System for Backup on Redundant Fileservers (Sistema di Backup sui Fileserver Ridondanti)

Carlo Podda e Alan Scheinine, Area HPC, CRS4

CRS4
Centro di Ricerca, Sviluppo e Studi Superiori in Sardegna
Polaris, Edificio 1
Loc. Pixina Manna
09010 Pula (Cagliari), Italy

E-mail: carlo@crs4.it, scheinin@crs4.it

May 2005

Abstract

A pair of computers have been configured as redundant file servers. These computers are also used to backup files. User files, exported using NFS, are placed on fast hard disks (type SCSI), whereas much larger, though slower hard disks (type SATA) are used for backup of these files. This technical report describes the implementation of the system of file backup.

Un paio di computer sono stati configurati come file server ridondanti. Questi computer sono usati anche per il backup (archiviazione) dei file. I file degli utenti, disponibili ad altri computer tramite NFS, risiedono su dischi (hard disk) veloci (tipo SCSI), mentre dischi molto più grandi però lenti (tipo SATA) sono usati per il backup di questi file. Questo rapporto tecnico descrive l'implementazione del sistema di backup dei file.

Contents

1 Introduction

Two computers of type ‘server’ (dual CPU, a large quantity of disk space, Gigabit Ethernet) were installed in the Arcosu cluster in a redundant configuration. The pair provides redundancy for services such as Sun Grid Engine, as well as redundancy for the file systems exported with using NFS. In addition, this same pair of servers performs backup of both user files and software packages. For user files the backup is done incrementally with daily snapshots, whereas for software packages only one backup copy is kept rather than a series of snapshots. This all-in-one solution in which various functions (system services redundancy, redundant file servers and backup) are hosted by just two servers was chosen in order to minimize the cost.

Rapid access of users’ files, exported using NFS, is obtained by using RAID 0, by channel-bonding two Gigabit Ethernet connections on each server, and by distributing the user files between both servers. If one server is not available, then a copy of the user files originally hosted by the unavailable server are exported from the remaining server.

2 Hardware

The user files are exported using NFS. In order to provide high-speed disk access, on each server the user files are on a pair of disks that use the SCSI protocol, configured as RAID 0 (stripping). Moreover, user files are divided between the two servers (partition ‘/els1’ on server ‘aiodda1’ and partition ‘/els2’ on server ‘aiodda2’) in order to allow a job to benefit from the bandwidth of two servers when using more than one file. On each server the partition /elsN has a size of 141 GigaBytes. The servers are connected to a Gigabit Ethernet switch using channel-bonding (trunking) of the two Gigabit channels available on each server.

If one server of the pair is not available, then both partitions /els1 and /els2 are hosted by the remaining server, though one of the partitions will be one day out-of-date, having the data from the nightly backup. In addition to the two SCSI disks (of 73 GB each) configured in RAID 0, there are four 73 GB SCSI disks configured in RAID 5 (redundancy using parity). These four disks host the operating system and software packages such as PGI, MPI, FFTW, StarCD, R, DX, and many other programs. The software packages use 30 GB of disk space. This system does not provide very high reliability for user files due to the fact that backups are done just once per day. On the other hand, this file space is intended to be used for short-term storage of user data.

The disk space available for a job can be viewed as being comprised of three types with differing levels of long-term reliability. The level with the fastest access is the local disk on each node, which provides scratch space. For newer nodes the size of scratch space is 160 GB. This file space is not exported using NFS because users tend to park data on nodes and with many nodes some will surely have disks that fail during the year. The middle level consists of /els1 and /els2 on the two servers. These file systems are for user files during program development and testing. For important data and programs, the user must arrange to have space assigned on high quality file servers available at CRS4. The file systems /els1 and /els2 provide work space for storage of files during an interval of a few days to few months. Each user and each group has a disk quota in order to avoid having the file servers aiodda1 and aiodda2 become archival storage for user files. It is expected that one of these file servers will fail suddenly (and will need to go off-line for repair) at most once every two years, so that the loss, about once a year, of data being generated by running jobs will be tolerable. The amount of user space, 282 GigaBytes for the two servers combined, is not large when one considers that each computational node has 160 GigaBytes. The limitation in disk space at the file server is due to the cost of SCSI hard disks, about ten times the cost of SATA hard disks. Though SATA II with native command queueing (NCQ) may prove to be as fast as SCSI (depending on the device controller), at the time the servers where purchased, SCSI was faster.

The backup is done to SATA hard disks, four disks of 250 GB each configured in RAID 5 are installed on each server. An important consideration is the storage space multiplication factor due to backups done as daily snapshots. On each server, the space available for backup of user files is 288 GB. With 141 GB of user files on each server, if every user changed their directory contents each day, then the backup could have at most two snapshots. As well as backup of user files and software packages, the SATA disks hold anonymous ftp (including distributions that can be installed using “kickstart”) and backup of other servers of the Arcosu cluster, and as a consequence, the SATA disks are fully used.

For various clusters at different sites, the details of the sizes of the file systems would vary, depending on specific needs. Nonetheless, the choices made for these two servers are a useful an example. At the time of the preparation of this report, SATA disks of 400 GB have become available for a cost similar to the price we paid for 250 GB disks a year ago. But on the other hand, if user files change often and if the number of snapshots desired is closer to seven rather than two, then the amount of space being backed-up must remain similar to this implementation. Why not consider more disks? The cabinet used to hold the ten disks has the dimensions of a rack U5 cabinet, whereas, larger cabinets are unwieldy and relatively expensive. The configuration described here can be a cost-effective size for one “unit” file server with backup.

Though there are other, more flexible solutions for assigning disks to servers, such a fiber channel or iSCSI, if in the end the file access is by way of NFS, then one can consider the balance between disk bandwidth and Ethernet bandwidth. For the SCSI disks purchased for this project, U320 10.000 RPM, the read bandwidth for a pair configured as RAID 0 had a maximum of 122 MB/s, which corresponds to the peak bandwidth of Gigabit Ethernet. The actual servers have main boards with two Gigabit Ethernet channels combined using channel-bonding and have a pair of SCSI disks in RAID 0 for the user files. But in addition, there are four SCSI disks in RAID 5 that host the software packages for the users, and when reading from disk RAID 5 can be as fast as RAID 0. So from the point of view of the Ethernet bandwidth, this configuration can be considered a balanced choice for a "unit" file server.

3 Filesystem Partitions

An overview of the partitioning of the hard disks into file systems will be used to describe the backup procedure. The file systems are shown in Fig. 3.1.

The partition `/BackupS` is for backup of files important for the system administrators. The cross backup of the root directory `/` in Fig. 3.1 refers to specific subdirectories whose contents might be different on the two servers, for example, `/etc`, `/usr/local` and `boot`. The partition `/BackupS` also contains backup copies of other file systems on the Arcosu cluster, such as the data on our web server.

The partition `/Archive64` contains software packages compiled for the 64-bit Opteron CPU. These software packages are used by all Opteron-based nodes on the Arcosu cluster. A complete copy of `/Archive64` is made each night on the twin server.

The partition `/Vari` contains various file sets. For example, it contains the anonymous FTP directory which is used for remote installation of operating systems. As such, the anonymous FTP directory contains various distributions of Linux. This partition also contains software documentation and installation notes. Most of the software documentation is mirrored on the HPC web server to provide access for the users of the cluster. A complete copy of `/Vari` is made each night on the twin server.

We see in Fig. 3.1 that the arrows that indicate the direction of backup are in opposite directions for `/Archive64` and `/Vari`. The server that NFS exports `/Archive64` is different from the server that provides ftp access to `/Vari` in order to distribute the load.

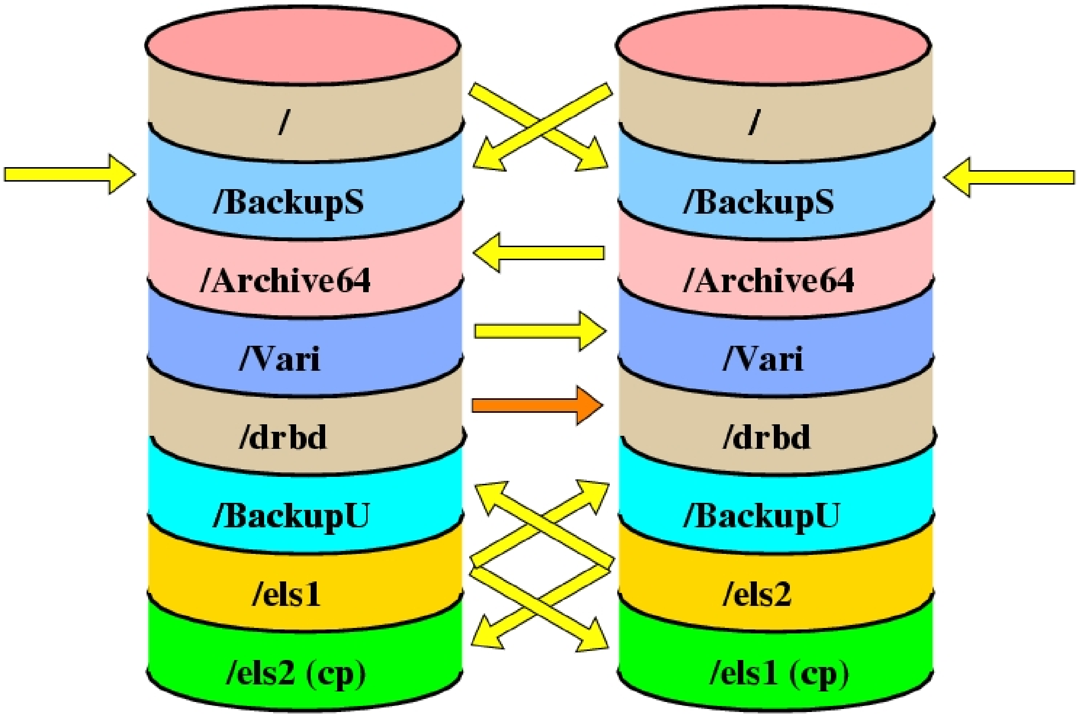

Figure 3.1: File systems and backup data flow. The partition `/BackupS` is for backup of files important for the system administrators, in particular, the configuration files of the other twin as well external file systems such as the data on our web server. The partition `/Archive64` contains software packages compiled for the x86_64 architecture, which is duplicated on the other twin. The partition `/Vari` contains various groups of files such as documentation and the contents of anonymous ftp. The partition `/drbd` provides real-time mirroring using the drbd software package. The partition `/BackupU` contains daily snapshots of user files. The backup is done between two servers so that if one server dies, the other server has the backup information. There is also a cross backup of user files that is a single copy rather than daily snapshots, shown at the bottom of the figure.

The partition `/drbd` contains a file system that is mirrored in real time on the second server. The software DRBD is described at it WWW site (http://www.drbd.org/) as follows.

> DRBD is a block device which is designed to build high availability clusters. This is done by mirroring a whole block device via (a dedicated) network. You could see it as a network raid-1.

Only one copy of the drbd file system is writable at a given time; upon failover using the software Heartbeat, if the primary, writable copy becomes unavailable then the write-only copy become writable. Presently at version 0.7, DRBD cannot be considered of production quality. On the other hand, in our experiments DRBD has proven to be reliable. It is used for the file-based state of the queue system for the cluster, Sun Grid Engine (SGE). For recording the state of SGE either a real database can be used or a set of files can be used. The latter can be copied or mirrored, hence is suitable for use on the `/drbd` partition. Using the software Heartbeat to signal when the primary SGE server is down and the SGE server on the second machine must be activated, the /drbd file system on the second machine is also changed from being a mirror to being writable.

The partition `/BackupU` contains the backup of user files, in particular, daily snapshots of the files in `/els1` and `els2`. The backup, done during the night, crosses between the servers in the sense that `/els1` on `aiodda1` has the backup written to `/BackupU` on `aiodda2` and vice versa. So that if `aiodda1` goes down then a copy of `/els1` remains on the other server in `/BackupU`. In addition, there is a cross backup of user files so that the server aiodda2 has a copy of `/els1` in `/els1 (cp)`, and vice versa. If one server goes down then the other server exports the copy from the faster SCSI hard disks, though from the collection in RAID 5 which is not as fast as the collection in RAID 0 used for the original locations of `/els1` and `/els2`. The copy `/els1 (cp)` replaces `/els1` and is writable by the users whereas the backups in `/BackupU` are read-only daily snapshots.

Though not shown in Fig. 3.1, if `aiodda1` is not available and `/els1 (cp)` is exported, then daily backups continue to be made to `/BackupU` on `aiodda2` from `/els1 (cp)` rather than from `/els1` on `aiodda1`. The same logic applies to `/els2 (cp)` when `aiodda2` is not available.

In addition, if `aiodda1` becomes unavailable then later becomes available, if the time of absence was more than a certain interval (now set at 180 minutes), then the contents of `/els1 (cp)` on `aiodda2` are written to `/BackupU` in directory
`/BackupU/els1/partial_save/`*date*, where *date* is a file name that indicates the date and time at which the copy was made. When `aiodda1` becomes available it takes-over the export of `/els1` and `/els1 (cp)` is no longer visible. The file system `/els1 (cp)` on `aiodda2` is not copied to `/els1` on `aiodda1` even though `/els1 (cp)` might have been changed by users if `aiodda1` had been down for a long period of time. Since, `/els1 (cp)` is a nightly copy of `/els1` and not a real-time copy, copying it back to `/els1` could cause the loss of some files. Copying `/els1 (cp)` back to `/els1` might create confusion rather than simplify the situation. The copy in
`/BackupU/els1/partial_save/`*date* is available to the users as a read-only file system so that users can choose files from `/els1` before `aiodda1` went down or from
`/BackupU/els1/partial_save/`*date* that contains a copy of `/els1 (cp)` at the moment `aiodda1` returned. The same logic applies to `/els2` when `aiodda2` is not available. The choice of 180 minutes is motivated by the desire to avoid the backup of `/els1 (cp)` when a server goes down for a short period of time, for example, for minor changes in the

configuration. Of course, rebooting a file server for changing the configuration or other maintenance would be done during a programmed maintenance period, that is, when the users have been advised in advance of the work on the file server.

Acknowledgments

This work is partially supported by MIUR, (the Italian Ministry of Instruction, University, and Research), project 212 of the request for proposals D.D. 1105 of 9 October 2002. Additional support is provided by the Sardinian Regional Authorities.

A Incremental Backup Using Rsync

The transfer of files is done using "rsync", aside from mirroring of one disk partition with the software "drbd". The source for rsync is
`http://rsync.samba.org/`
Recent versions of rsync can implement an incremental backup by using hard links (available on a Unix file system). A hard link uses very little disk space. Each backup looks like a complete snapshot, however, if a file has not changed between different snapshots, then a hard link is created rather than copying the file. For this mode, the rsync command needs a reference snapshot at the destination in order to determine if the file already exists at the destination. Two sources of information for creating an incremental backup using rsync are the following:
`http://www.mikerubel.org/computers/rsync_snapshots`
`http://www.pollux.franken.de/hjb/rsback`

At the present time (May 2005) the incremental backup is done once daily, during the late night hours. There is no special tool for restoring files from backup, instead, the users have read-only access to the backed-up files. For example, all files in the directory /els1 are backed-up. ('els' signifies 'embarrassingly large scratch'.) The directory /els1 is exported using NFS and is hosted by fast disks that use the SCSI protocol. The backup region for these user files is /BackupU, which is hosted by high capacity disks that use the SATA protocol. Under the directory /BackupU/els1 there are subdirectories daily.0, daily.1, daily.2 etc. with daily.0 being the most recent. Under daily.N one sees the same directory hierarchy as one sees under /els1. The directory /BackupU is exported using NFS as a read-only file system. To recover a file the user would need to look under /BackupU/els1/daily.0, /BackupU/els1/daily.1, etc. in order to find the version that the user considers most appropriate to recover; the file is then simply copied.

One detail concerning disk writing is that twice we have had a hard disk of type ATA break after an hour of writing continuously using rsync. We now use the bandwidth limiting option 'bwlimit' to reduce the disk's duty cycle.

An example of a script that we use for incremental backup is shown below.

```
#! /bin/bash

bwlimit=2400
rsyncexec=/usr/local/Backup/bin/rsync
src_computer=’root@aiodda1:’
src_directory=’/els1/’

finalvalue=5
iter_value=$finalvalue
archiveroot="/BackupU/els1"

origin="${src_computer}${src_directory}"
logfile="/usr/local/Backup/log/on_aiodda2_els1"
interval=daily

rm -rf ${archiveroot}/${interval}.${finalvalue}

while test $iter_value != 0
do
  oneless=$(($iter_value - 1))
  if test -d ${archiveroot}/${interval}.${oneless}
  then
    mv ${archiveroot}/${interval}.${oneless} \
      ${archiveroot}/${interval}.${iter_value}
  fi
  iter_value=$oneless
done

echo "‘date‘: starting rsync" >> $logfile
cmd="$rsyncexec -avz --delete \
  --exclude=’/lost+found’ --bwlimit=$bwlimit \
  --link-dest=${archiveroot}/${interval}.1 \
  ${origin} \
  ${archiveroot}/${interval}.0/"
echo "Command: $cmd 2>> $logfile" >> $logfile
$cmd  2>> $logfile

if test $? != 0
then
```

```
   echo "‘date‘: rsync finished with error(s)" >> $logfile
   exit 1
else
   echo "‘date‘: rsync finished" >> $logfile
fi
exit 0
```