Astronomical Data Analysis Software and Systems XIII
ASP Conference Series, Vol. 314, 2004
F. Ochsenbein, M. Allen, and D. Egret, eds.

Distributed Data Storage

Justin Kanoa Withington

Canada France Hawaii Telescope, Hawaii, USA

Abstract.
Distributed storage is the technique of storing a single data set across multiple hosts. This paper focuses on massive localized systems for distributed storage which are directed at overcoming the capacity and performance limitations of single-host, or monolithic, storage systems. Furthermore it will focus on applications in astronomy and draw examples from several existing astronomical distributed storage systems.

1. Methods

Several aspects of distributed storage systems are described. Where appropriate, examples from existing storage systems are given. For this paper distributed data storage systems at five astronomical facilities were surveyed.

Table 1. Example Systems Surveyed

Location	Aggregate Capacity	Number of Nodes
CFHT	18 Terabytes	12
CADC	55 Terabytes	21
SDSS	44 Terabytes	24
JAC	11 Terabytes	21
ESO	20 Terabytes	15

2. Cost

Central to leveraging the cost benefits of a distributed storage system is the relationship between the base cost of a host platform and the point at which it becomes impractical to add additional low cost local storage.

For example, the lowest cost class of random access storage device today is the IDE disk drive. As inexpensive disks are added to the base system, the cost per gigabyte will tend to decrease until capacity approaches four terabytes, at which point the cost per gigabyte is about $2.60. As capacity grows past this

point the cost per gigabyte starts increasing because the size of the disks and the number of attached disks begin to require specialized equipment.

This "sweet spot" depends on changing market conditions and can be periodically recalculated in an effort to determine the optimum node size.

3. Scalability

Scalability is perhaps the most liberating quality of a distributed storage system. Capacity can be dynamically added or removed and nodes are not limited to a previous design strategy so expansions can also take advantage of the moving sweet spot mentioned earlier.

A well designed system can scale in a perfectly linear fashion. That is to say that there is no upper limit to the maximum capacity of such a system. Furthermore, the cost/capacity relationship is also linear. Given the estimates of three terabytes at $2.60 per gigabyte we could estimate a 100 terabyte system today would cost roughly $260,000 and consist of about 25 nodes.

4. Data Tracking

Data tracking is simply the maintenance of a single point of reference for the location of all the elements in the data set. Two common ways of accomplishing this are through the use of a database and the use of a virtual file system.

A database provides a great deal of interpretive flexibility. A database also requires some kind of interface for clients to look up the location of data. Some software might need to be modified to the use the database interface.

A virtual file system symbolically represents the entire data set in a hierarchical fashion similar to a normal file system where each data point is a link to the actual data on a storage node. An advantage of a virtual file system is that users do not need to be retrained since the file access mode is familiar. Multiple virtual file systems can represent the same data set differently to suit particular applications. For example, one virtual file system might represent the data organized by instrument and another might represent the same set of files by program ID.

The choice of database or virtual file system depends primarily on the usage pattern of the data set. For example, the CADC is an archive and a self contained system in the sense that the only clients of the storage system are the CADC themselves and software written by the CADC to access the system. In this situation it is most reasonable to track the data in a database as it will be faster and more flexible for specialized interfaces. At CFHT, a diversity of engineers and pipelines use the storage system so a more traditional data representation is desirable.

5. Data Organization

Every distributed data storage volume consists of a collection of subvolumes located on the storage nodes. Each storage node may host one or more sub-

volumes. Data organization is the method used by the system to store the files across the subvolumes.

At one extreme the system might treat each file as a distinct data element and disperse them serially or pseudo-randomly across every available subvolume. On the other extreme, the system might only recognize classes of data as elements. For example, the system at SDSS tracks stripes of observations as directories. Under each directory is raw and processed data not directly managed by the storage system. JAC's WFCAM storage system organizes each extension of the camera onto a dedicated subvolume, with another dedicated subvolume for the processed data of each extension.

Such high level organization of data is attractive because it is less complicated to design and manage. It only works well however when the data itself is fairly well organized or homogeneous. It is less practical for a system which is managing diverse data types or data from multiple instruments.

Table 2. Data Organization

Location	Data Organization	Data Set Type
CFHT	file	heterogeneous
CADC	file	heterogeneous
SDSS	subdirectory	homogeneous
JAC	subvolume	homogeneous
ESO	file	heterogeneous

6. Redundancy

Some order of redundancy is a necessity in a massive storage system. Even if the data can be easily restored from an archive, the administrative overhead and operational downtime of responding to the inevitable failures can be impractical unless there is some layer of protection. The two most common techniques are data duplication and the use of internally redundant subvolumes.

In data duplication each data element is stored twice. The higher level system takes care that each copy is on a different subvolume and preferably on a different node. In this way the loss of any single subvolume and perhaps any single node will not result in the loss of any data from the overall collection. This is a simple and effective method but also costly, the storage system must be essentially twice the size of the data set.

One alternative is to make the subvolumes internally redundant, for example by using RAID level 5 arrays. Such an array can tolerate the loss of any single member disk with a usable capacity of $c(n\text{-}1)$ where c is the capacity of the smallest disk and n is the number of member disks. In other words the cost of redundancy decreases as the number of disks in the array increases.

This tendency of RAID level 5 arrays to be rather large is the basis of the cost/risk trade off. A data duplicating system will tend towards using the small-

est possible subvolume - usually a single disk, whereas an internally redundant subvolume will tend to be as large as possible - usually eight or more disks. While an internally redundant subvolume is less likely to fail, if it ever does fail the effect is more likely to be catastrophic.

The purpose of the storage system is usually the deciding factor. If it is an archive, as in the case of ESO and CADC, then data duplication is really the only appropriate method. The other systems are used for data processing and operate in parallel with an archive, for which purpose internally redundant subvolumes are adequate.

Table 3. Redundancy and Subvolume Type

Location	Redundancy Type	Subvolume Type	Primary Purpose
CFHT	internally redundant	RAID 5	analysis
CADC	data duplication	single disk	archive
SDSS	internally redundant	RAID 5	analysis
JAC	internally redundant	RAID 5	analysis
ESO	data duplication	single disk	archive

7. Conclusion

While the technique of building storage clusters at this scale is relatively new and the underlying technology in a state of rapid change, sound design methodologies can be developed. Existing systems provide examples of design considerations and a proof of concept for future systems.

Acknowledgments. The following individuals graciously contributed data for the examples used in this paper:

Dan Yocum, Sloan Digital Sky Survey

Nick Reese, Joint Astronomy Centre

Andreas Wicenec, European Southern Observatory

Severin Gaudet, Canadian Astronomy Data Centre