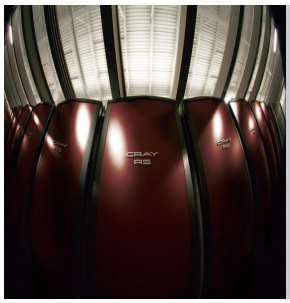

Sandia National Laboratories

Exceptional service in the national interest

A Storage/IO/Workflow View on Future Interconnects

Jay Lofstead

Scalable System Software
Sandia National Laboratories
Albuquerque, NM, USA
gflofst@sandia.gov

ExaComm Workshop @ ISC 2016

June 23, 2016

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

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.

My Perspective and Priorities

Sandia National Laboratories

- Storage
  - Disk and Burst Buffers
- I/O libraries
  - ADIOS, HDF5, NetCDF
- Workflows
  - Online (Integrated Application Workflows)
  - Disk for staging intermediate data

Number 1 Request

Sandia National Laboratories

STOP DUMPING ON US!!!

- Data staging (1996, 2006-present) worked to take data movement outside of application comm channels. Asynchronous collectives is taking this away from us!
  - We have GBs/node to push out and need significant windows we can own the network to get our jobs done.

- Network APIs change on a per-platform basis making optimizing data movement difficult
  - We have invented multiple network/RDMA abstractions to insulate our code from your whims!

More Seriously :-)

Sandia National Laboratories

- IO community working on concentrating IO-related comm into isolated machine areas, but need the windows to get data into those areas so we can get our work done with minimal wall clock time interference.
  - First hand observation has shown moving IO asynchronous reduces IO time to 0, but wall clock time is 30-100% longer because of interference. We have worked hard to get out of the way, but need our slice of the network to get our work done.

Pervasive Storage

Sandia National Laboratories

- With storage becoming pervasive in HPC platforms, we need ways to offer off node access with minimal node interference. Can't help the network, but can avoid memory if designed well.

- Byte-addressable or even block addressable off-node memory will make this harder

Programming APIs

Sandia National Laboratories

- We work in user space with LARGE data
  - Small messages are MBs, large are GBs (or 10s to 100s of GBs)
  - Why does the TCP API offer automatic packetization for large data blocks while HPC transport APIs do not? As a user, this would allow us to give the network large data blocks while the network layer interleaves other network traffic optimally or with QoS.

- A good, but not great, user-level API is good enough for us
  - With each network having its own API, we have to regularly abstract against a moving target. Can we have a simple API in user space that works?
  - MPI Inter-communicators are abysmally bad and hard to use. Can this be addressed (e.g., how do I identify a remote node I wish to connect to? How do I specify a list and make a bunch of connections at once?)

DDoS

Sandia National Laboratories

- Help addressing DDoS
  - Hardware on both sides of the software layer can handle 1 M+ messages/sec. Software cannot.
  - Parallel File System metadata servers are faster when clients manually introduce admission control with partial serialization
  - New Services in the compute area will require similar functionality. Can we get some sort of back pressure rather than hammering the service node? Can we offload the message interrupts so that the service node can get the service done more efficiently?

Biggest IO Bottleneck is Interconnect

Sandia National Laboratories

- MPI Two Phase Collective IO
  - At scale, this can take 99%+ of the IO time.
  - 3-D domains with a 3-D decomposition is really hard.
  - We use data staging to try to address this (as long as we don't have our data movement network slots taken away from us).