nodes, a maximum of 4 drives to be used for file pre-migrations, and a maximum of 6 drives to be used for recalls.

The tape stage area is in “sharing” with pure-disk areas. LHCb currently has 0.76 PB of disk in a dedicated GPFS filesystem, out of which only about 40 TB at maximum are guaranteed for the tape stage area. The rest is guaranteed for data usable for analysis and for users’ disk space. However, when there is free space available in the filesystem, the stage area is dynamically expanded by the system. For example at present there are about 120 TB of disk in use as tape stage area. Furthermore, even if the guaranteed stage area is quite small, it is distributed across all the disks of the filesystem, i.e. it always provides maximal throughput performance (about 2 GB/s with the current setup). This is

Figure 6. CNAF-outbound CMS traffic in the PhEDEx production instance, from CNAF to more than 70 CMS Tiers (each color is a different destination site), in the 12 months prior to May 2012.
a very important feature that avoids wasting a lot of disk space for the stage area only with the aim of providing a large sustainable throughput.

Figure 7 shows the cumulative data movements in pre-migrations and recalls and the instantaneous throughput to and from tape during the last 12 months. The system has demonstrated to be very flexible and stable for all the data access operations needed by the LHCb experiment. Basically no major interventions on the system have been ever needed.

Figure 7. LHCb experiment: (a) pre-migrate and recall cumulative data movements, and (b) instantaneous throughput during the last 12 months for pre-migrations (green) and recalls (red).

6. Conclusions
GEMSS is the storage solution used in production at the INFN-CNAF Tier1 as an integrated mass storage system for all the LHC and non-LHC experiments. It has demonstrated, from the user perspective and during these last years of production, to be high reliable and extremely stable, achieving the performances requested to a Tier1 site.

Latest improvements in GEMSS have increased its level of reliability and performance. In particular we are referring to the GEMSS optimized recalls and to the introduction of the GEMSS DMAPI server. Moreover new administrative tools and the integration with the Tier1 standard monitoring system have improved the management of the whole system, which automatically collects and creates activity reports, warning the system administrators in case of anomalous conditions. It is nowadays a matter of fact that GEMSS has represented a big step ahead, contributing in a prominent way to the consolidation and the stability of the global operation of the Tier1 center.

We are confident that the system will be able to fulfill the future storage requirements for the Tier1, in particular for what concerns the increasing demands of performance that applications will require during the next years.

References
[1] Bird I et al. 2005 LHC computing Grid Technical design report (CERN LHCC-2005-024) also available online http://cdsweb.cern.ch/record/840543/files/lhcc-2005-024.pdf
[2] IBM General Parallel File System Administration and Programming Reference Version 3 Release 2 SA23-2221-01
[3] IBM website references for Tivoli Storage Manager info and documentation https://www.ibm.com/developerworks/wikis/display/tivolidoccentral/Tivoli+Storage+Manager and http://www-01.ibm.com/software/tivoli/products/storage-mgr/
[4] Corso E et al. StoRM: A SRM Solution on Disk Based Storage System Proceedings of the Cracow GridWorkshop 2006 (CGW2006), Cracow, Poland, October 15–18, 2006.
[5] Tate J, Lucchese F and Moore R Introduction to Storage Area Networks SG24-5470-03 IBM
RedBook available online https://www.redbooks.ibm.com/redbooks/pdfs/pg245470.pdf

[6] Info about Lustre available online http://wiki.lustre.org/index.php/Main_Page
[7] Info about Castor available online http://castor.web.cern.ch/
[8] Info about dcache available online http://www.dcache.org/
[9] Info about xrootd available online http://xrootd.slac.stanford.edu/
[10] Bencivenni M et al. A comparison of Data-Access Platforms for the Computing of Large Hadron Collider Experiments IEEE Transactions on Nuclear Science, Volume 55, Issue 3, Part 3, pp. 1621–1630 (June 2008) ISSN: 0018-9499
[11] Info about StoRM available online http://storm.forge.cnaf.infn.it/
[12] Donno F et al. Addendum to the SRM v2.2 WLCG Usage Agreement. available online http://indico.cern.ch/getFile.py/access?contribId=14&sessionId=0&resId=0&materialId=paper&confId=6552
[13] Carbone A et al. Performance studies of the StoRM Storage Resource Manager Proceedings of “Third IEEE International Conference on e-Science and Grid Computing”, December 10-13, 2007, pp. 423–430 ISBN: 978-0-7695-3064-2
[14] Bonacorsi D, Gutsche O CMS Computing experience from the WLCG STEP’09 challenge to the first data taking of the LHC era International Symposium on Grid Computing 2010 (ISGC’10), Taipei, Taiwan, March 5-12, 2010
[15] IBM General Parallel File System Advanced Administration Guide Version 3 Release 2 SC23-5182-01
[16] IBM Tivoli Storage Manager for SAN for Linux Version 6.2: Storage Agent User’s Guide SC23-9799-01
[17] Systems Management: Data Storage Management (XDSM) API X Open Document Number: C429 ISBN: 1-85912-190-X Catalog number C429 UK ISBN 185912190X Available Online http://pubs.opengroup.org/onlinepubs/9657099
[18] IBM Tivoli Storage Management Concepts IBM Redbooks Series SG24-4877.
[19] Andreotti D et. al. INFN-CNAF Tier-1 Storage and Data Management Systems for the LHC Experiments proceedings di CHEP 2010 Taipei Taiwan under publishing Journal of Physics: Conference Series (JPCS) IOP Publishing.
[20] Info about Nagios available online http://nagios.org
[21] Info about LHC Era Monitoring available online http://lemon.web.cern.ch/lemon/index.shtml
[22] Andreotti D et al. INFN-CNAF Tier1 Storage and Data Management Systems for the LHC Experiments 2011 J. Phys.: Conf. Ser. 331 052005
[23] Jones R, Barberis D The ATLAS Computing Model J. Phys.: Conf. Ser. 119 072020
[24] Tuura L et al. Scaling CMS data transfer system for LHC start-up proc. of Computing in High Energy and Nuclear Physics (CHEP07), Victoria, BC, Canada, September 2007 - J.Phys.Conf. Ser.119 072030 (2008) doi: 10.1088/1742-6596/119/7/072030
[25] Bonacorsi D. et al. The CMS LoadTest 2007: An Infrastructure to Exercise CMS Transfer Routes among WLCG Tiers proc. of Computing in High Energy and Nuclear Physics (CHEP07), Victoria B.C., Canada, September 2-9, 2007
[26] Brook N. LHCb Computing Model LHCb Reference CERN-LHCb-2004-119, LHCC Reference CERN-LHCC-2004-036, LHCC-G-084.