UNM SCHOOL of ENGINEERING
Department of Computer Science

A peer-to-peer architecture for supporting dynamic shared libraries in large-scale systems

Matthew G. F. Dosanjh, Patrick G. Bridges
Suzanne M. Kelly, James H. Laros III

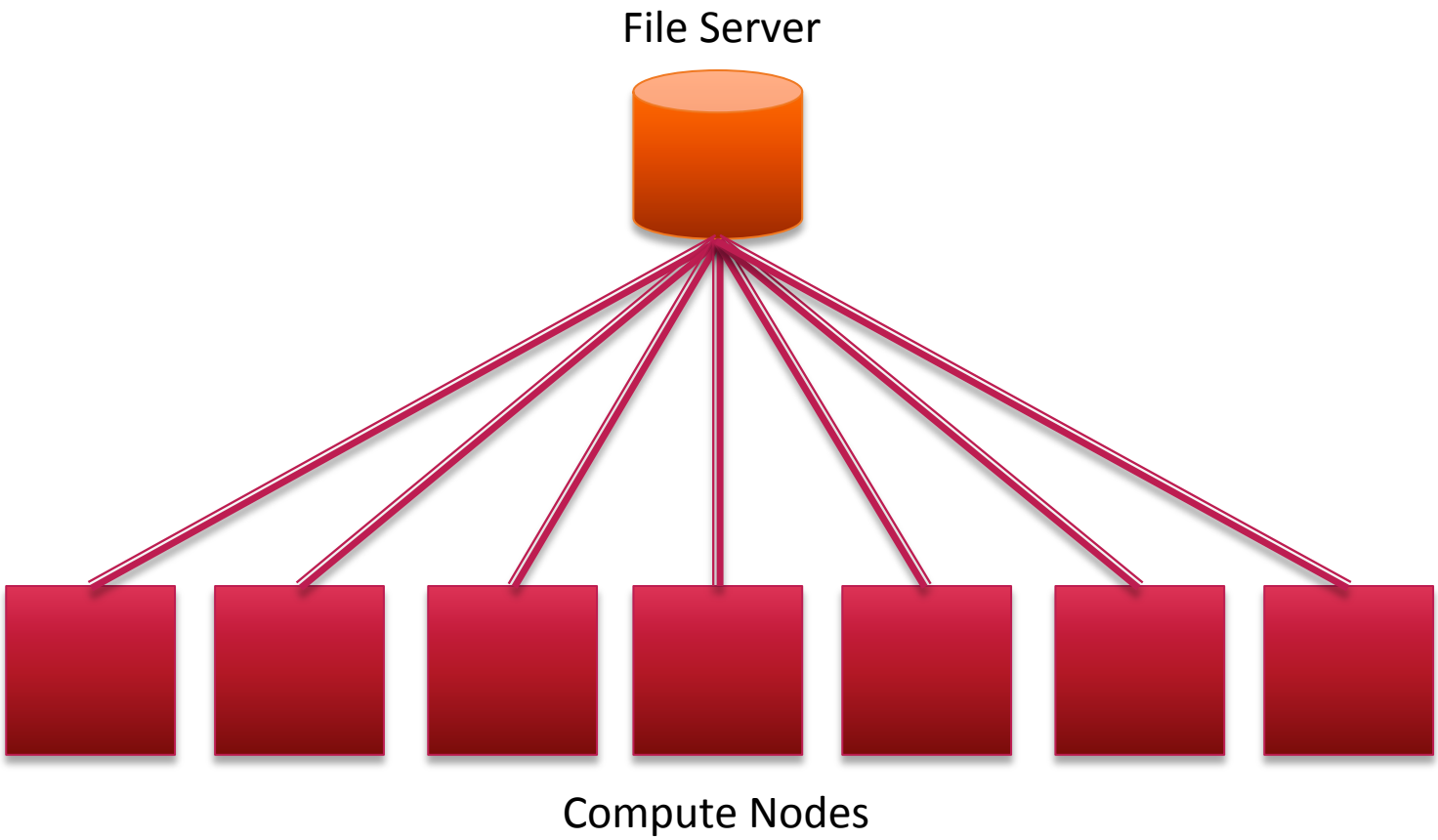
Distributed Techniques & HPC

- Larger, more data intensive systems have communication bottlenecks
- Cause severe performance issues
- Distributed techniques have dealt with these problems
- Can we use these techniques in an HPC context?

Example Problem

- Increasing demand for Dynamically Shared Libraries (DSLs) on HPC platforms
- DSLs load from shared storage
- Synchronized read requests

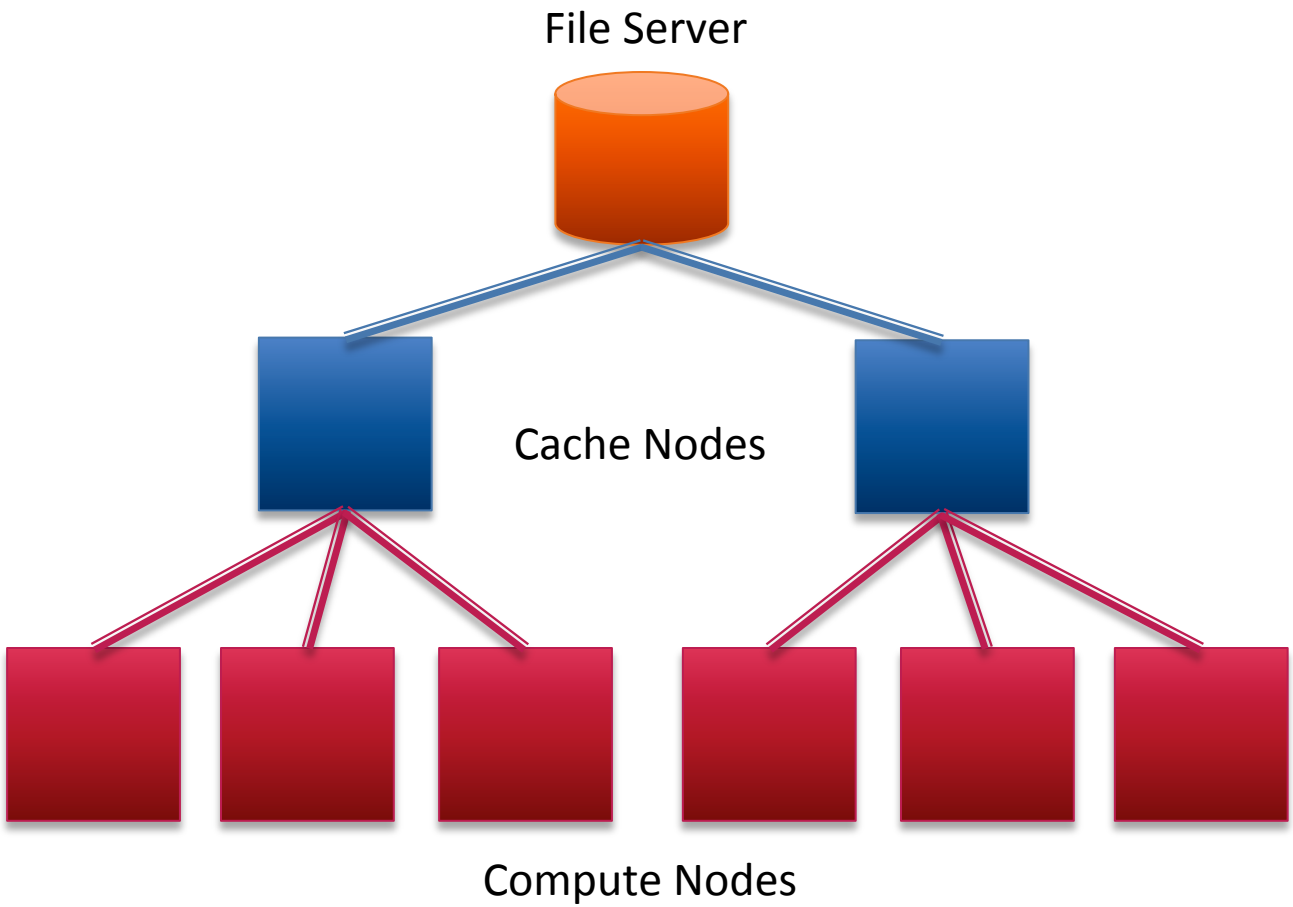
Naive Case - Data Flow



Example Problem

- Increasing demand for Dynamically Shared Libraries (DSLs) on HPC platforms
- DSLs load from shared storage
- Synchronized read requests
- Naïve solution bottlenecks the file server

Cache Nodes - Data Flow



Example Problem

- Increasing demand for Dynamically Shared Libraries (DSLs) on HPC platforms
- DSLs load from shared storage
- Synchronized read requests
- Naïve solution bottlenecks the file server
- Caching data on service nodes is not ideal
  - Requires extra service nodes
  - Can take ~30 minutes on 16k processing elements

P2P File Sharing Has Done That

- Technique for data distribution
- Uses clients' copy of data to service requests
- Reduces dependence on a centralized server
- Examples: Napster, Skype, BitTorrent

BitTorrent for DSL Distribution

- A two part architecture
  - Trackers store and host peer information
  - Nodes find peers by connecting to tracker
- Almost ideal for DSL distribution
  - Open Source
  - Library Support
  - Private Network Support
  - Extensible protocol
    - Peer Exchange
    - Magnet Links

BitTorrent Challenges

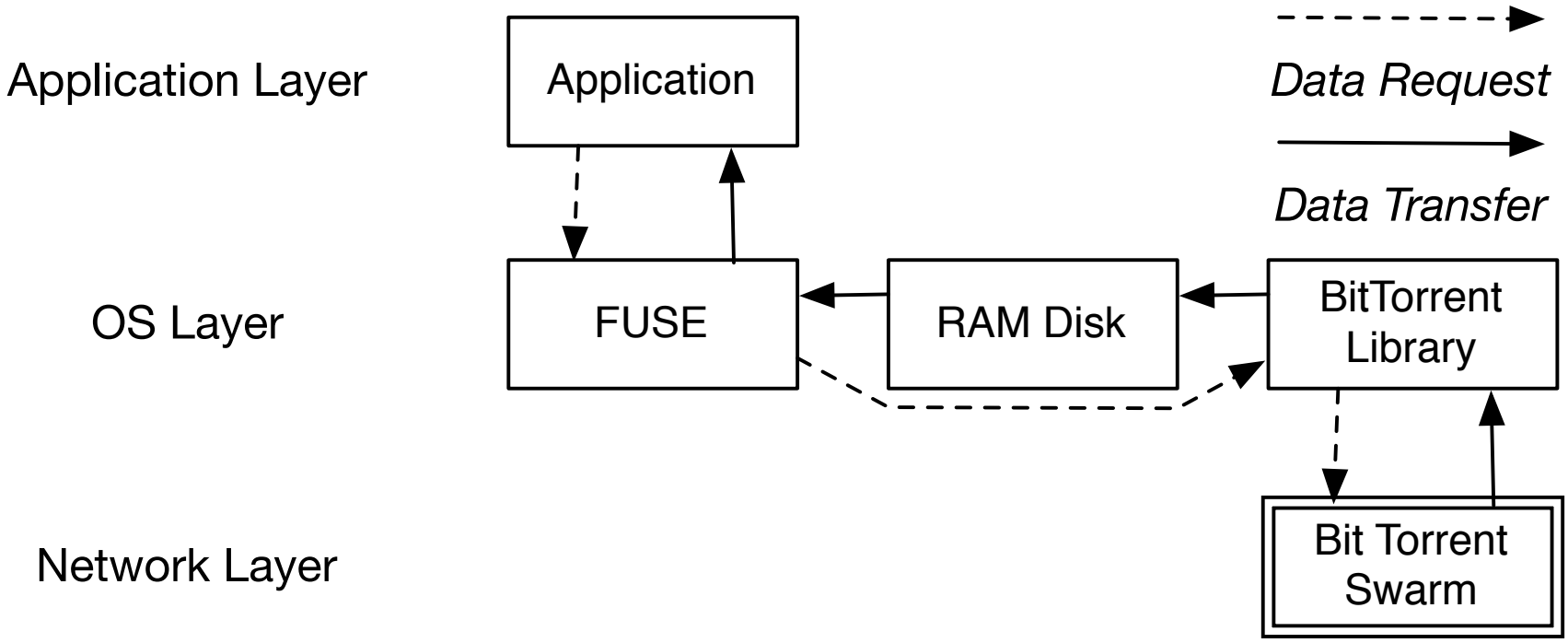
- Designed for untrusted networks
  - Peer Throttling
  - Up/Down Ratios
  - Upload Limit
- Designed for distributed connections
  - Overhead of Protocol
  - File Availability Information

- These problems can be fixed!

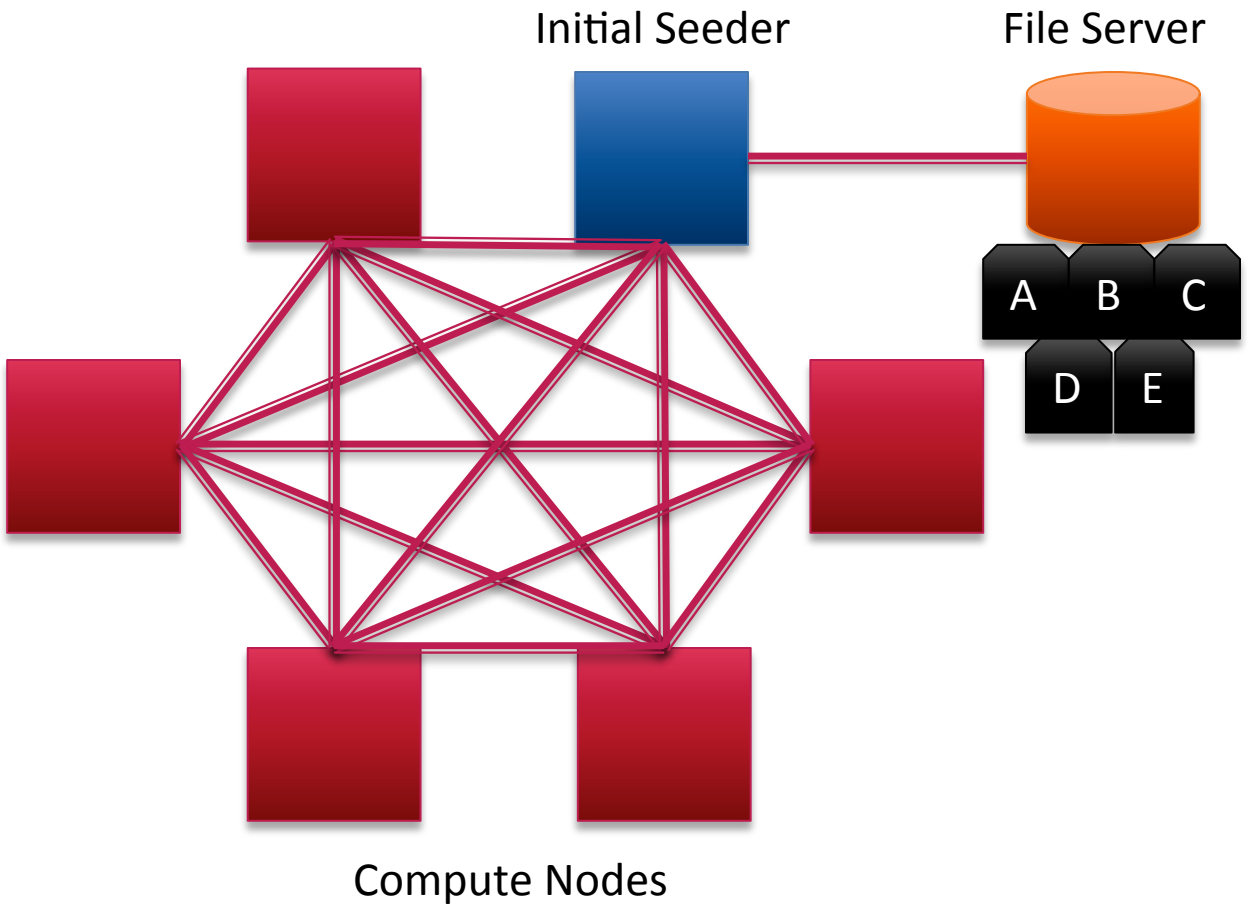
BitTorrent Set-Up

- User specifies a set of libraries
- Setup tool creates and registers a description file
- Launcher
  - packages and distributes the description file
  - sets up seeder nodes
    - Requests service nodes
    - Pushes data onto service nodes
    - Starts seeder client
  - Mounts FUSE file system on compute nodes

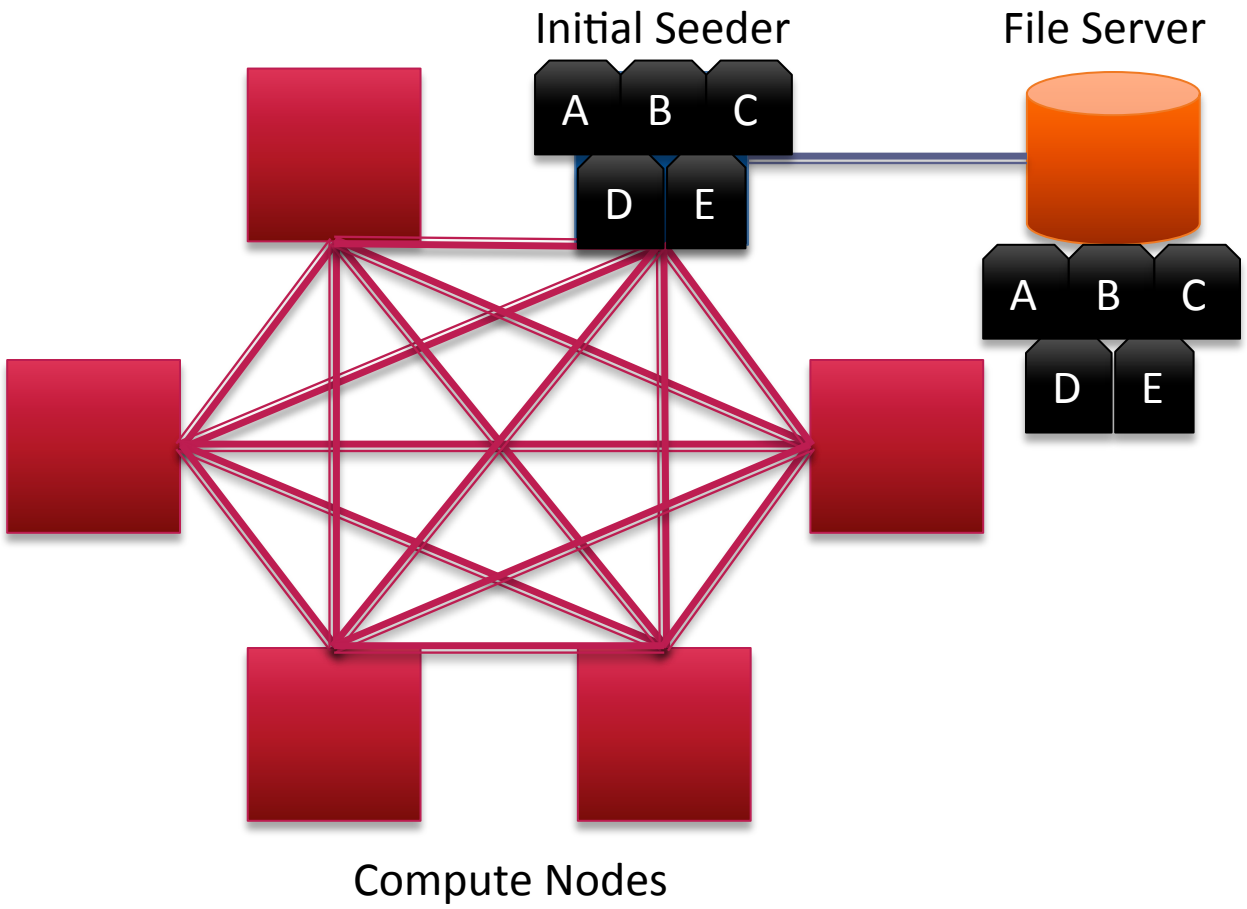
Architecture



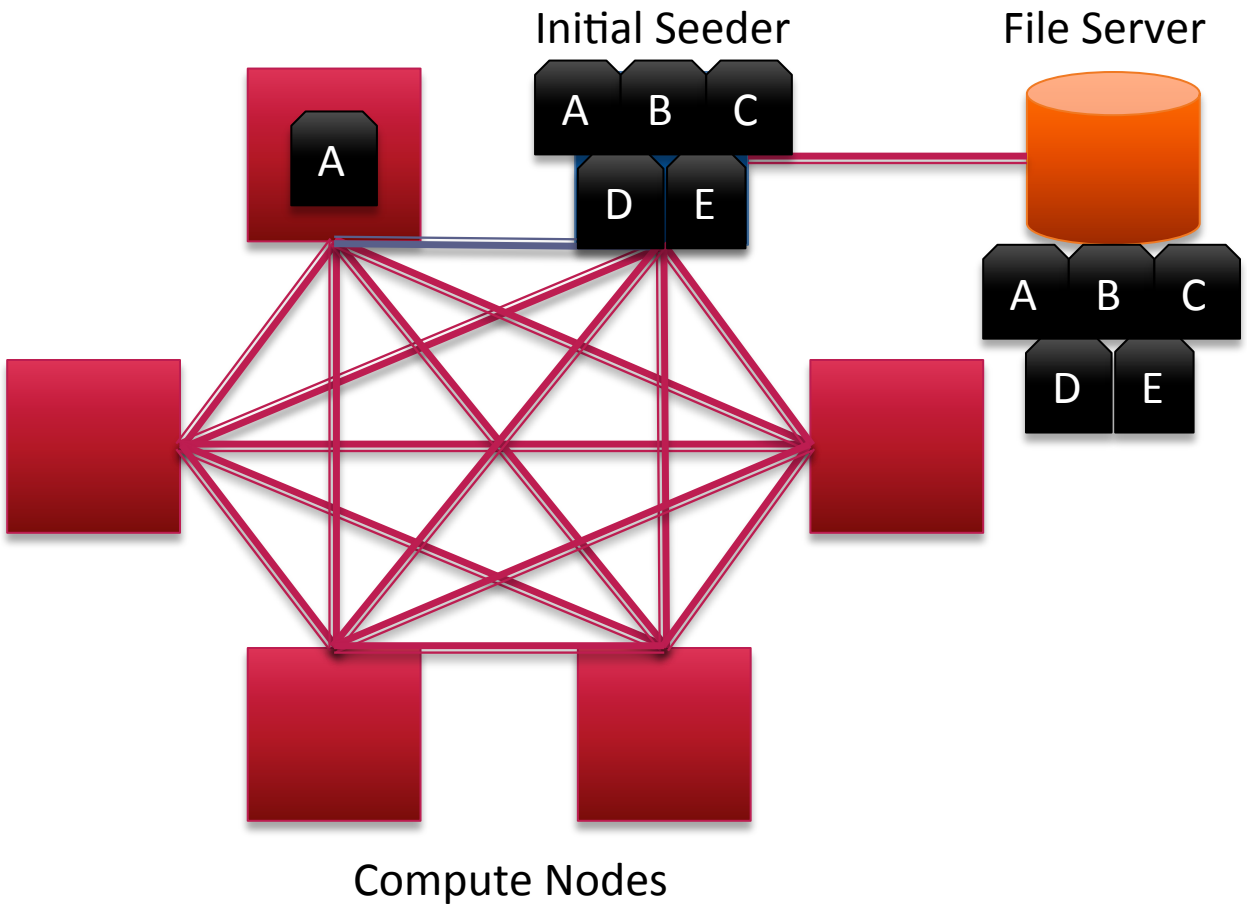
BitTorrent - Data Flow



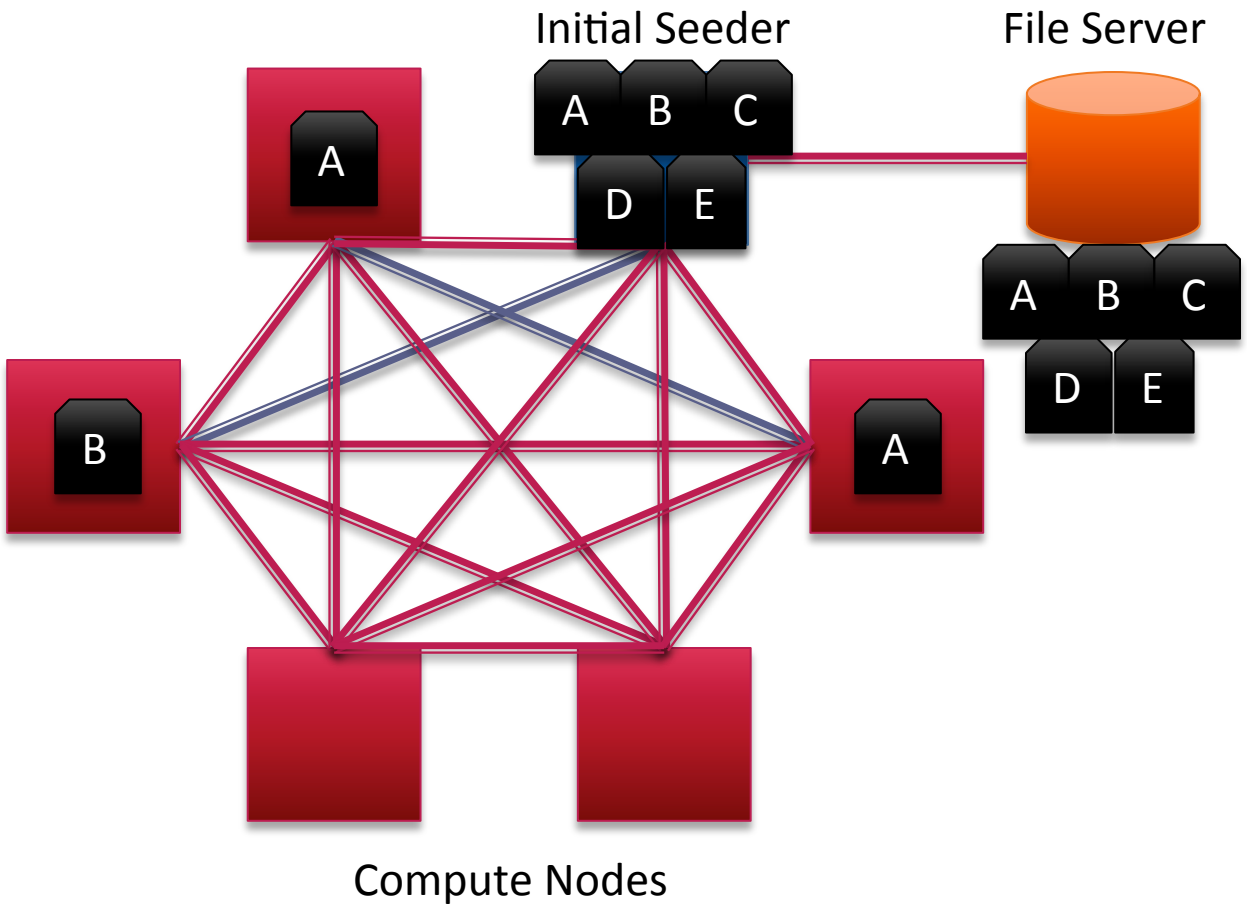
BitTorrent - Data Flow



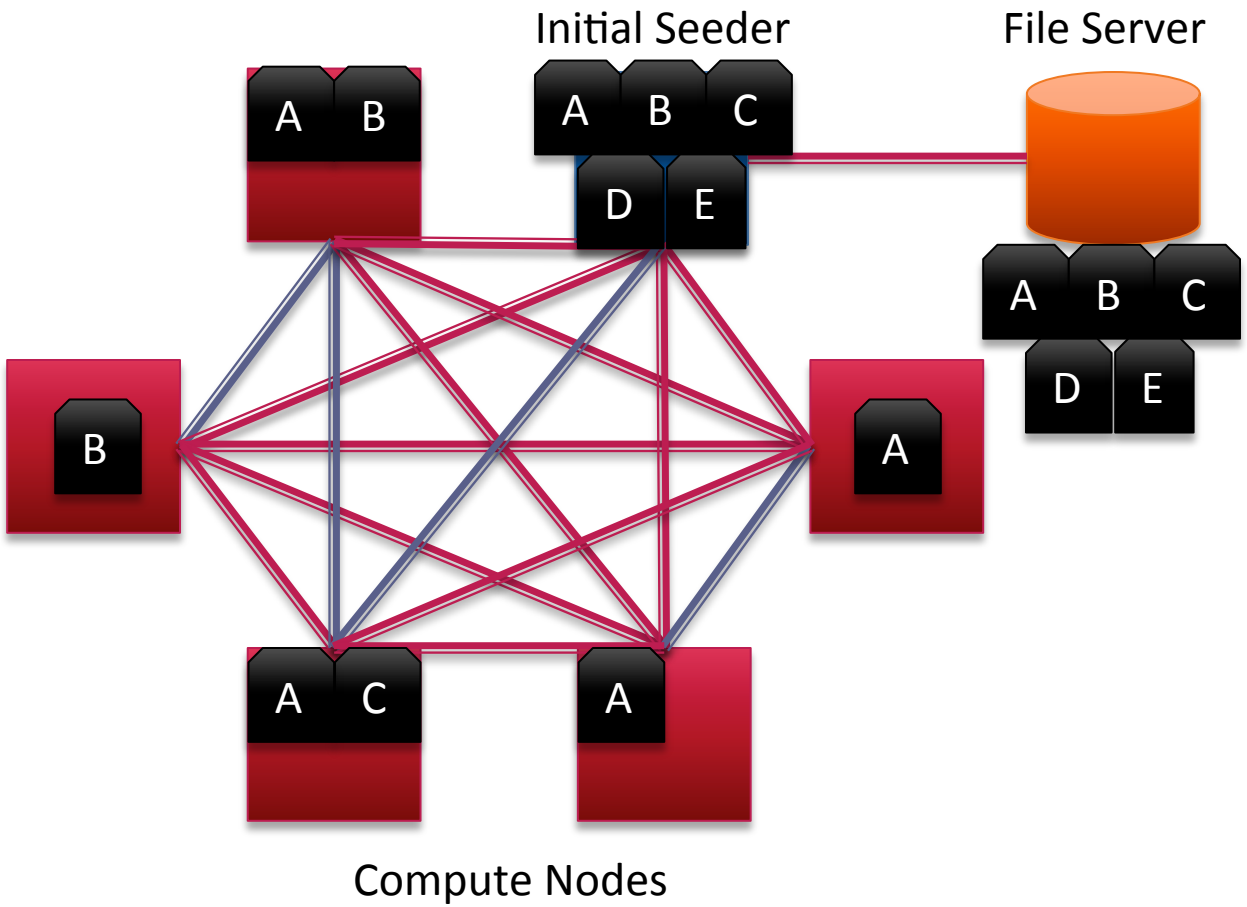
BitTorrent - Data Flow



BitTorrent - Data Flow



BitTorrent - Data Flow



Proof Of Concept

- Implemented working prototype using cTorrent
- Copies data to local RAMDISK.
- The system invokes cTorrent's command line interface
- Several limitations
  - Extra layer of indirection
  - Extra time spent opening and closing torrent system
  - Harder to optimize
  - Keeps multiple copies of data in memory

Proof of Concept – It Works

- Ten Nodes
- Simultaneous Requests
- Each Test Doubled Size

Example Problem - Continuing Work

- Optimization of cTorrent based system
- Build a more closely coupled system using libTorrent
- Larger scale testing

Future Work

- Other problems distributed computing has solved
  - Writable Filesystems - NoSQL
  - Power Management - Data Centers
  - Etc.
- Can we adapt these to HPC?

Acknowledgements

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 (NNSA) under contract DE-AC04-94AL85000.

This work was funded by NNSA's Advanced Simulation and Computing (ASC) Program.

Questions?