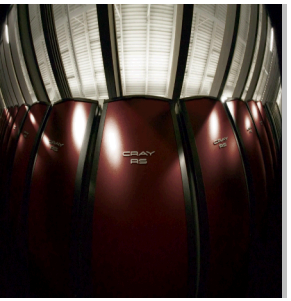
SAND2013-3094P



Data Management Issues for Exascale Platforms

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

Georgia Tech Alumni Day

April 15, 2013

Sandia National Laboratories

Exceptional service in the national interest

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.

HPC Data Management

Sandia National Laboratories

- Simulations generate LOTS of data periodically now
  - E.g., 10s of TB every several minutes
- Older APIs (NetCDF, HDF, PnetCDF)
  - Focus on data model, portability
- Next Generation (ADIOS)
  - New data model, portability maintained
  - Reworked for performance for MPP, large parallel storage arrays

- What do we need for the future?

Petascale to Exascale Changes

Sandia National Laboratories

Petascale	Exascale
~16 cores/node	100s of ‘cores’ per node
~250 GB/sec storage array	1-10 TB/sec storage array
1-2 GB/core	< 1 GB/’core’
RAM/HDD/Tape	RAM?/NVRAM/SSD/HDD/Tape

Independence vs. Coordination

Sandia National Laboratories

- Older APIs
  - Generally fully coordinated (mostly failed petascale)
- ADIOS
  - Generally fully independent (mostly successful petascale)

- Small per process data sets still problematic
  - If small enough to fit in one proc's memory, worth it?
  - Too much overhead to store small blocks

File System Matters

Sandia National Laboratories

- ADIOS demonstrated: Manage the Filesystem
  - Control for metadata contention (file creation/open)
  - Control for false sharing (multiple procs on same stripe, not same spot)
  - Control for stripe spanning (overhead of switching stripes)
  - Control for other users - 'interference' is highly transient
  - Maximize parallelism – even beyond file system capacity
- Avoid single points of contention
  - Single MDS, single index collector, single 'master' for anything

Rich Metadata Critical

Sandia National Laboratories

- Space aware is key – not enough space for full index
  - Data characteristics
    - Min/max
    - Mean
  - Collected index blocks
    - Dense collection of metadata for rapid processing

Memory Hierarchy Depth

Sandia National Laboratories

- Currently have RAM->HDD->Tape
- Moving to RAM->NVRAM->SSD->HDD->Tape

- Layout matters for parallelism; blocking factors vary by tech
  - Moving back down may lose layout information
- Locality, energy, performance all matter
- Data migration automatic or manual? (HSM?)

Break SE Best Practices

Sandia National Laboratories

- Software Engineering Best Practices not always best
  - Compile time checking can be problematic
    - Adds complexity to API
      - HDF API calls to navigate hierarchy
      - PnetCDF ‘modes’ for definitions or read/write
    - Reduces options for implementations (too detailed API)
      - Can you change to using staging with complex API semantics?
      - How about switching to asynchronous IO?
      - What about a carefully synchronous step-by-step implementation?
    - Less dynamic flexibility possible (must recompile to make changes)
      - Likely must recompile to make changes to behavior
      - Cannot add/remove non-data elements like attributes

IO Stack – Not an IO API Only

Sandia National Laboratories

- Fully integrated storage hierarchy key
  - Simple, flexible API
  - Options for data movement (staging, disk, or other)
  - Flexible formats based on technology (adapts automatically)
  - Address parallelism maximally
  - Interoperate with existing POSIX interface somehow

Sandia National Laboratories

Questions?