# Uncovering Errors: The Cost of Detecting Silent Data Corruption

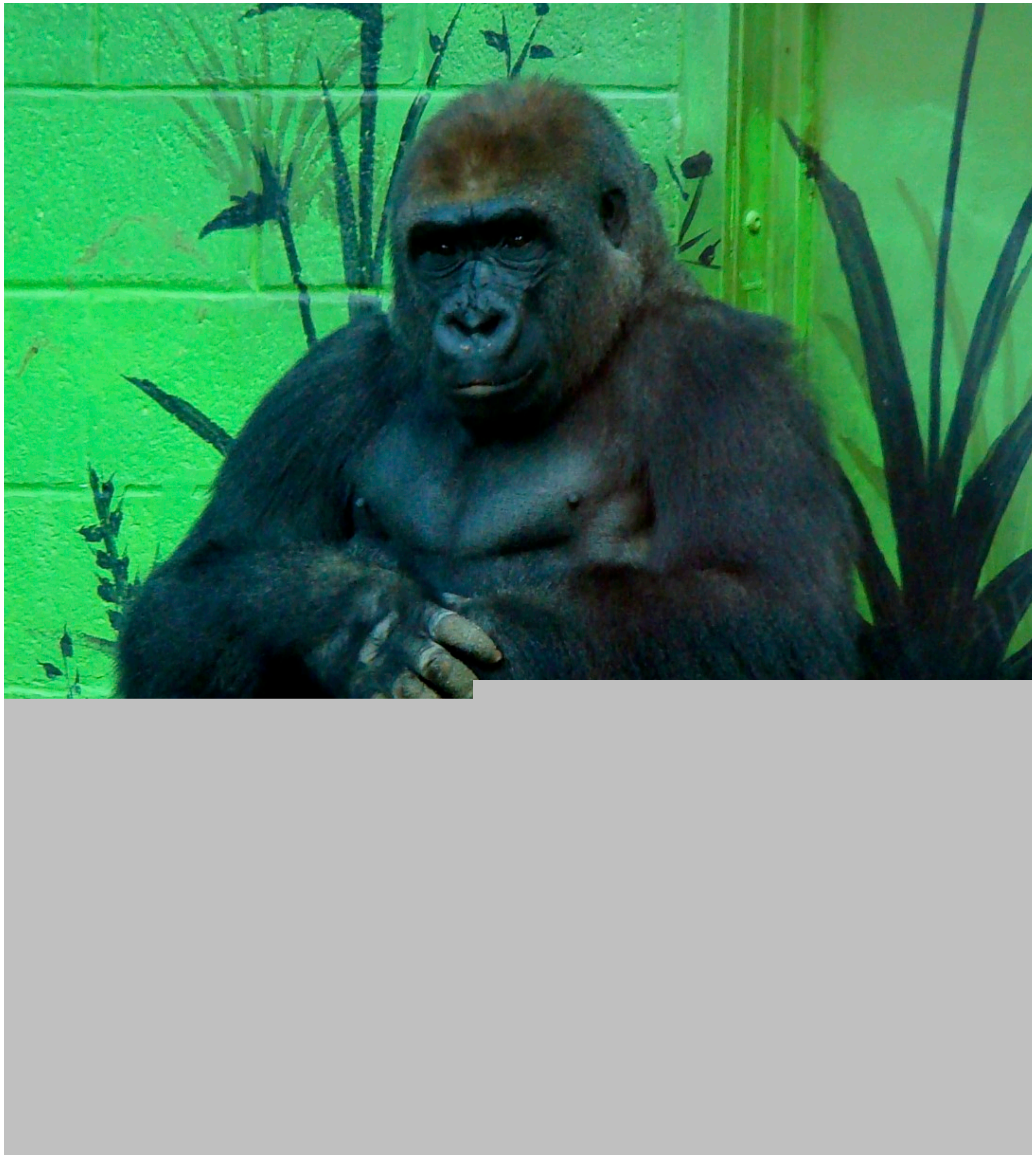

Sumit Narayan, John A. Chandy
Department of Electrical and Computer Engineering
University of Connecticut
Storrs, CT 06269

Samuel Lang, Philip Carns, Robert Ross
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439



November 15th 2009
Petascale Data Storage Workshop (PDSW) 2009
Portland, OR

# Data Corruption





Sumit Narayan
University of Connecticut, Storrs, CT

# Silent Data Corruption

- What is *silent data corruption*?
  - Fault not detected by the system
  - Fault not communicated to the user/application
- Why does it occur?
  - misdirected writes
  - torn writes
  - data path corruption
  - bit-rot

# Silent Data Corruption

- Other sources
  - Software bugs
    - File system, device drivers, software RAID
  - Firmware bugs
    - RAID controllers, disk drives
- Faults can occur during reads or writes, or both!
- Is RAID a good solution? – No!

# Data Corruption is Real

- 1.5 million disk drives in NetApp database in 32 months#
  - Data corruption detected
    - 8.5% of all nearline disk drives
      - 13% of errors went undetected
    - 1.9% of all enterprise class disk drives
      - 38% of errors went undetected
- Another study of the same database over 41 months*
  - 400,000 checksum mismatches

# Bairavasundaram et al. in ACM SIGMETRICS Performance Evaluation Review 2007
* Bairavasundaram et al. in USENIX FAST 2008

# Data Corruption is Real

- Experiments at European Organization for Nuclear Research (CERN)^
  - Wrote 2 GB on 3,000 nodes every 2 hours for 5 weeks
    - 500 errors on 100 nodes
      - 10% - sector sized or page sized
      - 80% - 64K regions
- Data corruption not limited to consumer-based drives, also present on enterprise-grade disks

*^ Bernd Panzer-Steindel, CERN/IT*


Sumit Narayan
University of Connecticut, Storrs, CT

# Data Integrity

- Common algorithms used to check data integrity
  - Cyclic redundancy checks (CRCs)
  - Adler's Algorithms
  - Fletcher's Algorithms
- ECC in hard drives
  - Not sufficient
- Other algorithms
  - Secure hash algorithms (SHA), message digest algorithms (MD5)

# Data Integrity in HPC

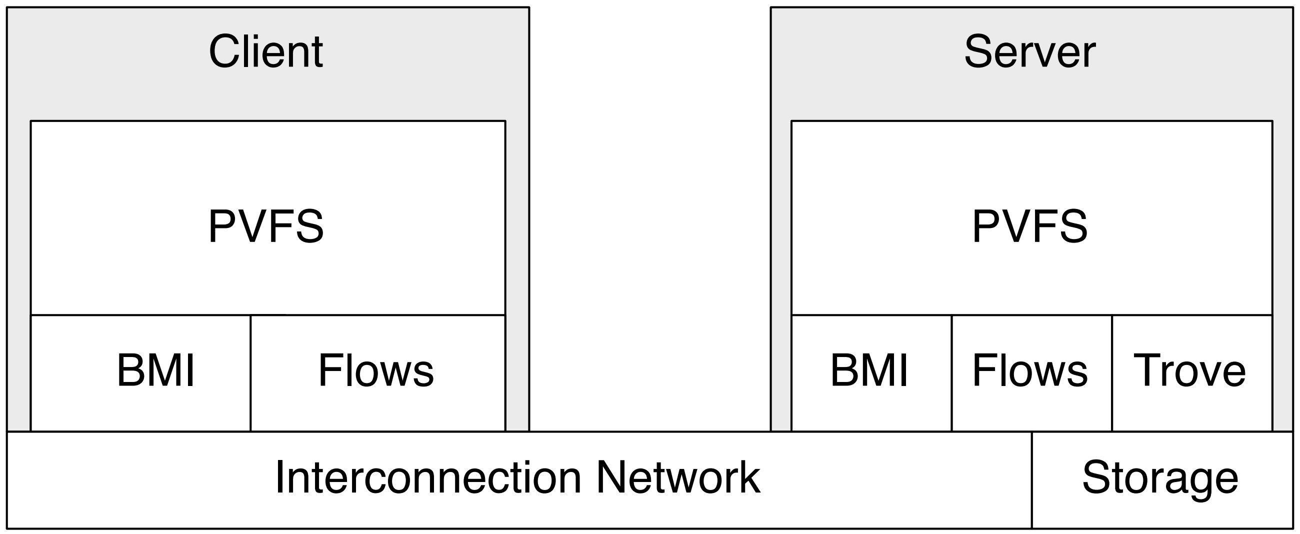
- Same analysis of same data must give same results
- 1000s of disks and billions of data blocks
- Complex workload characteristics
  - Aligned/unaligned access patterns
  - Varying sizes
  - Concurrent
  - Reads/Writes

Loss of data + Loss of time = Loss of mind!


Sumit Narayan
University of Connecticut, Storrs, CT

# Parallel Virtual File System (PVFS)

- Parallel file system
  - Separate metadata and data servers
  - Data stored on local file systems
    - bstreams



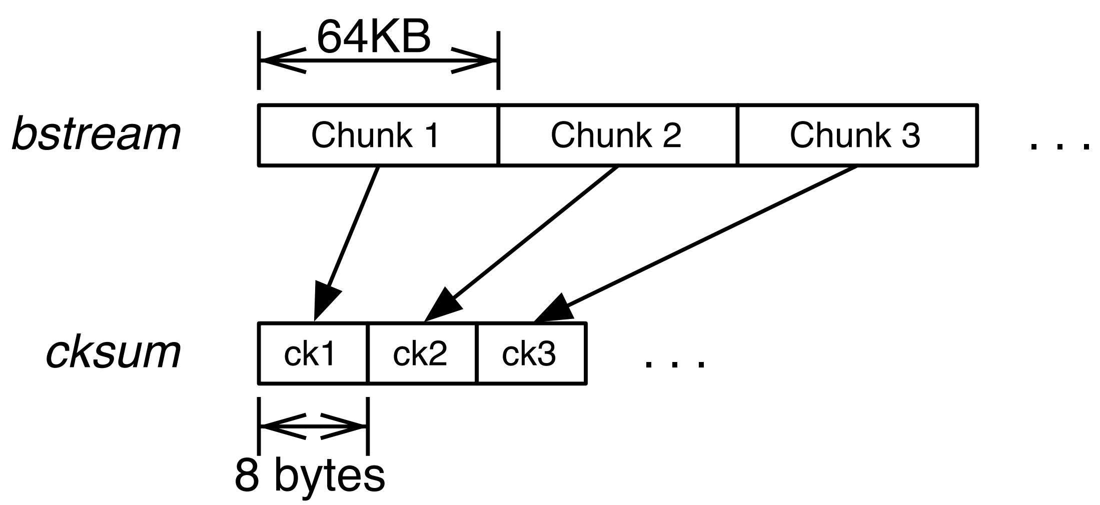
# Design

- Where do we store CRCs?
  - Metadata servers
    - single point of contact per file
    - atomic updates – major bottleneck
  - Storage servers
    - take advantage by distributing computations
    - finer granularity
    - local cache


Sumit Narayan
University of Connecticut, Storrs, CT

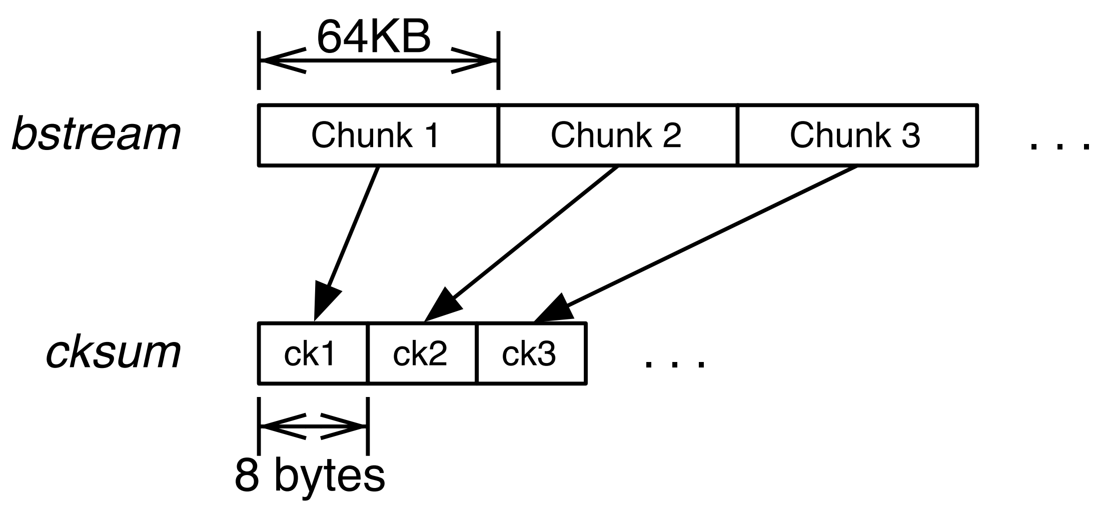
# Design

- PVFS (version 2.8.1)
- CRCs
  - separate file for checksum
- Check code of 8 bytes for every 64 KB
  - Check code for 32 MB of data in 4K block of disk

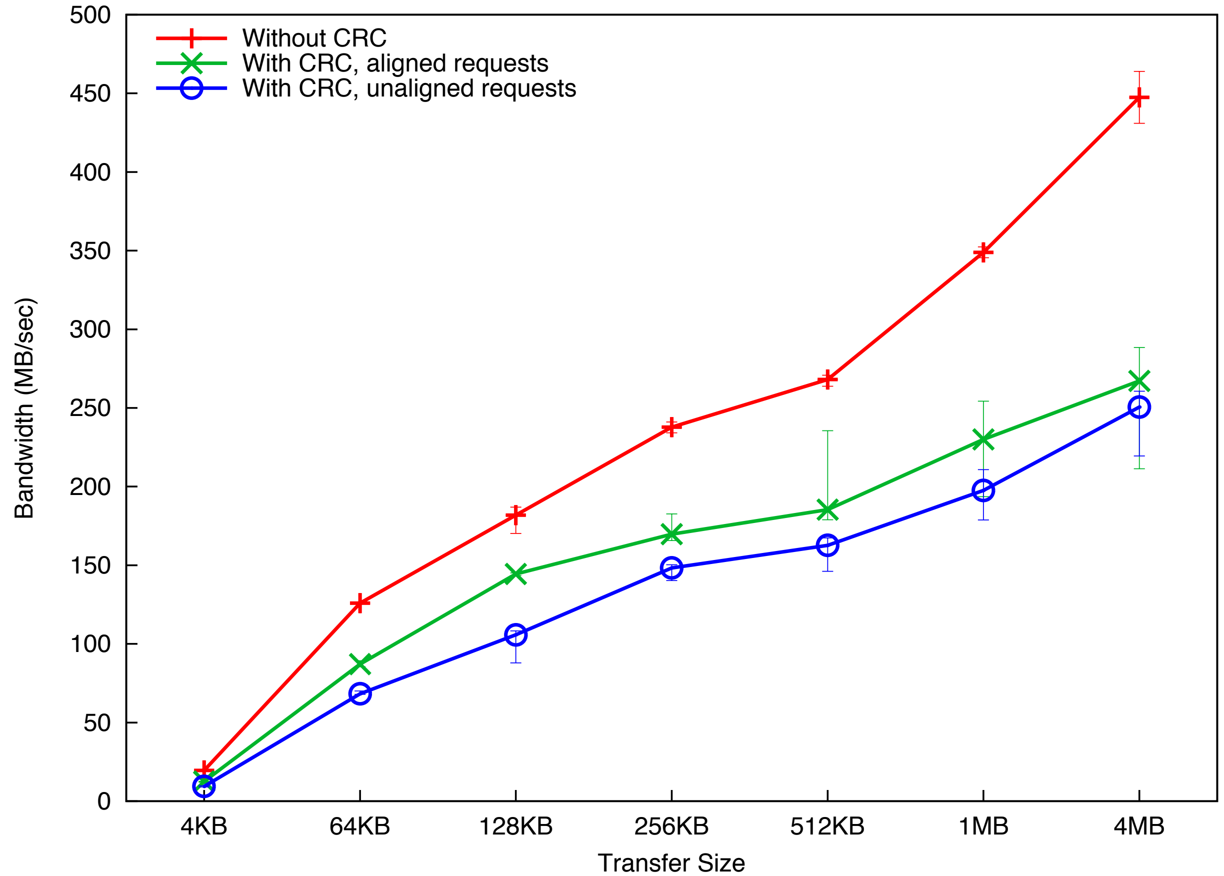


# Design

- Aligned Requests
- Unaligned Requests
  - read-verify-modify-write
- Multiple threads to operate on CRCs
  - Reads – read-ahead checksums
  - Writes – calculate checksums



# Results – Single Client



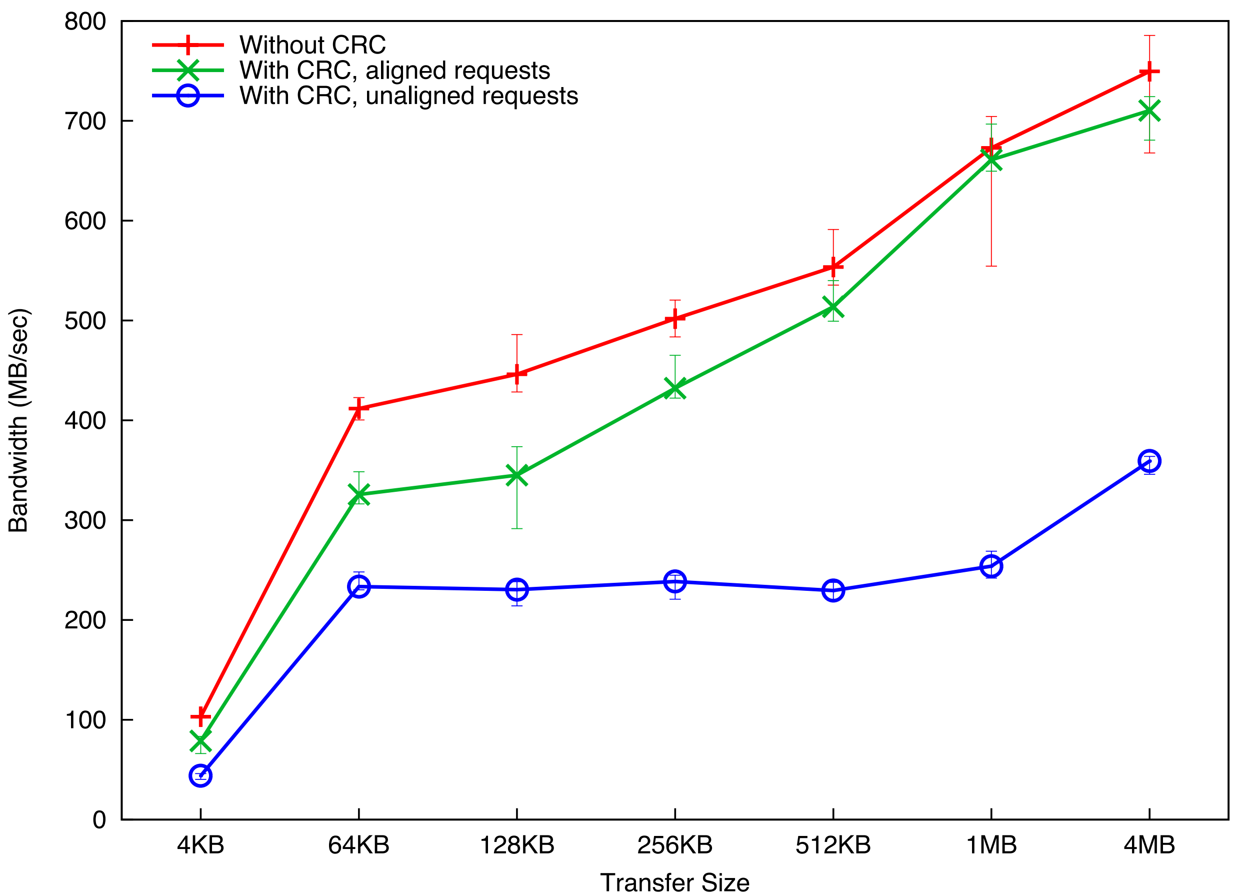

Write

- IOR benchmarking tool
- 8 GB file
- 8 storage servers

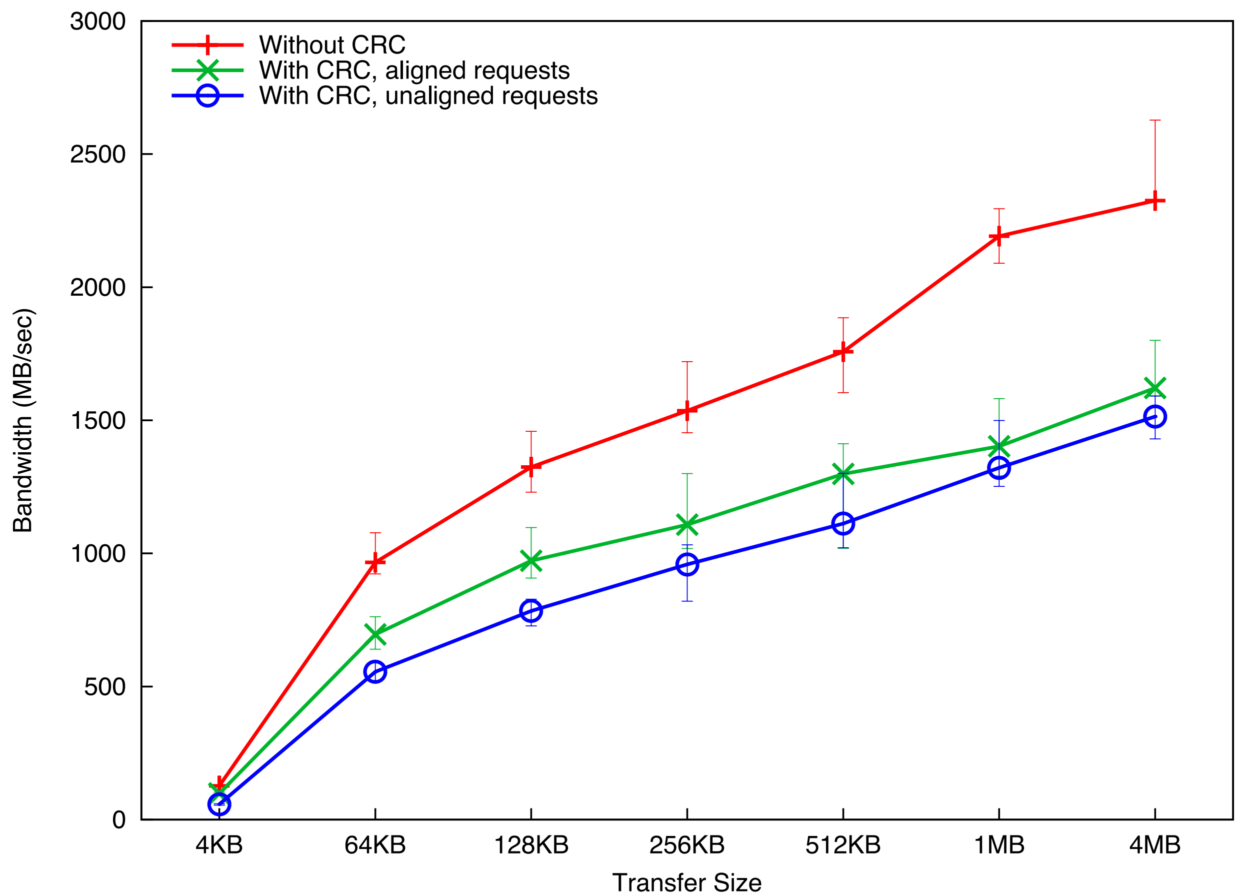


Sumit Narayan
University of Connecticut, Storrs, CT

# Results – Multiple Clients



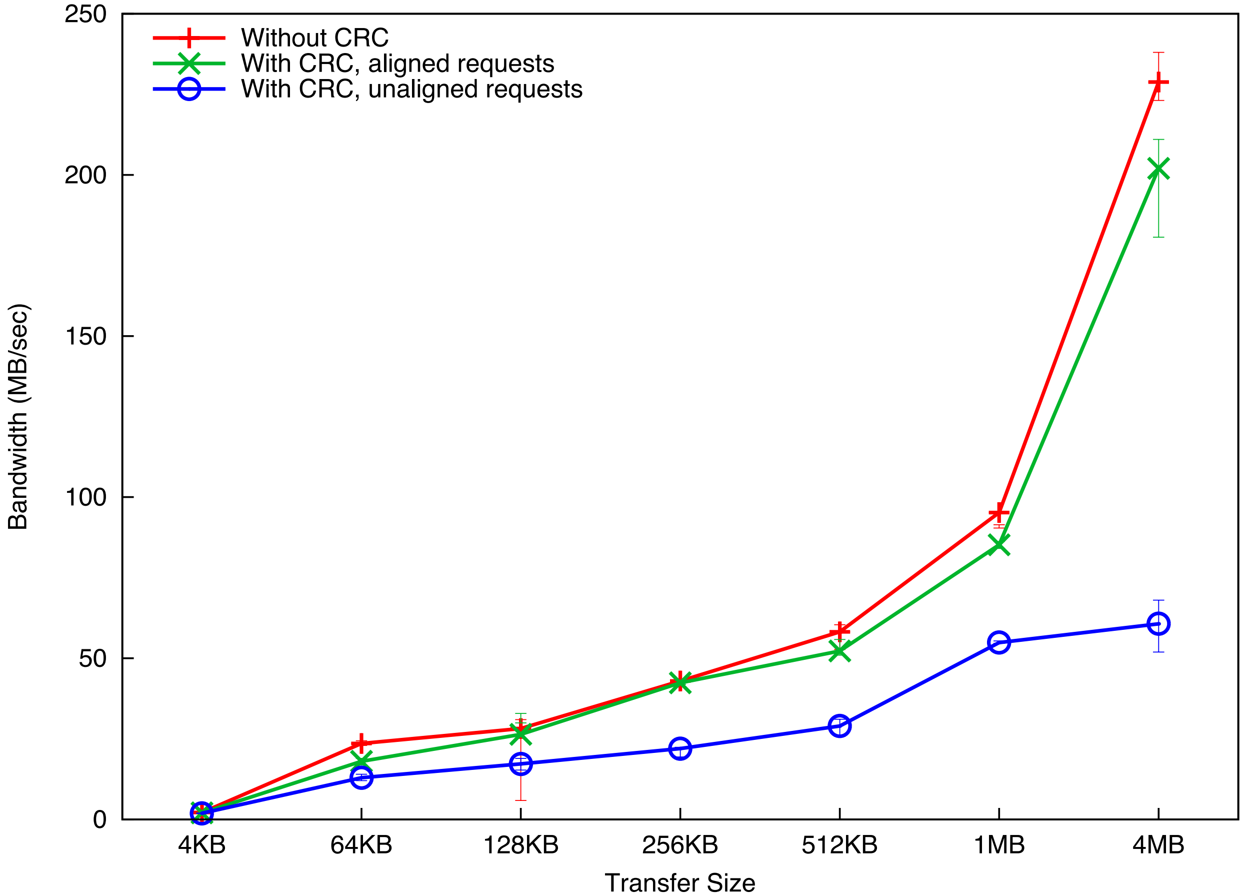

Write



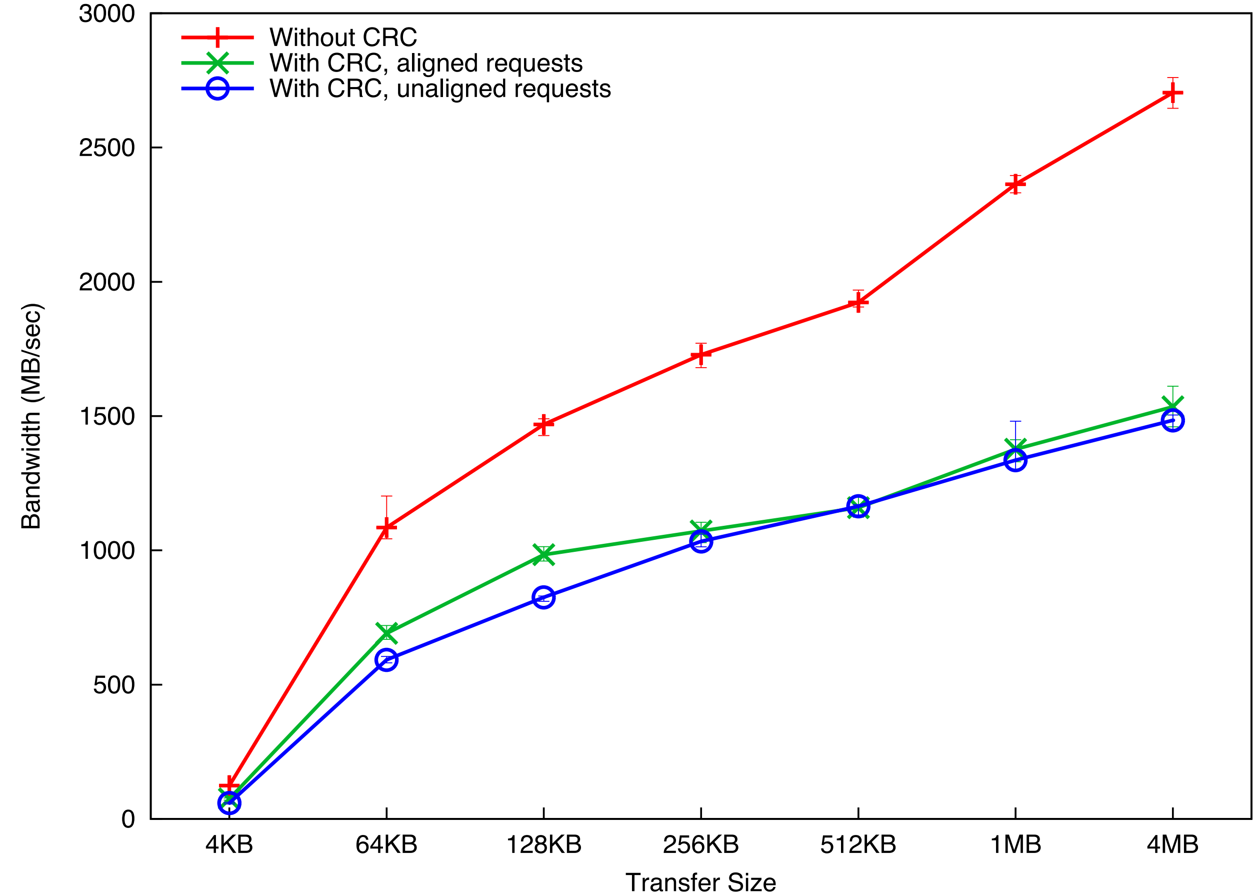

Read

- IOR benchmarking tool
- 8 GB file
- 8 clients, 8 storage servers
  - 1 GB / storage node

# Results – Multiple Clients (Sync)




Write




Read

- IOR benchmarking tool
- 8 GB file
- 8 clients
  - 1 GB / storage node
- Data sync enabled

# Related Work

- Google file system
  - multiple copies
  - in-line checksum
- Local file systems
  - ZFS, Btrfs
- Lustre, GPFS, PanFS, Ceph

# Summary

- Integrity checks necessary for all storage devices
- Provided integrity checks for parallel file system
  - Aligned request
  - Unaligned request
- Separate file for checksum
- MPI hints / POSIX attributes
- Flash-based storage
  - Limited write cycles

# Future Work

- Pipeline architecture for calculating CRCs on storage nodes
- In-line storage of CRCs in bstreams
- End-to-end checksums
- Varying parameters - checksum chunk size, algorithms etc.
- GP-GPUs
    - Reed-Solomon coding and decoding for Extended RAID on GPUs - PDSW'08

This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract DE-AC02–06CH11357.



Sumit Narayan
University of Connecticut, Storrs, CT