I/O and Storage Breakout

Scott Klasky, moderator

Mathew Wolf, scribe

Scope of discussion

- Focused on 4 key topics in the I/O space
  - Scientific Data Formats
  - Data Models
  - Database Technologies
  - Hierarchical Storage

Before we get there: Cross-cutting issues

- Resiliency
  - We will face failures of both hardware & software.
  - What are the "threat" models that are most important to address?
- CAP
  - Consistency, Availability, and Partitioning(/parallel)
    - Choose two – and we've already chosen Parallel
- Data ≠ Files
  - We have to break the idea that data access is through a standard file system

Scientific Data Formats

- Current issues:
  - There are many domain (non-?) standards
- Current state of the art
  - ADIOS-BP, FITS, HDF5, netCDF, ROOT,…
- Requirements for Exascale
  - Self-describing data
  - Explicit & implicit parallelism support
  - Data reductions & compression (tie to data model)
  - Graceful transition to other platforms (also w/novel architectures)
- Metrics
  - User adoption
    - Ease of use, volume of data stored, number of users
  - Throughput
    - Both read & write values. Peak will be low, so % of peak is important
  - Broad adoption outside of DOE
- Cross-linkages
  - Hierarchical storage, OS development (billion files?)
  - Data archiving community

Data Models

- Current issues
  - End users fit their data model to whatever is allowed by their desired file format. Or whatever sounds good at the time. (Is this bad?)
- Current state of the art
  - Community adoption of larger schemas
  - Structural primitives (grids, array descriptors, viz meshes, etc.)
  - Relational data models (provenance + experimentalists)
- Requirements for Exascale
  - Matching & conversion between data models will be critical
  - Decouple the logical view of data and the physical instantiation
- Metrics
  - Acceptance (?)
- Cross-linkages
  - Every app report
  - Every viz & analysis report
  - Every simulation project ever

Database Technologies

- Current issues
  - Lack of clarity what the role of databases should be at leadership scale
- Current state of the art
  - Used for raw data in some fields (astrophysics, high energy physics)
  - Industry data investments (bigTable, Cassandra, Hbase, etc.)
  - Metadata storage for HPC (in mysql, oracle, etc.)
- Requirements for Exascale
  - Complex queries against the new data objects will be important
  - Separate logical from physical view
- Metrics
  - IOPS are insufficient. Queries returned are where we're really interested
- Cross-linkages
  - Obvious industry ties. Experimental communities w/substantial investments.

Hierarchical Storage

- Current issues
  - Usability. Users don't understand hierarchies.
  - Extreme variability in performance from user perspective
- Current state of the art
  - File systems, I/O staging systems, asynchronous I/O, transactional memory
- Requirements for Exascale
  - Explicit handling of very deep memory hierarchies
    - "from registers to tape"
  - Ease of use
  - I/O system should be integral to design – separate network, dedicated buffers, etc.
  - More flexible view of computation and storage I/O
- Metrics
  - Performance (overheads, latency, throughput)
  - Power (Watts per byte stored? Per query answered?)
- Cross-linkages
  - HECURA, NNSA exascale effort, industry data centers
  - **Can't** just be a funny exascale system. Need to cross-leverage.

Take Aways

- We aren't (just) writing files any more.
- Input and Output aren't clean tasks any more
  - Sometimes you have to move computation to the data, or vice versa.
- Deep memory hierarchies are at least as scary as the changes to the processor designs.