Multiple Resident AFS Fileserver at Garching

Hartmut Reuter, Computing Center of the Max-Planck-Gesellschaft (MPG) and Institut für Plasmaphysik (IPP), Garching bei München, Germany

Introduction

Last year at Tours I presented IPP's brand-new Cray-EL fileserver [1] which at that time was undergoing its acceptance tests. Now, a year later, this server is in full production and I can present our experiences.

The fileserver consists of a Cray-EL with the following equipment:

Processors	4
Memory	64 MW
IOPs	5
HiPPI interface	1
fast disks	20 GB
SCSI disks	50 GB
ER90 tape drives	4
GRAU ABBA/E robot	1
D2 cartridges	502
Total Capacity	37 TB

The robot system can be upgraded to 2700 cartridges or 200 TB total capacity.

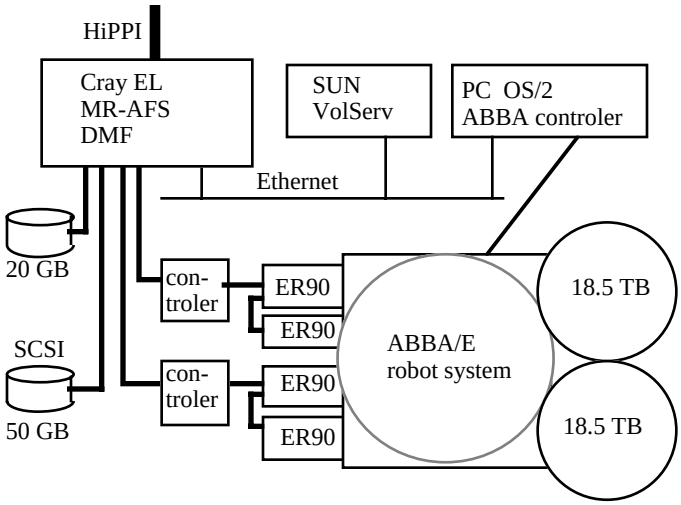

Figure 1: Fileserver Configuration.

The Cray-EL runs with the following software releases:

Unicos	8.0.4
DMF and MSP	2.2.3
MR-AFS	3.3a

The only user application running on the EL is a program called "arc" which allows the backup of the local filesystems of workstations into DMF.

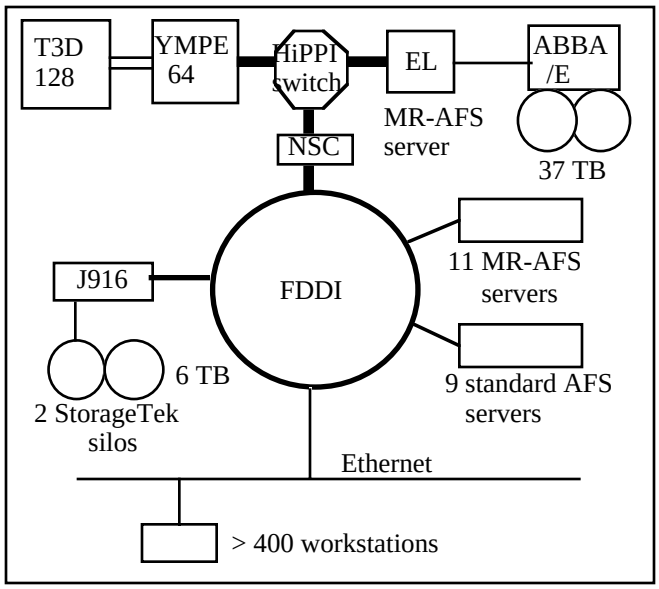

Figure 2: Network Configuration at IPP.

Multiple Resident AFS

Our site has a rather heterogenous workstation environment, along with some Crays (a YMP attached to a T3D, a Jedi and the fileserver-EL). The only transparent way to share data between all these machines is AFS (DFS being not yet available on all our platforms).

The issue of data migration is not covered by standard AFS. Therefore we are using Multiple Resident AFS - a set of extensions to the standard AFS source code written at Pittsburgh Supercomputer Center by Jonathan Goldick, Chris Kirby, Bill Zumach and others [2][3][4].

"Multiple Resident AFS" (MR-AFS) is presently the only distributed filesystem which provides distributed data migration. By "distributed data migration" I mean that any fileserver in the cell may migrate data to and from any archival server.

By means of a database service all MR-AFS fileservers know about all shareable resources such as disk space or archival storage. These shared resources are administrated democrati-

cally by all fileservers. That means the resource itself consists only of a tiny server process which provides remote access to the data stored there, while all "intelligence" such as data management, space management and recovery procedures are left up to the fileservers.

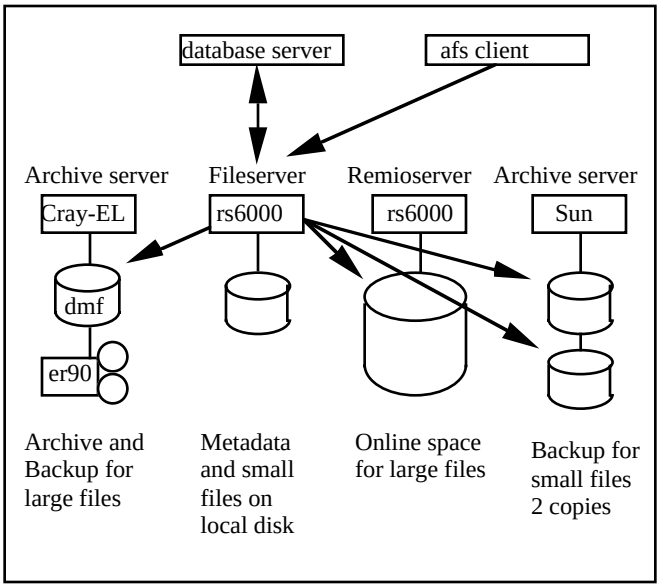

Figure 3: Fileserver storing files on different residencies.

This concept does not cover the classical data migration between disk and tape. This classical data migration is considered as an extra layer which must be provided by the operating system of the machine which hosts the archival server. For the classical data migration therefore standard solutions such as DMF in the case of Cray or Unitree or Epoch can be used.

The independence of these two data migration layers allows all of our presently 12 MR-AFS fileservers to offer data migration, not just the server on the Cray-EL.

On the other hand DMF is not the only archival software we use. For the small files (< 64 KB) where migration onto tape would be mainly overhead we offer a disk to disk migration making use of some cheap and slow 9 GB disk drives.

Experiences

The fileserver has been in real production since January. We configured two DMF partitions, one used by MR-AFS the other one by "arc", to backup the workstation disks and also the AFS home directories of the users.

During the first five months of production at times the server ran not as smoothly as we had hoped it would. We saw a number of hardware and software problems which reduced the availability of the system to a sometimes unacceptable degree. But with so much new hard- and software the beginning is probably never that easy.

For about three months the system has been running much more stably. The performance, however, is still not satisfactory, at least for the MR-AFS fileserver. But after getting the system stable we are confident we will solve the performance problems too during the next half year.

The DMF partitions contain now (migrated) files with the following total sizes:

MR-AFS-partition	0.7 TB
"arc"-partition	1.7 TB

With two tape copies per file we now have at least 4.8 TB valid data on the tapes, but probably more because there are many soft deleted files in the DMF partitions. The data in the "arc"-partition are much greater than the contents of the workstation partitions that were backed up because we presently keep multiple versions to allow for restoring old file versions, too.

In the following I am going to describe the problems and experiences we had with the different susbsystems. There were a lot of problems in the beginning as you will see, but I must say that all our partners were very responsive in solving them.

The Cray-EL

The Cray-EL ran much less stably than the old YMP we have. During the year a lot of boards and power supplies were replaced. For several months now the system has been stable.

The Robot System

Initally the fileserver was equipped with the EMASS DataLibrary as a robot system. During the acceptance tests a year ago, however, this robot system turned out not to be reliable enough. The frequency of failures grew from week to week and after a while EMASS realized that the DataLibrary would never become reliable enough to pass our acceptance test. Therefore they replaced the DataLibrary by an EMASS/GRAU ABBA/E system, which was easy for them since they had just bought the majority of this German robot company .

This new robot system turned out to be extremely reliable - we haven't had a single failure since it was installed - and in addition it is much faster than the DataLibrary was. Another interesting feature of the GRAU robot system is that you can store a large number of different media types in the same robot system, from tiny DAT tapes up to our 75 GB D2-tapes.

The ER90 Tape Drives

We have seen transfer rates seen up to 18 MB/s for big files during dmget operations. Thus the promised 15 MB/s sustained transfer rate is realistic for the ER90 drives.

Initially we had bought only two ER90 drives because of their very high price. Unfortunately these drives turned out to be rather delicate. So it often happened that one of them was down.

In October last year we saw a very dangerous error: one of the servo-engines which moves the head had an electronic failure with the result that the data written by that drive were not readable any more. The read after write check had been successful because both heads were displaced together.

As of February we have had two additional drives. They were necessary to allow for a real production environment When we had only two drives each of them had its own interface

in the IOS system. Now two of them are attached to each interface using daisy-chaining. There was a problem in the IOS software which sometimes caused device interrupts to be forgotten. This problem has recently been solved, too, so that now we see a really stable production environment with respect to the ER90 drives and the robot system.

The HiPPI Interface

We experience problems with the data transfer over the HiPPI whenever one of the Crays on the HiPPI switch is down (before we got the Jedi we had another EL at the HiPPI switch). This seems to be a weak point in the HiPPI protocol, which allows for switching to an interface that had been configured down by the host it belongs to. These dead connections have a certain timeout, but during that time all other traffic is blocked and a lot of packets are lost.

DMF

The old version of DMF could start writing only at the beginning of a tape. With the 75 GB D2-tape cartridges this would have led to a tremendous waste of capacity because the disk partitions we have are much smaller than these 75 GB. Therefore it was necessary to partition the cartridges to 70 virtual tapes of one GB each.

The old MSP was not smart enough to understand that it is not necessary to dismount and mount the cartridge in order to switch from one virtual tape to the next on the same real cartridge.

The new DMF and MSP are much better adapted to ER90 tapes. The hit rate for finding subsequent dmget requests to be served from the same tape without a dismount in between is now, of course, much higher.

We are running DMF with two independent MSPs in order to get two tape copies from each file. The hardware failures on the ER90 drives and some errors in the IOS software brought on a situation where we lost the contents of some tape partitions. It turned out that the old DMF didn't have any convenient way like the "dmmove" in the new DMF to recover from such a situation. There was therefore a lot of hand-work to be done. However, the support by Cray was great: they did all the tedious work for us.

In a fileserver system with hundreds or thousands of dmget requests per day a kind of "fair share scheduler" for dmget requests is required. Queueing techniques known from batch environments should be used to guarantee reasonable response times for single dmget requests even if someone else has put hundreds of requests into the queue.

Another urgent requirement is more transparancy. It should be possible to view the queue in order to estimate the waiting time and - for the system administrator - to understand better what load DMF currently has. I admit that in the case of MR-AFS requests to DMF this still wouldn't give full transparency, because here the server processes running under root would appear as requestors rather than the individual AFS users. To solve this problem some extensions to MR-AFS would be necessary, too.

MR-AFS

The support of the MR-AFS software by the Pittsburgh Supercomputer Center was excellent. You could inform them of a problem in the morning and in the evening they sometimes had a solution. The only limiting factor was the time difference between Europe and the PSC which caused the response to a request sent in the morning not to arrive before the afternoon.

Of course such a giant software system has a lot of errors in the beginning. Standard AFS servers are installed in some hundred sites, MR-AFS software only in a hand-full. It also turned out that we were the first ones to really heavily use the system. So a lot of errors showed up at Garching for the first time.

We still have a number of unsolved problems, but over all the MR-AFS software is quite stable. The medium time between fileserver crashes is more than a month and such a crash generally has an effect only on a small part of the whole AFS cell.

But in the first half year of production we had some severe problems which also caused data losses. There was a bad error in a recovery procedure, which instead of healing ill AFS volumes (what in DFS is called file sets) resulted in disappeared files. Such errors come up only in an unstable environment, when recovery is likely to be initiated.

Another error which caused some files to be damaged was due to a fix for a problem caused by Unicos. The AFS client software unfortunately sends the contents of a file in reverse order, the last chunk first. That means that a big file starts with a giant hole which later will be filled. Unlike all other known Unix systems Unicos really writes this hole with zeroed blocks. This caused the first write to take up to more than a minute and led to time outs on the RPC-connections. Unfortunately the work-around for this problem had a small error which damaged about 30 files before it was recognized.

The Backup Concept

In standard AFS, where all files of a volume (file set) are stored in a local disk partition, backing those data up is relatively easy. With MR-AFS the files may be stored on remote shared disks or in DMF, too. Therefore a full backup of all data belonging to a volume generally is not possible unless you want to wait for all the files in DMF to come back onto disk.

In order to protect the files against loss due to disk crashes, at least one additional copy is made of each file after a delay of not more than an hour. Small files (< 64 KB) are copied on two low cost disk partitions and bigger files are copied into a DMF partition on the Cray-EL. Therefore only the metadata of the volume and those very new files which haven't already been copied need to be included in the nightly backup. These dumps are themselves put into DMF.

In order to allow for a consistent restore of a volume after a disk crash this kind of nightly dump is still not sufficient. The reason is that between a dump was made and disk crash occurs, the user might have removed or overwritten files which are referenced in the dumped metadata. Therefore the volumes have

to be cloned at the same time the dump is made in order to increment the file reference counts for the files stored in DMF or on the cheap disks. Only the next night, when the procedure is repeated, those file references on already deleted files get decremented and the files really disappear.

What we have learned is that a consistent backup of a distributed filesystem with distributed data migration is quite complex and needs a lot of planning. But on the other hand, it's only a small amount of data which needs to be dumped each night (less than a promille).

Performance

Our main intention during this first year of tests and production with the new fileserver was, of course, to get things going. Therefore performance is going to be investigated only now that the basic needs have been satisfied.

There are different aspects of performance. One which is very important for the experiments at our institute is the transfer rate at which big files can be read or written from or to the fileserver. The Cray-EL fileserver itself is equipped with a HiPPI interface which is connected through a HiPPI-switch to our FDDI networks. The best performance values we see are between IBM rs6000 workstations as clients and servers. Depending on the chunk size defined on the server we see values of up to 1.5 MB/s for writes and 3.0 MB/s for reads. Since the Cray-EL has the same kind of SCSI disks and a faster network interface the transfer rates should not be worse. But unfortunately they are. This is not a general result for Crays: a test fileserver on our YMP shows a performance which is about the same as on the IBM rs6000s. Therefore I am hopefull that we will findthe bottleneck on the EL.

Another aspect of performance is how long it takes to get a file back from tape. The giant data rate of the ER90 drives does not help very much because most files are not big enough to make the data transfer itself the most important part of the response time.

The startup time is very long: it takes about fifteen seconds until the robot starts to move. This time is probably spent while three databases are inspected: the DMF database, the VolServ database on the SUN system, and finally the ABBA database on a PC under OS/2. With our old fileserver [5] which does not need those database accesses the mount request starts immediatly within a second after a file is requested. The throughput of our system with four ER90 drives turns out to be in the order of 100 dmget-requests per hour. The average wait time is about five minutes and may become much longer, whenever the queue grows faster then it can be served.

Another experience is that the performance of DMF can suffer badly on fragmented databases. Therefore it is very important to compress the database frequently which on the other hand is not possible without interrupting the service.

Future Plans

One of the goals we had was to provide users with infinite disk space. We first moved the home directories of our users to MR-AFS fileservers where inactive files could be migrated. The result was not convincing: in a Unix environment most files are much too small. Either you don't migrate them at all, then you better keep the home directories on Standard AFS servers, or you will hear lots of complaints about blocking commands and unacceptable response times.

As a result we soon returned the user's home directories to non-migrating fileservers. Instead, we offered each of them a secondary volume in a migrating tree. But the users also put too many and too small files here. That was fine as long as they only stored their files, but whenever they tried to get them back they had to wait incredibly long time because they had induced thousands of dmget requests. It is a matter of education to convince them to "tar" their thousands of small files before putting them into a migrating system.

Therefore we will probably change our policy again and allow data migration only for files larger than 20 MB. In terms of MR-AFS that means we have to separate large and small files into different residencies with different migration policies. We are still in the very early stages with this kind of tuning. But it is clear that here is a wide area for improvements.

The StorageTek silos which presently are filled with the tapes belonging to our old IBM /370 based fileservers are already attached to the Jedi. As soon as the data from the old fileservers are copied into MR-AFS this capacity can be used by MR-AFS. Then we will probably increase the minimal size limit for files which get their archival residency on the ER90 tapes in order to make better use of the high transfer rate of these devices.

For the smaller files the tape mount time is the limiting factor. The cheaper StorageTek drives will therefore allow us to run more dmget streams in parallel.

Probably in the far future the question of the fileserver software to be used will have to be evaluated once more. At least in the present situation it is not clear whether the Pittsburgh Supercomputer Center will be able to give the software support any more after the authors have left PSC this summer. For the current release which now runs quite stably we do not see huge problems because we already have achieved some knowledge of the program structure which may be sufficient to analyze and correct errors. But a port of the software to some future release of AFS may be to big a step for the man-power we have. Therefore the requirement for a "Multiple Resident DFS" or some thing equivalent remains very important.

References

[1] Hartmut Reuter, "Super AFS Fileserver at Garching", *CUG Proceedings*, October 1994

[2] Daniel Nydick, Kathy Benninger, Brett Bosley, James Ellis, Jonathan Goldick, Christopher Kirby, Michael Levine, Christopher Maher, Matt Mathis, "An AFS-based mass storage system at the Pittsburgh Supercomputer Center", *Proceedings of the Eleventh IEEE Symposium on Mass Storage Systems*, October 1991

[3] Jonathan S. Goldick, Kathy Benninger, Woody Brown, Christopher Kirby, Christopher Maher, Daniel S. Nydick, Bill Zumach, "An AFS-Based Supercomputing Environment", *Proceedings of the Twelfth EEE Symposium on Mass Storage Systems*, April 1993

[4] Christopher Maher, Jonathan S. Goldick, Christopher Kirby, and Bill Zumach, "The Integration of Distributed and Mass Storage Systems", *Proceedings of the Thirteens IEEE Symposium on Mass Storage Systems*, June 1994

[5] Hartmut Reuter, "The HADES file server", *Proceedings of the Eleventh IEEE Symposium on Mass Storage Systems*, October 1991