

Parallelism in PSI

The Parallel Storage Interface
at Los Alamos

Unix-Like Interface

- Unix syntax and semantics
 - » ls, cd, chmod, etc
- Easy learning curve
- Non-Unix features
 - » Trash can
 - » Command test facility

Why Increase Parallelism?

- Parallel generation => parallel archiving
- Without it, we fall further behind
- Cray @ 90 MB/s vs. archive @ 10 MB/s
10-to-1
- HP @ 30,000 MB/s vs. archive @ 200 MB/s
150-to-1
- File counts are increasing.

Biggest Challenge: Whole Tree Transfers

- File counts > 100,000
- 100's of GB
- Any combination of file sizes
- Users have little time to learn about HPSS
- The big button:
psi store -R *dir* (unconditional)
psi store -R -cond *dir* (conditional).

LANL Transfer Modes

- Disk COS's
 - » Decrease time to store first byte
- Direct tape COS's
 - » Reduce disk cost
 - » Increase sustained throughput
 - » Fetch performance same as store perf.

PSI Goal - Everything in Parallel

- File transfers
- Attribute querying ('stat' on client & server)
- Attribute modification (chmod, etc)
- Small file aggregation (1 million/month).

PSI File Transfer Parallelism

- 1. HPSS parallel file transfers
- 2. Multi-host parallel transfers
- 3. Simultaneous file transfers.

File Transfer Parallelism (cont)

- Efficient scheduling of resources
 - » Speed of host disks, CPUs, network
 - » Tape drives required
- High performance client movers.

PSI Client Mover

- Multi-threaded (pthreads)
- Multi-host transfers
- One mover per client host per file transfer
- No per-stripe synchronization (no shmem)
- Independent input and output (incl. piping)
- N-ahead disk I/O
- Multiple network interfaces supported.

PSI Client Mover Performance

- Disk I/O
 - 90-95% of disk bandwidth
- Network interface
 - 95-99% of network bandwidth.

PSI Scheduler

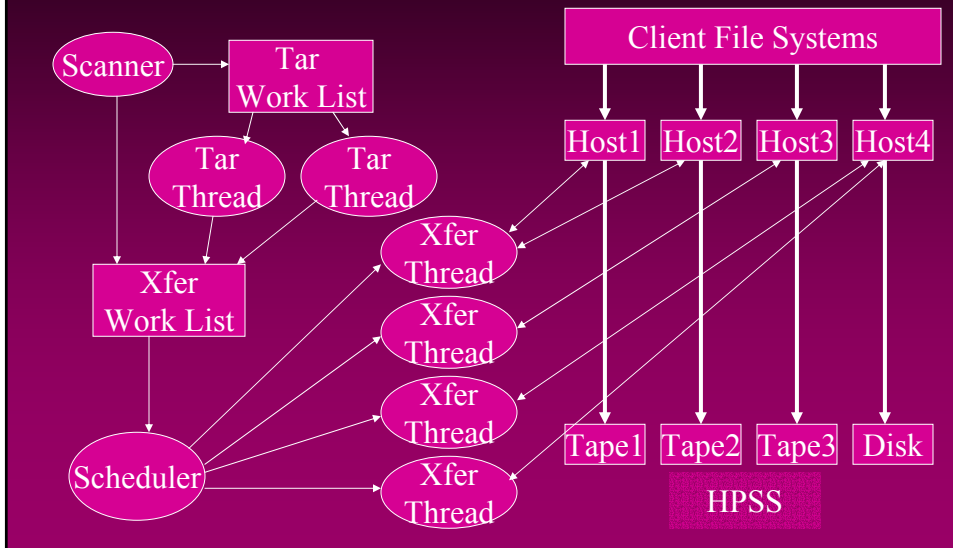
- Estimates performance of each transfer component (disk, network, CPU)
- Each transfer assigned to most lightly used host(s)
- Optimization - store by COS, fetch by cartridge and segment.

Scheduler (cont)

- Cognizant of COS and tape drive limits
- Redirection from disk COS to tape COS for disk response, tape speed
- Future - Cooperative Scheduling.

PSI File Transfer Control

Store With tar



Single-Command Performance

Command: `psi store filelist`

- 1 file x 4-way 9940B => 184 MB/s
- 2 files x 4-way 9940B => 361 MB/s
- 3 files x 4-way 9940B => 538 MB/s
- 20 files x 1-way 9940B => 850 MB/s
- 1 x 8-way 9940B, 5 x 4-way 9940B,
6 x 2-way 9940B => 1041 MB/s

PSI Future Work

- Parallel tar
- Cooperative Scheduling

Parallel Tar

- Multiple tars - one per directory
- Selectively tar up SMALL files
 - » Reduces amount of metadata
 - » Reduces cond. metadata queries
 - » Allows large files to be fetched directly
- Only for whole directory transfers
- Large files can be transferred while tar is processing small files.

Performance vs. Resource Consumption

- Historically - limited user/job to small fraction of resources
- Result - less conflict, less performance
- Q: How to improve user/job performance and still avoid conflict?
- Or, how to give one job most of available HPSS resources without penalizing others?

Cooperative Scheduling

- Answer
 - » 1. Monitor available tape drives
 - » 2. Give up resources readily when other jobs need them
 - » 3. Avoid unstable algorithms
- Applies to user jobs, MPS, repack, etc.

Wish List

- /dev/null mode for HPSS movers
 - » Facilitate network performance tests

Summary

- More to parallel archiving than running single-file transfers