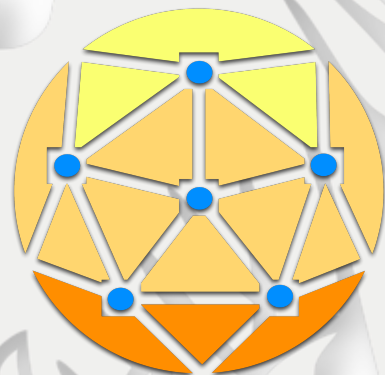# StashCache

Data delivery network for OSG Connect users

Anna Olson • University of Chicago

XRootD Workshop
29 January 2015

Open Science Grid

Outline

- Why?
- How?
- Testing
- Results
- Next Steps

Target community & current setup

- Individual researchers and small project groups in OSG have few options for moving data to grid sites
- Current methods:
  - HT Condor file transfer; http (curl, wget)
    - Very good for small input data sets < 5 GB / job
    - **Inefficient** if same input data is used repeatedly
    - **Scaling issues** as input data set size grows
- Can XRootD provide a better delivery system?

OSG Stash service: origin of job data

- Distributed network filesystem using Ceph
  - x2 replication, 90 TB usable storage for OSG users
- OSG Connect Users:
  - Transfer data into Stash using Globus Online
    - or other tools: scp, rsync, xrdcp
  - POSIX access from OSG Connect login host
  - User and project namespace exported to XRootD and http.   c.f. http://stash.osgconnect.net/

XRootD proxy caching

Similar to http proxy caching
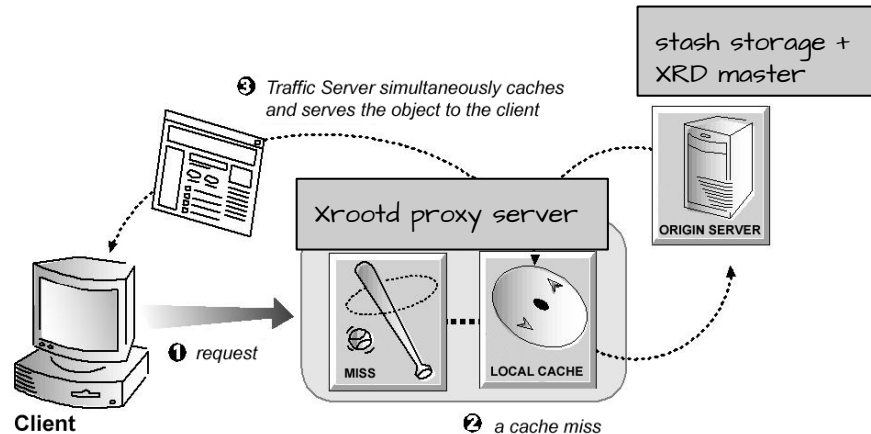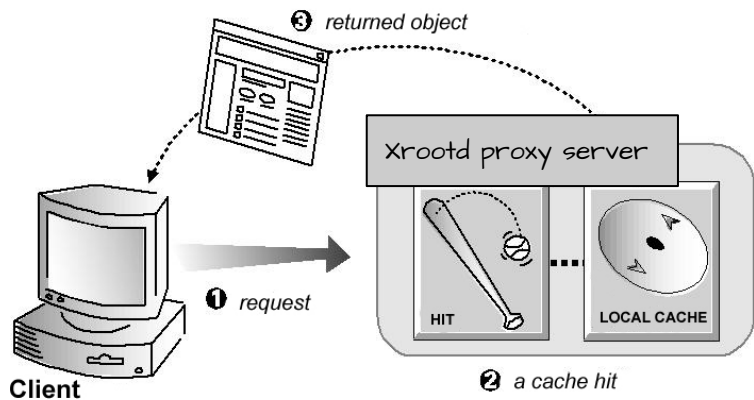
Uses Stash origin namespace: `user/userid/filename.dat`

figure credit: Apache traffic server proxy caching: http://goo.gl/Yz9RSJ

StashCache locations & compute sites

Stash origin: ⭐

OSG Caches: 🟠

Deployment issues

- Simple in principle, but...
  - ulimit needs to be high (4096)
  - Port 1094 needs to be open to all traffic
  - XRootD cache cleaning sometimes leads to crashes
    - Cache remains full
    - This bug will be fixed in the next release!

Hide details from user: `stashcp`

- Simple user client to transfer data from network to job sandbox
- Available on all OSG sites with OASIS mounted
- Examples:

```
$ stashcp -s user/olsona/samples/iowa/iowa_01.fna -l
samples/

$ stashcp -s user/olsona/samples/iowa/ -l samples/

$ stashcp -r -s user/olsona/samples -l .
```

Questions going in

Does Stash Cache outperform STASH?

- What kind of performance do we see from Stash Cache?
- What are its limitations?
- What is the best way to get the optimal location?

Testing & measurement methods

- Send jobs out to OSG
- Each job pulls a number of files using either stashcp (for Stash Cache) or wget from STASH
  - Locations: BU, UChicago, STASH, UCSD
  - Files downloaded to either job sandbox or /dev/null
  - Single or multiple jobs sent out at a time
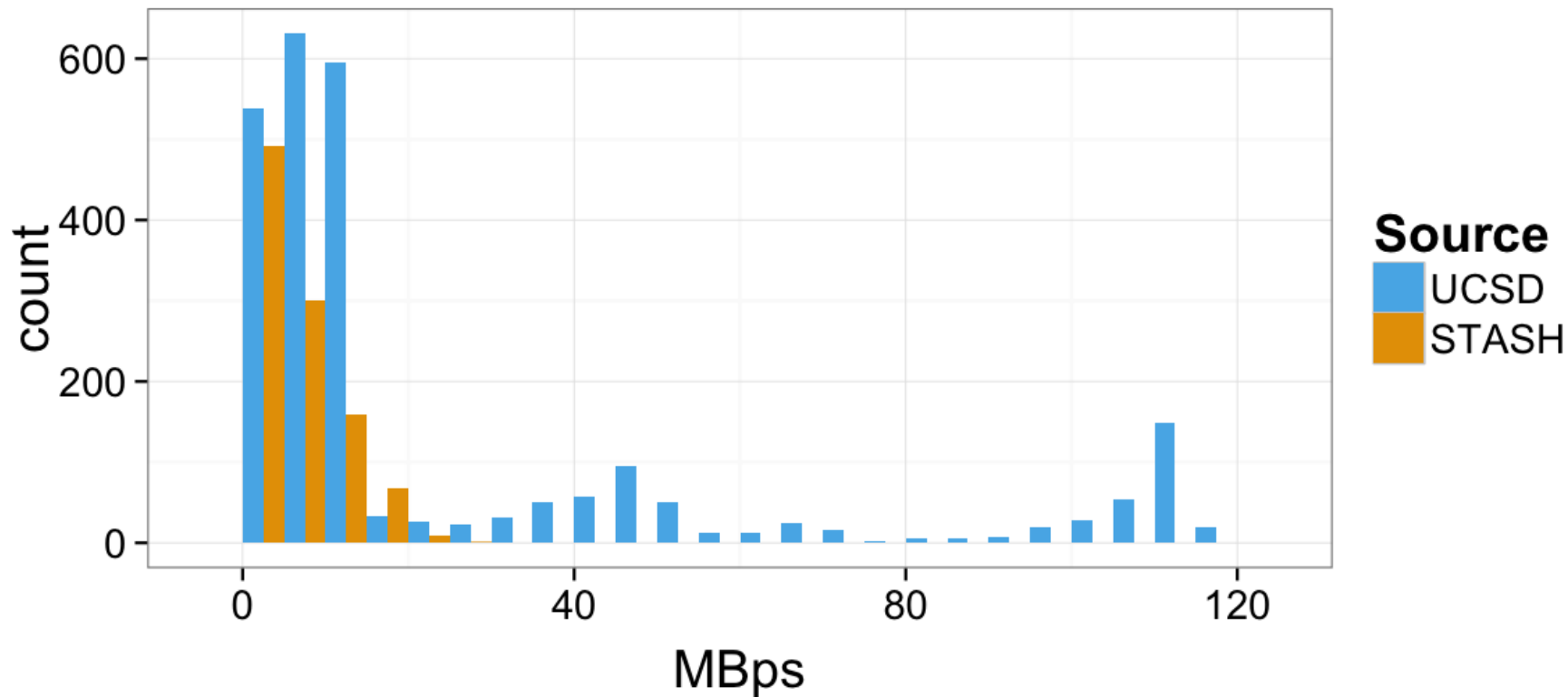- Source, destination, file size, and download time are recorded

Caching tests

- Pull same files multiple times
- 100 jobs sent out, each pulling 10 files in series
  - Multiple jobs could be pulling from the same source!
- Available sources: BU, UChicago, UCSD and STASH
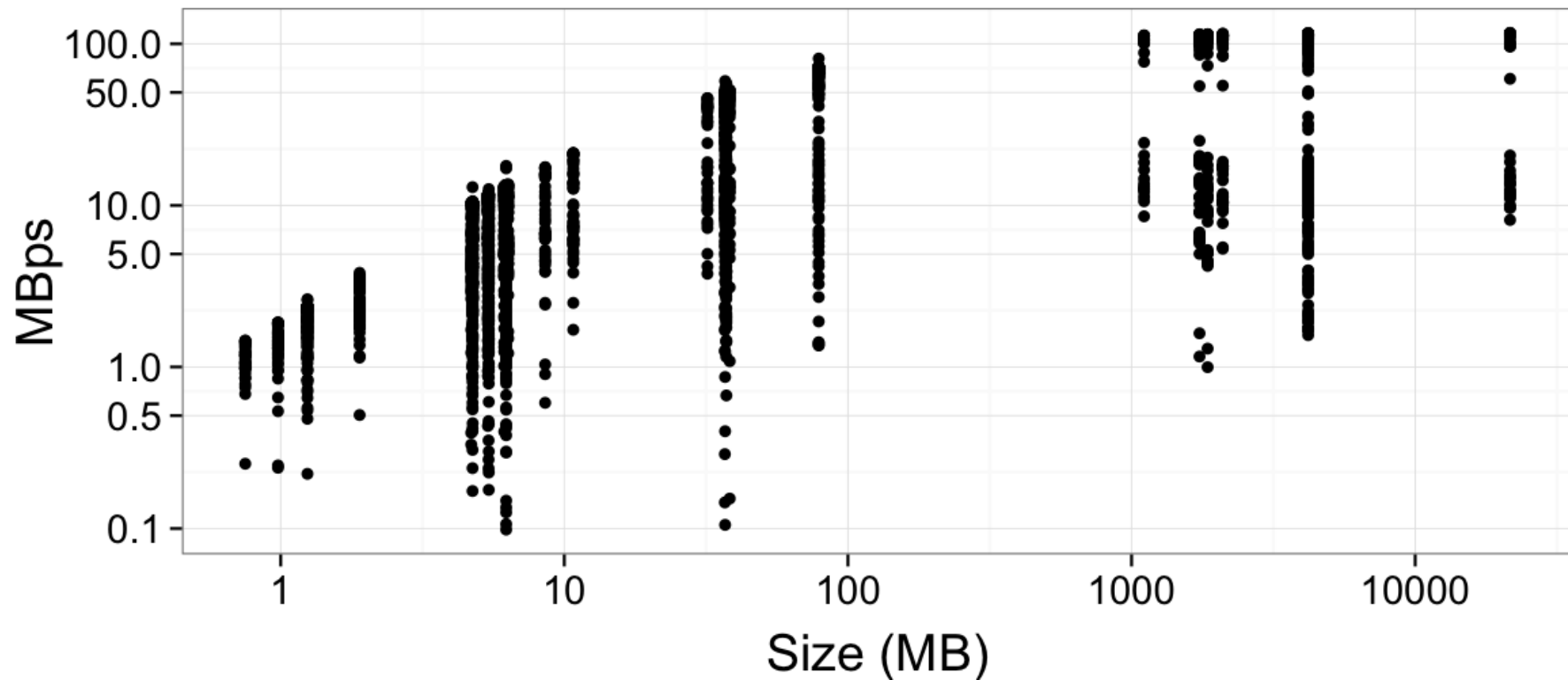- File size ranged from 750KB to 21GB
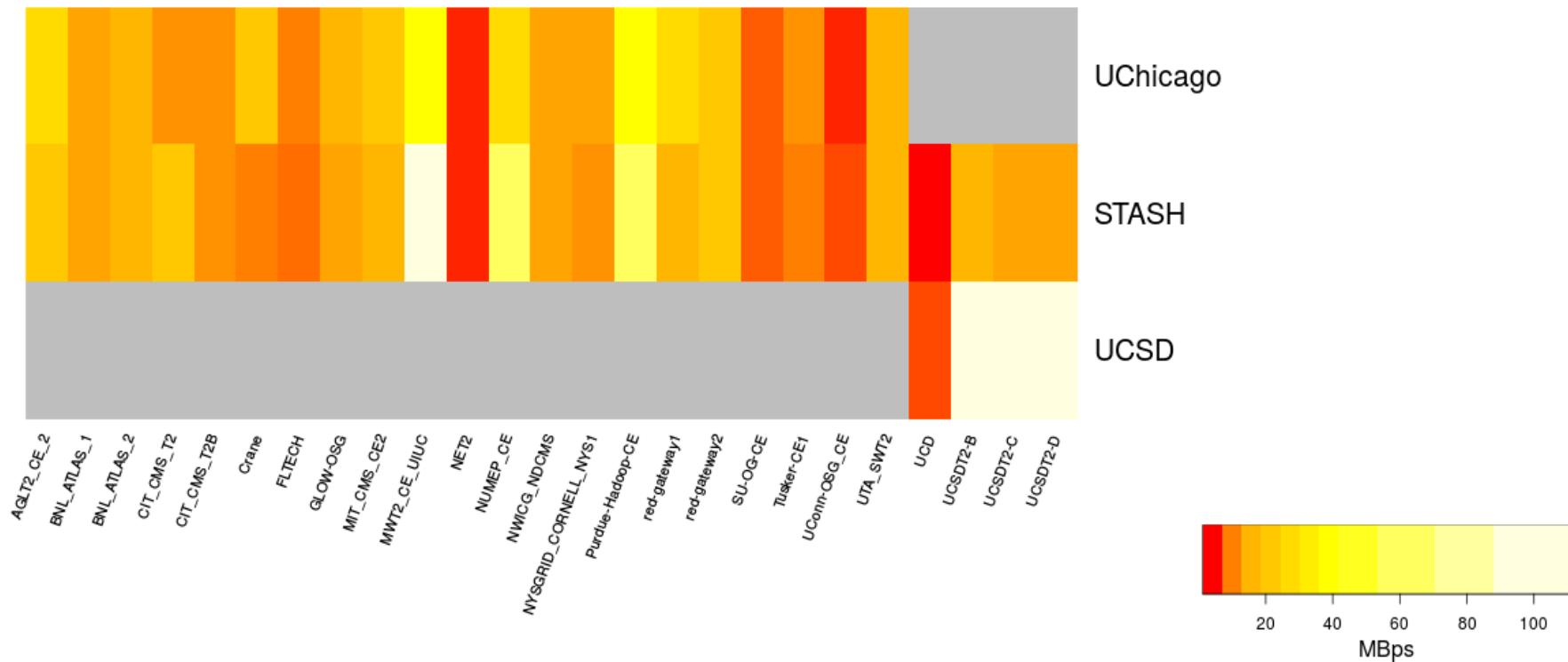
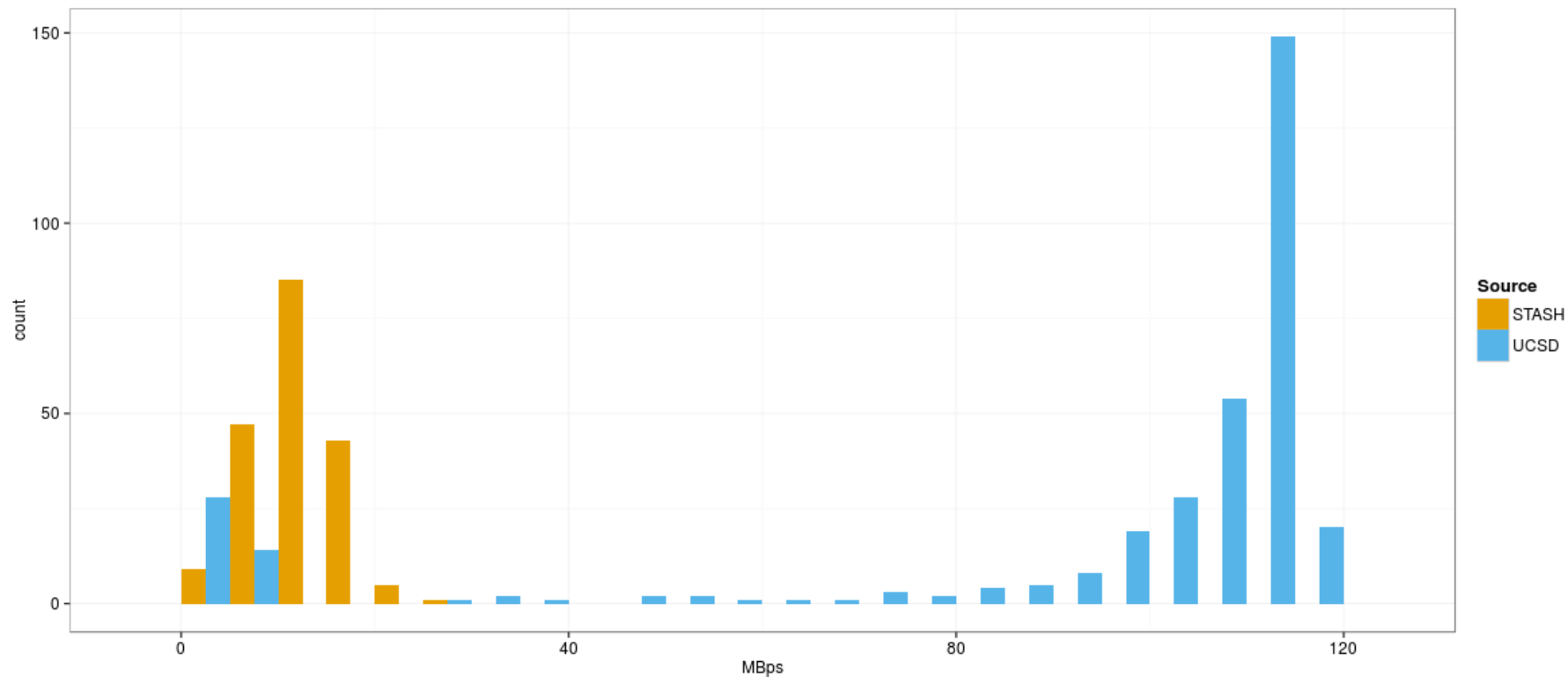# Median download speed: all files
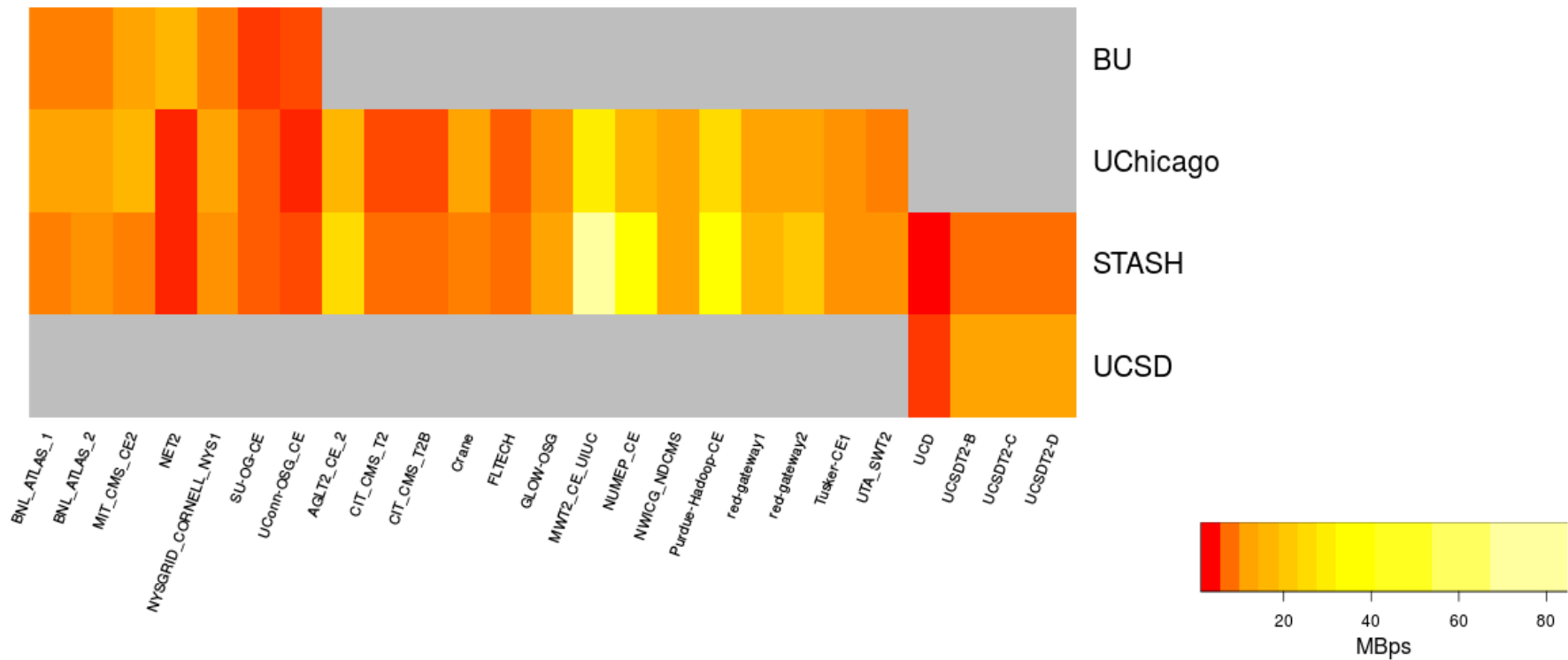
Distribution of speeds: UCSD

Speeds by file size: UCSD
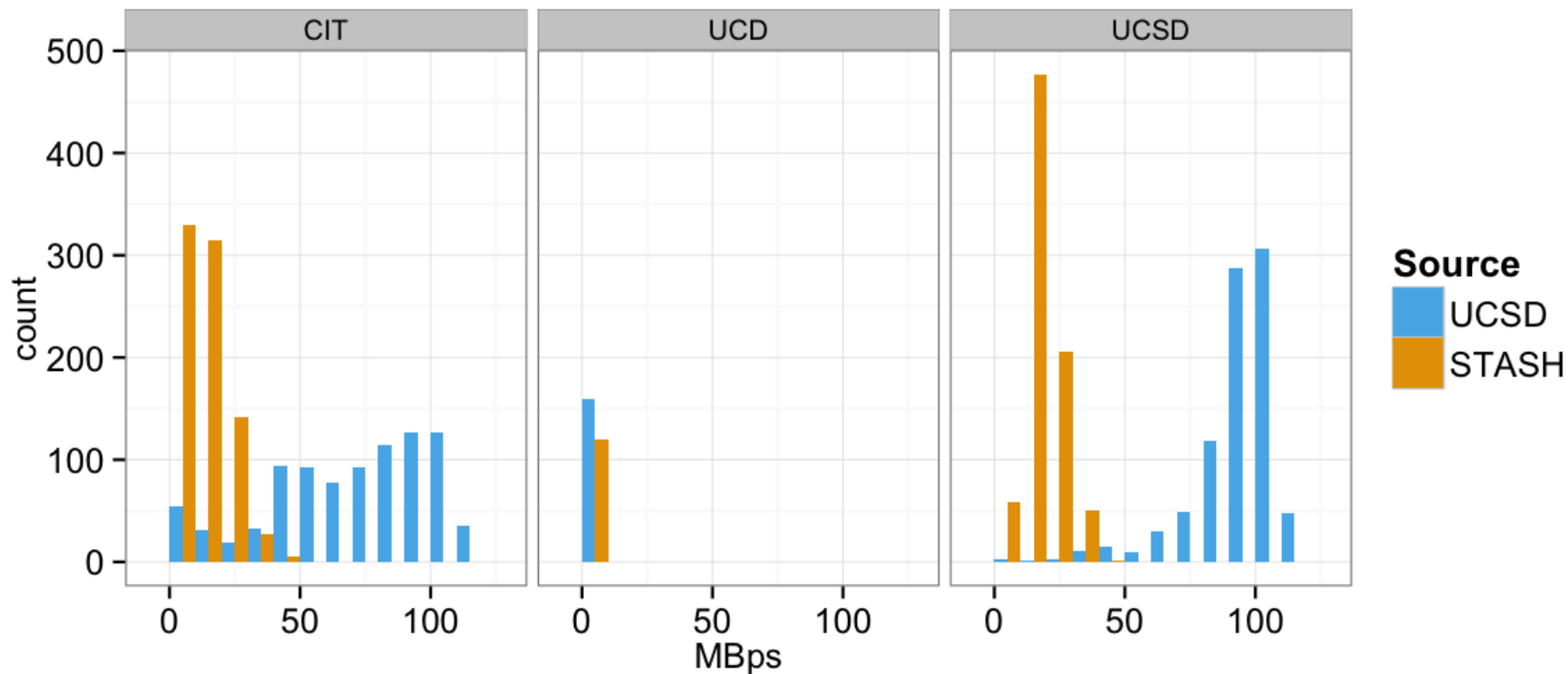
Median Download Speed: big files
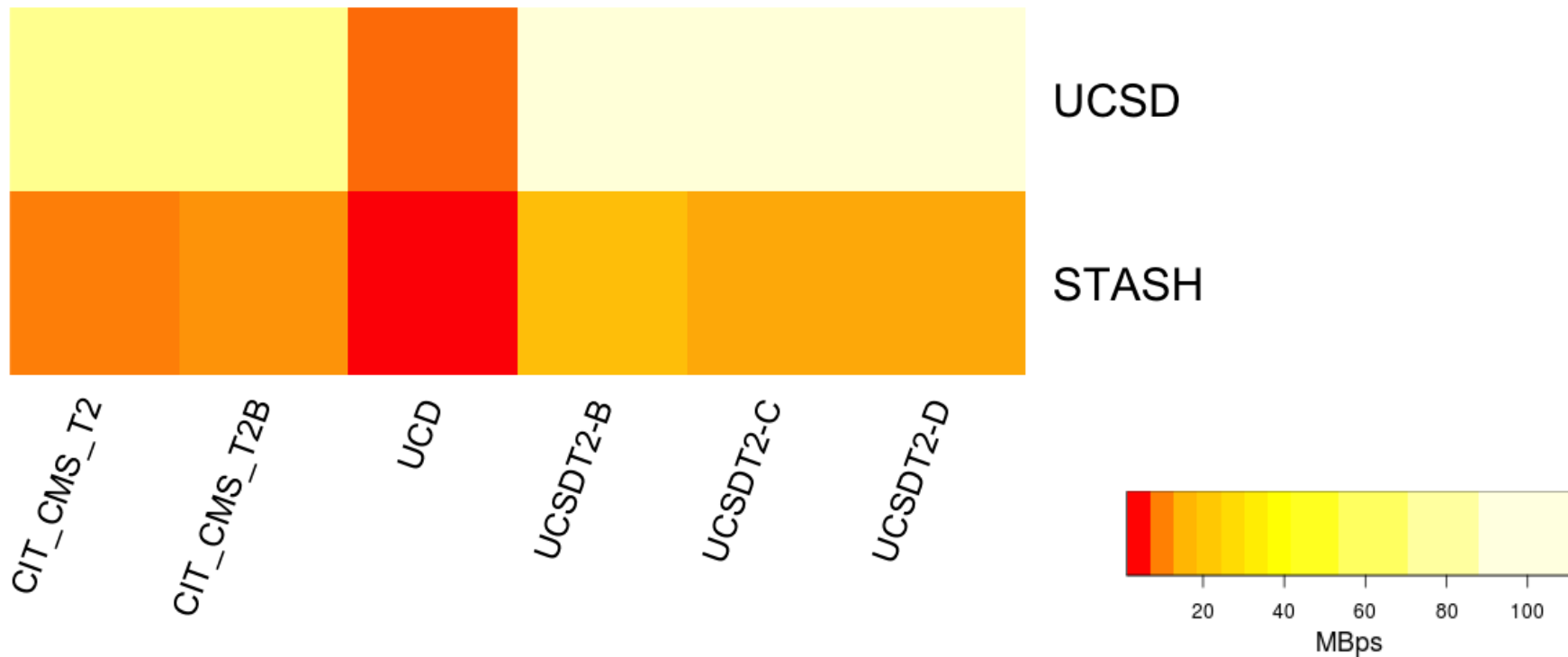
Distribution of speeds: big files

Single-pull tests

- Only one job active at any given time
  - Pulls 5-10 files in series
  - No competition
- Sources: UCSD, STASH
- Files were copied to /dev/null
- File size ranged from 134MB to 4.3GB

Distribution of speeds

Comparison of median speeds

Conclusions

- Caches that share a LAN with OSG nodes seem to work very well
- Simply looking for the nearest cache location is not always best
- Simple setup is not so simple (but it's still very minimal!)

Next steps

- Instrument each stash copy, monitoring and mining using flume/hadoop/ES/Kibana
- Deploy more caches at OSG sites as deployment is refined
- Testing with common Galaxy application ➡ provides alternative to iRODS infrastructure currently in use on OSG

More information

- Project: http://goo.gl/qZaqMS
- olsona@cs.uchicago.edu

Network of proxy caches

- Origin servers
  - OSG Stash
  - UCSD (to come)
- Proxy caches
  - AAA XRootD proxy cache
  - Simple server:
    - few TB
    - 10 Gbps NIC
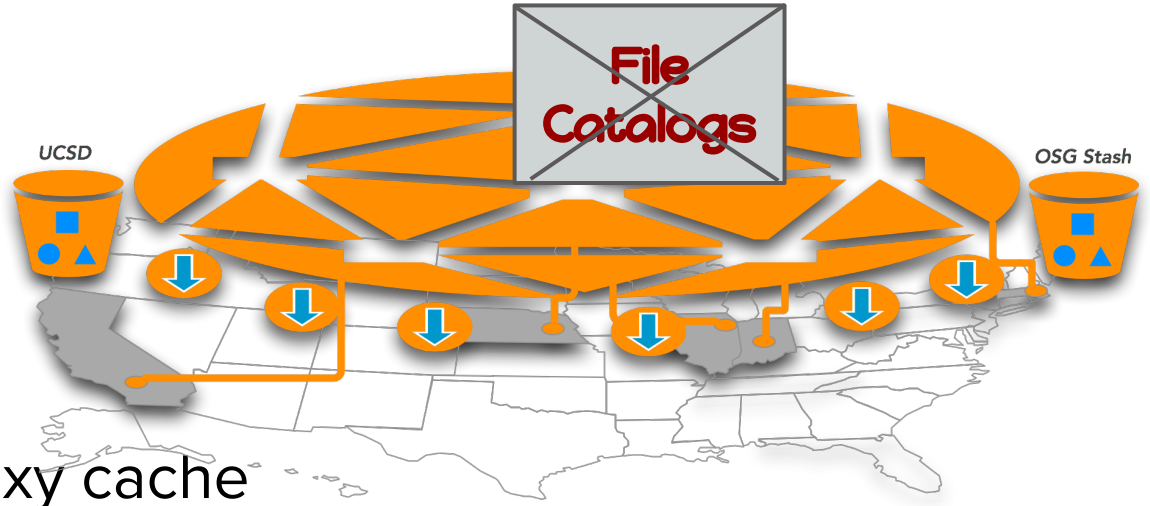
Initial caches: UC, IU, UCSD, BU

Application - genome analysis

Sequence Alignment of DNA, RNA and Proteins

- Find the relation between different species (phylogenetic tree)
- Predict the structure and function of proteins and RNA

Application: genome analysis

- The size of database is 12 GB for the current test case. BLAST search on 12 GB database requires more than 12GB RAM
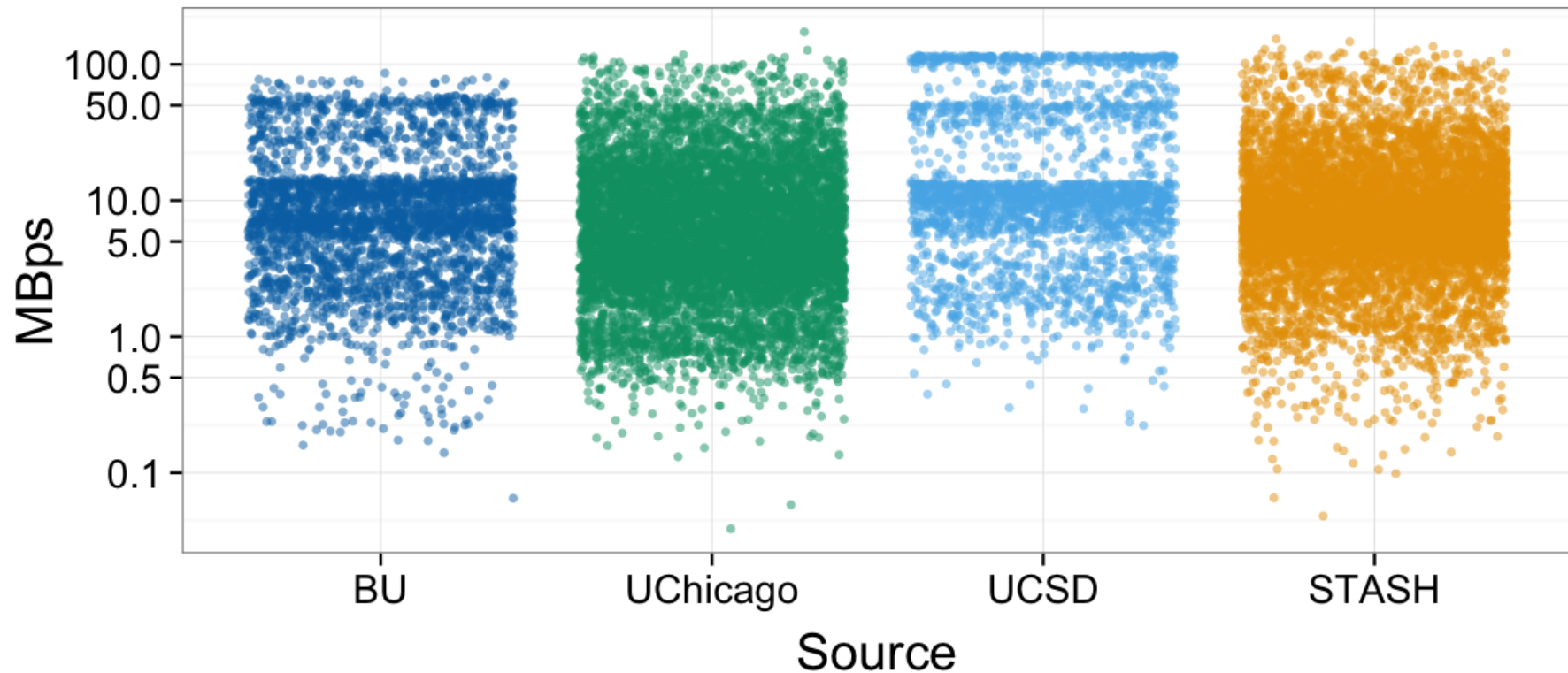
Workflow

- Split the database into several files < 1GB
- Use stashcp to copy the data files from stash to the worker machines
- Perform blast analysis on each file and collect the results

Speeds by File Size
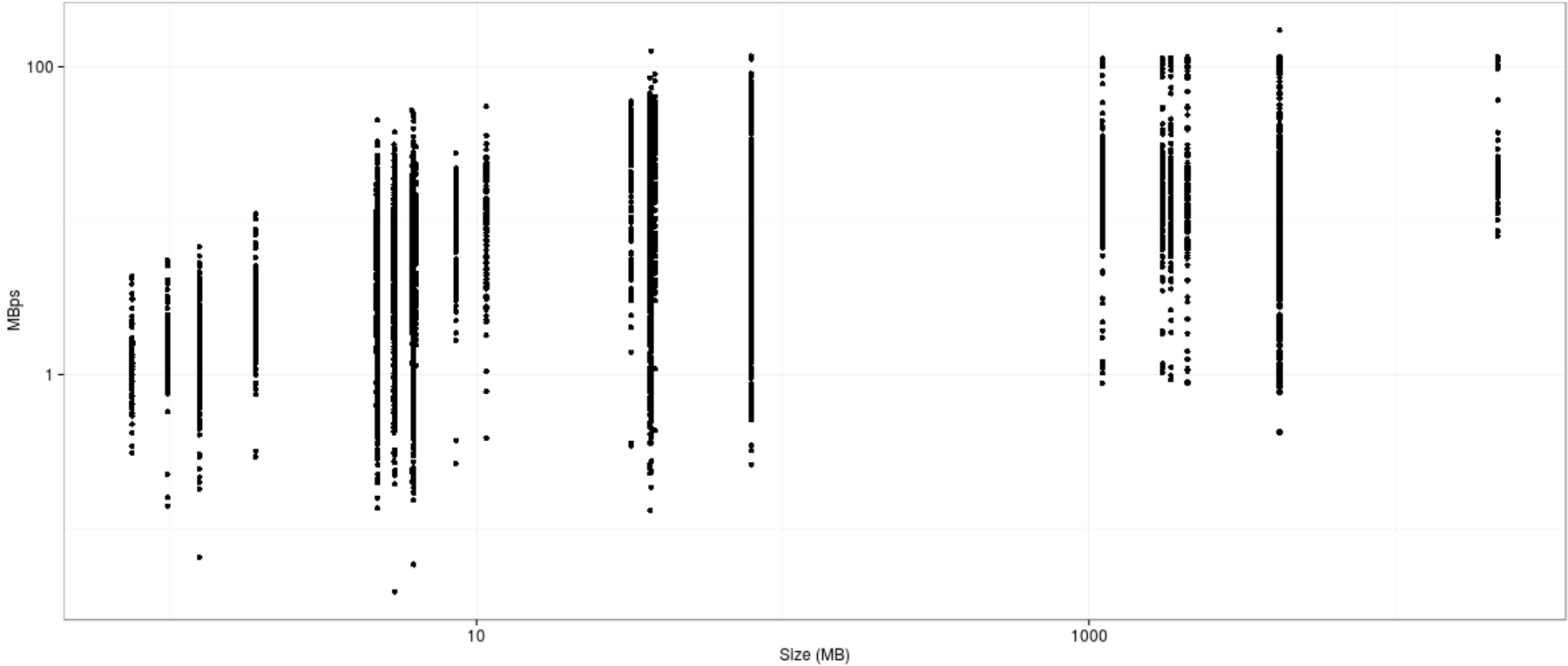
Speeds by source: all files

Speeds by source: big files