BLUE WATERS

SUSTAINED PETASCALE COMPUTING

Parallel Synchronization of Multi-Pebibyte File Systems

Andy Loftus (aloftus@Illinois.edu)

NCSA

GREAT LAKES CONSORTIUM
FOR PETASCALE COMPUTATION

CRAY

Outline

1. Motivation
2. Design
3. Architecture of psync
4. Preliminary performance review
5. Enhancements
6. The future of parallel sync

1. Motivation

- Upgrade filesystem firmware
- Minimize downtime

Home
120M inodes
220 TiB

Steps:

1. Evacuate Home
2. Upgrade Home
3. Repopulate Home
4. Repeat for Projects

Projects
463M inodes
1.2 PiB

1. Motivation - Goal

rsync

Custom parallel file copy tool

+ Already exists
+ Correct ★

- Serial
- Slow

Goal:
Parallel rsync

- Doesn't exist
- Ensure correctness

+ Parallel ★
+ Fast ★

2. Design

rsync

- Files only (no dirs)
- Properly handles all file types
- Already handles metadata

Parallel Management

- Parallel file copies (rsync)
- Parallel tree walk
- Directory sync
- Hardlinks (handle with care!)

3. Architecture of psync

- Distributed Task Queue
  - Python Celery
  - RabbitMQ (message broker)
  - Redis (centralized logging)
- Tasks
  - Sync a file
  - Scan a directory
  - Sync a hardlink
  - Sync a dir (metadata – mtime must be done last)

3. Architecture of psync – Directories

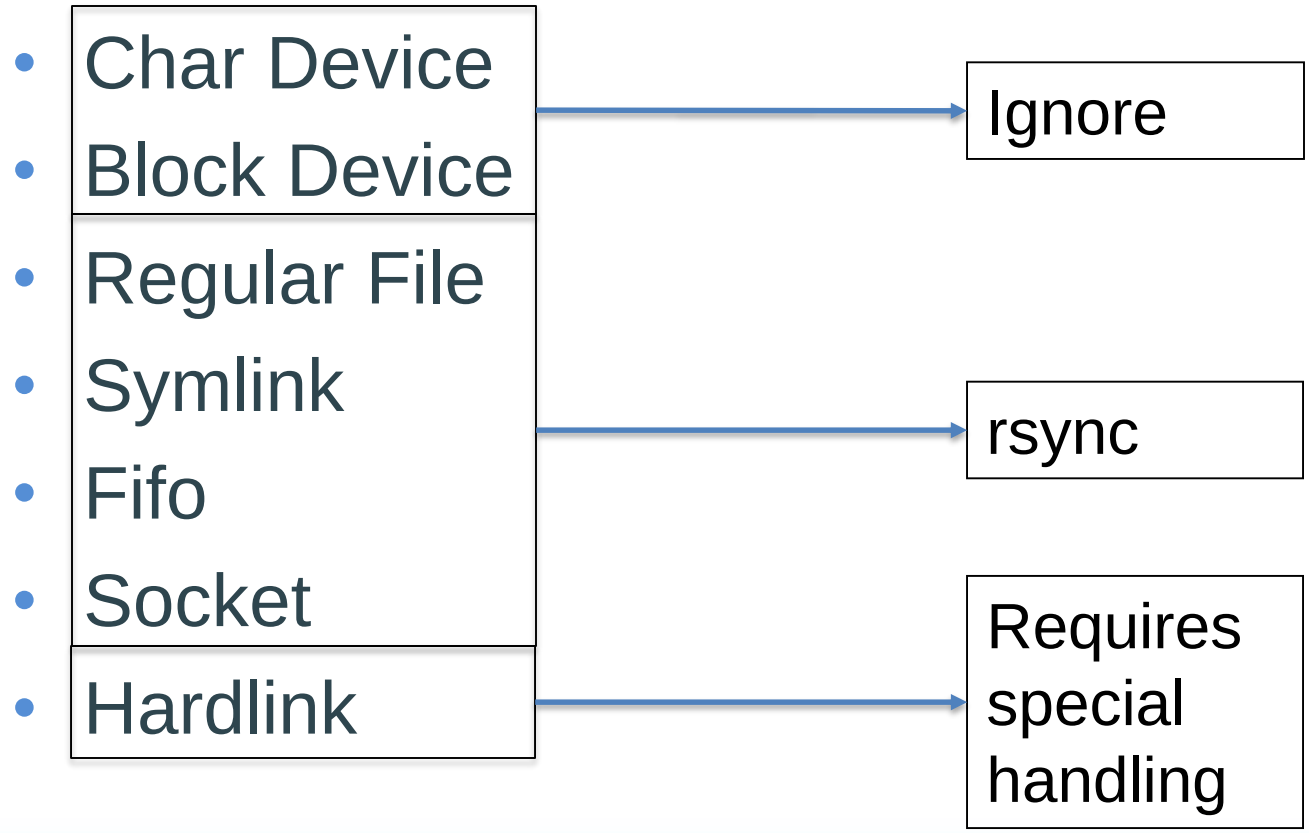
ScanDir

- Scan a single directory non-recursively
- Delete from target on the fly
- Subdirs processed in parallel

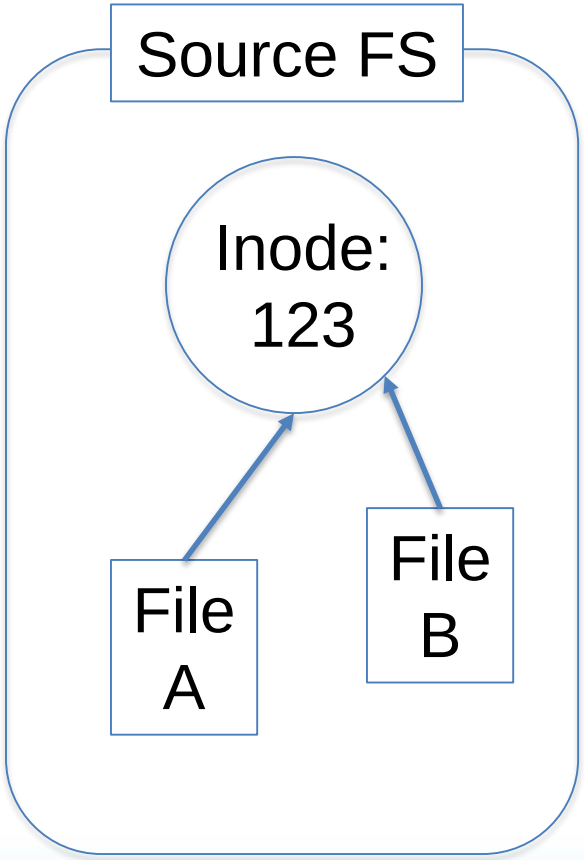
SyncDir

- Sync dir mtime AFTER all contents have been processed

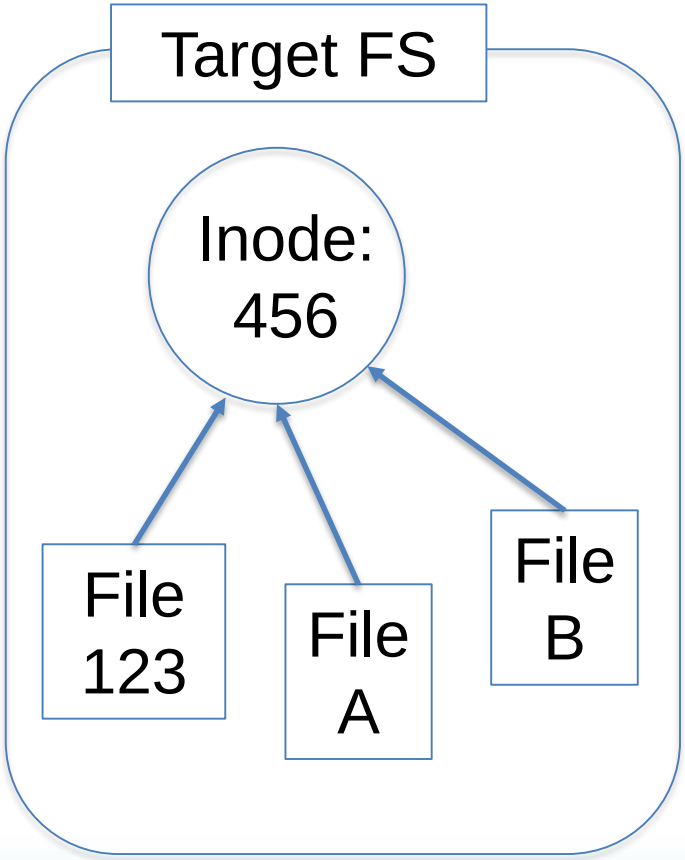
3. Architecture of psync – Special Files



3. Architecture of psync – Hardlink Resolution



1. Check for tmpfile
2. No? Copy to tmpfile
3. link target file



4. Preliminary Performance Review

- Blue Waters – Cray XE6 compute nodes
  - Dual AMD Interlagos 6276 CPUs @ 2.3GHz
  - 64 GiB RAM
- Compute to LNET routers
  - Cray Gemini high speed network (~6 GiB/s)
- LNET routers to Lustre
  - QDR Infiniband
- Cray Sonexion 1600 filesystem appliance

4. Preliminary Performance Review

Initial Sync

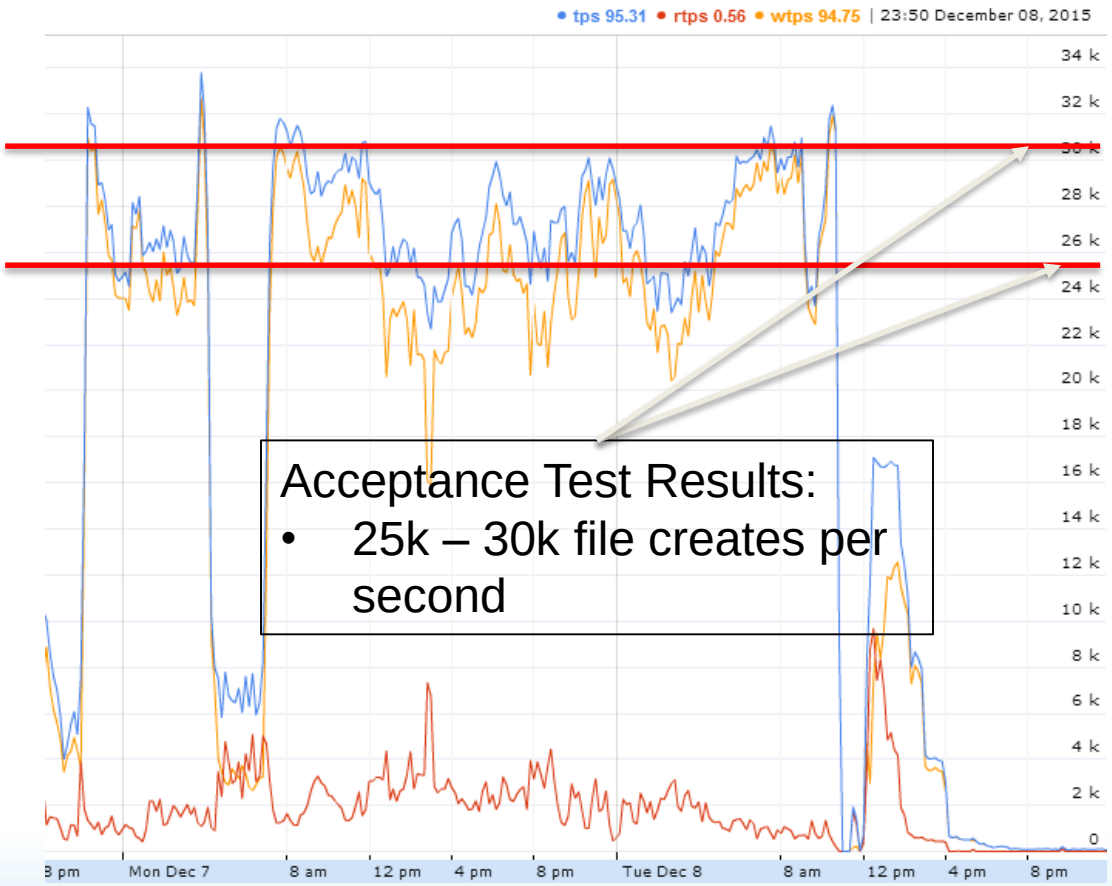
- 2384 procs
- 398 nodes
- Idle file system
- 2.25% per hour
  - Based on inode count
- ~44 hours

Re-sync

- 800 procs
- 100 nodes
- Live filesystem
- ~16 hours
  - Affected primarily by total number of inodes

Home File System – Mostly Small Files

4. Preliminary Performance Review



Graph shows MDS IOPS during psync run on isolated file system

Psync scaled to 2384 processes pushed MDS into peak IOPS range

5. Enhancements

- File restriper
  - Re-stripe files based on policy (like mrsync, retools[1])
- Other filesystems
- Support for non-root use

6. The future of parallel sync

Psync

- NCSA
- 2015
- Python
- RabbitMQ
- Redis
- rsync

PCP

- SISG
- 2015
- Python
- MPI
- file open

Lustre Data Mover

- SDSC
- 2016
- Python
- RabbitMQ
- Memcached
- file open

FCP

- ORNL
- 2015
- Python
- MPI
- Work Stealing

6. T

		PCP	Lustre Data Mover	Psync	FCP
Technologies		mpi	msg queue	msg queue	mpi
Correctness	Special Files	?	?	Y	
	Hardlinks	N	N	Y	
	rsync verified	?	?	Y	
	checksum	Y	?	Y	Y
	re-sync	partial	Y	Y	N
Single pass		N	N	Y	N
Delete		?	Y	Y	
Scalable		Y	Y	Y	Y
Dynamically scalable		partial	Y	Y	N
Filesystem agnostic		Y	N	modular	Y
Auto-detect filesystem		Y	N	N	
Run as root		limited ?	Y	Y	
Run as non-root		Y	?	possible	
Monitoring	stats	N	Y	partial	Y
	progress	N	Y	Y	Y
Limit by	Filename globbing	Y	N	N	
	file age	N	Y	Y	
Lustre specific	Stripe aware	Y	Y	Y	Y
	Restripe capable	Y	Y	N	
	liblustre	Y	Y	N	
	"safestat"	Y	Y	N	
Features	Pause / resume	Y	Y	Y	Y
	parallel single file copy	Y	N	N	Y
	incremental backup	Y	N	N	
	multiple MDS support	N	Y	N	
	dynamic mountpoint discovery	Y	N	Y	
	parallel filesystem walk	Y	N	Y	Y
	fadvise NoCache	N	Y	N	
	dataset checksum	N	N	N	Y

References

1. Retools - http://people.nas.nasa.gov/~kolano/projects/retools.html
2. PCP - https://github.com/wtsi-ssg/pcp
3. Lustre Data Mover - https://github.com/sdsc/lustre-data-mover
4. FCP - https://github.com/olcf/pcircle
5. Psync - https://github.com/ncsa/psync