Selective Backup and Cloud Computing

Joulien Tatar
UC Irvine

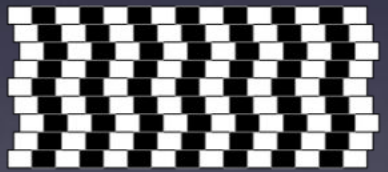
Selective Backup: Motivation & Technical Overview

- UCI HPC cluster consists of private (mostly) and public data storage servers.
  - ~2PB cumulative (...and growing quickly)
  - Can’t backup everything (yet) so decided to allow users to selectively backup what is most important.
- Hardware: 480TB on two Hitachi JBODs (~$50K)
  - 60 8TB disks
  - Intel 48-core server,
  - SSD disks for metadata
  - FS: BeeGFS (speed) with ZFS (data protection)
- Software:
  - Custom written scripts
    - Utilize rsync and GNU parallel

Selective Backup: Design*

- Users interaction with system through two configuration files:
  1. Backup list
     - Paths of files/directories specified in order of priority
     - Backup user config options
  2. Exclude list
     - rsync exclude patterns apply in addition to explicit files/directories
- User configuration files parsed to create executable rsync arguments on per user basis
- Parallel rsync streams initiated using GNU Parallel (each user is a stream)
  - Number of streams based on CPU load

https://rsync.samba.org



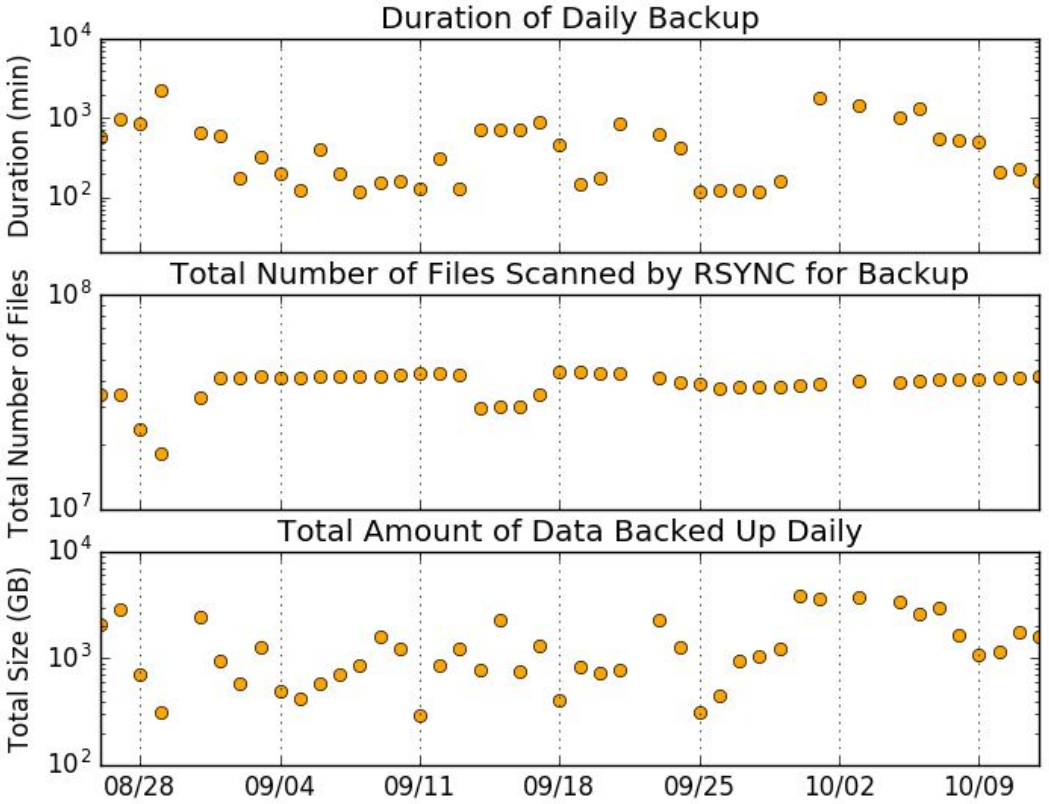
https://www.gnu.org/software/parallel

* Designed and written by Joseph Farran, lead UCI HPC Architect and Sys Admin

Selective Backup: Capabilities (First pass)

Typical Operation:

- ~1690 user accounts
- 1TB/day
- ~30 million files considered
- Run for 4 to 5 hours/day



Singularity Containers

singularity.lbl.gov

Goal:

- Allow HPC users to bring and share their own Linux work environment to any other Linux host
  - Example: If custom simulation software requires Ubuntu running python 2.7, gcc 4.8 and ROOT 5.34, but HPC cluster runs CentOS 7, python 3.0, gcc 6.2, and ROOT 6.06, user can build an singularity container (a single image file) with Ubuntu and the appropriate software stack needed.
  - Once singularity container exists, it can be used on any Linux host that has singularity installed.

Singularity for Cloud Computing (Workflow Verified)

- Build singularity container on local host with all data analysis software dependencies
  - *singularity create container.img; singularity bootstrap container.img centos.def* (centos.def contains list of dependencies)
- Start any Linux compute instance on HPC/Amazon/Google etc and install singularity
- Transfer container and analysis data to cloud instance
- Run jobs
  - *singularity exec container.img python simulate.py*
- Transfer analysis output.
- Shut VPC down.