x
Loading

Search

Hi, goldj

LINUX MAGAZINE

Today's HPC Clusters Resource Center

- Analysis
- Interviews
- Webinars
- White Papers
- Case Studies

Parallel Platters: File Systems for HPC Clusters (Part One)

High Performance Computing (HPC) clusters are easier, and cheaper, than ever to put together. If you have an interest in pulling together your own cluster, or maybe you just want to understand more about cluster technology, it's necessary to grok the differences between clusters and standard systems.

Jeffrey B. Layton, Ph.D.
Monday, October 15th, 2007

High Performance Computing (HPC) clusters are easier, and cheaper, than ever to put together. If you have an interest in pulling together your own cluster, or maybe you just want to understand more about cluster technology, it's necessary to grok the differences between clusters and standard systems. One of the differences is storage requirements. While a standard server or PC can get by with a single disk and a standard file system, clusters need something a little more advanced.

In an effort to scratch the proverbial I/O itch, this multi-part article is designed as a high level survey that just touches upon file systems (and file system issues) used by clusters. My goal is to at least give you some ideas of available file system options and some links that allow you to investigate. Before I start, however, I want to discuss some of the enabling technologies for high performance parallel file systems.

Enabling Technologies for Cluster File Systems

As I discussed in a previous article Infiniband (IB) has very good performance and the price has been dropping steadily. DDR (Double Data Rate) Infiniband is now pretty much the standard for all IB systems. DDR IB has a theoretical bandwidth of 20 Gbps (Giga-bits per second), a latency less than 3 microseconds (depending upon how you measure it), an N/2 of about 110 bytes and a very high message rate. The price has dropped to an average cost of about $1,200-$1,400 per port. In addition, Myrinet 10GigE is also a contender in this space.

At the same time, most applications don't need all of that bandwidth. Even with the added communication needs of multi-cores there is plenty of IB or 10GigE bandwidth left over. To make the best of all the interconnect capability, vendors and customers are using the left over bandwidth to feed the data appetite of cluster nodes.

File System Introduction and Taxonomy

To make things a little bit easier I have broken file systems for clusters into two groups: Distributed File Systems, and Parallel File Systems. The difference is that parallel file systems are exactly that," parallel" as they utilize multiple data servers. While distributed file systems that use a single server, are not necessarily parallel, they can give the user" parallel access" to a file system.

Due to the length restrictions of this article, I can only touch on a few file systems. Please don't be alarmed if your favorite is not covered. It's not intended to dissuade anyone from considering or using that file system. I simply chose to limit the number of file systems I covered to keep the size of this piece reasonable. I have tried to cover the popular systems, but popularity depends on application area as well. Finally, the discussion of a file system is not to be considered an endorsement by me, or *Linux Magazine*. If you think your file systems deserves more attention, by all means, contact me and tell me how you use it, and why it works for you.

Distributed File Systems

The first set of file systems I want to discuss are what I call distributed file systems. These are file systems that are

Community Tools

ShareThis
Recommend This [?]
(2 votes, average: 5 out of 5)
Loading ...
Users Who Liked This [?]

Tags:
dfs hpc
Tag This!
No Comments

Micropoll

What are the high performance network plans for your next cluster?

- ○ Will be moving to 10 GigE
- ○ Staying with basic GigE because cost/performance is fine
- ○ Waiting because 10 GigE is still too expensive
- ○ Waiting because I need to see 10 GigE benchmarks
- ○ Trying to decide between InfiniBand and 10-GigE

Vote

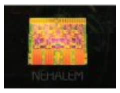
Loading ...

- Video
- Papers
- Webinars

In
Ri
On
Dou
Ric
the
Org

The

In
Te


Sa
Vi
In

network based (i.e. the actual storage hardware is not necessarily on the nodes) but it not necessarily parallel (i.e. there may not be multiple servers that are delivering the file system). I think you will recognize some of the names of distributed file systems.

NFS has been the primary file system for clusters because it' s there, and is pretty much" plug and play" on most* nix systems. It was the first popular file system that allowed distributed systems to share data effectively. Consequently, it can be viewed as another of the enabling technologies for HPC clusters (i.e. Getting user directories on the nodes was trivial). Moreover, it is the only file system *standard* for sharing data over a network.

NFS version 3 (NFSv3) is probably the most popular version of NFS. It was released around 1995 and added several features including support for 64-bit file sizes and offsets (so it can handle files larger than 4GB), asynchronous write support, and TCP as a transport layer. Around 2003, NFS version 4 (NFSv4) was released with some improvements. In particular, it added some speed improvements, strong security (with compatibility for multiple security protocols), and— perhaps most important— NFS became a stateful protocol.

Many companies make, market, and support NFS systems, or Network Attached Storage (NAS) devices. IBM, NetApp, EMC, HP, ONStor, Scalable, BlueArc, and many other vendors including small hardware shops, that make, market, and support NAS devices. Since the NFS protocol is a standard these devices are all interoperable and so it is largely a" plug and play" technology.

As convenient as they are, NAS devices are not without issues when it comes to HPC clusters. For example, they don' t scale well, particularly for large systems, either in terms of capacity or performance. Also, they have limited performance as they are a single storage server. To overcome some of these problems, vendors have made specialized hardware to improve scalability and performance, but you can only do so much to improve scalability and performance within the limitations of the NFS standard.

In general, a NAS device has a single NFS server that is connected to a private cluster network that all of the cluster clients can access. Behind the single NFS server, sometimes called a filer head, is hardware storage. The filer head exports the file system that the clients mount. When a client wants to access the data, the request is sent to the filer head that then sends back the data. The performance limitations are primarily due to all data requests having to flow through a single point. Some vendors have come up with solutions in the filer head to improve performance. Plus, they have used some fairly hefty hardware to allow the storage to scale to some fixed amount (usually in the Terabyte range). There is also a class of solutions called Clustered NAS (more on that topic further down) that improve performance, but can also have scalability and performance limitations.

The good news for NFS and NAS is that many codes don' t require lots of I/O for good performance. These codes will run very well using NFS as the storage protocol even for large runs (100+ nodes or several hundred cores). NFS provides adequate performance until the input and output files for these codes become extremely large, or if the code is run across a very large number of processors (in the thousands).

As I mentioned previously, NFSv4 added the ability for NFS to become a stateful protocol. (In general, NFS was designed to be *stateless* so the server would not have to maintain state for client connections, thus making the implementation much easier. A *stateful* protocol is desired for certain aspects of file serving that can' t be done in a stateless way, such as file locking in a distributed environment.) NFSv4 does seem to have limited adoption based on the number of sites that are using it. Even with all the industrial strength features, NFS still lacks the performance and scalability required by many large clusters, but that is about to change.

pNFS

A number of vendors are now working on version 4.1 of the NFS standard. One of the biggest additions to NFSv4.1 is Parallel NFS (pNFS). You might think this is an attempt to kludge NFS for better performance and scalability, but this isn' t the case. It is a well-planned, tested, and executed approach to adding a true parallel file system capability to the NFS protocol. The goal is to improve performance and scalability while making the changes within a standard (recall that NFS is the only *true* shared file system standard).

Moreover, this standard is designed to be used with file based, block based, and object based storage devices with an eye towards freeing customers from vendor lock-in. The NFSv4.1 draft standard contains a draft specification for pNFS that is being developed and demonstrated now. Vendors working together to develop pNFS include Panasas, NetApp, IBM, Sun, and many others. Obviously the backing of large vendors means there is a real chance that pNFS will see widespread acceptance in a reasonable amount of time.

The basic architecture of pNFS is shown below in *Figure One*.

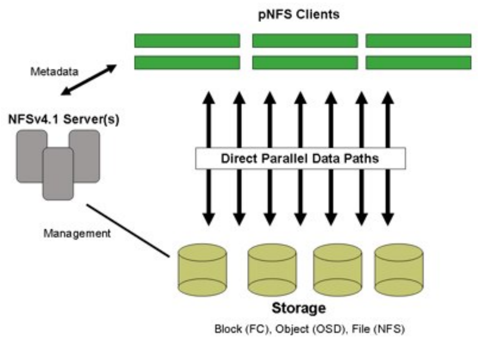

pNFS Clients
Metadata
NFSv4.1 Server(s)
Direct Parallel Data Paths
Management
Storage
Block (FC), Object (OSD), File (NFS)

Figure One: The Basic Architecture of pNFS

The configuration consists of some NFSv4.1 servers (shown on the left) that act as metadata servers. On the bottom of the figure is the storage that is connected to both the metadata servers and to the pNFS clients. The clients are also connected to the NFSv4.1 servers. When a client requests data, the metadata managers are contacted. They communicate with the storage to determine where the data is located (in the case of read) or where the data could be located (in the case of write). Then the metadata managers pass back a layout of where the data is located on the storage devices.

The client (s) then contact the storage devices directly. This design eliminates the metadata server from being a" middleman" throughout the entire transaction and the clients can then access the storage devices in parallel to improve throughput. The vendors will only need to write what is called a layout driver for the pNFS client. This feature allows the client to communicate with the storage pool.

One of the features of NFSv4.1 is that it avoids vendor lock-in. Part of this is because pNFS will be a standard. In fact, it will be the* only* parallel file system standard. The pNFS standard provides for three major types of storage, (block, file, and object based) allowing various types of vendors to ship storage that can integrate into a pNFS file system. This design goal allows you to chose whatever storage you want as long as there are layout drivers for it.

So why should multiple vendors support NFSv4.1? They are competitors, after all. The answer lies in fact that standards ultimately help sell hardware. With pNFS, the vendors can now support multiple Operating Systems without having to port their entire software stack to the new OS. They only have to write a driver for their hardware rather than porting an entire software stack to a new OS. So for the vendor pNFS increases the possible customer base for their storage hardware and lowers the cost of software development. Open standards are a Good Thing ™.

While pNFS is on its way to becoming a standard, one can' t predict exactly when it will happen. It appears that by the end of 2007 or the beginning of 2008 we may have a real parallel file system standard in the form of pNFS. If you want to learn more about pNFS you can go to the pNFS Web site. You can also go to the Panasas Web site and follow links to presentation and information about pNFS.

If you want to experiment with pNFS now, the Center for Information Technology Integration (CITI) has some kernel patches for Linux 2.6 that use PVFS2 for storage.

Clustered NAS

Since NAS boxes only have a single server (single filer head), Clustered NAS systems were developed to make NAS systems more scalable and to give them more performance. A Clustered NAS uses several filer heads instead of a single one. The filer heads are then connected to storage via a network or the storage may be directly attached to each filer head.

Clustered NAS systems have two primary architectures. In the first architecture, several file heads each have some storage assigned to them. The other filer heads cannot access the data, but all of the filer heads" know" which filer head has which data. When a data request from a client comes into a filer head, it determines where the data is located. Then it contacts the filer head that owns the data using a private storage network. The filer head that owns the data retrieves the data and sends it over the private storage network to the originating filer head which then sends the data to the client. This first approach is used by NetApp (NetApp-GX).

In the second approach, the filer heads are really gateways from the clients to a parallel file system. For these types of systems, filer heads communicate with the client using NFS over the client network but access the parallel file system on a private storage network. The gateways may or may not have storage attached to them depending upon the specifics of the solution. This approach allows the ClusterNAS to be scaled quite large because you can just add more gateways— which also increases aggregate performance because there are more NFS gateways. This approach is used by Isilon. It is also used by Panasas, IBM' s GPFS, and other parallel file systems when they are running in a *NFS mode*.

The problem with either approach to Clustered NAS devices is that you have limited performance to the client because you are using NFS as the communication protocol. Most of the Cluster NAS solutions use a single GigE connection so you are limited to about 90-100 MB/s at most to each client.

I want to stop here since this is a logical break between Distributed File Systems and parallel file systems. In the next piece, I will discuss some of the more popular parallel file systems for clusters. These file systems can be (and are) deployed today. But don' t underestimate the power of pNFS in the near future. Beware the standard.

Jeff Layton is an Enterprise Technologist for HPC at Dell. He can be found lounging around at a nearby Frys enjoying the coffee and waiting for sales (but never during working hours).

Read More

1. Before You Commit: Four Key Questions To Ask About InfiniBand
2. The Personal Cluster: Coming To A Desk Near You
3. Fun and Games
4. Storage Convergence: Fibre Channel over Ethernet
5. HPC Hardware is Free

Comments on Parallel Platters: File Systems for HPC Clusters (Part One) »

Comments via RSS

No comments yet.

Comments via RSS
You are logged in as **goldjay**. Logout »
Have a Gravatar? Your Gravatar pic will appear next to your comments. [?]

Your Comment

You may use <a href="" title=""> <abbr title=""> <acronym title=""> <b> <blockquote cite=""> <code> <em> <i> <strike> <strong> in your comment.

Add comment

About | Contact Us | Privacy Policy | FAQ | RSS