Supporting Data-Intensive Distributed Computing in an Exascale Era

Ioan Raicu
Computer Science Department, Illinois Institute of Technology
Math and Computer Science Division, Argonne National Laboratory

August 7th, 2013
MAGIC Meeting: 2020-2025 Scientific Computing Environments
• **Today (2013): Multicore Computing**
  - $O(10)$ cores commodity architectures
  - $O(100)$ cores proprietary architectures
  - $O(1000)$ GPU hardware threads

• **Near future (~2019): Manycore Computing**
  - $\sim1000$ cores/threads commodity architectures

Supporting Data-Intensive Distributed Computing in an Exascale Era

Pat Helland, Microsoft, The Irresistible Forces Meet the Movable Objects, November 9th, 2007
• **Today (2013):** Petascale Computing
  – O(100K) nodes
  – O(1M) cores

• **Near future (~2018):** Exascale Computing
  – ~1M nodes (10X)
  – ~1B processor-cores/threads (1000X)
Exascale Computing Architecture

• Compute
  – 1M nodes, with ~1K threads/cores per node

• Networking
  – N-dimensional torus
  – Meshes

• Storage
  – SANs with spinning disks will replace today’s tape
  – SANs with SSDs might exist, replacing today’s spinning disk SANs
  – SSDs might exist at every node
Segregated storage and compute
- NFS, GPFS, PVFS, Lustre, Panasas
- Batch-scheduled systems: Clusters, Grids, Supercomputers
- Programming paradigms: HPC, MTC, HTC

Collocated storage and compute
- HDFS, GFS
- Data centers at Google, Yahoo, and others
- Programming paradigms: MapReduce

Others from academia: Sector, MosaStore, Chirp

Supporting Data-Intensive Distributed Computing in an Exascale Era
Future Storage System Architecture for Extreme Scale HEC

Network Fabric

Compute & Storage Resources

NAS
Parallel File System
Network Link(s)
Some Challenges to Overcome at Exascale Computing

• Programming paradigms
  – HPC is dominated by MPI today
  – Will MPI scale another 3 orders of magnitude?
  – Other paradigms (including loosely coupled ones) might emerge to be more flexible, resilient, and scalable

• Storage systems will need to become more distributed to scale ➔ Critical for resilience of HPC

• Network topology must be used in job management, data management, compilers, etc

• Power efficient compilers and run-time systems
Expected checkpointing cost and MTTF towards exascale

- System MTTF (hours)
- Checkpoint Overhead (hours)

System Scale (# of Nodes):
- 65,536
- 131,072
- 262,144
- 524,288
- 1,048,576
Simulation application uptime towards exascale

Application Uptime %

Scale (# of nodes)

- No Checkpointing
- Checkpointing to Parallel File System
- Checkpointing to Distributed File System

2000
BG/L
1024 nodes

2007
BG/L
106,496 nodes

2009
BG/P
73,728 nodes

2019
~1,000,000 nodes
Decentralization is critical
- Computational resource management (e.g. LRMIs)
- Storage systems (e.g. parallel file systems)

Preserving locality is critical!
- POSIX I/O on shared/parallel file systems ignore locality
- Data-aware scheduling coupled with distributed file systems that expose locality is the key to scalability over the next decade

Co-locating storage and compute is GOOD
- Leverage the abundance of processing power, bisection bandwidth, and local I/O

Supporting Data-Intensive Distributed Computing in an Exascale Era
Critical Technologies Needed to achieve Extreme Scales

- Fundamental Building Blocks (with a variety of resilience and consistency models)
  - Distributed hash tables (aka NoSQL data stores)
  - Distributed Message Queues
- Deliver future generation distributed systems
  - Global File Systems, Metadata, and Storage
  - Job Management Systems
  - Workflow Systems
  - Monitoring Systems
  - Provenance Systems
  - Data Indexing

Supporting Data-Intensive Distributed Computing in an Exascale Era
More Information

• More information:
  – http://www.cs.iit.edu/~iraicu/
  – http://datasys.cs.iit.edu/

• Contact:
  – iraicu@cs.iit.edu

• Questions?