

*date:* March 4, 2011

*to:* General Release

*from:* Lee Ward

*subject:* I/O and Network Breakout Summary

## Scope

We are responsible for the high performance IO and file system within the multi-program, parallel machine as well as the networking infrastructure used to move data and user sessions within the NNSA complex.

The IO portion of our domain can be any portion of the software stack, from the application libraries and supporting middleware to the support file systems found within the operating system. We also are responsible for working with hardware storage vendors in support of our mission. This can be a wide range of engagement. We meet frequently with vendors to discuss hardware changes that better support our mission and we have prototyped hardware solutions in order to gauge mission impact with experimental solutions.

The networking portion of our domain is typically restricted to hardware. We work with the vendors to suggest improvements in support of our mission, evaluate and test new vendor offerings, and provide support to our operations infrastructure in identifying and deploying state of the art enterprise and long-haul networking technologies.

## Current State of the Art

By far, the greatest demand on our IO systems, deployed in high performance computing, is for the purpose of checkpoint and restart of long running jobs with compute needs that outlive current mean-time-to-interrupt times for the machines we deploy. Because of the highly synchronized manner in which our applications function the IO pattern for checkpoint-restart is bulk-synchronous, the distributed application dumps state from all it's component nodes at the same time and to the same globally shared storage.

---

<sup>1</sup> Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Currently, to support this activity we primarily utilize the Lustre file system, an open-source and community supported project, the PanFS file system, from Panasas corporation. We have in the recent past used the GPFS file system, from IBM corporation, and may do so again in the future. All of these file systems function in a roughly similar fashion. They centrally coordinate, and manage, a large set of storage devices hosted on the high-performance machine, or near by.

To date, these solutions have worked well, particularly when the applications do not attempt to coordinate access to shared files. Some applications do share files, though, and cannot, typically, achieve better than ~50% of the potential IO bandwidth available.

As we move toward the exascale computing realm our classic solutions will be hard-pressed to address our needs, unaided. The non-shared files approach that delivers our best performance, now, will be impacted by the sheer number of compute clients that are predicted for the exascale machine. The controllers, in some cases, do not scale well and the control protocols, in others, look to be challenging at scale. For the shared-file case, the same problem is present but it is exacerbated by the need to manage a globally-coherent address space at an unprecedented scale.

Application middleware in our complex is, primarily, a small collection of scientific data file management and abstraction libraries. Typically, the Hierarchical Data Format (HDF version 5) library, or some variant of the Network Common Data Form (NetCDF) interface, or the hom-grown EXODUS format in one case. Adoption of these abstractions is scattered within the applications. Developers seem to welcome the useful abstractions but sometimes are unwilling to pay the performance cost of maintaining them. They can be troublesome to tune for a particular machine and file system. As we move to exascale, those that require globally coherent address spaces will be particularly challenging to keep attractive.

Our infrastructure networking challenge is highly constrained by a need to remain compatible with commonly deployed technologies. This form of networking is available, and must interconnect with, our external enterprise networks which is always based on commodity protocols and hardware solutions for both economic and technological reasons. We could not afford to generate, maintain, or motivate external support for the translation equipment or the long haul routing and transmission hardware if we turned to custom solutions. We would be challenged to design and deliver competitive solutions for the enterprise and long-haul as such function has never been within our scope of responsibilities in the past. We must allow the market to dictate our options, though we can influence the high-end of that market.

As we progress toward exascale we intend to track the market and purchase the most cost-effective solutions for our needs. We cannot, now, often and reasonably move our datasets between the various locations in our complex easily and we do not foresee a better scenario as we approach exascale.

## Research Needs

The current file system solutions were designed to work in a hierarchy that had, never great but better, ratios of bandwidth at their interfaces. The petascale realm has altered those for the worse, exponentially so. Everything indicates this will continue as we approach exascale. In the main, we must address this problem as well as the predicted shortcomings of the current file system's need for centralized management.

We believe this is best done by researching file systems that can smoothly tier and avoid, or mitigate the need to manage more than a few storage components at any time, irrespective of the number of clients that are simultaneously accessing the storage. Newly arrived, and attractively priced, solid state storage may be incorporated as a "burst" buffer for our bulk-synchronous checkpoints. We can quickly write a checkpoint into this storage and then, when the application has re-entered the long compute phase, drain the solid state store to higher capacity, cheaper, more classical rotating magnetic stores. We must address the centralized control problem as well. For this, modern peer-to-peer and "cloud" services would seem to offer an inspiring model. While not directly applicable to our environments they do contain interesting and, undeniably, scalable alternatives that we might leverage as part of our solutions.

Such a file system represents an architecturally integrated hierarchy of storage devices. As such, the opportunity to integrate what has been the traditionally separate data archive capability will occur. This seamless integration should be attempted as well since we already contemplate support for a high degree of heterogeneity and robustness in the inherently unreliable network and stores.

Our application developers have strong need for functional, fast, intuitive scientific data management libraries. More, strong interest in data provenance and an ability to make use of data sets simultaneously with creation and population motivates us to take a deeper look at our middleware stack. Perhaps something new, perhaps only significant modification and extension of the existing libraries. It is unclear, yet, which or how.

We estimate initial prototypes will be available by approximately 2015 with reasonable funding, and subsequent hardening and production quality implementations by 2020.

## **Development Needs**

For file systems and storage at exascale, at least, our only answer cannot be simply a wonderful new file system that relies on the answers to questions nobody has even thought to ask until recently. We must not plan on an indeterminate future in that way. It would be irresponsible.

Therefore, we believe we must develop and deploy proven and likely technologies from our, and others, current research in order to ease the blow to existing file system solutions. Much of the non-scalable control problem with these file systems can be mitigated by making the real machine appear as something far smaller, far more like what existing file systems were developed for. A project called the IO Forwarding Software Layer can accomplish this, and has been used in current production petascale environments with promising results. We believe that we can mitigate much of the single-shared-file issue with a layer of software that translates this mode of access into, what the underlying file system believes to be, many non-shared files. This solution is known as the Parallel Log-structured File System (PLFS) has been in use within our complex for a little more than a year and, again, with promising results. Finally, we believe relatively minor modifications to our Scalable Checkpoint and Restart (SCR) low-level library will allow us to address the bulk-synchronous nature of our checkpoints using solid state storage as a "burst" buffer, minimizing the storage budget for the anticipated exascale machine.

These technologies do not make a modest research program irrelevant, though. Some of us believe we can reach low exascale with judicious work on and deployment of these technologies. Some do not. All agree, though, that they buy us needed time and allow a significant, perhaps imperfect, risk mitigation strategy for the research we propose.

Our current archive solution, the High Performance Storage System (HPSS) will require continued maintenance and thought as we move to exascale. It's architecture accommodates a virtually unlimited number of parallel channels to and from the archive. The problem will be in making the current implementation better meet that virtual without requiring a major rewrite or refactoring.

The middle ware scientific data management libraries must be heavily worked on to address performance issues. Even today, many applications shy away from these libraries for performance reasons. Not surprising given that they were designed and implemented for the single workstation on the desktop. It is no stretch to imagine they will be completely unacceptable at exascale.

We estimate that the identified portions of our development will be useful by 2015, some already are in early stages.

## **Co-Design Needs**

IO and infrastructure networking are completely cross-cutting, we will need help gathering requirements and evaluating how well we meet application needs throughout. We will necessarily place demands on the compute and service operating system as well as the machine interconnect.

In particular though, we will need to work with the folks who specify and design the hardware and the operating systems on both the compute and service nodes in order to have our needs met, and the applications developers, tools and environment, and visualization folks to generate acceptable solutions for them.

## **Co-Design Opportunities**

We can immediately identify opportunities to work with the DOE Office of Science ASCR program in pursuit of a common solution for both. As well, we should examine partnerships with the Hierarchical Data Format (HDF) folks to address our needs relative to performance of the HDF5 library. A previous partnership has generated the IO Forwarding Software Layer and, potentially, we might wish to extend it to include support for coming platforms in some fashion.

## **Potential Partnerships**

Our scope is common within the HPC community and we believe we could meaningfully partner with DoD, academics, and DARPA in pursuing our goals. We have already begun a partnership with the DOE Office of Science ASCR program.