

BLUE WATERS

SUSTAINED PETASCALE COMPUTING

Help! My filesystem is too big and I can't
back it up!


Automating backups on a 4PiB filesystem.

Andy Loftus (aloftus@Illinois.edu)



GREAT LAKES CONSORTIUM
FOR PETASCALE COMPUTATION

CRAY



BLUE WATERS
SUSTAINED PETASCALE COMPUTING

I NSF NCSA GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION CRAY

What is a backup?

Revision History – Point in time restore


vs

Single copy – HSM

CUG 2014 2

HSM provides only the capability of Single Copy.
This talk will focus on Backups as Revision History.

This is a custom backup solution.
Why reinvent the wheel?



BLUE WATERS
SUSTAINED PETASCALE COMPUTING

Motivation

- Daily
- Automated
- Generic

CUG 2014 3

1. Daily: Want a solution that runs, and completes, every day. Key: It's got to be fast.
2. Automated – daily execution and dynamically adjust to changing environment
 1. Ie: new and deleted users and groups
 2. Counter example - Amanda pdf on Lustre wiki requires a manual configuration of Disk List Entries.
3. Generic - Not tied to any specific filesystem or implementation - portable
 1. Example: Cray Sonexion is a black box. Direct access to low level lustre storage is not allowed.
 2. Don't want to focus on the filesystem structure, just the data.

BLUE WATERS
SUSTAINED PETASCALE COMPUTING

INSF
NCSA
GREAT LAKES CONSORTIUM
FOR PETASCALE COMPUTATION
CRAY

Problem

Lots of data to scan
Lots of data to copy
Lots of data to transfer

↓

Lots of data ...too much to deal with at once

CUG 2014 4

What makes this a hard problem?

- /home – 2 PiB capacity
 - 537 users
 - 16 misc (VM's, boot image servers, etc)
- /projects – 2 PiB capacity
 - 130 projects

Lots of data

- SCAN to find changes
- making a COPY of the files is the actual backup
- TRANSFER the files to a safe, preferably remote, location

BLUE WATERS
SUSTAINED PETASCALE COMPUTING

I NSF NCSA GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION CRAY

Solution

PARALELLISM

CUG 2014 5

Any solution must take advantage of the parallelism that already exists in the system.

Challenge: Implementing a fully parallel backup solution.

BLUE WATERS
SUSTAINED PETASCALE COMPUTING

1 NSF
2 NSERC
3 GREAT LAKES CONSORTIUM
FOR PETASCALE COMPUTATION
4 CRAY

Challenges

1. Partition the filesystem
2. Parallel execution
3. Parallel transfer
4. Handling full backups

CUG 2014 6

1. Must define non-overlapping targets that can be backed up in parallel.
2. Distribute the backup tasks to run in parallel.
3. Efficiently transfer the backup archives to long term storage.
4. Full backups essentially comprise a copy of the entire filesystem. Using a policy of full backup every 30 days, this becomes a recurring problem.

Solution must be:

- Automated (performed by software)
- Dynamic (resumable/restartable)

BLUE WATERS
SUSTAINED PETASCALE COMPUTING

INSF
NCSA
GREAT LAKES CONSORTIUM
FOR PETASCALE COMPUTATION
CRAY

Challenges – 1. Partition the filesystem

```

graph TD
    home((home)) --> staff((staff))
    home --> vendor((vendor))
    home --> sciteam((sciteam))
    staff --> user1((user))
    staff --> user2((user))
    staff --> user3((user))
    vendor --> cray((cray))
    vendor --> nvidia((nvidia))
    cray --> user4((user))
    cray --> user5((user))
    nvidia --> user6((user))
    nvidia --> user7((user))
    sciteam --> user8((user))
    sciteam --> user9((user))
    sciteam --> user10((user))
  
```

CUG 2014 7

Task: Define non-overlapping chunks that can be backed up in parallel.

The filesystem structure naturally provides good splitting points at user home directories and project specific directories.

Manually define topdirs. Then the software scans exactly 1 layer below for basepaths. The basepaths are the small, non-overlapping chunks that get backed up in parallel.

As new users or projects are added, or deleted, the changes are discovered automatically.

All the goals have been met:

1. DAILY - Small chunks backup faster. Small can mean either few inodes or low capacity use, or both.
2. AUTOMATED - Automated discovery of changes (ie: new users, new projects, etc...)
3. GENERAL – Haven't relied on any filesystem-specific or implementation specific details.

BLUE WATERS
SUSTAINED PETASCALE COMPUTING

INSF
NCSA
GREAT LAKES CONSORTIUM
FOR PETASCALE COMPUTATION
CRAY

Challenges – 2. Parallel execution

The diagram illustrates a parallel execution challenge. On the right, a box labeled 'Manager Node' contains an orange box labeled 'Backup tasks'. On the left, a vertical stack of four white boxes labeled 'Worker Nodes' is shown. Arrows point from the 'Backup tasks' box to each of the four 'Worker Nodes' boxes, indicating the distribution of tasks. A vertical ellipsis between the second and third worker nodes indicates that there can be more than four worker nodes.

- Backup is a sequence of independent tasks
- Distribute tasks to worker nodes

CUG 2014 8

Running backups in parallel requires accounting for two key points:

1. Ensure each backup can run independently (for resiliency, pick up where last part left off)
 1. Works by saving state in a file
2. Running backup tasks in parallel (backups require root)

Cannot schedule jobs using Moab because policy prevents root jobs. Use RPyC to build in a custom solution.

BLUE WATERS
SUSTAINED PETASCALE COMPUTING

INSF
NECA
GREAT LAKES CONSORTIUM
FOR PETASCALE COMPUTATION
CRAY

Challenges – 3. Parallel transfers

The diagram shows a cloud labeled 'Globus Online' on the left. Two arrows originate from it: one labeled 'Source' pointing to a blue cylinder labeled 'Lustre', and another labeled 'Destination' pointing to a blue rectangle labeled 'Long Term Storage'. Between 'Lustre' and 'Long Term Storage' is a large blue double-headed arrow labeled 'DATA'. Below 'Long Term Storage' are three icons representing different storage media: a tape, a disk, and another tape.

CUG 2014 9

- Some efficient transfer solution usually already configured on a cluster
- 3rd party means the backup software is not tied to a specific implementation
- Backup software only needs to interact with the third party transfer mechanism

BLUE WATERS
SUSTAINED PETASCALE COMPUTING

INSF
NECA
GREAT LAKES CONSORTIUM
FOR PETASCALE COMPUTATION
CRAY

Challenges – 4. Full backups

Rolling fulls – spread out full backups over entire cycle.

Full backup if:
 $\text{Basepath_ID} \% 30 == \text{DOY} \% 30$

CUG 2014 10

Two approaches:

1. One full and incrementals forever.
 1. GOOD: minimizes data to be transferred
 2. BAD: complex mgmt, saves all data forever
2. Periodic fulls (in one form or another).
 1. GOOD: Simple
 2. BAD: Recurring fulls

Idea: Only a few full backups run each day.


Cycle = how many days between full backups. 30 day retention = 30 day cycle

683 basepaths / 30 day-cycle = 22.8 full backups per day

Algorithm: Choose which basepaths get a full backup on a given day.

Assign each basepath a positive integer id.

Basepath % 30 = initial DOY



BLUE WATERS
SUSTAINED PETASCALE COMPUTING

I NSF NCSA GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION CRAY

**I need a supercomputer
to backup my supercomputer!**

CUG 2014 11

Backups run on 28 IE nodes, outside of the cluster but mount the filesystem.

BLUE WATERS
SUSTAINED PETASCALE COMPUTING

I NSF NCSA GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION CRAY

Software Design

Goals	Principles
<ul style="list-style-type: none">• Resilient• Scalable	<ul style="list-style-type: none">• Modular• Event driven

CUG 2014 12

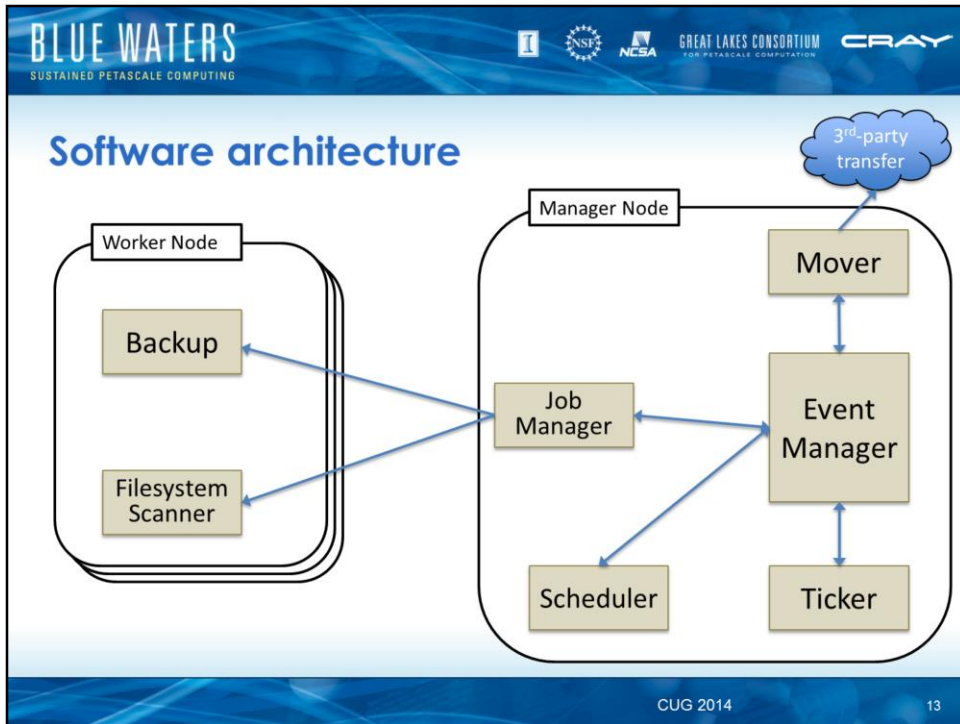
Recall motivations: daily , automated, generic

Design the software to support these driving factors.


- Resiliency
 - Ability to tolerate failures (node crashes, network failures, and filesystem failures)
- Scalable
 - Scale for increased performance and/or larger filesystems by simply adding new nodes.

Goals promote modular, event driven architecture.

- Modular
 - Simplicity in both design and ease of maintenance.
- Event driven
 - Events (not time) drive all actions. Activity proceeds based on messages passed between modules.



1. Modules
2. Events (messages)
3. Event Manager as post office
4. Modules register for notification of event types
5. Steps of a backup, move forward based on events



BLUE WATERS
SUSTAINED PETASCALE COMPUTING

I NSF NCSA GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION CRAY

Advantages of dar

- Separate catalog file
- Recognition of deleted files when using file lists
- Random access archive

CUG 2014 14

Dar used to do the actual backup

CATALOG


- archive metadata in a separate file
- Reference for incremental backups
- Small, stays on disk (full archive file sent to remote, low cost, long term storage)

FILELIST

- During Incremental, properly handles deleted files

RANDOM ACCESS – doesn't have to scan archive file in order

- Recall – backup as sequence of small steps & software design goal of resiliency (ability to pick up where it left off)
 - Example: create archive & extract catalog are independent steps



BLUE WATERS
SUSTAINED PETASCALE COMPUTING


I NSF NCSA GREAT LAKES CONSORTIUM FOR PETASCALE COMPUTATION CRAY

Results

9.6 hours	Average daily backup runtime
2.4 hours	Average full backup time
9 minutes	Average incremental backup time
404.5 million	Average number of inodes scanned daily
51.3 million	Average number of inodes backed up daily
1.61 PiB	Average bytes scanned daily
16.8 TiB	Average bytes backed up daily
10	Number of topdirs
683	Number of basepaths

CUG 2014 15

Averages from Feb 2014 over a 5 day period.



Future enhancements

- Performance related
 - Multiple files per Globus transfer task
 - Multiple tasks per worker node
- Capacity – better filesystem partitioning
 - Auto-split *large* basepaths
 - Generate list of files for backup
 - Changelog?
 - Robinhood: needs revision history for file deletes
 - Generic scan?
 - Lustre_rsync – not parallel, filesystem specific
 - find

CUG 2014 16

PERFORMANCE

- Bottleneck – inefficient use of Globus
- Currently only a single task per remote host
 - Different nodes do not have to have the same capabilities
 - Some may be able to run more tasks in parallel

CAPACITY

- Initial attempt – automated scan to breakup a project directory
 - Complicated, not scalable
- Robinhood – No revision history, deleted files not detected
 - Thought - data-mining operation beside robinhood, effectively creating snapshots of the filesystem
- Find – Could be pheasable, stat is fast and light, scalability is a concern

Thank you!

- Questions?

Andy Loftus (aloftus@Illinois.edu)