Integrating JASMine and Auger

Sandy Philpott
Thomas Jefferson National Accelerator Facility
12000 Jefferson Ave.
Newport News, Virginia USA 23606

Sandy.Philpott@jlab.org
757-269-7152
http://cc.jlab.org

Spring06 HEPiX – CASPUR

JASMine

Jefferson Lab Mass Storage System - Tape and Disk

2 STK Powderhorn Silos

1.5PB used out of 2PB capacity offline storage

Components

- Data Movers
  - 20 9940B tape drives and local staging disks
  - Legacy 9940A drives – read only, tape copies
- Disk Cache - 60TB online storage
- Library Manager

Auger

Jefferson Lab Batch Computing System

LSF6, fair share scheduling

175 dual Intels, PIII and Xeons

- PIIIs run 3 jobs
- Xeons use hyperthreading, run 7 jobs

1100 job slots

Increase total job throughput – keep the CPUs busy!

Integration Overview

Auger batch farm is JASMine's largest client (other clients are the physics experiments, interactive users, HPC, SRM grid users…)

Goal: Keep farm CPUs busy!

- don't start a job until its data is online in the cache
- don't delete a job's data from the cache until the job is done with it (until the data has been copied locally)

Integration

In the early implementations, the mass storage system and batch farm were independent entities, with little coordination between them.

A farm job would run, specify its data, then wait for it to arrive online.

Auger now prestages a job's data, so the job doesn't use a job slot and tie up a CPU sitting idle until it can really run.

Integration (cont)

But make sure prestaging doesn't preload too much data…

- In early implementation, a user who submitted many jobs would cause too much data to be prestaged
- FIFO, so data would overrun the cache, and be gone before it was used!

Auger now provides simple, smart staging…have just enough data for each user prestaged to stay ahead of the LSF fair share scheduler (at least 1 file)

  - Algorithm based on ratio of free space to how much a user has already used (configurable)

Adaptive Cache

Simple round robin of multiple distributed cache disk servers wasn't enough

Use *adaptive* cache, considering not only

- total disk usage of the cache server

But also

- current load on the disk

Combine with

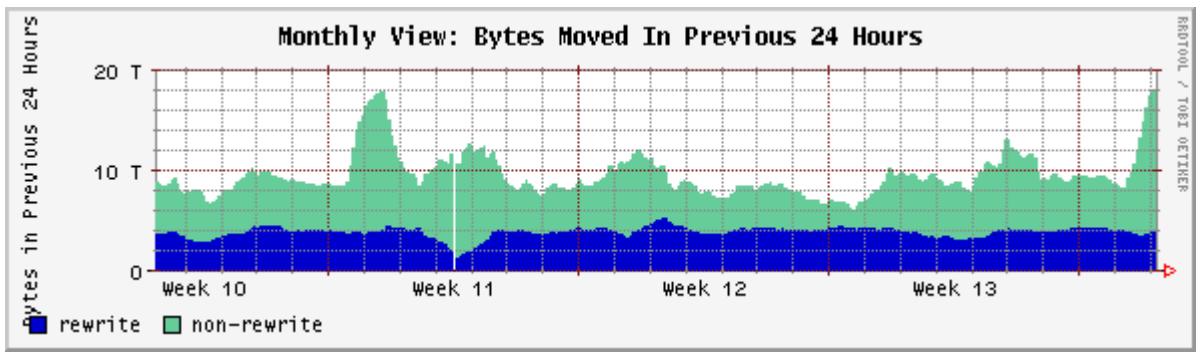
- Strict or relaxed allocation
  - Strict – files may not exceed allocation limit
  - Relaxed – files may exceed limit if additional space available
- Strict or automatic deletion
  - Automatic - FIFO deletion policy
  - Strict – delete only if user requests deletion

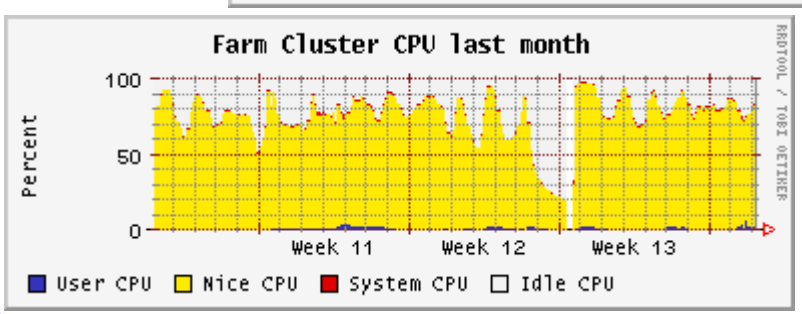
Throughput

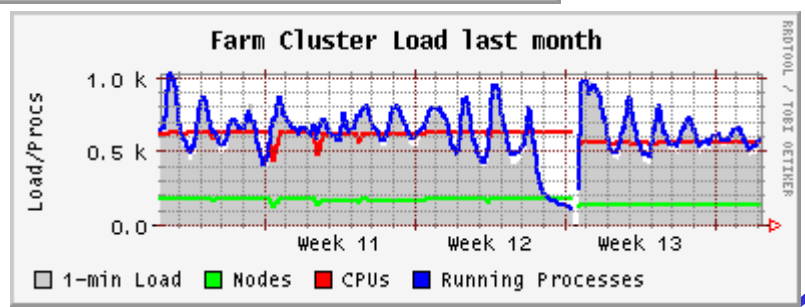
JASMine routinely moves 8 -10TB/day.

It has seen maximum of 20TB/day.

These numbers are to/from tape; they are higher if the online disk files are included…







Conclusions

JASMine is

- distributed
- scaleable
- modular
- efficient

and allows for further growth with minimal changes.

Adding simple Auger features for smart prestaging and adaptive cache policies take advantage of JASMine's capabilities.