SAND2008-0700P

Some Open Problems in Supercomputing I/O

Subtitle 28 pt

January 29 2008

Lee Ward
Principal Member of Technical Staff

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Introduction

- We'll discuss modern file system architectures
- Requirements for the next generation of "leadership" class machines
  - These, typically, set the stage for midrange and enterprise solutions
- Then turn to open problems and wishes

Architectures

- Network File Systems
  - Well, they all are that
- Distributed File Systems
- Parallel File Systems

Sandia National Laboratories

Properties

- Goal is POSIX compliance
  - Or, at least, POSIX compatibility to some degree
- Standard open, close, read, and write semantics
- Files have a single address space that is
  - Globally shared
  - Coherent
    - Last writer wins

Network File System

- Best known is NFS
  - Stateless until version 4
  - Which brought coherency problems
- Simple, straightforward, easy to deploy and manage
- Standard I/O libraries can be (almost) unaware of these
  - With additional lock managers, or with NFSv4, they can be

Distributed File Systems

- Best known is GFS (Redhat)
- Files are distributed to the storage servers
  - Typically do not span servers
  - When they do, it's to increase address space not speed
- Metadata may be centralized or distributed
- Frequently used in the enterprise
  - Strong POSIX compliance allow standard I/O libraries to be oblivious

Parallel File Systems

- Representatives; GPFS, Lustre, PanFS, PVFS
- Same features, and similar execution as distributed file systems
- Adds simultaneous, parallel, transfers to multiple, independent storage nodes
  - Needs a strong, fast network to do this
- Interestingly, PVFS is stateless
  - Which sacrifices coherency
    - How often do we really need that though?

Middleware for Parallel File Systems

- FS much like distributed normally; Allow oblivious standard I/O libraries to be used
- With ***many*** extensions to support relaxed, or non-existent coherency, stripe and stride, etc.
- Middleware such as MPI-2 I/O leverages the network and compute client to aggregate and make more efficient file access.
  - Collective operations
  - Data sieving
  - Peer caching

Requirements for PetaFLOPS

- 10^4 to 10^5 clients and network links
- Network link speed on the order of GB/s, bidirectional
- Ability to manage billions of files
- Perform I/O at 400GB/s to TB/s
- Need 20 – 40 PB of storage space

- Say, 20,000 disk drives?

HECURA/IWG

- High End Computing University Research Alliance, Interagency Working Group
- Created in response to U.S. Presidential commission report that noted under funding in high tech
- For I/O, advises funding agencies like NSF, DOE/Osc, DOE/NNSA, DoD, and NASA of gaps and progress in the field
- Open problem areas; metadata, measurement, QoS, future architectures, protocols, archive, management and RAS, security

Metadata

- Address issues in storage name space and file metadata
- Open problems
  - Scaling and partitioning of the service
  - File system and Archive integration and coherency
  - Exploitation of hybrid storage devices
  - Transparency and access methods

Measurement and Understanding

- Primarily simulation related
- Open problems
  - System workload in the enterprise
  - Standards for benchmarks
  - Testbeds
  - Application of visualization and analysis tools to large scale traces

Quality of Service

- Provides (semi?) deterministic performance to system shared resource
- Open problems
  - End-to-end QoS
  - A standard API

Next Generation I/O

- Where do we go from here?
- Open problems
  - Understanding abstractions, naming, organization
  - Architectures
  - Self-* components; Assembly, reconfiguration, healing
  - Managing millions of components
  - Hybrid devices
  - Small record access

Communication and Protocols

- Impact from networks and protocols
- Open problems
  - Active networks
  - Alternative transport schemes
  - Coherent schemes

Archive

- How to never have to delete anything
- Open problems
  - APIs and standards for interface, searches, attributes, staging, …
  - Long term, attribute-driven, security
  - Data reliability and management
  - Metadata scaling
  - Policy-driven management

Managament and RAS

- Config, deploy, and reliability, availability, servicability
- Open problems
  - Automated analysis and modeling
  - Formal failure analysis
  - Scalability
  - Power consumption and efficiency

Security

- Authentication and authorization
- Open problems
  - Long-term key management
  - End-to-end encryption
  - Overhead and scaling
  - Tracking of information flow, provenance
  - Ease of use, ease of management, quick recovery, APIs for same

Conclusion

- Network, distributed, and parallel file systems are relatively new or stagnant
  - Conformant API and semantics are 40+ years old
- Disks aren't getting appreciably faster
- We're in a corner; Relying on the network and trying to aggregate more and more components
- The field is ripe for a paradigm shift
  - In architectures, protocols, and physical devices