Building the Perfect BRICK

Rationale for the Research Computing Storage BRICK

by Harry Mangalam, Frank Wessel, & Tony Soeller
Research Computing, NACS

Digital information storage is increasingly important at all institutions, especially academic ones. Experimental equipment generates increasing amounts of data and therefore research is in the forefront in requiring more storage and better ways of dealing with it. Remote sensing streams, gene expression data, medical imaging, and simulation intermediates are now easily ranging into the 10s, 100s, and often into the 1000s of GBs. As well, class work, lab notes, administrative documents, email, and generic digital multimedia contribute to the digital flood.

Some of this data is reproducible at low cost; some is once-in-a-lifetime. Other data is extremely valuable either because of the cost of reproducing it or it deals with sensitive financial or medical records. This proposal does not address the storage of legally binding documents of the highest sensitivity and security. There are commercial vendors who supply such technologies and they are typically an order of magnitude more expensive than the storage that we address. We address the storage in the 'pretty' spot of this terrain - pretty cheap, pretty secure, pretty available, pretty fast, pretty accessible, pretty flexible. The plan is to use these devices as building blocks of a larger infrastructure, and because it is also a fairly accurate industry term, we are calling the device described here a *BRICK*.

We have compared available commercial solutions to our requirements and found many of them to be too expensive, too power-hungry, or too platform/protocol-specific for this charge. It is notable that Sun recently announced a product (the Sun X4500 *aka* Thumper) that is fairly similar to our proposal. It is included in our comparison table below.

Not only is size increasing but people are communicating this data to their colleagues at increasing rates. Typically this is done via email attachments but there is some evidence that researchers are using URLs to pass pointers to data as opposed to the data itself. NACS has a charge to see that this is done securely, easily, quickly, with generous allocations as to bandwidth and storage limits. One of the ways to do this is to match storage demands from schools with local bricks that are still maintained by NACS. They could be co-located in remote server closets and managed by NACS, the local administrators or a combination of the two.

In designing our , we have kept things as generic as possible. We use common parts so replacements and upgrades are inexpensive. Our approach only recently became possible because of improvements in hardware - the SATA family of disks and controllers especially. This technology has pushed the price of enterprise-level reliability down by allowing 'fail-in-place', hot spares, and rebuild-on-the-fly approaches to increase overall reliability using less reliable parts (see IBM article). The disks we propose using are not the 'enterprise-level' disks that are typically used, but generic SATA disks. There have recently been 2 studies published (from Google and CMU) that compare disk failure rates in very large (100K) disk populations and both make the point that 'Enterprise-qualified' disks are no more reliable than the SATA disks we specify, although they do provide better single-disk performance. We anticipate disk failures and have allocated 2 hot-spare disks for each RAID (itself redundant) to anticipate disk failures. Even with these generic SATA disks, single-disk failures are uncommon. Having 3 disks fail at once (the

number that would lead to data loss) would be rare in the extreme. Despite this reliability, this project does not consider disaster recovery directly. That could certainly be implemented with these bricks but not without more thoughtful planning and additional software such as the High Availability clustering tools or using a number of geographically dispersed *BRICKS* as the basis for a distributed storage system such as the CleverSafe system

These bricks will be 'pretty fast' – they can accept or supply network data at the rate of 2 Gb ethernets, about 140MB/s. The disk subsystems should be able to handle I/O at roughly 1/3 more than that, so even at saturating ethernet loads, there is still usable I/O for on-board processing such as servicing web or database queries. They are not meant to be high performance storage devices although they could be configured to operate about 2x as fast as described, at reduced reliability or capacity or both. The recently released SATAII PCI-e controllers improves the I/O even more. Even with cheap components, such controllers when paired with SATAII disks and appropriate filesystem types can provide block read speeds of >400MB/s and block writes of >200 MB/s (personal experience).

The current plan is for these bricks to run Linux although most Unix-like OS's will work; the protocol level software that will oversee the disks is all Open Source as well. We plan to make the storage available for the following protocols although we will start with the first 2 or 3 and expand as needed.

- PCs running MS Windows via SAMBA
- Linux & Unix file service via NFS or AFS.
- Desktop and Laptop backup via rsync, unison, BackupPC, Bacula, or Amanda.
- Webfiles-like access via WebDAV
- File Versioning control via CVS or Subversion.
- Databases via PostgreSQL, MySQL

We also want to make our configuration experience easily available to others who would either want NACS to make this storage available on a for-pay basis or to enable them to clone the *BRICK* for their own use. This should enable a fairly naïve user to buy the hardware from an approved vendor, install the described packages in a single command, and overlay a configuration package from NACS that would enable them to turn on the services they want. This approach would save money for NACS, for the campus, and especially if the approach was adopted UC-wide for such storage.

This approach should drive the cost of storage very close to the cost of generic hardware, trading expected failure and replacement costs for the extreme cost of never-fail components. Note that by 'generic' equipment, we do not mean the cheapest possible. Most of the components we will use will be comparable to those used by second tier vendors, but they will not be brand-named, and therefore will be considerably cheaper than true 'enterprise-qualified' parts. We realize that the human costs of administering and replacing hardware are among the most expensive costs so we do not want to make such a *BRICK* cheap up-front and abhorrently expensive to maintain.

Thanks to the Linux revolution, a number of vendors sell the kind of equipment we require. We have chosen ServersDirect.com as an example, but others can provide similar pricing. The final decision will be a careful review of recent purchasing, others' experiences, etc. We include ***Table 1*** below as an example of the costs of such a product. The summary is that from published prices, our 12TB *BRICK*, configured similarly, is about 2/3 the price of the 12TB Sun product. The generic 24TB *BRICK* is about

57% the price of the 24TB Sun. The Sun product does come with some additional features such as the 'Lights-Out Integrated Manager" but only has 2 years of support for the stated price. The ServersDirect product can additionally be configured with 4 DualCore Opterons and up to 32GB RAM which could make an 8-way system available for a large compute or database server.

A large consideration is the ease with which the server can be brought to service and configured. All the software packages we've mentioned could run on either Solaris or Linux. However on Linux, the pre-compiled software packages are available with the **apt-get** command and so can be installed via a single line. Many of the Solaris packages would mostly have to be compiled and installed by hand. Sun does include or provide pre-compiled packages for some packages such as NFS, SAMBA, Postgresql, kerberos, subversion, cvs, and rsync, but it does not support nearly as many packages as Linux. Solaris does provide a native implementation of ZFS, a modern file system and volume management implementation, but it will also soon run on Linux as well, as Sun has made it Open Source and the Linux port is already underway.

The advantage that Linux has in terms of available binary packages is reduced when you consider the configuration process is largely constant across platforms unless specific configuration programs have been written as well. In most cases they have not, but this is the one of the goals of this project, and most of the configurations would support Linux and Solaris equally well. This is important as some users will feel that the extra money spent on Sun hardware and software is well-spent. Sun has a reputation for reliability and stability, and the decreased cost of the generic platform and increased flexibility will result in some edge cases where stability is compromised.

Regardless of the hardware vendor chosen, this approach is roughly 1/10th the cost of storage from companies such as Network Appliance or EMS and for the kind of storage much of the university requires, its cost and reliability are very attractive. The funds requested would cover the purchase of one of the systems below which would provide the testbed for the services described above.

Table 1. Comparison of the BRICK vs Sun X4500.

Size	***4-socket MB 12TB***	***2-socket MB (~Sun) 12TB***	***Sun X4500 12/24 TB***	***4-socket MB 24TB***
Rack Units	5U	5U	4U	5U
PS	Redundant	Redundant	Redundant	Redundant
Gb eth	2	2	4	2
CPU	2xOpt852 2.6GHz	2xOpt 275dc* 2.2GHz	2xOpt285dc* 2.6GHz	4xOpt 850 2.4GHz
RAM	8	16	16	16
raw TB	12	12	12 / 24	24
Support	3yr	3yr	2yr	3yr
OS	Linux	Linux	Solaris	Linux
Bays	24	24	48	48

Size	*4-socket MB 12TB*	*2-socket MB (~Sun) 12TB*	*Sun X4500 12/24 TB*	*4-socket MB 24TB*
Cost (08.28.2006)	$18772	$18,085	$26,396 (12TB) $55,996 (24TB) (w/ UCI discount)	$31,869

- dc = DualCores; 2 CPUs per die

Discussions on the RCS Storage Brick Proposal

Harry Mangalam, Frank Wessel, Research Computing Support, NACS, UC Irvine
September 22, 2006

Thanks to Joseph Farran, Francisco Lopez, and John Mangrich for taking the time to evaluate and critique the RCS Storage Brick proposal. After the discussion, all of us came to a better understanding of what it does, what it's supposed to do, and what it should do better in order to be more valuable to the larger IT community at UCI. We also discussed the priority of the Brick vs moving more quickly on the Grid Computing initiative (see point 6).

In order for the Brick to be of use to NACS and a wider community than just researchers, RCS needs to address the following points, most of which can only be tested with the device itself.

1. Disk I/O - how fast can the system move data in and out of the subsystem under various RAID configurations? Since it can be configured to stripe, mirror, parity-check in several configurations, this should be an early test of whatever system we get. About a year ago, on a dual Opteron machine similar to the proposed configuration, I was able to obtain bonnie++ benchmarks at >100MB/s for both read and write on a 3ware 9500 RAID5 array. This was performed under Linux, using the XFS file system with default settings. Different applications will perform best using different filesystems, RAID types, and other parameters. Determining these parameters will be part of the Storage Brick evaluation process.
2. For it to be useful for remote administration by NACS or others, it should support IPMI2 for out-of-band access to the BIOS. Most modern server boards support this either on-board or by plug-in boards, but this must be a priority. The RAID card should also support out-of-band management. This reduces the choice to Areca, as 3ware only supports host-based web administration.
3. Among the most important utilities for a large disk partition used in a campus environment, the Brick must support quotas, logical volume management, support for different filesystem types, filesystem growth over multiple disks and preferably over separate network nodes. In Linux, there are 2 approaches - LVM2 and IBM's EVMS, both open source. LVM2 is the better supported, but IBM's seems to be more capable. I'll evaluate both again.
4. Also important for integration into the UCI campus computing environment is the ability for users to be integrated using LDAP and Kerberos. Linux supports both these

authentication schemes but it needs to be shown how easily it can be done and administered.

5. Remote administration of day-to-day chores. If this Brick is going to be useful to department administrators as well as commandline commandos, there needs to be a web or other GUI interface to do so. Webmin is one approach that has good support and traction in the wider community, but there are also tools based on VNC and desktop tools such as the KDE desktop administration tools. Webmin includes a number of modules for things as diverse as nagios, mysql & postgres databases, mail services, etc, so it looks like the obvious choice.
6. Also discussed was that although the grid computing initiative is attractive, the people present indicated that increasing storage requirements was as large a problem as grid computing and the point was made that the grid project also needs storage devices. Therefore investigation into the Storage Brick supports both projects.

Additional people on campus who have expressed positive interest in this project are Domingos Begalli of Physical Sciences, Steve Carlyle of Biological Sciences, and Garr Updegraff of the Registrar's office.