

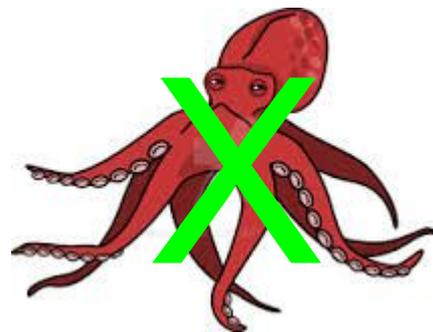
# Update on Xcache

Wei Yang  
Andy Hanushevsky

# What is the Xcache

Squid like Cache but speak the “root” protocol:

- XROOT\_PROXY similar to “http\_proxy”
- Caching at sub-file level, or whole file
- Multi-thread, async data fetching
- Designed for both large and small static data files
- Clusterable for scaling up
- ROOT and HTTP protocol between client and Xcache
  - ROOT between Xcache and data source, will support HTTP



Goal: improve remote data access efficiency.

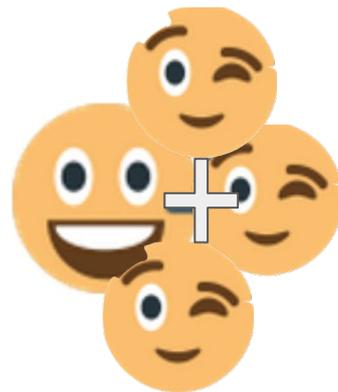
# Plugin Architecture

Plugin enable additional functions to base Xcache.

- Intercept the request and do something about it.

For example:

- Make decision on what to cache, and what to NOT cache
- Map a logical file name to the best data source
  - For example, querying RUCIO for a list of data sources in a metalink, sorted by GeoIP, then use the metalink as data source
- Where to place the cached data file
  - Identify the same files at different data sources and share a cache entry

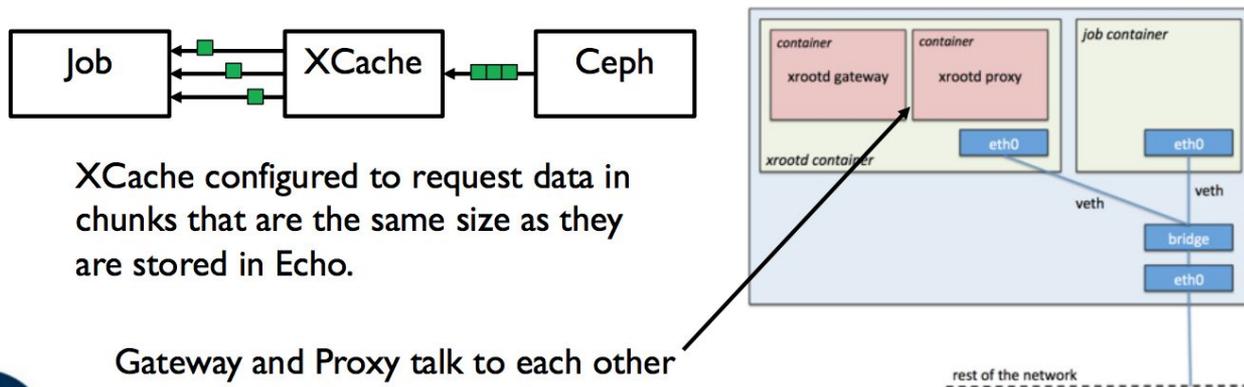


# Activities on Xcache Applications

1. OSG Stash Cache and CMS user analysis (ask Frank W.)
2. [Xcache to optimize direct IO access to CEPH storage \(RAL\)](#)
3. Xcache for CVMFS
  - [Successfully and transparently replaced squid in CVMFS](#)
4. [RUCIO metalink plugin for Xcache \(SLAC\)](#)
  - Access data using LFN.
  - Also an example of developing plugin in Xcache
5. [Eval and integration with ATLAS workflow \(Edinburgh\)](#)
6. Volatile RSE - has an initial discussion, need follow up investigation
7. Xcache for ESS
  - Initial test shows that they can work together

# RAL XCache Setup

- A Ceph plugin has been written for XrootD.
- Object stores don't support vector reads.
- XCache is used to optimize access for jobs using Direct I/O.
- Container on each WN runs XrootD gateway which talks to Ceph and XrootD proxy which users will talk to.

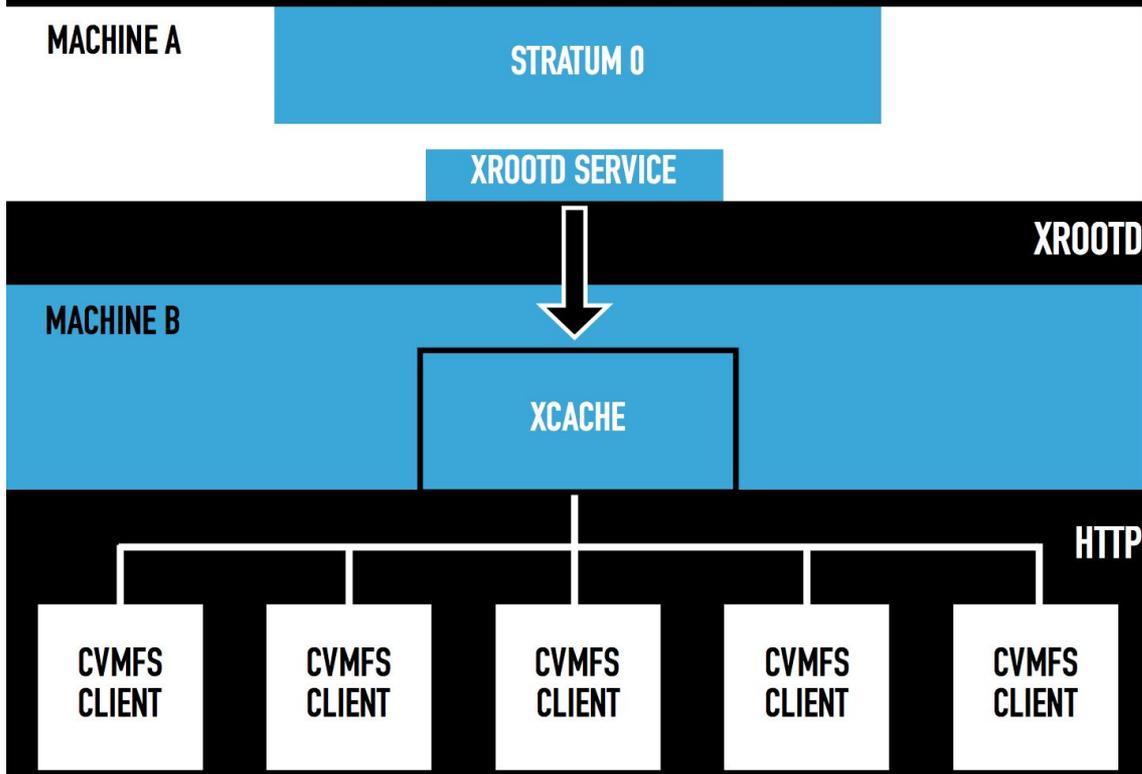


XCache configured to request data in chunks that are the same size as they are stored in Echo.

Gateway and Proxy talk to each other using Simple Shared Secret Key.



# SETUP



## Xcache to replace Squid (scale up)

- Export BNL Stratum-1 via xrootd
- Setup SLAC Xcache for S-1
- SLAC batch nodes point to Xcache instead of squid
  - Only need two lines of config change in the cvmfs client.

Next to investigate:

- Replace CVMFS cache on WN by a centralized site level cache
  - Replace stratum-1s by Xcache clusters
- CVMFS as a content delivery network for some data files

# Xrootd Read-Only Cache

Example: What you can do with a Plugin

## Cache file blocks

### SQUID-like:

Xcache URL:

**xroot\_proxy**="root://local.cache"

### Optional

Update RUCIO:  
add, delete files



root://local.cache//scope:file

MetaLink XML file:

<"1" root://RAL//dir1/scope/fa/6b/file>

<"2" root://FZK//dir2/scope/fa/6b/file>

root://local.cache//root://RAL//dir1/scope/fa/6b/file

RAL: /dir1/scope/fa/6b/file

root://local.cache//root://FZK//dir2/scope/fa/6b/file

FZK: /dir2/scope/fa/6b/file

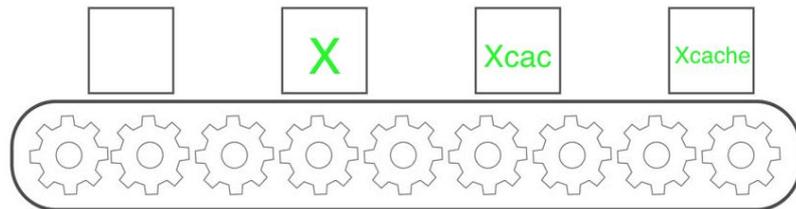
/scope/fa/6b/file



# Moving forward on (continuous) development

0. Core development (@UCSD/SLAC) ended a year ago.
1. HTTP protocol between Xcache and data source
  - Xcache currently can do HTTP between client and Xcache
  - With HTTP between Xcache and data source, will improve data source coverage to ~100%
2. Cache Eviction
  - Experiment data are static and never change. No need for cache eviction
  - But maybe needed as a general purpose cache. However,
  - Are we opening a can of worms here? What if the file is changing during cache access?
3. Read-only or Read-Write
  - Read-only means data upload will have to take a different path
  - Read-Write? write-back, write-through - trade off: efficiency vs data integrity (in RUCIO)
  - What about deletion?

# Packaging and deployment



## Flexibility of Xcache

- Xcache can be configured in many ways to satisfy special needs, or
- Combine with plugin library for special functions

But it complicates the deployment effort - need standardization

### 1. Xcache in CVMFS and packaged in Docker and Singularity

- Good for standard along deployment, or evaluation

### 2. **SLATE** project for automated / orchestrated deployment

- Packing Xcache in Docker - we may even have several flavor of Xcaches
- Deploy edge service via Kubernetes
- Adapt a DevOps model
- A US ATLAS goal to deploy at T1/T2s by FY18Q2 - we have two full days 🤖

# Use case models

## 1. How can we enable experiment to use Xcache

- Geographically distributed Xcache to accelerate user analysis
- Especially if user no longer need to be involved in finding data sources.
- This is essentially what CMS did.

## 2. New storage models for small sites or CPU / Opportunistic only sites

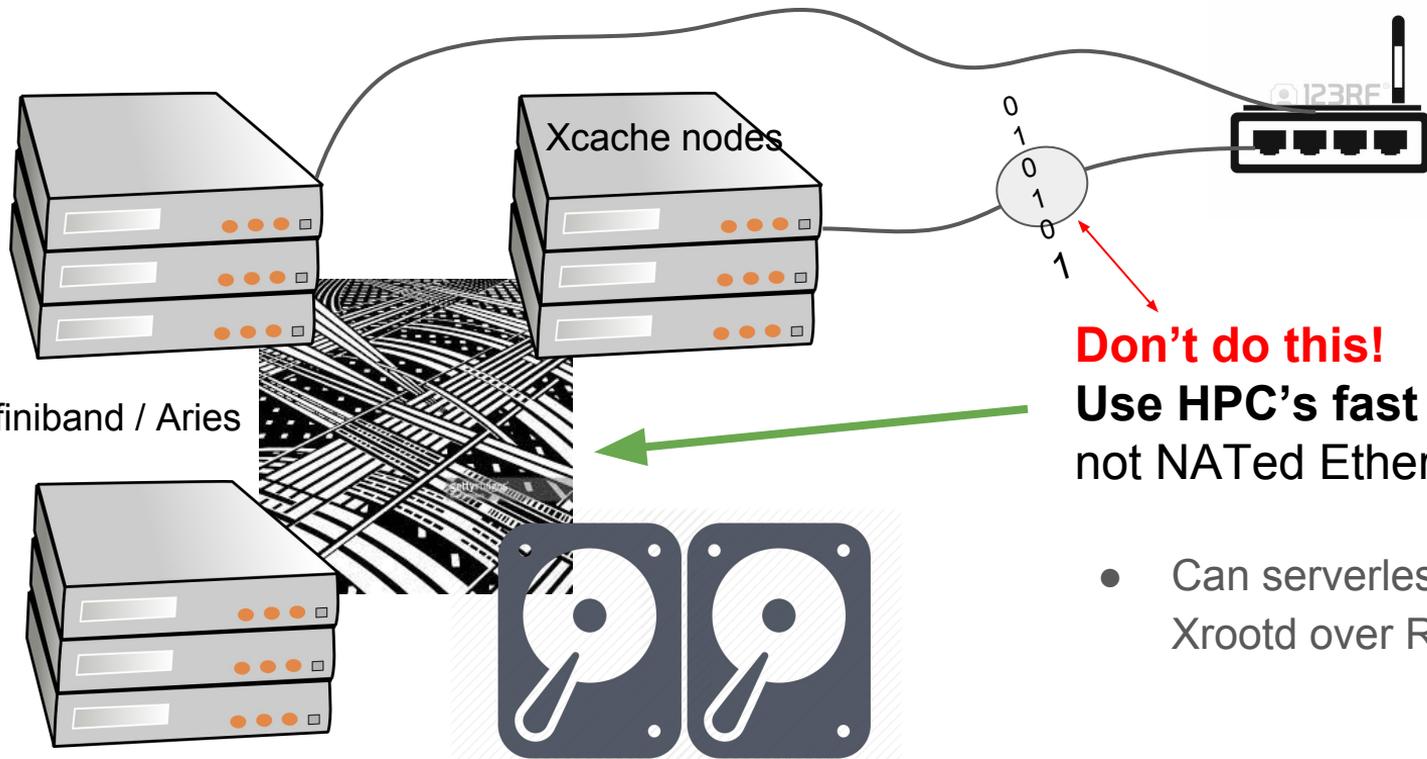
- Working on RUCIO volatile storage implementation

## 3. Cache in front of “Data Lakes”

## 4. Xcache for HPC? Interesting idea but need careful planning:

- HPCs have good infiniband-like network (e.g. Aries on Cori), but mediocre Ethernet topology
- Put cache space on Shared Filesystem?.

# Xcache for HPCs? be careful



**Don't do this!**

**Use HPC's fast interconnect  
not NATed Ethernet**

- Can serverless cache help? and Xrootd over RDMA?