StashCache: A Distributed Caching Federation for the Open Science Grid

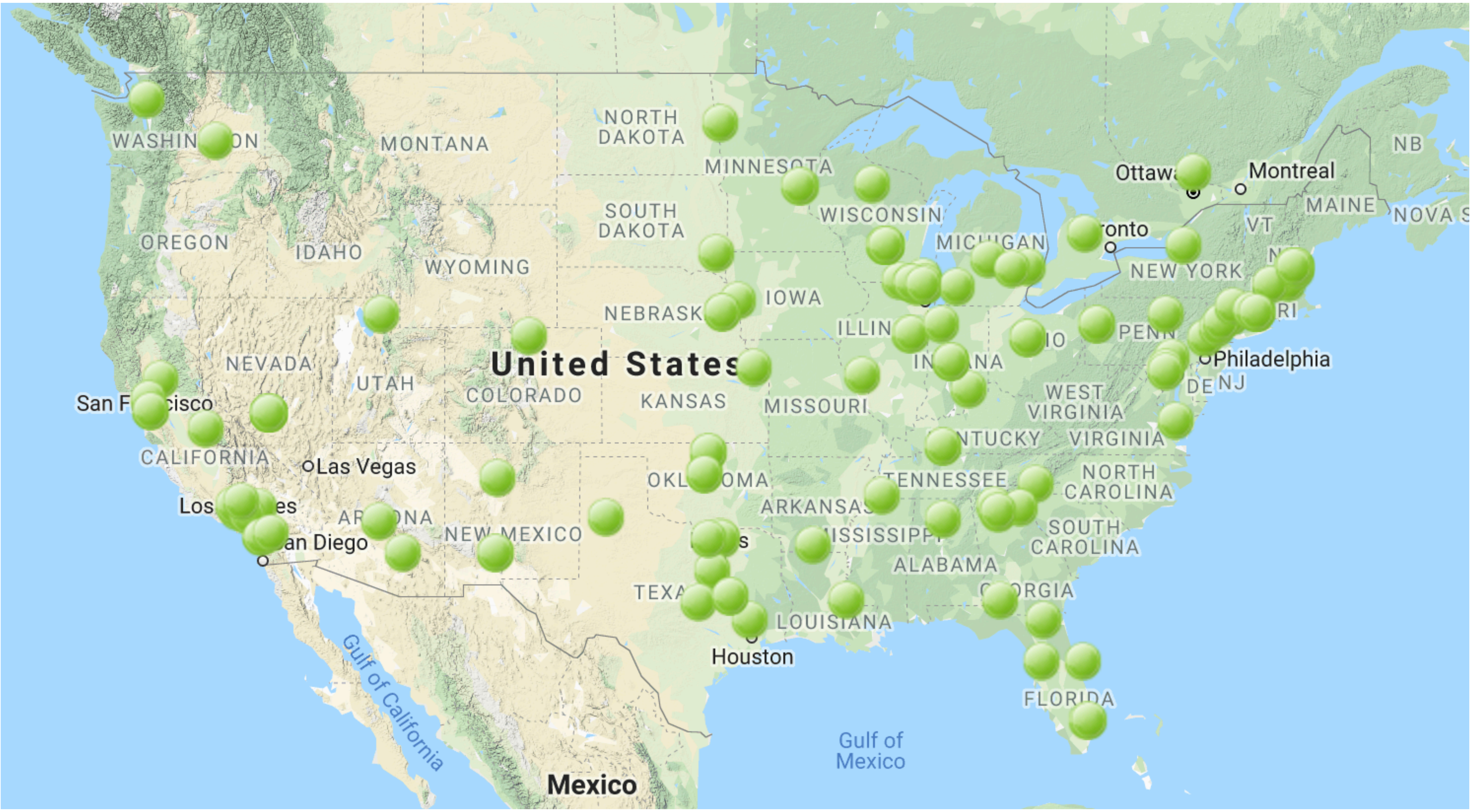
Derek Weitzel
University of Nebraska—Lincoln
On behalf of Marian Zvada, Ilija Vukotic, Rob Gardner, Brian Bockelman, Mats Rynge, Edgar Fajardo Hernandez, Brian Lin, Mátyás Selmeci

Imagine a Scenario

- Jane the biologist wants to run BLAST
- She can run it on your laptop, but it’ll take weeks to complete the queries.
- She chooses to run the jobs on cyberinfrastructure like the Open Science Grid

DHTC on the OSG

- The Open Science Grid provides DHTC services to hundreds of researchers across 100+ sites



Opportunistic Users

- Jane doesn’t own any hardware on the OSG

- Sites allow opportunistic usage of computing resources, but not storage

- Opportunistic users do not have a dedicated method for distributing data throughout the U.S.

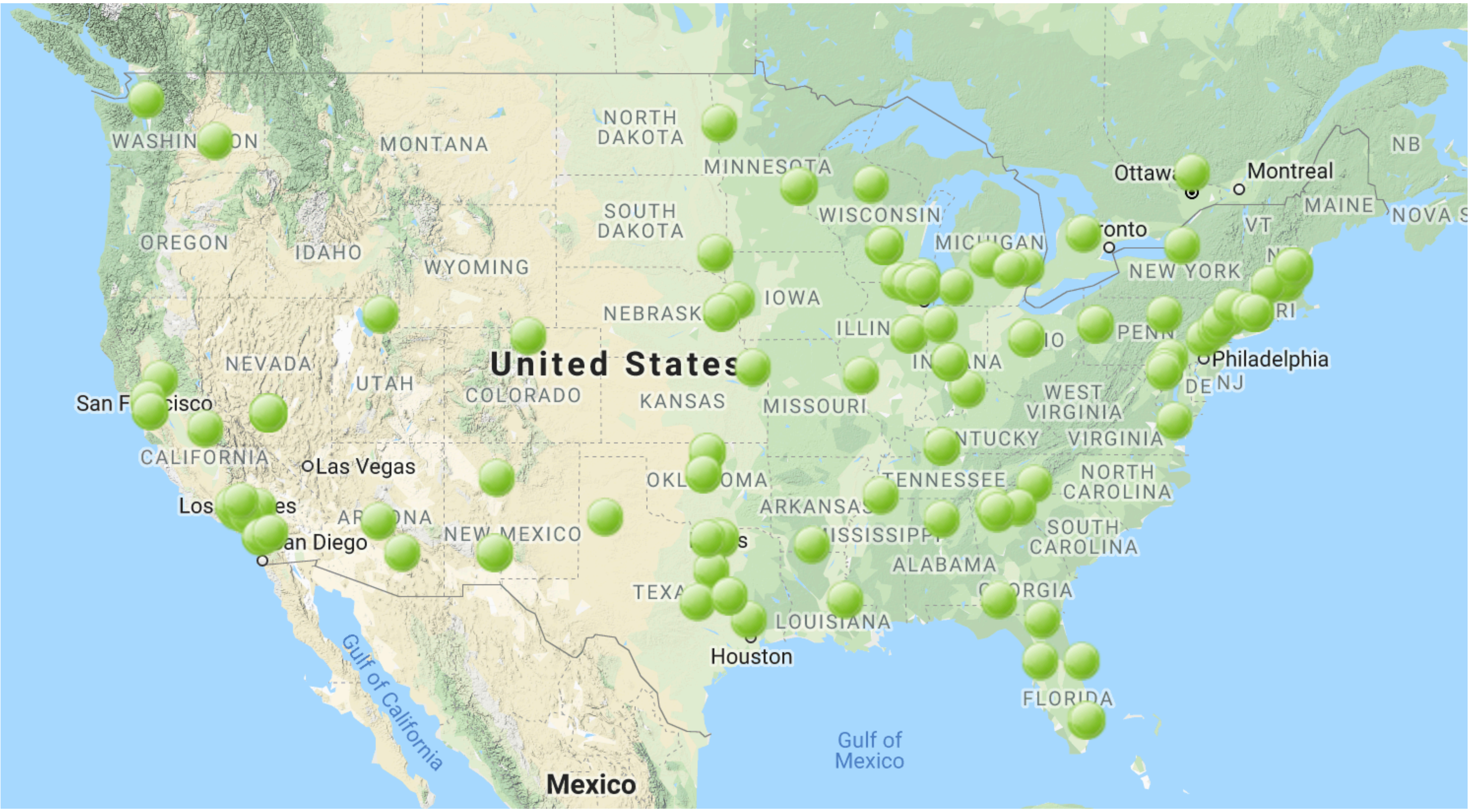
- Non-Opportunistic users have built entire custom frameworks to geographically distribute data

- If Jane transferred her database to all of the sites her jobs may run (20+), then many jobs will pull the data

Opportunistic Users

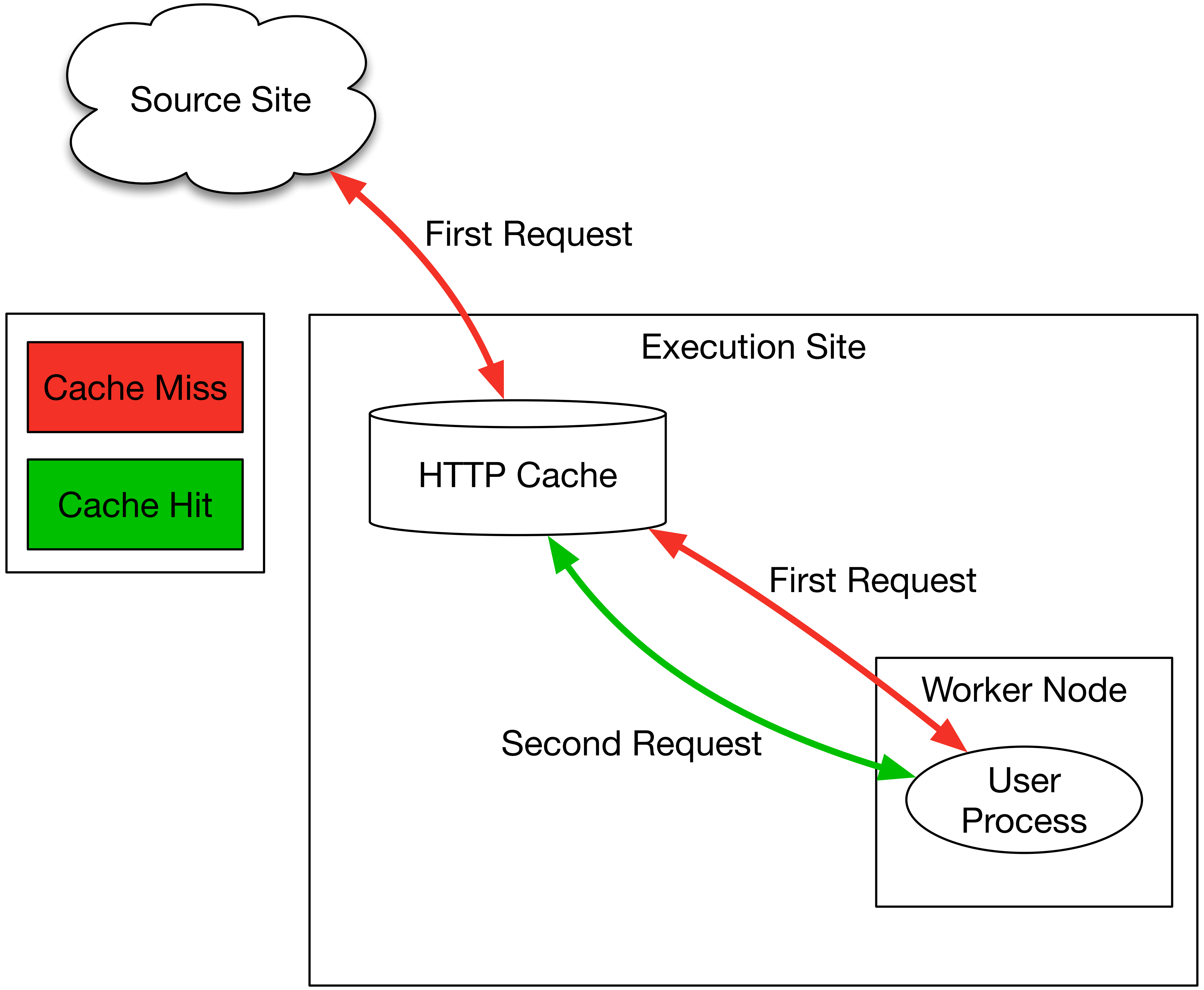
- They still need to distribute data with their processing, even though they don't own nearby storage or the computing

- It needs to distribute to multiple sites across the U.S.



HTTP Proxies

- Each site has an HTTP Proxy deployed to assist in data transfers
- HTTP proxies where originally designed for small files less than a few MBs
- They are configured to not cache files larger than several GBs (site dependent)



Source Site

First Request

Cache Miss

Cache Hit

Execution Site

HTTP Cache

First Request

Worker Node

Second Request

User Process

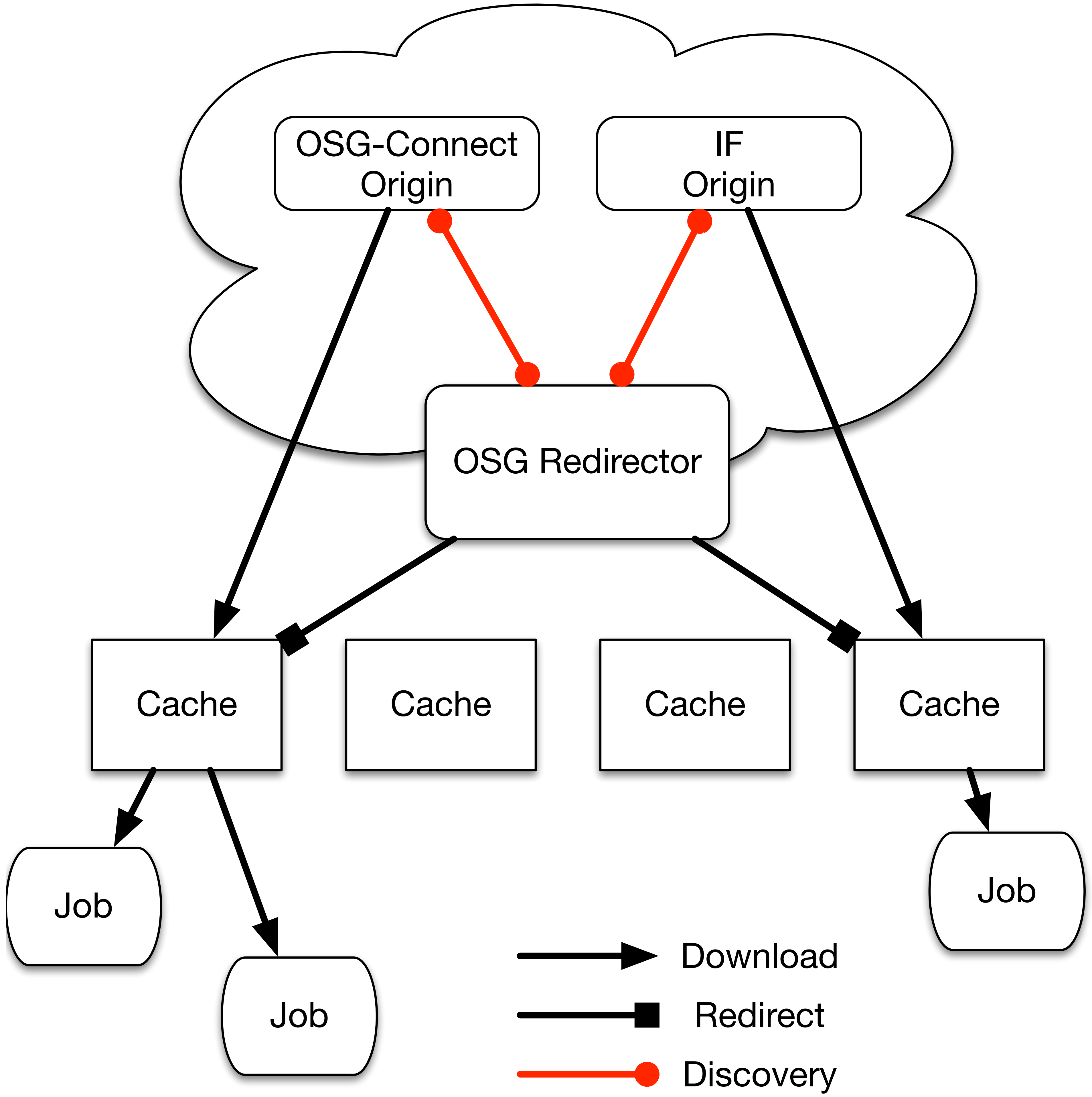
StashCache Overview

- Distributed Regional Caches
- StashCache is based on the XRootD technology
- XRootD provides the services for data discovery and caching

StashCache

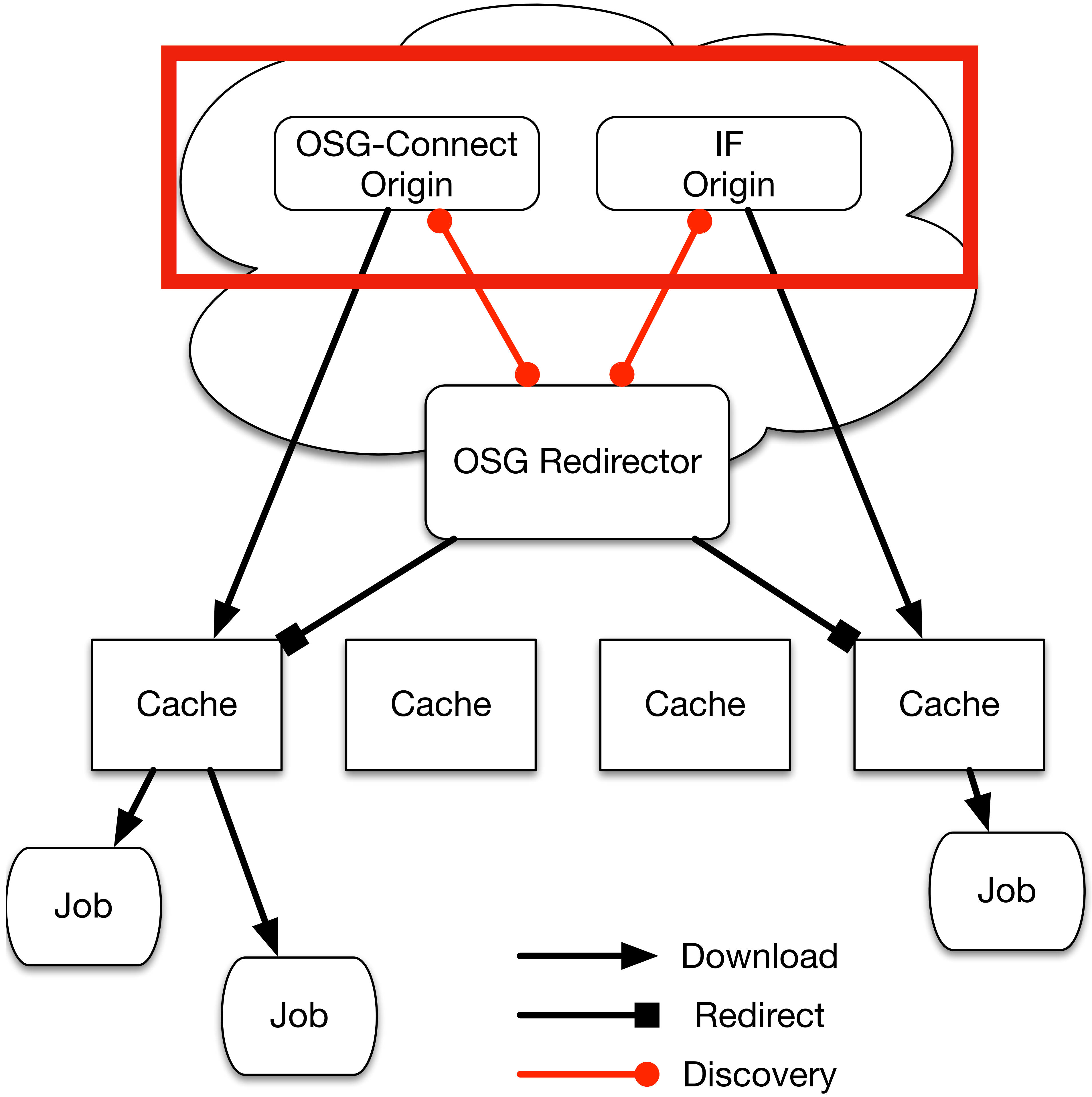
Four Components of StashCache

1. Data Origin
2. Data Cache
3. Redirector
4. Clients



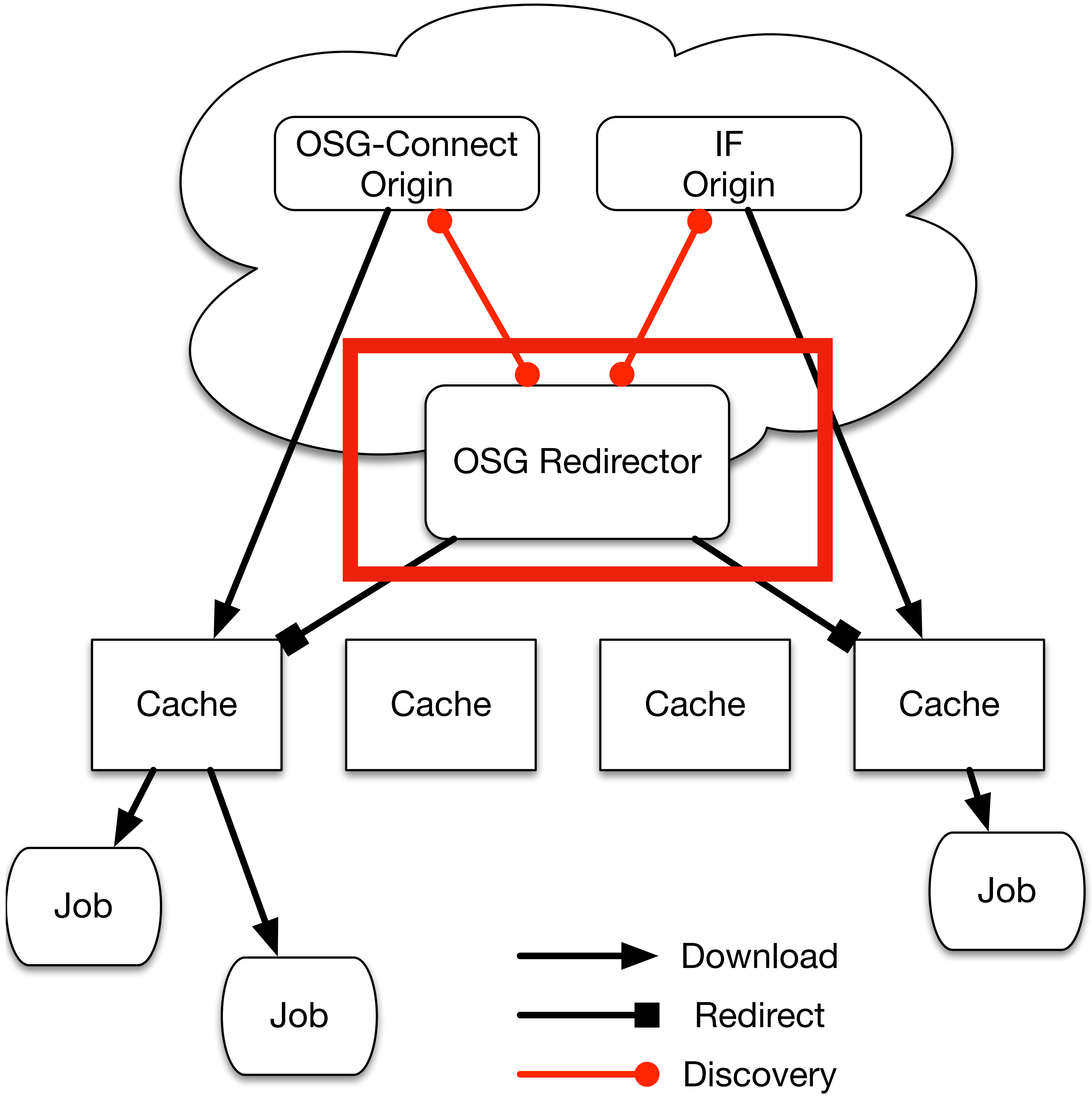
StashCache - Origins

- Data Origin is the authoritative source of data
- Each organization has their own Origin
- The Origin answers file existence questions from the Redirector
- The Origin sends requested data to the Caches



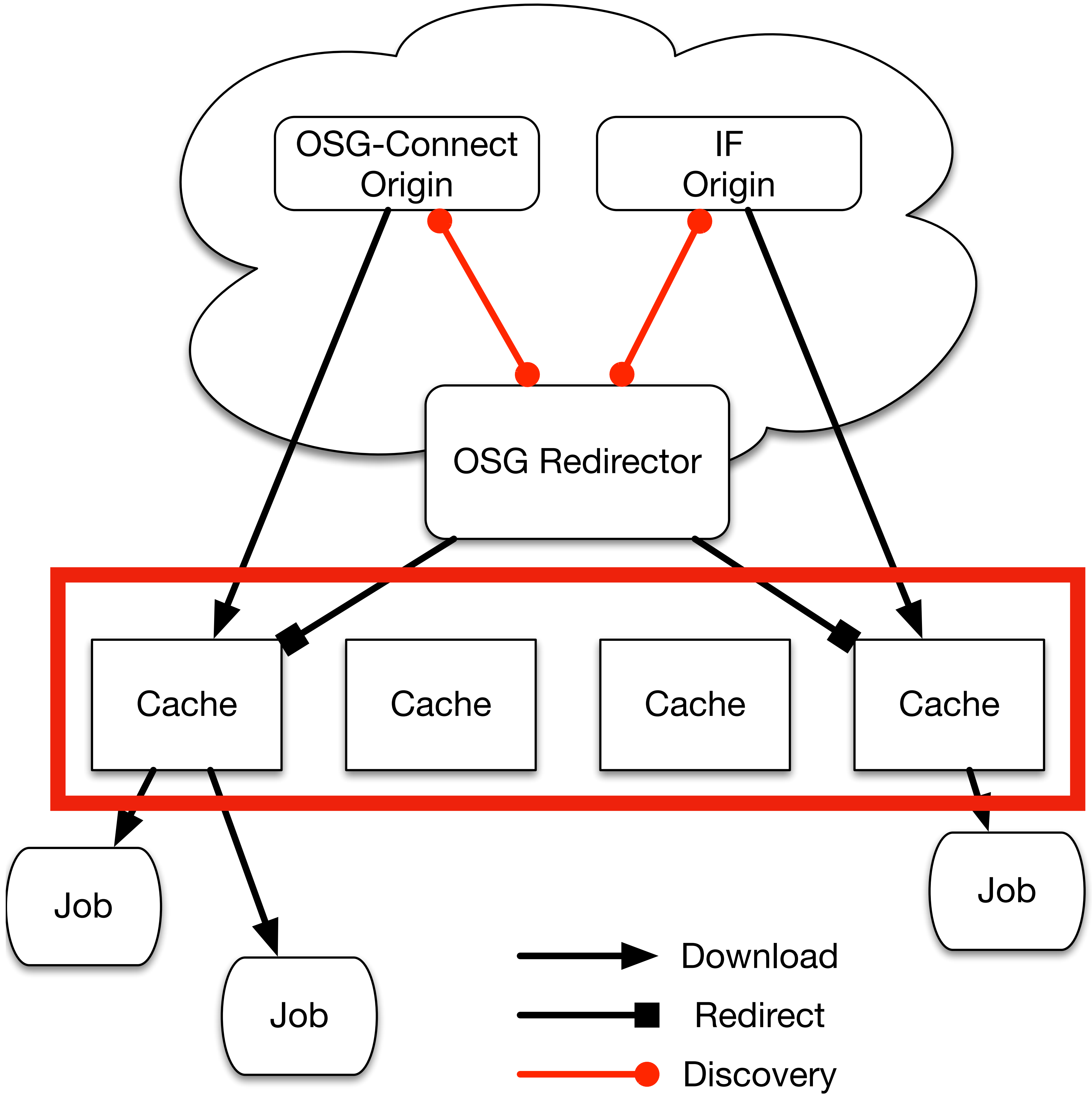
StashCache - Redirector

- Redirector responds to data location requests
- Queries all of the origins to find source of data
- Redirects clients to the Origin serving the requested data



StashCache - Caches

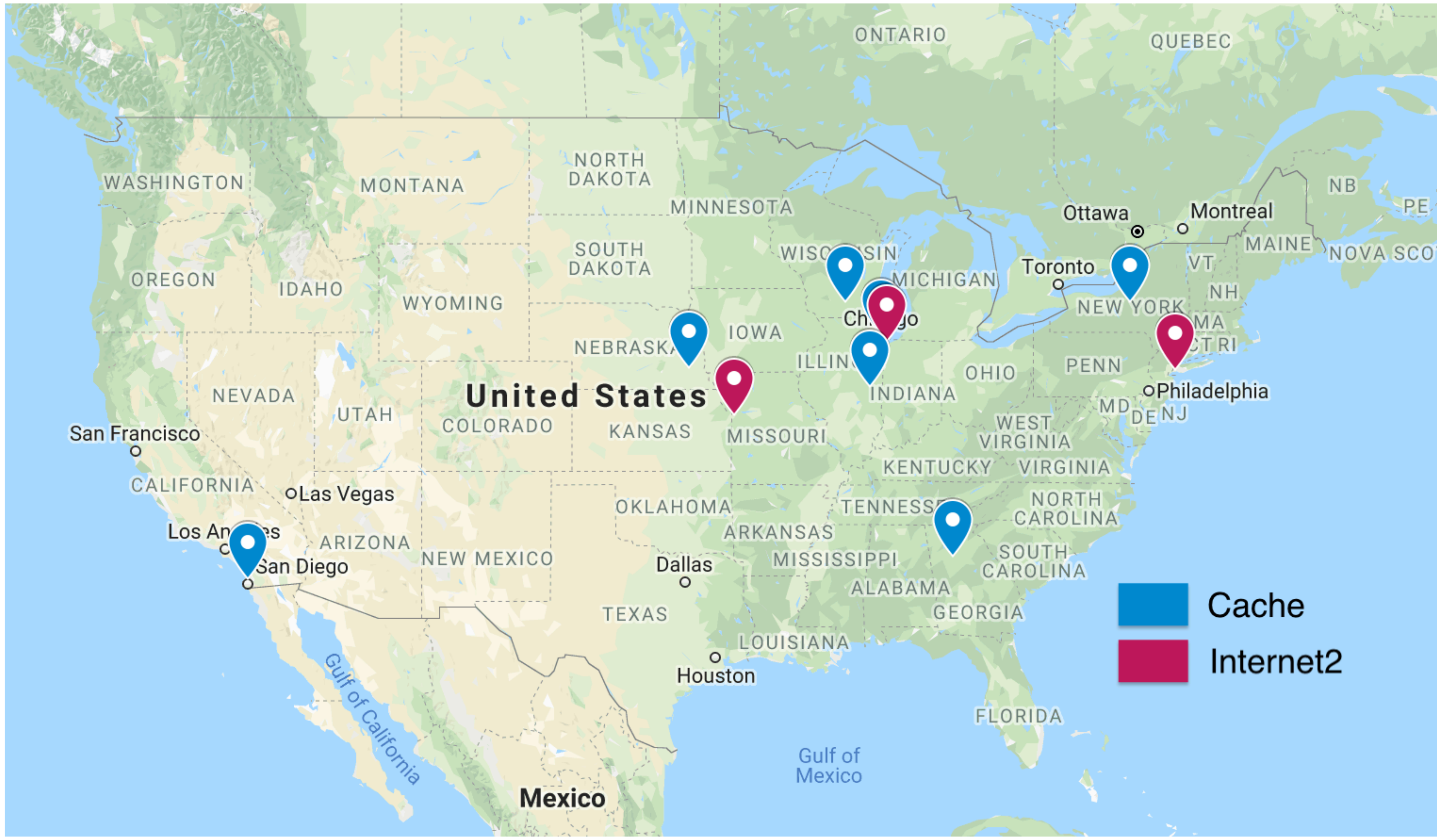
- Caches listen for requests from the clients
- Caches ask the redirector for the location of the data
- Downloads data from Origins
- Stores the data locally for future requests



StashCache - Caches

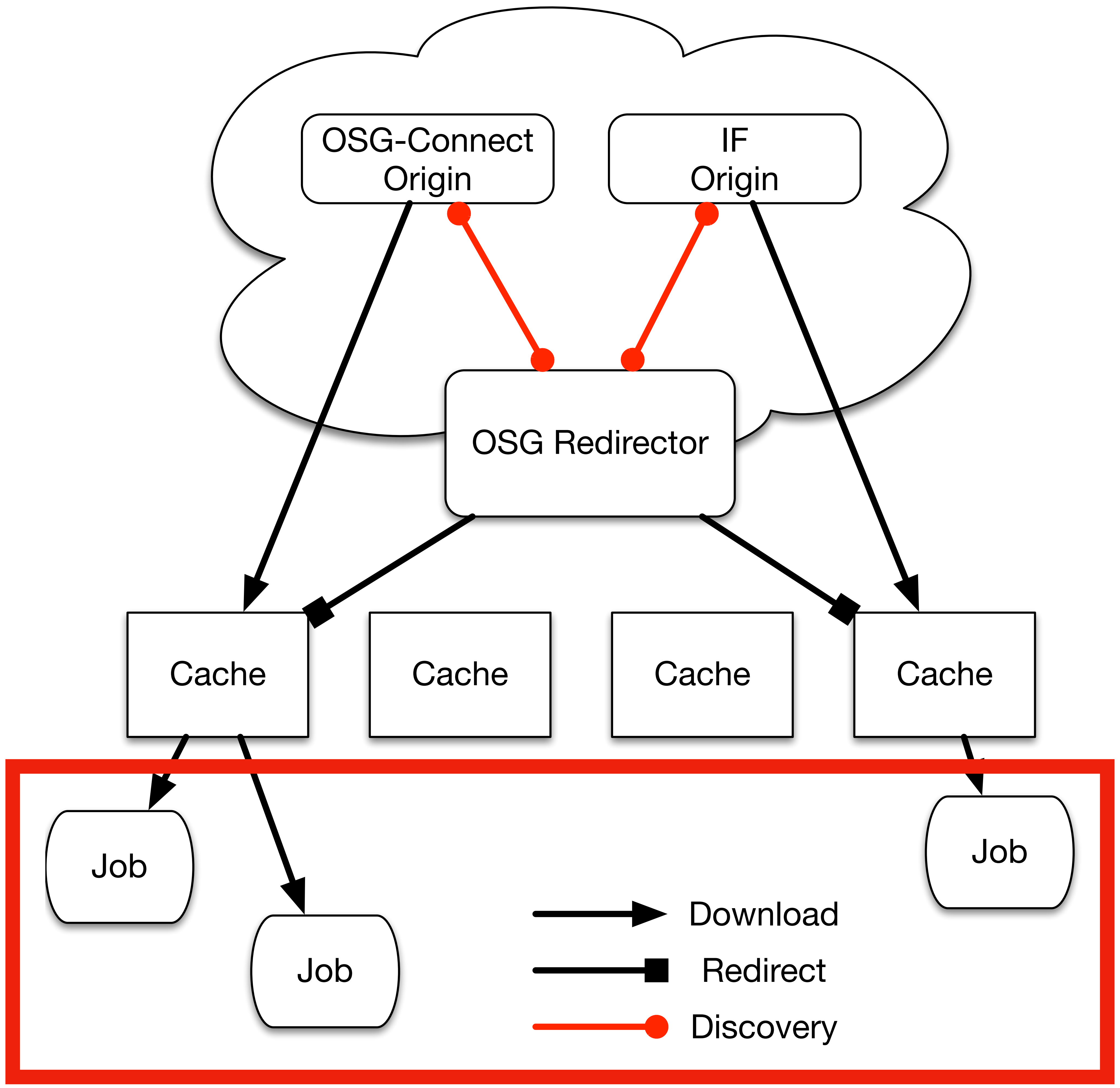
- Caches located throughout the U.S.

- Caches also located on the Internet2 backbone



StashCache - Clients

- Clients act on behalf of users to download data from caches
- Clients find the “nearest” cache through GeoIP



StashCache - Clients

- Two clients are available for StashCache:
- **CVMFS:** A transparent directory on the worker node which pulls data on demand from Caches.
- **StashCP:** Similar to “scp” tool. Copies entire file from StashCache federation to worker node.

Clients - CVMFS

- Presented as a directory on the worker node

```
$ ls /cvmfs/stash.osgstorage.org/user/bio_jane/public/blastdb/
yeast.aa  yeast.aa.phr  yeast.aa.pin  yeast.aa.pnd  yeast.aa.pni  yeast.aa.psd  yeast.aa.psi  yeast.aa.psq
```

- Metadata such as directory structure is stored in CVMFS
  - File size, permissions, and checksums of files
- Data is pulled from StashCache caches
- Only the parts of data which are read are downloaded.
- An indexer is constantly running to scan several Origins creating the metadata for CVMFS.

Clients - StashCP

- StashCP does not require the indexer to find the file, therefore it is instantly available
- The indexer can take 8+ hours to scan an origin
- StashCP will directly download the entire file from the Caches.
- Uses GeoIP to find the “nearest” cache

Experiments

- Compare HTTP Proxy and StashCache
- Use data sizes representative from what is actually used on StashCache
- Use real sites
- Download each file 4 times, 2 times for HTTP to show caching improvement and 2 for StashCache for caching.

Experiments

- Choose data sizes for the from the Percentiles of actual usage
- Created test data files of each of the percentiles
- Created experiments to download and time the download for each file size
- Additionally, added a 10GB data file to show future, larger data size potential

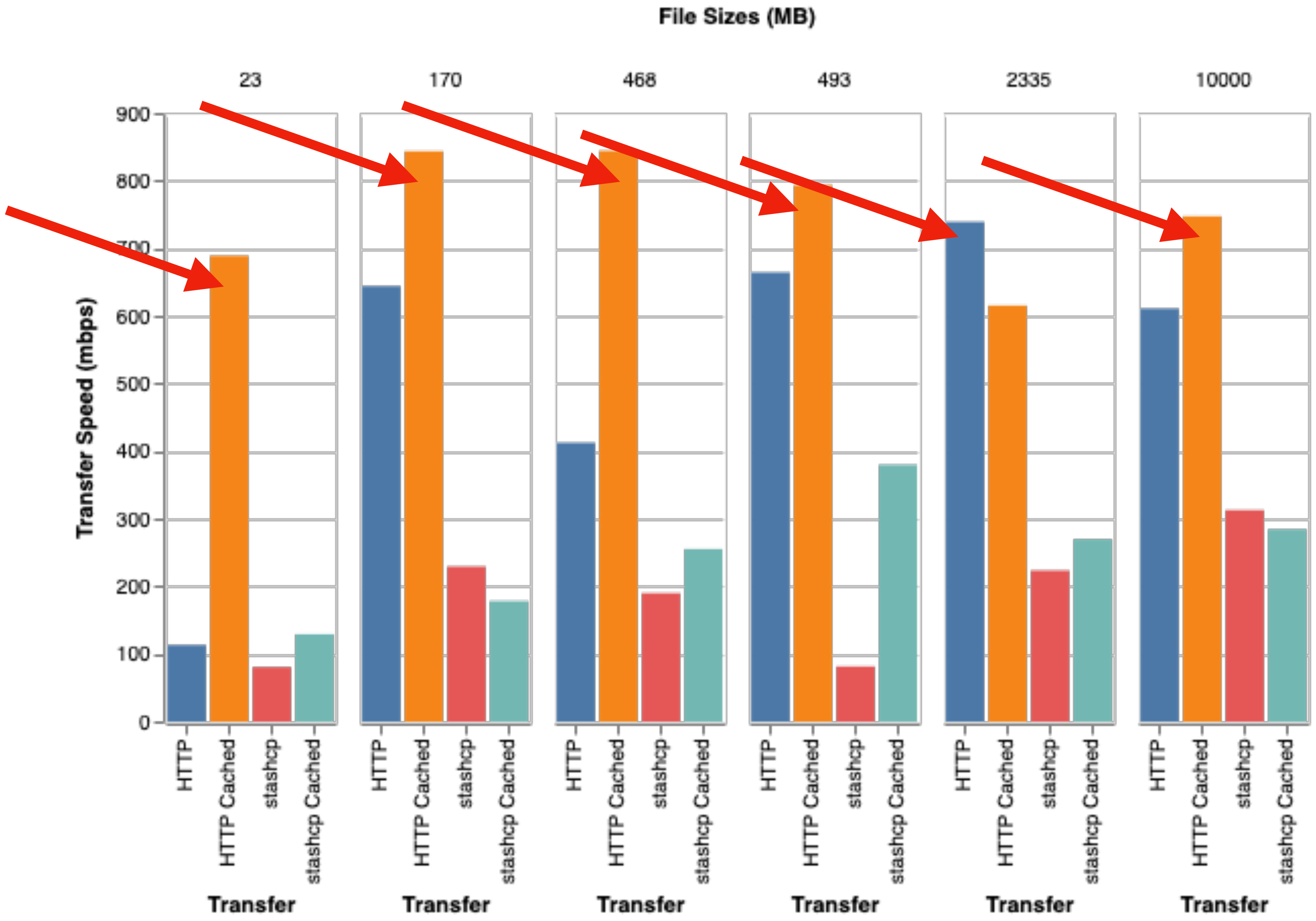
Percentile	Filesize
1	5.797 KB
5	22.801 MB
25	170.131 MB
50	467.852 MB
75	493.337 MB
95	2.335 GB
99	2.335 GB

Sites for Experiments

- Chose the top 5 opportunistic sites on the OSG for the previous 6 months:
    - Syracuse University
    - University of Colorado
    - Bellarmine University
    - University of Nebraska - Lincoln
    - University of Chicago

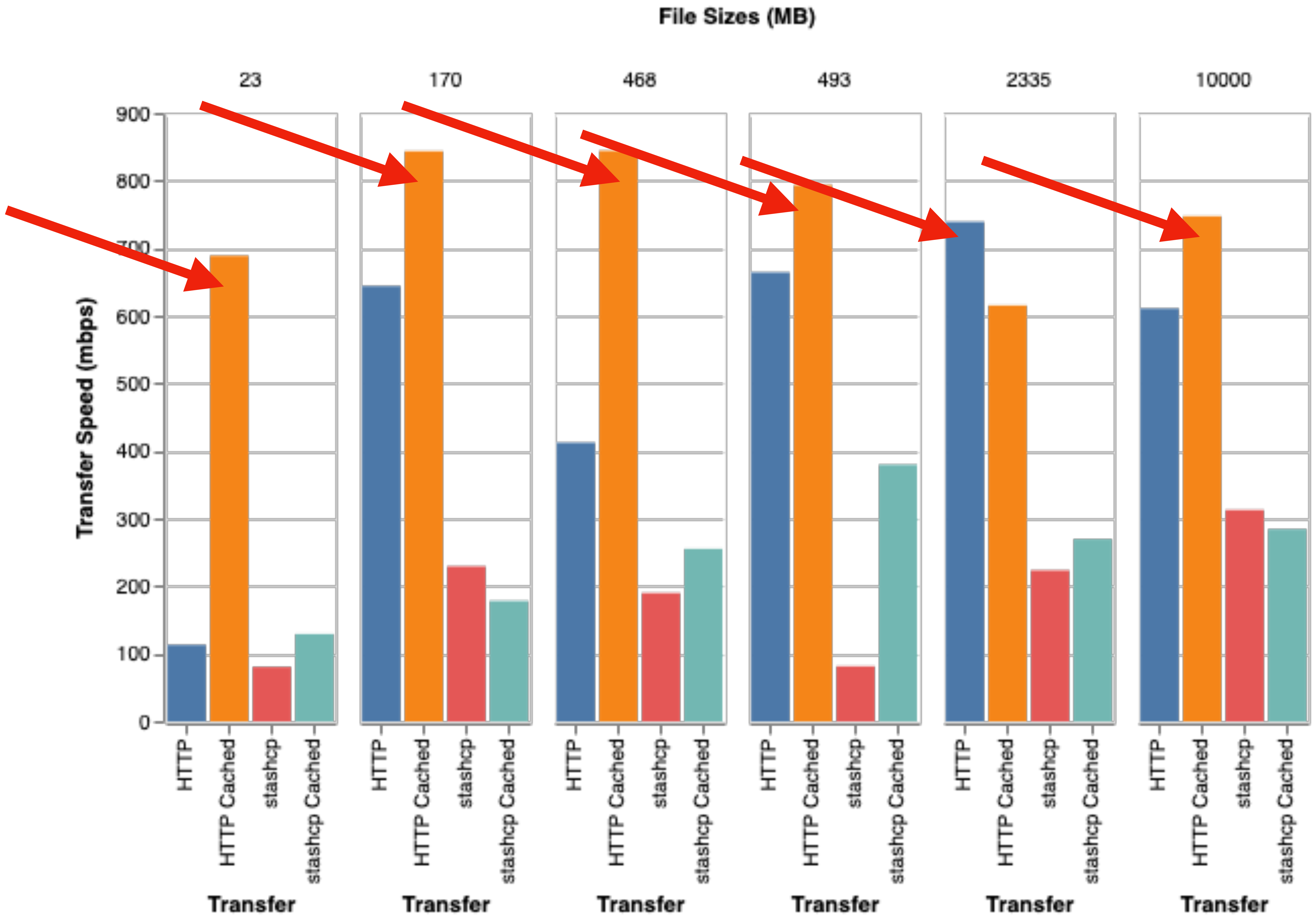
Notable Site - Colorado

- At Colorado, HTTP Proxies are always faster than other transfer methods



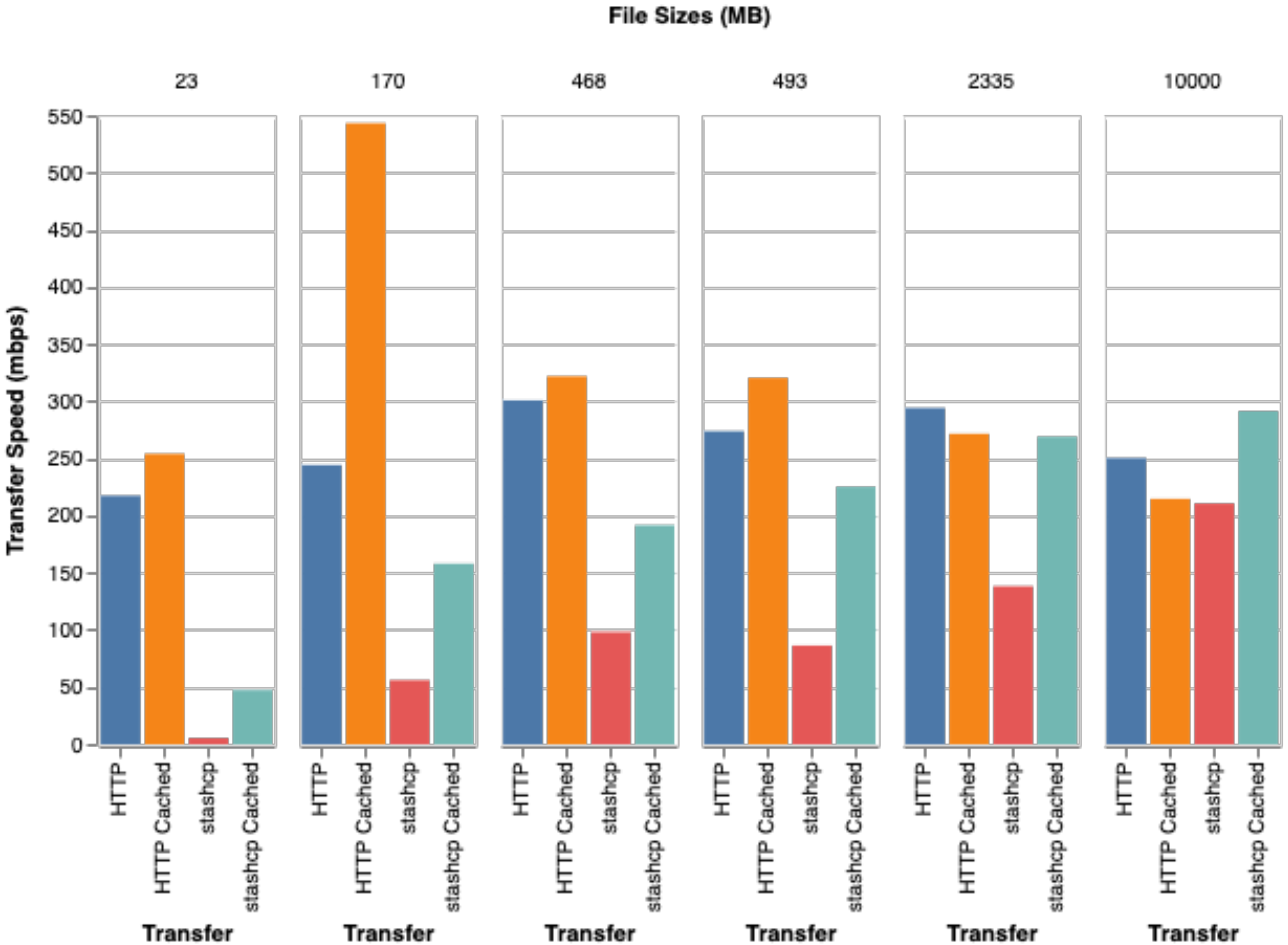
Notable Site - Colorado

- I contacted Colorado to ask about the discrepancy. They prioritize network between the proxy and the WAN over the worker nodes.



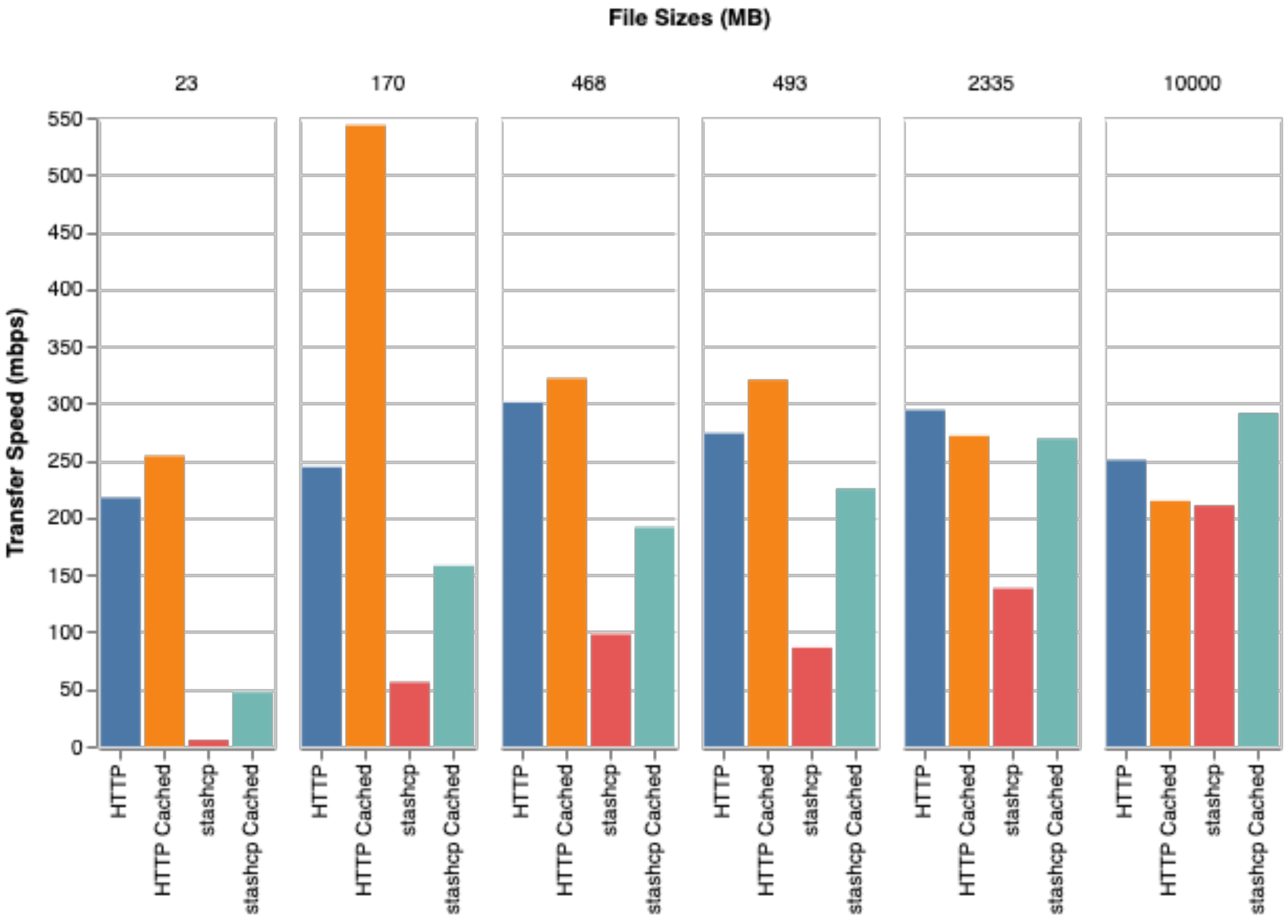
Notable Site - Syracuse

- For small files, HTTP wins. But as the file size increases, StashCache is better.



Notable Site - Syracuse

- StashCP used for these tests has a large startup time for GeoIP lookups



Large Files

- Percent difference between HTTP Proxy and StashCache.
- Negative values indicate the time to download decreased when using StashCache
- Sites vary widely.

Site	2.3GB	10GB
Bellarmine	-68.5%	-10.0%
Syracuse	0.9%	-26.3%
Colorado	506.5%	245.9%
Nebraska	-12.1%	-2.1%
Chicago	30.6%	-7.7%

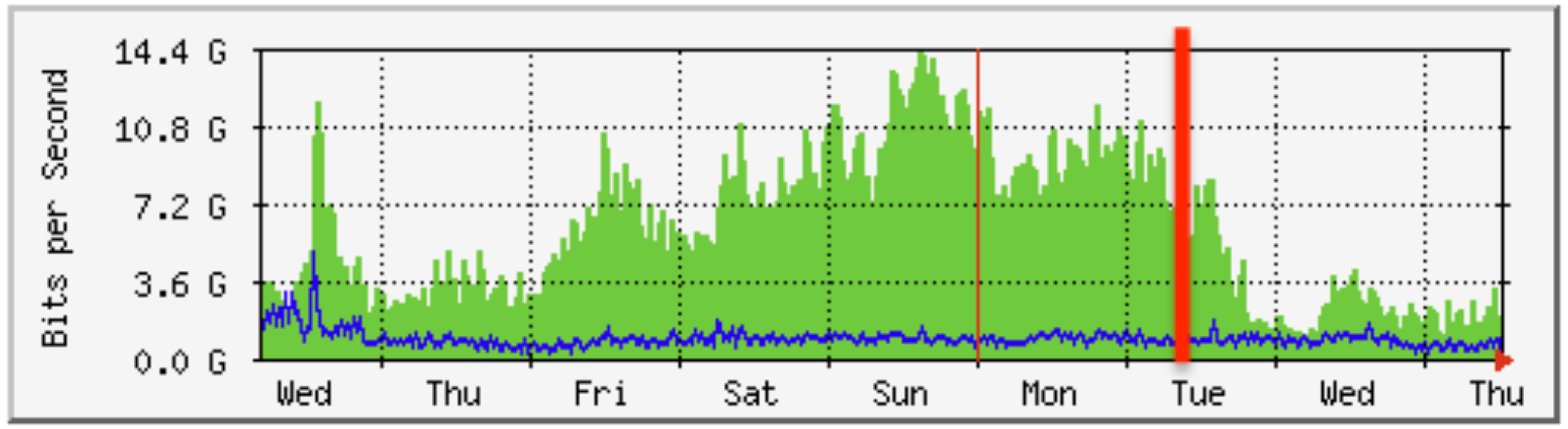
Results - Notes

- The HTTP cache never caches the 10GB file
  - The caches are configured to not cache files over a set size (configured by the site)
- Some sites have very fast WAN connections
- Each site has different behavior

Effect on WAN

- Syracuse installed StashCache while workloads were running
- Noticeable affect on WAN network traffic
- Cache freed WAN network for other users

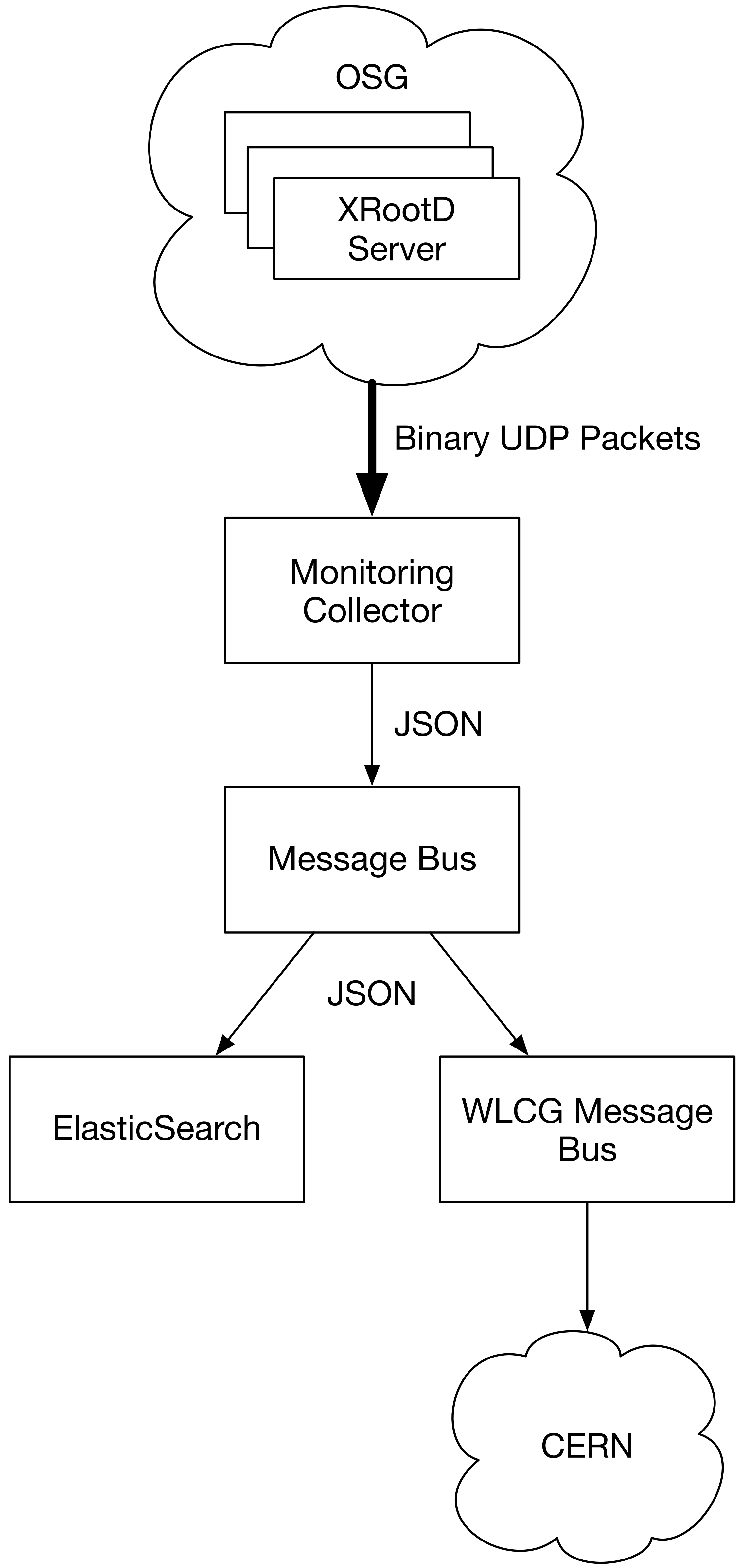
`Weekly' Graph (30 Minute Average)



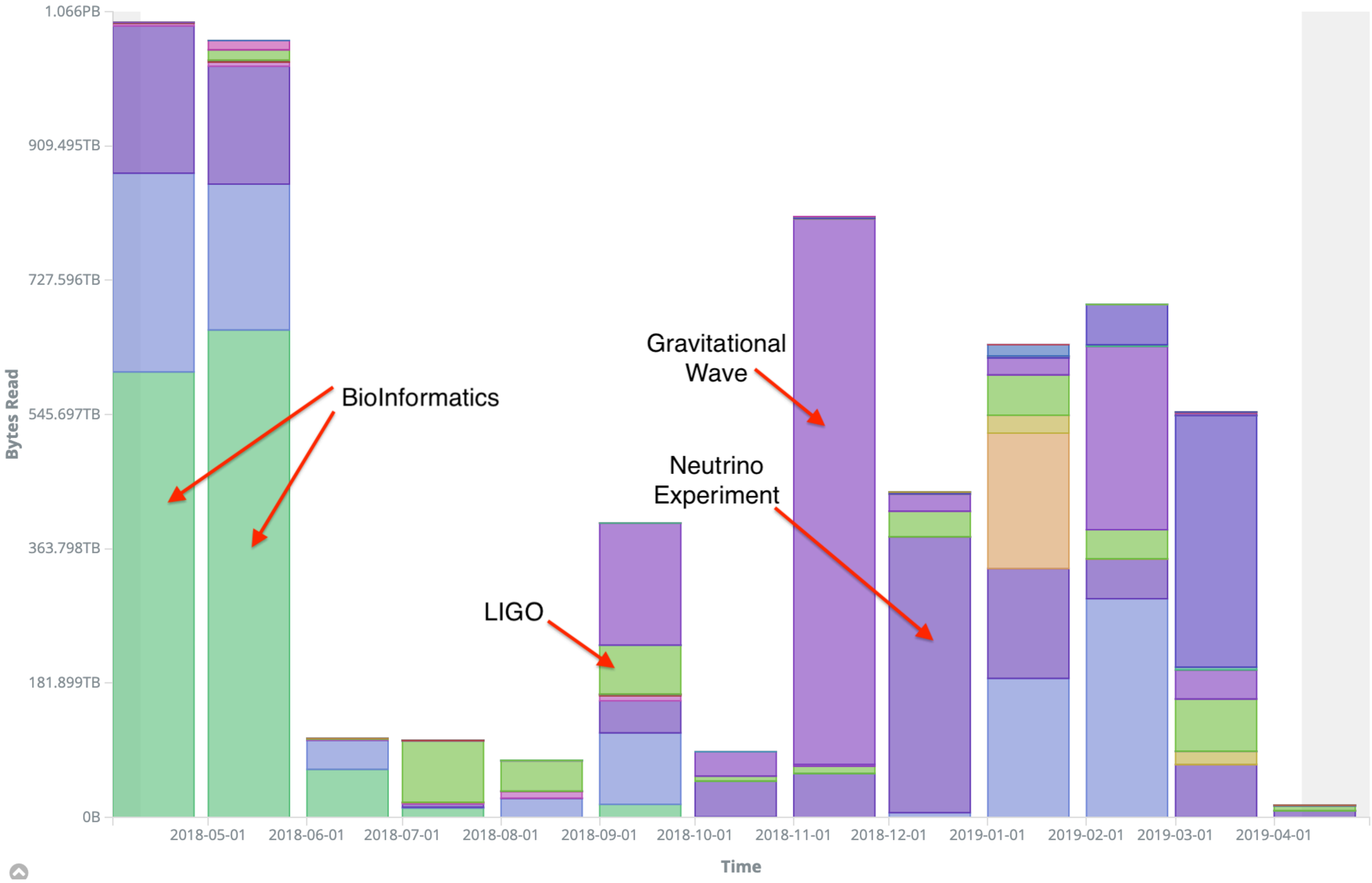
	Max	Average	Current
In	14.3 Gb/s (35.8%)	5794.3 Mb/s (14.5%)	1658.6 Mb/s (4.1%)
Out	4835.0 Mb/s (12.1%)	849.8 Mb/s (2.1%)	1172.9 Mb/s (2.9%)

Monitoring

- Each Cache sends monitoring information.
- Receive data such as what directories are downloaded, and from which domain



StashCache Usage



Top Users of StashCache

Experiment	Usage
Open Gravitational Wave Research	1.079PB
Dark Energy Survey	709.051TB
MINERvA (Neurtrino Experiment)	514.794TB
LIGO	228.324TB
Continuous Testing	184.773TB
NOvA	24.317TB
LSST	18.966TB
Bioinformatics	17.566TB
DUNE (Neutrino Experiment)	11.677TB

StashCache Packaging

- Origins and Caches are packaged as RPM’s and Docker containers (with kubernetes configs)
- They are distributed by the Open Science Grid
- Documentation available at: bit.ly/stashcache-docs

Conclusions for Jane

- Jane can use StashCache to distribute her blast database
- The database will be cached on the regional caches
- She can run Blast on

Conclusion

- StashCache provides a data distribution method for opportunistic users
- For small files, Proxied HTTP outperforms StashCache
- But, for larger files, StashCache outperforms at most sites