

Your data, our responsibility

Robert C Bell, Jeroen van den Muyzenberg, Peter Edwards

¹CSIRO Advanced Scientific Computing, Melbourne, Australia, Robert.Bell@csiro.au

INTRODUCTION

High Performance Computing (HPC) centres become holders of high volumes of data. Although computation is the initial focus, the users become somewhat attached to their data, and woe betides the computing centre that loses users' data. Users would like unlimited storage, and with high performance, visible from everywhere, secure from loss, and at zero cost.

HPC centres should meet the users needs within constraints of budget and technologies available, providing highly reliable storage, with high performance and sufficient capacity. But it is a little like the old truism: "you can have good, cheap or quick" – choose any two, but you can't have all three at once. In the same way, centres cannot provide storage with high capacity, high performance and low cost in the one system. Other considerations arise – reliability/ resiliency/ recoverability/ recovery time / bandwidth and i/o operations per second.

Centres have to make backups of files to provide some protection against loss. The facilities they provide and the policies that they apply to users' data affect the productivity of the users and their perception of the service.

This paper will outline some issues with data storage at HPC centres. It then addresses the need for backup and various techniques used. It will finish by showing how the rsync command can be used to provide full backups on every cycle at the cost of incremental backups, and how the Tower of Hanoi scheme can be used to automatically manage backup sets at appropriate intervals back in time. The techniques together provide a reduction in the storage needed for backups by factors of about five over conventional schemes, while providing superior coverage.

FILES SYSTEMS – POLICIES

File systems are the cornerstone of the storage facilities at HPC centres, along with the management strategies for them. A range of storage systems is provided to meet different purposes within the constraints upon the facility.

The storage facilities and policies can be a major factor in users' perception of the service. Too little space, or policies that are too restrictive can make the users work untenable. Having too many locations that a user has to deal with can make them unproductive, in having to copy and move data around. They don't have to do much of this on their personal systems – why is dealing with the HPC centre and its storage facilities so much harder and less productive of their time? A major example is the compute-centric HPC centre, where the target of user and management's attention is the computing platform, and data has to be explicitly moved from an archive to an intermediate platform and into the HPC system for input, and the reverse process has to be carried out for the output data. Our experience has been that the transport processes are some of the least reliable steps in the chain. Some centres provide less storage in users' home areas than users have on their own desktop or laptop systems – why?

For data-intensive users, how much better it is to provide a platform where they can directly work with the data, possibly across an automated tier of storage media, and take explicit action if needed to send computational tasks to the compute platform. This data-centric model has much to recommend it.

By way of example, on the CSIRO ASC SGI Altix system, which hosts the CSIRO Data Store, we provide a areas pointed to by shell variables \$HOME, \$WORKDIR, \$DATADIR, \$TMPDIR, as well as special areas for shared datasets. Each area is subject to management by various techniques: backup, quota, job-temporary, flush and migration.

It is essential to have a policy for management of every file system, and to make this known to the users. Some sites require users to sign assent to these policies.

HSM

What is unusual about the CSIRO ASC set-up is that \$HOME area is virtually infinite, because it is hosted on a Hierarchical Storage Management (HSM) system; we use SGI's Data Migration Facility (DMF), and this has been in use since 1991. More typical is an arrangement where the \$HOME area is quite small, with the size constrained by the capacity of the Centre to back it up.

CSIRO had an HSM system called the 'Document Region' on a CDC 3600 more than 35 years ago. The idea is that when file systems are filling, the HSM copies data from the primary storage area to secondary media (typically magnetic tape), and removes the data but not the metadata from the primary storage area. When a user or application needs to access the data, the data is automatically restored to the primary storage area.

Although an inconvenience to users, in that not all their data is immediately accessible, HSM has many advantages. HSM allows users to have access to vastly more data than they could have otherwise. It can lower the cost of storage. In these days of green issues, HSM can save energy – there is no power required to maintain storage on tapes, unlike the costs of running discs. HSM allows users to 'see' all of their data holdings where they work, rather than have to deal with a separate 'archiving' system. HSM with the secondary copies on tape can provide a greater level of protection; two or

more copies can be easily kept, and no system administrator action can remove data from an off-line tape: the same is not true for on-line disc.

HSM allows transitions between sites and machines to be carried out. CSIRO ASC and predecessors have moved the data four times between four sites and five hosts. Users have seen the 'same' \$HOME file system since 1990.

Finally, and this is most important in the context of this paper, HSM can get around the backup problem for large file systems. HSM allows the data to be trickled to secondary media. When a dump of an HSM-managed file system is done, then only inode information and the data of files that have not yet been written to tape need to be captured. For the CSIRO ASC Data Store file system, which in July 2009 contained 850 Tbyte under HSM management with 15.5 million inodes, the daily incremental dumps are typically only 2 Gbyte, and the weekly full dumps 75 Gbyte.

BACKUP

Having an established HSM filesystem does not obviate the need for backup. The HSM system itself needs to be guarded, and there are usually many other file systems that need backing up, on all the other servers in the centre – both system and user areas.

Backup is the process of taking copies of data, to allow restoration in the event that the original is lost, deleted or corrupted. [1] [2]. The threats and risks to data should be identified, and the backup strategy designed to guard against the major threats and risks. A typical approach might be to assess the likelihood of an incident, the consequences of the incident, and then deduce an overall measure of the risk, following by a policy and procedure to reduce the highest risks.

Backup is needed to guard against user error (commonly a few files accidentally deleted or over-written), systems administrator error (commonly whole file systems), hardware failure (commonly whole disc sub-systems, with multiple file systems), software failure (commonly RAID controlling software or systems utilities such as flushing scripts), and external failure (building collapse, fire, flood, terrorism).

Despite the need for backup, we find that centres usually provide backup of only a few file systems, and this is because the cost of providing backup of all user (and system) file storage areas can be prohibitive. Typically, backup is done by making dumps of a file system: this takes about as much space as the total occupied area on a file system. But one dump is insufficient, since when there is a loss, it is usually not discovered for a while, and what the user wants is a copy of a file from a previous day or week. So, dumps back in the past are kept: perhaps 4 weeks of weekly dumps, then perhaps 3 months of monthly dumps, 2 half-yearly dumps, and a few years of annual dumps [3]. This is messy to manage, and comes at a huge cost in storage, and in the repeated copying of the same data. And these backups do not serve as archives either, as old tapes tend to become unreadable. Off-line data is dead data, or at least dying. [4]

So, the traditional way of doing backups is resource-consuming both of storage capacity and personnel to run it. These days for desktop systems, external hard drives are the favoured way to go, with Apple's Time Machine. [5] [6] being especially attractive. One of the features of this is common to the backup schemes to be described here for UNIX/Linux systems.

Having a file system as a target for the backups, as at CSIRO ASC, can be a great advantage, and the following techniques are based on the use of such a facility. Having the target managed by HSM is an even greater advantage.

BACKUP – THE TOWER OF HANOI

CSIRO has been using the Tower of Hanoi scheme since 1998. The Tower of Hanoi arises as follows [7]. In Hanoi, so the legend goes, a group of monks are transferring rings from one pole to another, with a third pole being used as a temporary storage, with each ring being a different size. The rule is that each move must be made so that no ring is placed on a smaller ring. If we number the rings from number 1, the smallest, the moves are in the order: ring 1, ring 2, ring 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5, 1, 2, etc. Models have been available as children's toys.

Ring 1 is used every second time, ring 2 every 4th time, ring 3 every 8th time, etc. So, ring j is used every 2^j th move, starting on move 2^{j-1} . How is this related to backup? Simply, if we label each dump with a sequence number i , then we put each dump into a directory or tape or bucket or set j , where j is the Tower of Hanoi ring number to be moved at move i ; and keep one of each set number.

There are huge advantages to this scheme for dump set management.

- It is easily programmed to be automatic in determining which dump sets to use. There is none of the complication of days of the week, days in the month common to other schemes [8], [9].
- It can use a sequence number rather than just a date. This allows arbitrary spacing of dumps. When a system is busy, then more dumps can be taken to provide better coverage. When the system is slack, the dumps can be spaced further apart. When the system is down, and scheduled dumps are missed, it doesn't matter nor upset the dump sequence.
- The coverage of kept dumps spaces out over time. For example, if dumps are done daily, then at any time there will be better coverage than two dumps in the last two days, 3 dumps in the last 4 days, 4 dumps in the last 8 days, 5 dumps in the last 16 days, 6 dumps in the last 32 days, etc. The growth in storage goes as $\log_2 n$, where n is the number of dumps taken.
- The system self –heals.

We made several advances over the straight Tower of Hanoi. Firstly, we start with sequence number zero, and define this separately as set 0, and never remove it. This provides a baseline dump of the earliest state of the target file system. Secondly, to provide better coverage of recent times, we keep two of set 1. Thirdly, we make backups to a new area before discarding or recycling the old. This means we always keep at least the last 5 backups. Together, each of these costs an extra dump set. We use HSM-managed file systems as the targets of these backups, and so obviate the need for separate space and tape management.

The Tower of Hanoi scheme provides the best cover you can economically get, without knowing anything about where the vital copies are. It matches the heuristics that the most wanted files from backups are likely to be from the most recent backups, with a diminishing probability of files being needed the further back in time.

BACKUP – RSYNC AND --LINK-DEST

The admirable rsync command [10] is very useful for doing backups, since it does not transfer files from source to target that are already in the target, and has a clever algorithm to update files.

In 2007, one of the authors (JvdM) discovered a newish parameter available in, called --link-dest. The --link-dest option provided a way to do a backup from a source to a destination directory, but allowed rsync be aware of a previous directory on the destination side. If files were found to be the same in the source and previous destination directory, then instead of copying the file from the source to the destination directory, a UNIX hard-link would be created in the destination directory to the corresponding file in the previous destination directory. Hard links provide a way for a single copy of a file to be visible in multiple directories.

Rsync with --link-dest provides enormous savings in two ways. Firstly, it saves on bandwidth by not transferring the same files repeatedly. Secondly, the --link-dest facility provides the ability to make each directory look like a full backup of the source directory, at a cost of storing only one copy of files common to multiple directories. In a typical backed-up file system, where perhaps only 1% of the files and data are changed between backups, this means that only 1% of the data that would otherwise have been transferred is required to be transferred, and the storage on the destination side increases by no more than 1%.

The use of rsync with the --link-dest option provides the best of both worlds, in that it gives full backups at the cost of incremental backups. Repeated copies are not made, the system gives immediate access to individual files, and you can easily view the entire holdings of a file.

CONCLUSION

This paper has raised some issues of management of data for users at HPC centres. In particular, it has highlighted the complexity in storage services that are provided by many HPC centres. It highlighted the advantages of HSM, in particular for ameliorating the backup problem, and for acting as a target for backups of other systems.

The need for backups was discussed, and the marriage of the Tower of Hanoi scheme and the facilities of the rsync utility have been shown to be a very effective way to manage the backup requirement for large file systems. This scheme had been in operation at CSIRO ASC since May 2007, and at the time of writing was managing the backup of about 14 million source files and about 3.4 Tbyte, and was using about 35 million inodes on the target side. The run time for this backup is about 3 hours each night.

The scheme can save a considerable amount of storage. For example, for a home area on one of the CSIRO ASC systems holding about 875 Gbyte, the current backups occupy only about 1865 Gbyte after two years of backups, an expansion by a factor of only 2.1. A conventional backup scheme would be able to hold only two full backups in this space, whereas the Tower of Hanoi/rsync --link-dest scheme holds 13 backups – a saving of a factor of about five, given that the file system usage has grown over time.

Commercial backup solutions are of course available, and will no doubt work well in many circumstances. However, there are dangers to committing long-term information to a proprietary format that vendors usually use. The Tower of Hanoi/rsync --link-dest scheme is based on open source.

A recent book [11] addresses inexpensive backup solutions for open systems, but gives only a brief mention of Tower of Hanoi: it does cover the use of rsync and hard-links.

REFERENCES

1. <http://en.wikipedia.org/wiki/Backup> Backup
2. <http://www.taobackup.com/> *The Tao of backup*
3. http://en.wikipedia.org/wiki/Grandfather-father-son_backup *Grandfather-father-son backup*
4. Rothenberg, J (1995) *Ensuring the longevity of digital documents* Sci Amer, vol 272, no 1, 42-47
5. <http://www.apple.com/macosx/what-is-macosx/time-machine.html> *Time Machine*
6. http://earthlingsoft.net/ssp/blog/2008/03/x5_time_machine Porst, S-S (2008)
7. http://en.wikipedia.org/wiki/Tower_of_Hanoi *Tower of Hanoi*
8. http://en.wikipedia.org/wiki/Backup_rotation_scheme *Backup rotation scheme*
9. <http://folk.uio.no/johnen/bontmia/> Vollestad, J E (2003) *Backup Over Network To Multiple Incremental Archives*
10. <http://en.wikipedia.org/wiki/Rsync> rsync
11. Preston, W. C. (2007) *Backup & Recovery: Inexpensive Backup Solutions for Open Systems*