Will They Blend?: Exploring Big Data Computation atop Traditional HPC NAS Storage

Ellis H. Wilson III [1,2] Mahmut Kandemir [1] Garth Gibson [2,3]

[1]Department of Computer Science and Engineering,
The Pennsylvania State University

[2]Panasas, Inc.

[3]Department of Computer Science,
Carnegie Mellon University

July 3rd, 2014

Before We Begin: Get the Slides and Paper

Slides and Paper are Available At:

www.ellisv3.com

1. Introduction and Background
   - From 10,000 Feet: Considering Hadoop's Fit in HPC
   - Goals of this Research: MapReduce in HPC?
2. Converged Architectures for Hadoop on NAS
   - Overview of Architectures
   - Reliability and Performance Implications
   - RainFS
3. Performance Evaluation of Converged Architectures
   - Setup and Benchmarks
   - Performance Results

Motivation

- Divide between HPC and Big Data is increasingly foggy
  - Big Data processing framework MapReduce (MR) promises faster time-to-solution for data-intensive science
  - But MR often comes tightly coupled with the Hadoop Distributed File System (HDFS)
  - Standard HDFS requires local disks to the compute for distributed storage
- HPC typically already has it's own Parallel File System (PFS) solutions in place
  - Using Hadoop threatens to require large capital and maintenance investments
  - Totally dropping MPI and similar solutions for MR is impossible
  - Copying massive amounts of data from Network-Attached Storage (NAS) to HDFS and back is a common problem
  - Dividing your storage into two pools, NAS and HDFS, will exacerbate the Compute-Storage gap

Hurdles to Adoption of Hadoop in HPC

- Loss of Infrastructure Consolidation
- Forced Import/Export
- I/O Performance Degradation
- Loss of High-Availability
- No Modification to Files
- Inefficient Compute-Storage Coupling

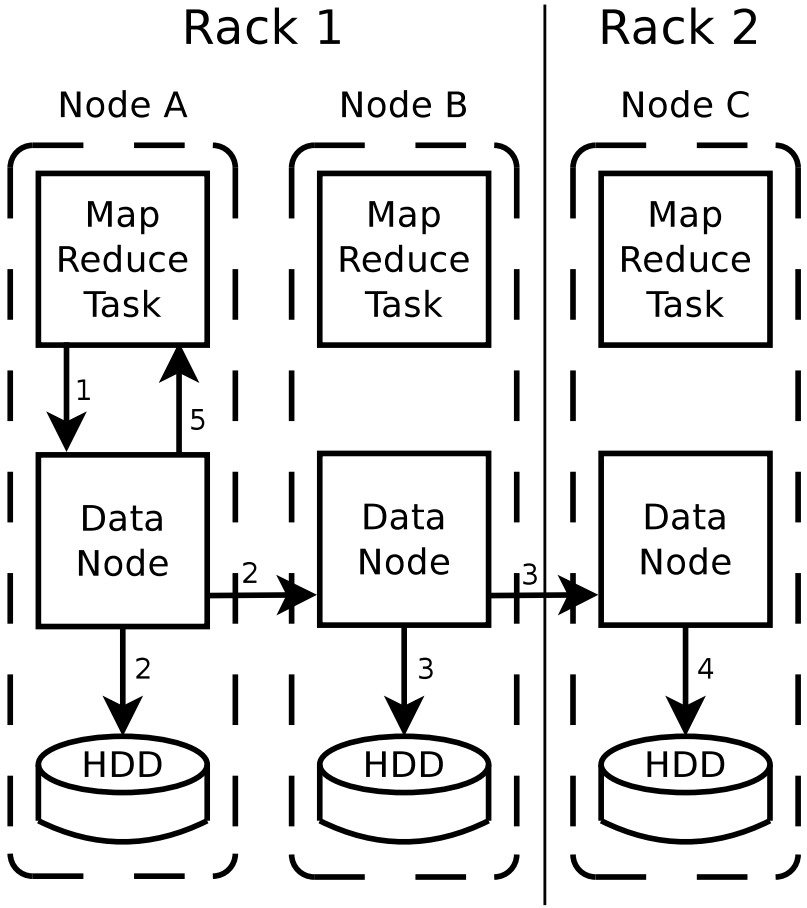
Goals of this Research

Three Main Goals/Contributions:

1. *Explore if/how one can enable MR to run on traditional NAS*
   - Enables reuse of existing storage – infrastructure consolidation
2. *Explore whether one can use MR alongside MPI and others without copying*
   - Improves utility of capacity, reduces network contention, fights the I/O Gap
3. *Identify the relative efficiencies and reliabilities of potential solutions*
   - Examine four different architecture approaches

First: Consider Traditional Hadoop

Typical Hadoop Architecture: Example of Write Path



Exploration of Four Possible Architectures

Possible Architectures:

- *Traditional* HDFS Pointed at a PFS
  - Configure HDFS with PFS paths rather than to local disks
- *HDFS as a Wire Protocol* in the PFS NAS Heads
  - Run DataNodes (DNs) on NAS heads instead of all clients
- *No HDFS*, MR Directly to the PFS
  - Run MR configured to send data directly to PFS
- *RainFS*: Replicating Array of Independent NAS File System
  - New Hadoop Filesystem designed specifically to intermediate between MR and PFS

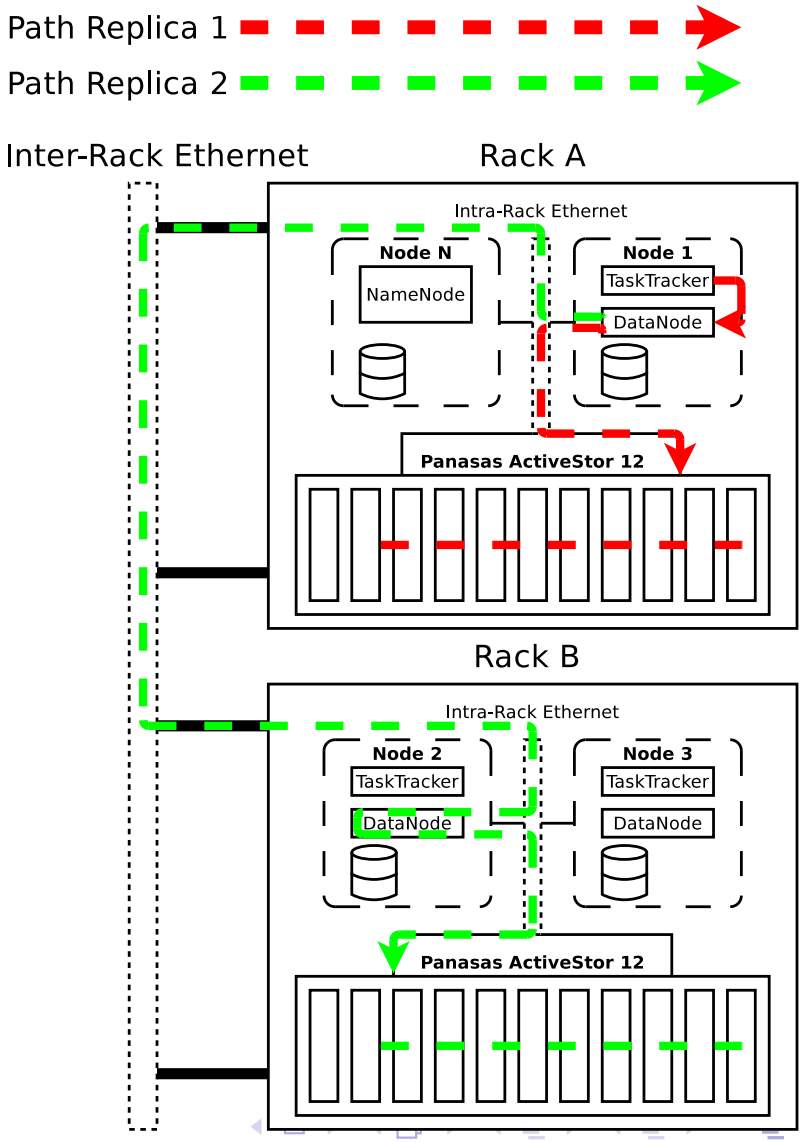
Architecture Details: Traditional HDFS

Pros:

- Simplicity

Cons:

- Performance Degradation: One full replica in network contention
- Reliability Limits: Duplication is the ceiling
- Copy Required: Distinct namespace



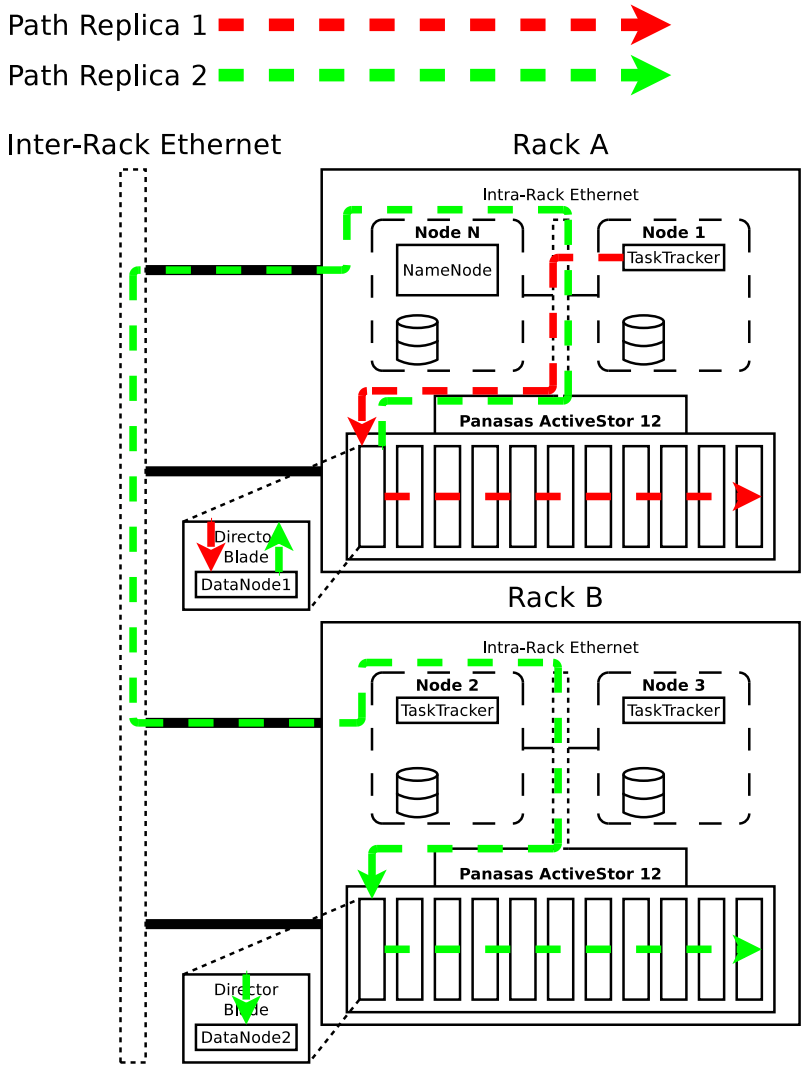
Architecture Details: HDFS as a Wire Protocol

Pros:

- HDFS becomes Yet Another Protocol
- Reliability limits go away

Cons:

- Performance Bottleneck: NAS Head limits throughput
- NAS Invasion: May not be possible (easy) with many NAS solutions
- Copy Required: Distinct namespace



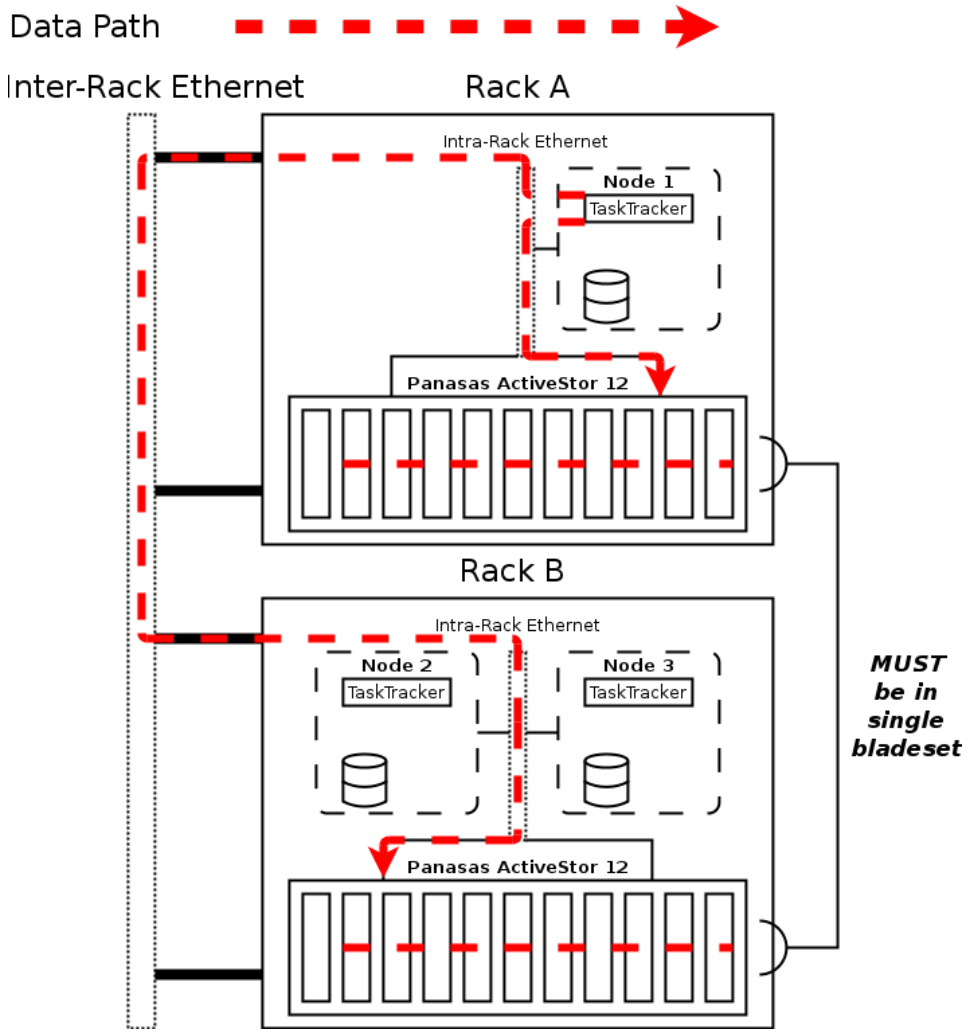
Architecture Details: No HDFS

Pros:

- High-Performance: Alleviates overheads and bottlenecks
- No Copies: Operates on typical POSIX namespace

Cons:

- Requires Single Namespace: No HDFS to intermediate between distinct NAS
- No Replication: Must tolerate solely RAID



Hadoop vs. HPC Storage: A Reliability Divergence

HPC Storage

- Enterprise storage solutions
- RAID 5/6
- ECC-enabled hardware (sometimes end-to-end)
- Redundant hardware (PSU/NIC/etc)

Hadoop Storage (HDFS)

- Commodity hard drives in compute nodes
- Replication performed across nodes/racks
- No ECC
- No Redundant hardware

Converged Reliability Guarantees

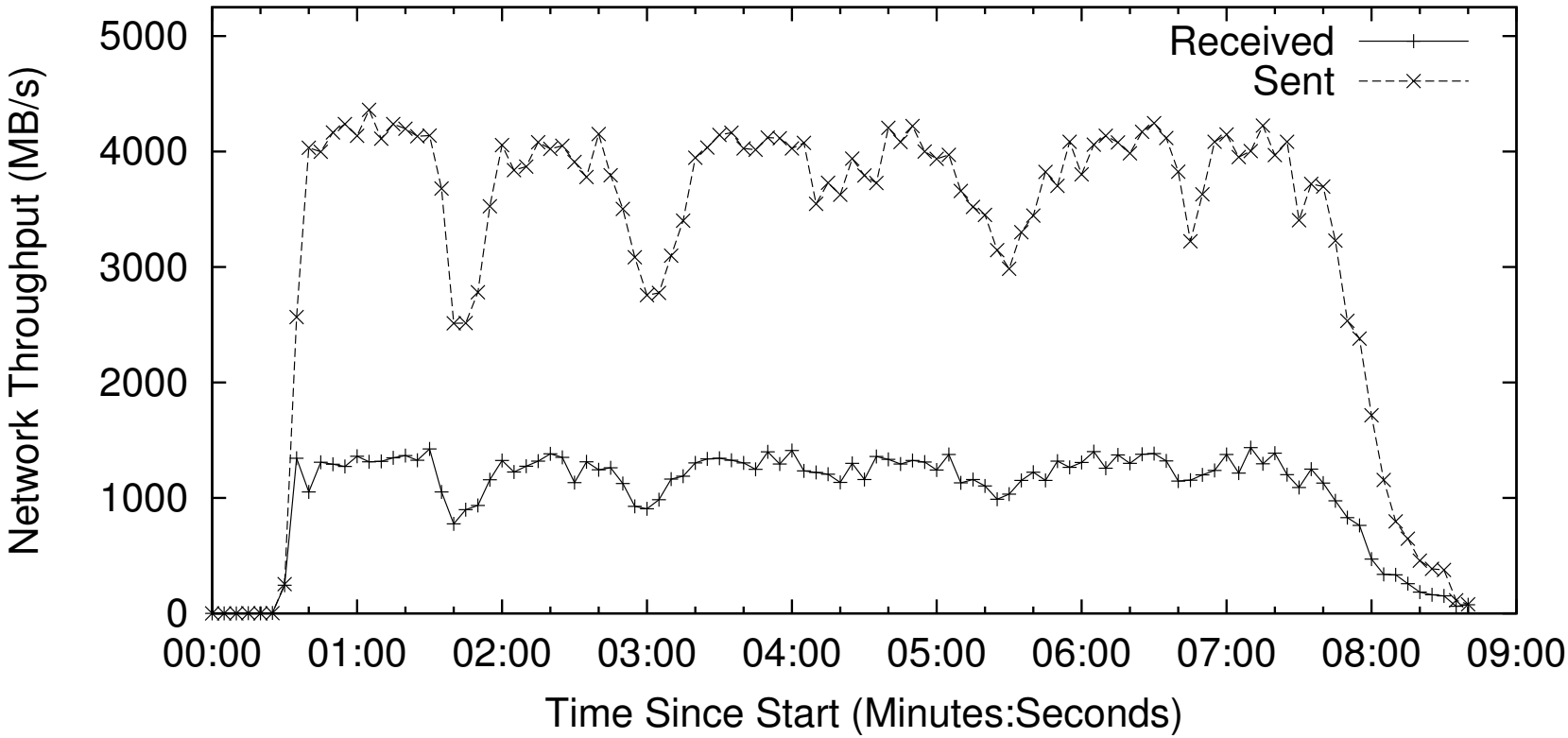
	RAID 5	RAID 6	RAID 5	RAID 6	RAID 5	RAID 6
	Repl. 1	Repl. 1	Repl. 2	Repl. 2	Repl. 3	Repl. 3
DN-on-Client	1 / 0	2 / 0	3 / 1	5 / 1	– / –	– / –
DN-on-NAS Node	1 / 0	2 / 0	3 / 1	5 / 1	5 / 2	8 / 2
No HDFS	1 / 0	2 / 0	– / –	– / –	– / –	– / –
RainFS	1 / 0	2 / 0	3 / 1	5 / 1	5 / 2	8 / 2

Two main failure modes for converged HDFS/HPC storage:

- Failure of a disk
- Failure of a rack

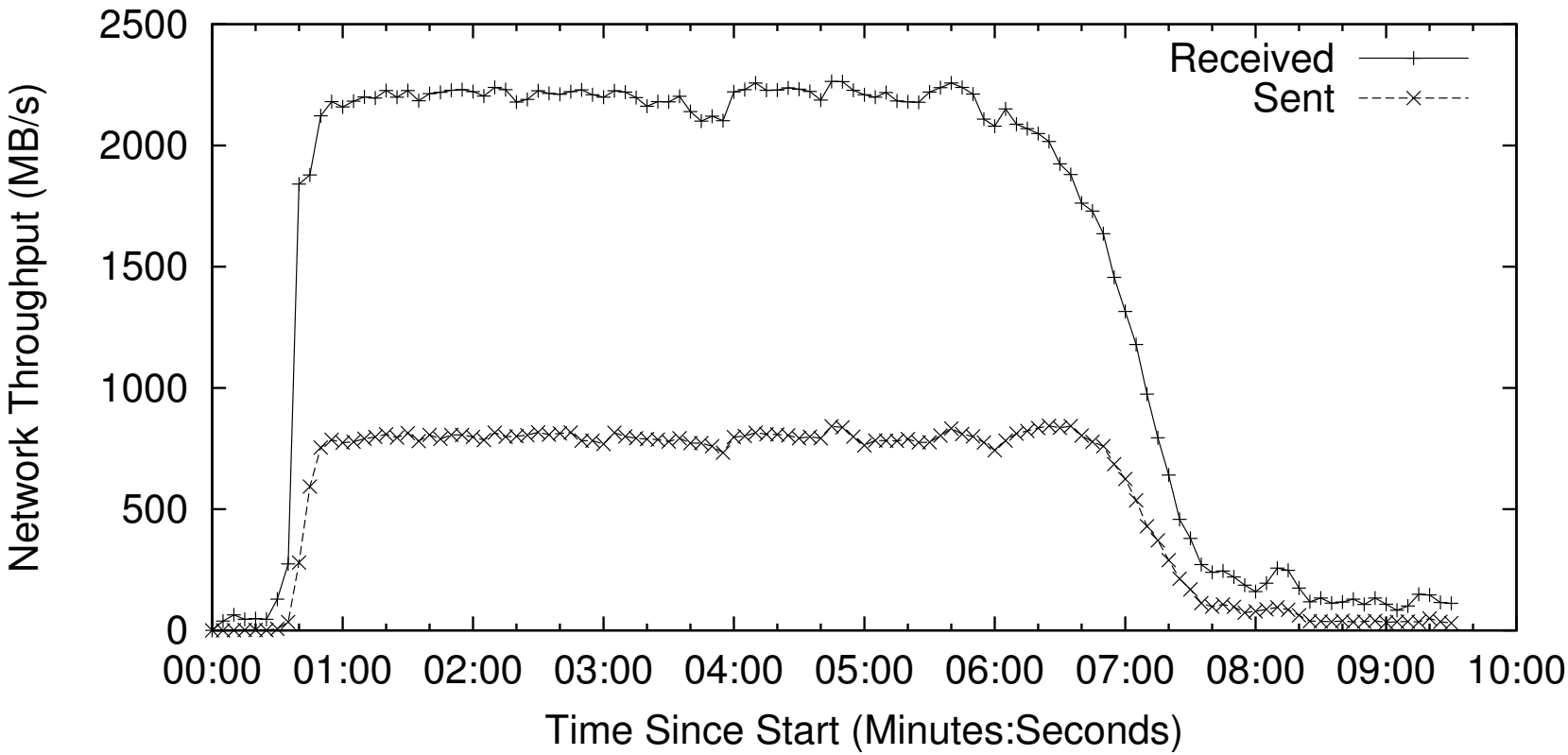
Locality Confusion: Write Transport

Errant Pass-Through Behavior on Write



Read Transport

Errant Pass-Through Behavior on Read



Design Desirata

Four main goals for RainFS:

1. **Client-Level Federation of NAS Systems:** Enable performance of all available NAS systems concurrently *and* maintain discrete failure domains
2. **Full Replication:** Restore replication ability in MapReduce
3. **No Data Pass-Throughs:** Writes/Reads should never go through another client node.
4. **A Fair Namespace:** Create a framework-agnostic namespace where no imports or exports are required.

Main Implementation Mechanisms

Symbolic Links (symlinks)

- Symlinks on master failure domain are pointed at replica zero on one of the NAS systems
- Placement of replica zero is randomly chosen, following replicas are round-robined
- MR can read from MPI output; MPI can read from MR output
- Key algorithms and their synchronization issues are covered in the paper

Hidden Metadata File

- Beside and named similarly to the symlink
- Manage where replicas exist, up/down state, etc
- Avoids dedicated, centralized metadata manager daemon

Setup and Benchmarks in Use

Hardware Environment

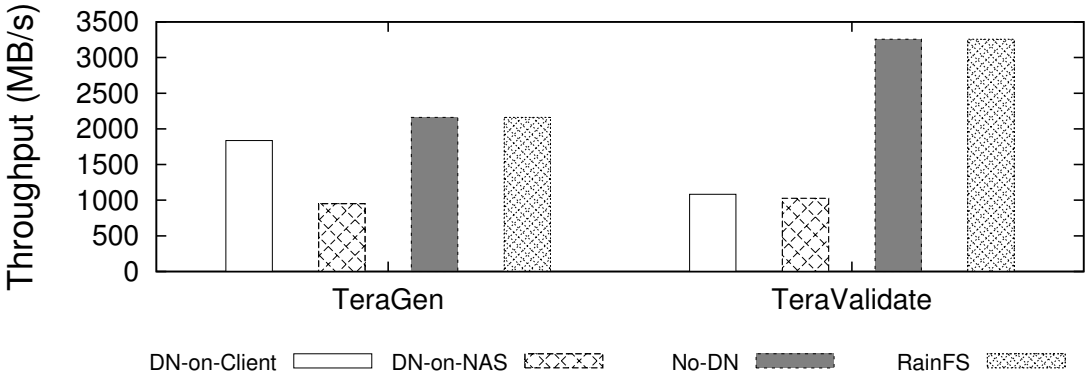
- Cluster of 50 multi-core machines at Carnegie Mellon
- CentOS 5.5 running as VM on KVM
- DirectFlow(tm) network attached protocol to:
- 5 shelves of Panasas ActiveStor 12

Benchmarks in Use

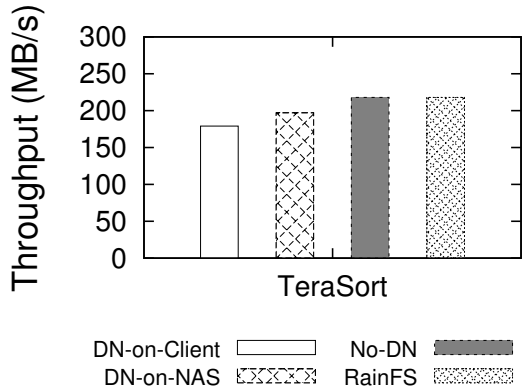
- Ubiquitous Yahoo! TeraSort Benchmark Suite
  - TeraGen - Write-intensive
  - TeraSort - Mixed, CPU-intensive
  - TeraValidate - Read-intensive

Impact of Architecture on Throughput Performance

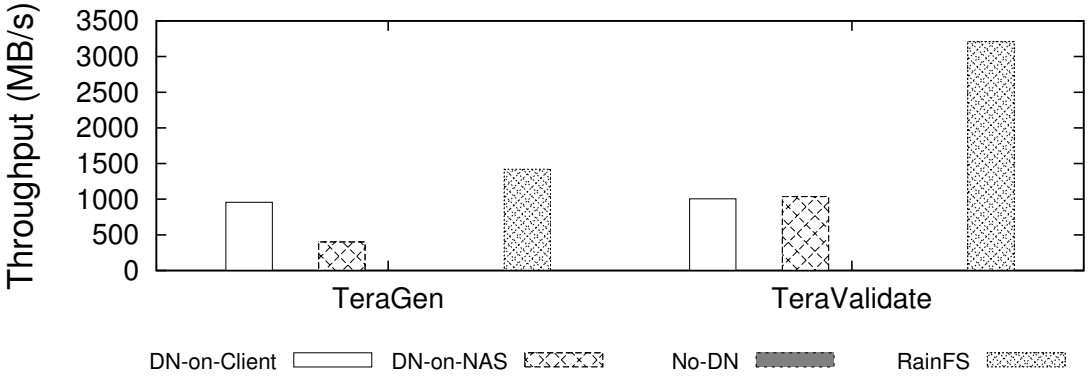
Yahoo! TeraSort Benchmark (50 clients, 500GBs of Data)



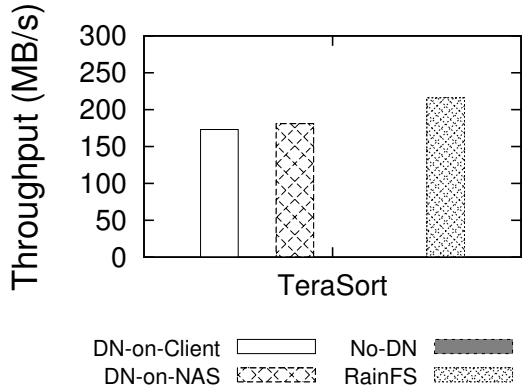
(a) Rep. Level 1: Write- and Read-Intensive



(b) Rep. Level 1: Mixed



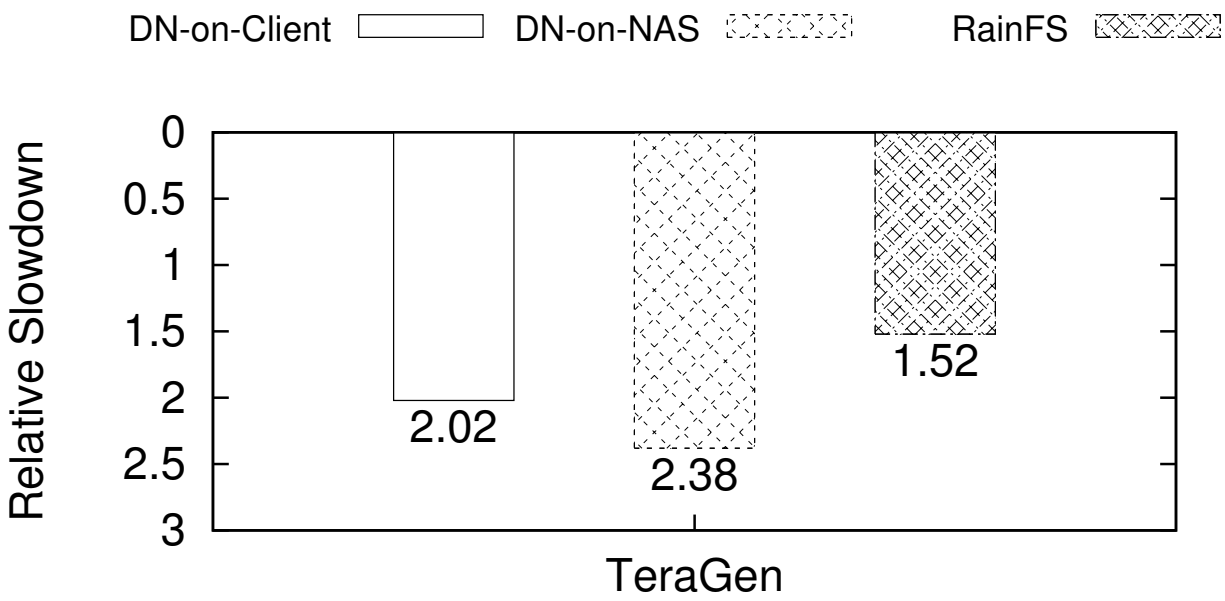
(c) Rep. Level 2: Write- and Read-Intensive



(d) Rep. Level 2: Mixed

Impact of Replication on Performance

Relative Throughput Slow-Downs for Replication per Architecture



Conclusion

Conclusions:

- Convergence of Big Data and HPC is happening – compute is easy, storage is hard
- Numerous pitfalls/caveats, especially relating to reliability
- Four different architectures using Hadoop MapReduce on NAS were explored – each have their own pros/cons
- RainFS demonstrates optimal performance with highest reliability

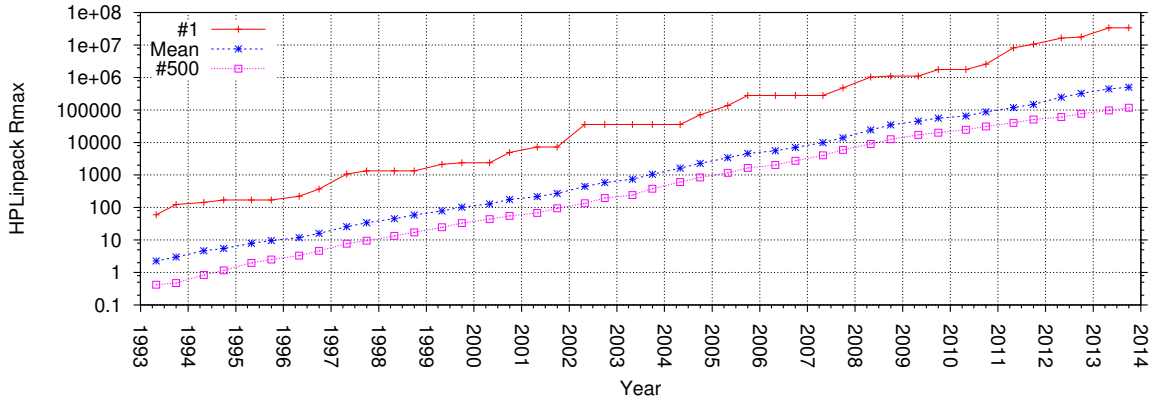
Questions?

Backup Slides

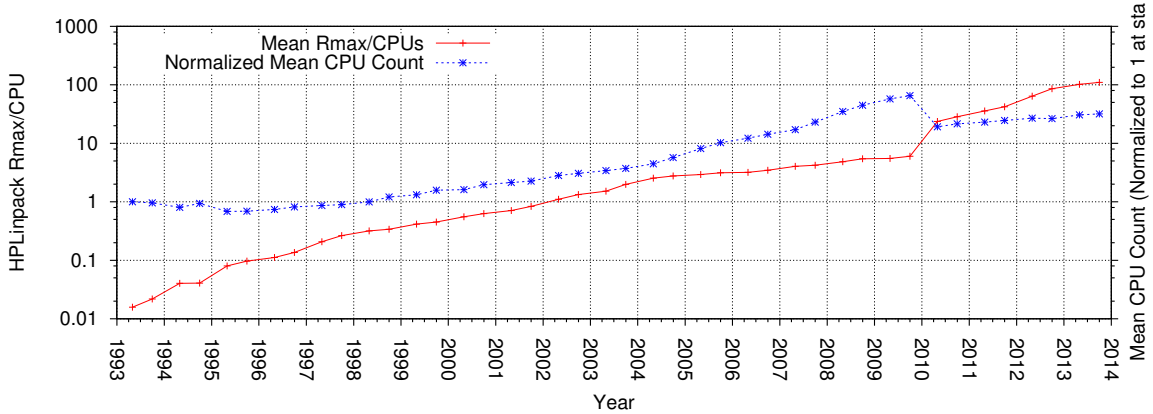
– Begin Backup Slides –

Super-Moore's Law CPU Scaling in Supercomputers

Top 500 Supercomputer HPLinpack Results over Time



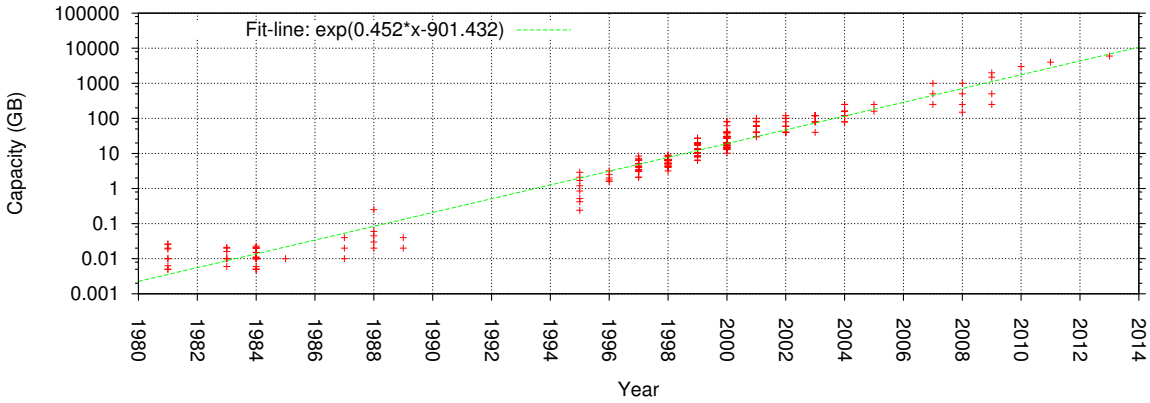
Doubling Performance Every 13.5 Months



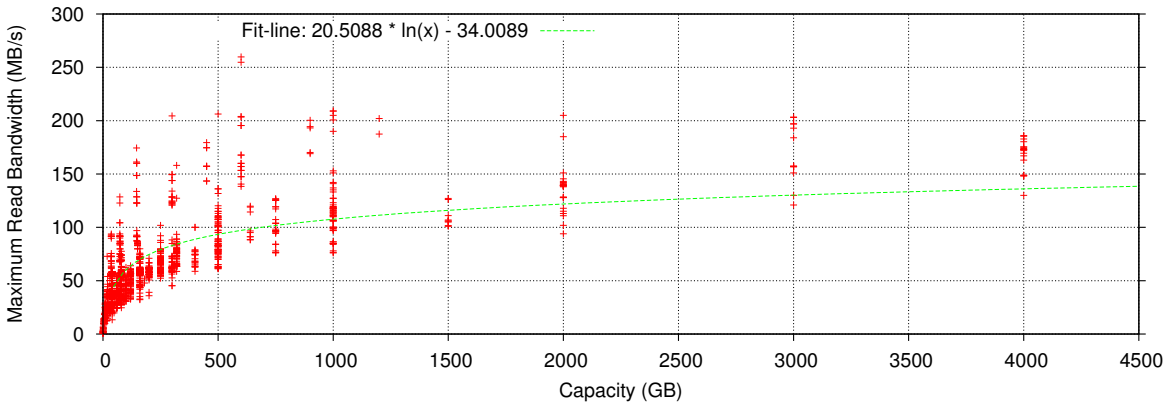
Doubling CPU Count Every 22 Months

HDD Capacity and Bandwidth Scaling

Historical Data from over 1500 Hard Drives



Doubling Capacity every 18 Months



Doubling Bandwidth only once per decade!

Taken In Perspective: A Grim Reality

Time taken to access all on-disk data:

- 1996 - 30 seconds
- 2006 - 30 minutes
- 2016 - **1 day**