SAND2011-8738P

I/O, Networking, and Storage

Lee Ward, Luis Martinez; SNL

Rob Ross; ANL

Dick Watson, Chris Morrone, Mark Gary; LLNL

Hal Armstrong, John Bent; LANL

Matt Crawford, FNAL

Working Group Description

- Scope
  - I/O middleware, ex. HDF and NetCDF
  - File Systems
  - Storage aggregation and management
  - Archive Systems
  - Enterprise and long-haul networking
- Most significant challenges
  - Bulk-synchronous checkpoints at scale
    - We're told this is all going to be solved
    - We're not told when ☺
  - Network encryptors

Short-Term Challenges & Path Forward

- Compute node aggregators
  - Too many compute nodes
  - Need a way to "hide" the compute nodes from the file system service
    - Ex. IBM BG architecture, Cray DVS
  - Have IOFSL project but it needs further work to fully accomplish the task
    - Derivative file handles, allows server to better manage local state
    - A write-back cache would allow data aggregation and buffering
- Performant N:1 streams
  - Metadata services don't currently scale well
    - We need to create fewer files
  - PLFS allows the application N:1 access
    - But opens many files to do it
    - Which is OK because it's no longer an application problem
- Overlapping I/O with compute

Short-Term Challenges & Path Forward

- Overlapping I/O with compute
  - Others are working, now, to enable the applications
  - We must respond with reasonable system behaviors in kind
  - Working on in-system checkpoint and restart
    - SCR
  - Another option is in-situ processing

Long-Term Challenges & Path Forward

- Exascale presents an inflection-point
  - Problems we've already dealt with will come back with a vengeance
    - Centralized metadata, organization may become intractable
    - Striping will involve managing, and coupling hundreds or thousands of components
  - Old annoyances become new problems
    - Large multi-core, sheer scale, accelerating memory bandwidth
- We will take two approaches
  - The "evolutionary" will anticipate changes to existing solutions and we will provide or motivate changes and new layers intended to make the new platform look more like the old, from the IO service providers view
  - The "revolutionary" will take a fresh look at the problems and perceived gaps in order to provide a new, and well integrated, option.

Long-Term Challenges & Path Forward

- Evolutionary
  - – The potential solutions are numerous, varied, and partially or wholly exist independent of one another
    - Deploying, then, builds a house of cards and a maintenance nightmare
    - We must integrate most or all into a small number of products
  - – We do not know just how far this approach will take us
    - All the way to exascale we hope, we think even
- Revolutionary
  - – A “fresh” approach reasonably considers solutions that have never existed
    - Properly, we call this activity “research”
    - Which is probably not something that should be involved in “product” plans
- These are supposed to mitigate each other
  - – But the degree of mitigation is debatable