Distributed File Systems for Exascale Computing

Dongfang Zhao
Department of Computer Science
Illinois Institute of Technology
dongfang.zhao@hawk.iit.edu

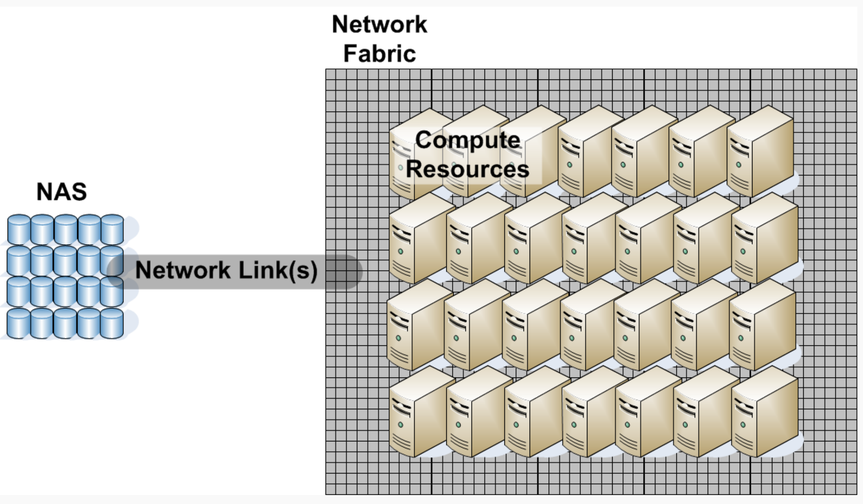
Ioan Raicu
Department of Computer Science
Illinois Institute of Technology
iraicu@cs.iit.edu

Goal

Develop both theoretical and practical aspects of building distributed files systems scalable to exascale supporting millions of nodes and billions of concurrent IO requests

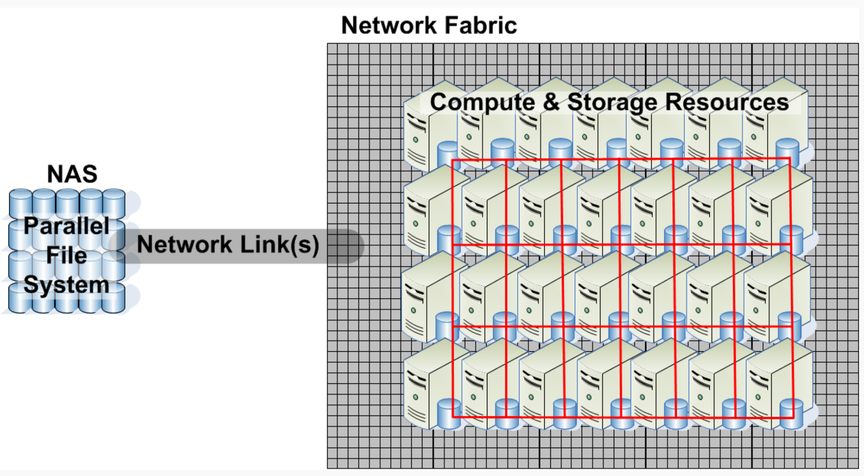
Motivation

Current architecture (i.e. compute nodes are remotely connected to storage nodes) would unlikely scale well at exascale



Proposed Architecture

- Distribute data into local persistent storage to explore data locality for computation
- Distribute metadata into local persistent storage to remove the bottleneck of centralized metadata management
- Coexist with remote parallel file systems



Building Blocks

- ZHT: distributed metadata management system
- HyCache: SSD/HDD caching
- IDAStore: GPU-based coding for data redundancy
- FFSNET: light-weighted data transfer protocol
- PAFS: provenance-aware distributed file system

References

[1] FusionFS project website: http://datasys.cs.iit.edu/projects/FusionFS/index.html

[2] Ioan Raicu, Ian Foster and Pete Beckman. Making a Case for Distributed File Systems at Exascale, *ACM Workshop on Large-scale System and Application Performance (LSAP)*, 2011

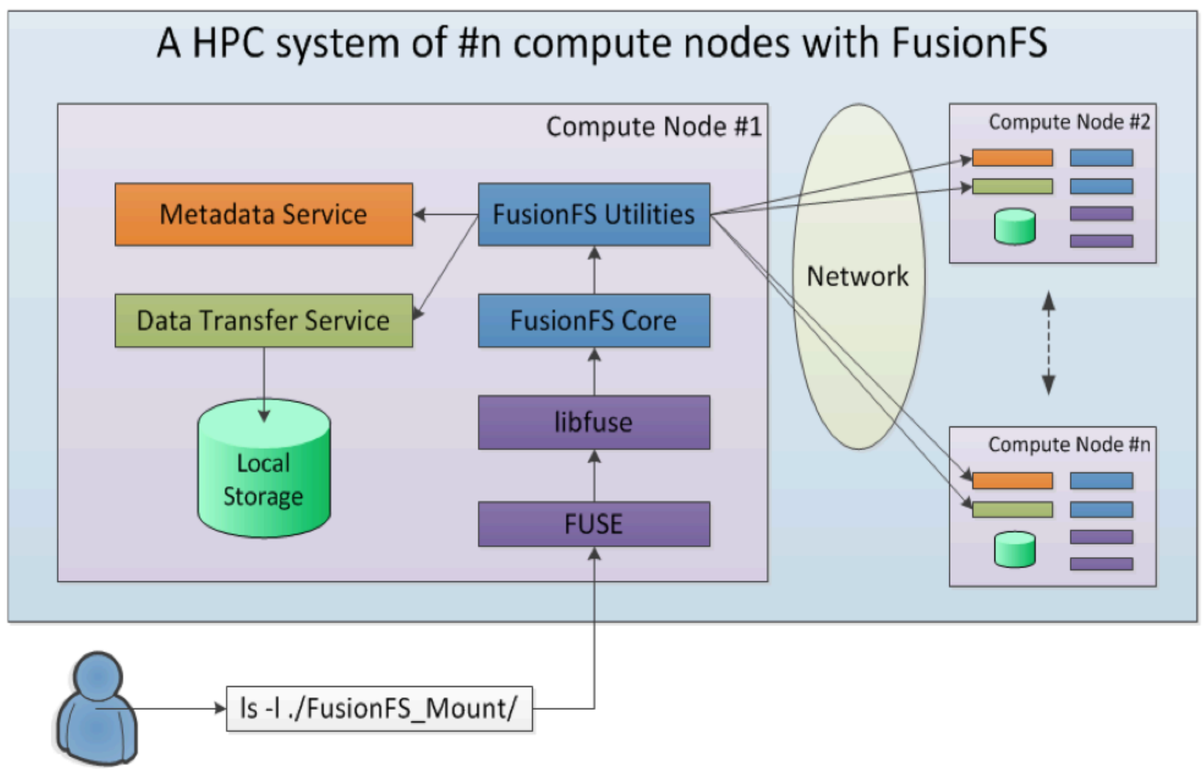
Acknowledgement

This work is supported by NSF grant OCI-1054974

FusionFS Overview

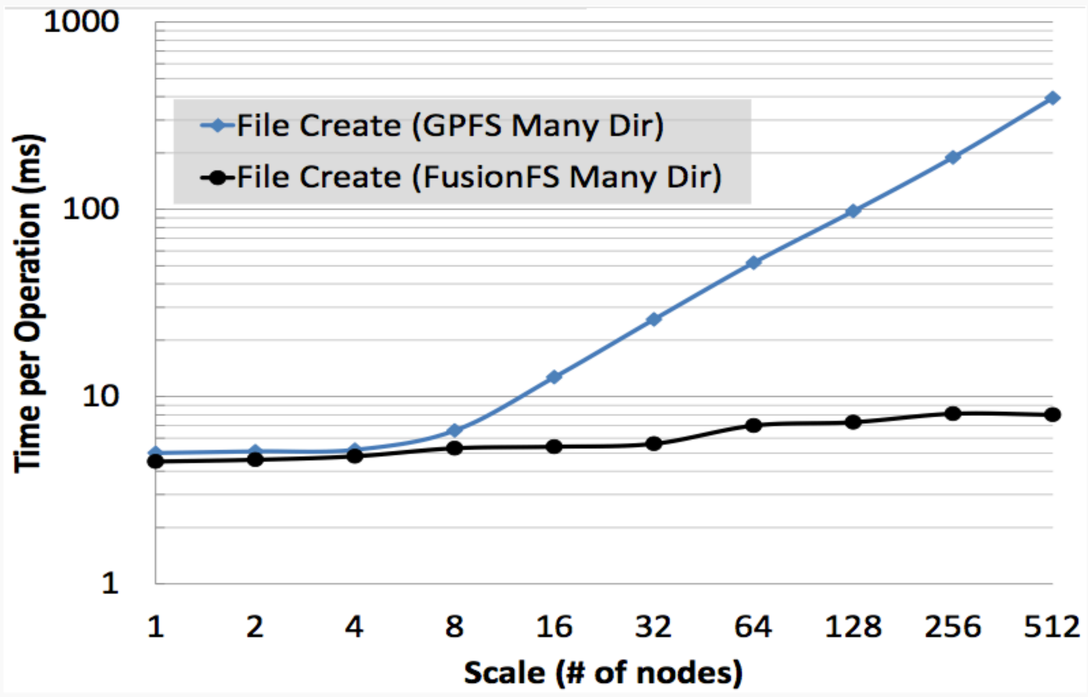
Features

- Distributed Metadata Management
- Distributed Data Management
- Data Indexing
- Relaxed Semantics
- Data Locality
- Overlapping I/O with Computations
- POSIX



Access Patterns

- ❑ **1-many read** (all processes read the same file and are not modified)
- ❑ **many-many read/write** (each process read/write to a unique file)
- ❑ **write-once read-many** (files are not modified after it is written)
- ❑ **append-only** (files can only be modified by appending at the end of files)
- ❑ **metadata** (metadata is created, modified, and/or destroyed at a high rate).



Metadata performance on IBM Bluegene/P

Current Status

- FusionFS prototype with POSIX has been developed
- FusionFS has been deployed on:
  - Linux cluster (512-cores)
  - IBM Bluegene/P (2048-cores)
- Benchmarks tested:
  - IOZone and IOR
  - Metadata: Excellent scalability (4ms@1-node → 7ms@512-nodes)

Next Release (before 2013)

- Support asynchronous file writes
- Test with real scientific applications
- Scale to 32K-cores

Long-Term Plan (before 2016)

- Scale to 1 million nodes
- Support fault tolerance with replications and erasure coding
- Improve FUSE performance and/or develop FusionFS kernel module