ATLAS XCache Study @ Edinburgh

Teng Li
GridPP
University of Edinburgh
16/10/18

Outline

- Motivation
- Summary of previous XCache study
- Recent update
- Summary and plans

Motivation: from GridPP's perspective

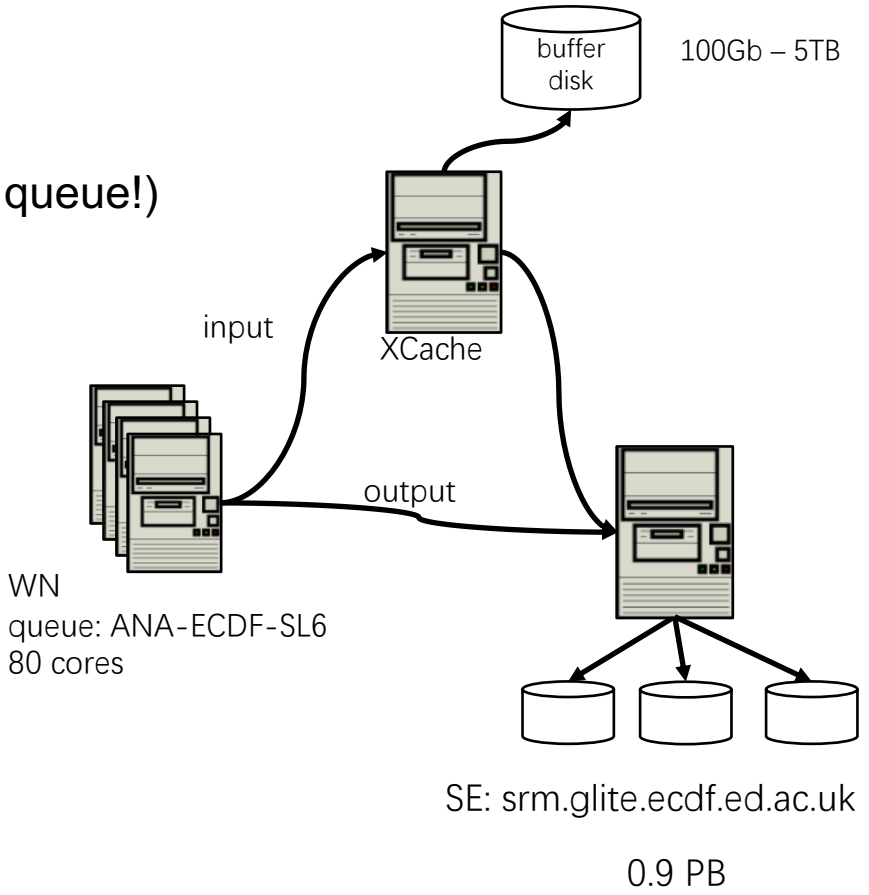
- On the edge of GridPP5 → GridPP6
  - Storage in smaller Tier2s may disappear or turn into volatile storage
    - SEs directly attached to remote CEs might be a short-term solution
    - Smart preemptive cache is already used (aCT + ARC Cache)
    - Opportunistic cache is under investigation
  - UK is investigating data lake solutions
    - Replicas across lake or stripe/ erasure code across Tier1/Tier2s
    - Data cache at different levels may be key components
      - Sitting closely to WNs to speedup data access
      - Potentially regional cache for reconstructed copies (EC)
        - Ideally working with DDM to make things "smarter" (e.g. rucio)

Motivation

- Investigation of opportunistic XCache
  - Test the feasibility of XCache
  - Try to identify dev issues
  - Measure caching performances
  - To guide future optimizations

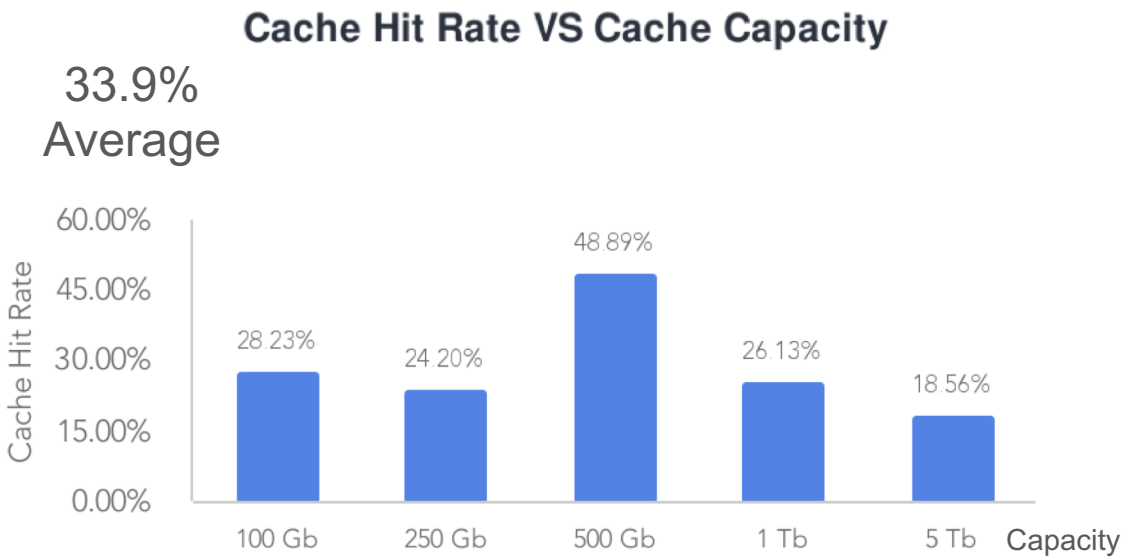
Summary of previous study

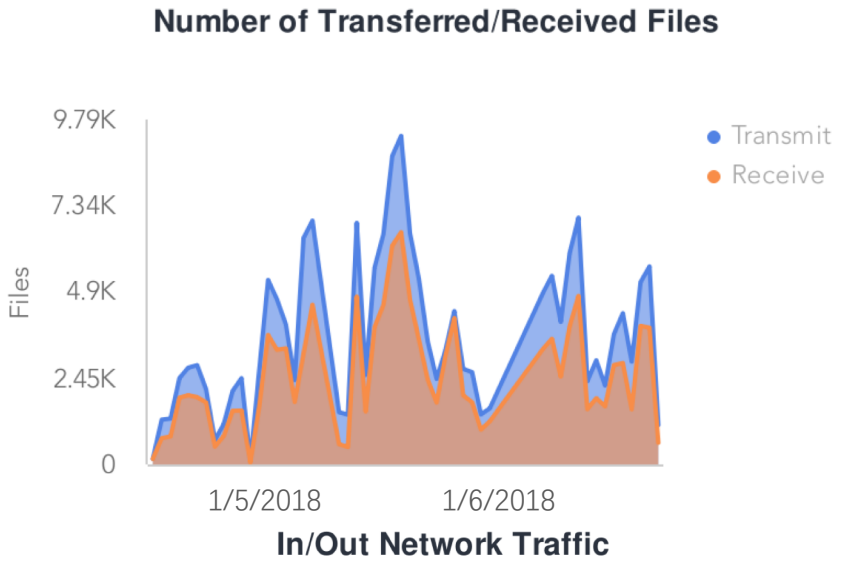
- Overview
  - Use an ATLAS analysis queue for testing
    - At very small scale (80 cores, we have a very small analysis queue!)
  - Simulating a CE attached to a remote SE (diskless site)
    - 0.9 Pb storage
  - Workflow
    - Input network traffic of WNs is redirected to XCache
    - Output network remains unchanged
    - Whole file mode is used
  - A XRootD client plugin is used to redirect the input url
    - root://srm.glite.ecdf.ed.ac.uk/file → root://xcache.url//root://srm.glite.ecdf.ed.ac.uk/file

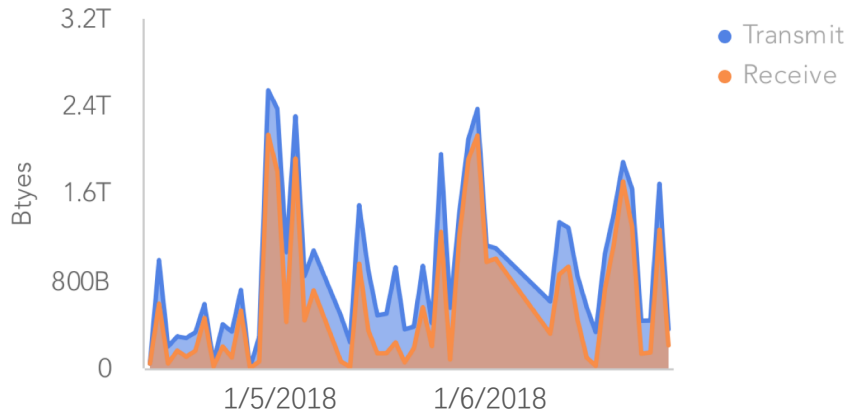


Caching performance

- 4 months of data is taken to measure the cache performance
- Average cache hit rate is 33.9%.
- Different cache capacities are tested. Peak value reached ~50%. (Only for reference, since errors are high in production environment)





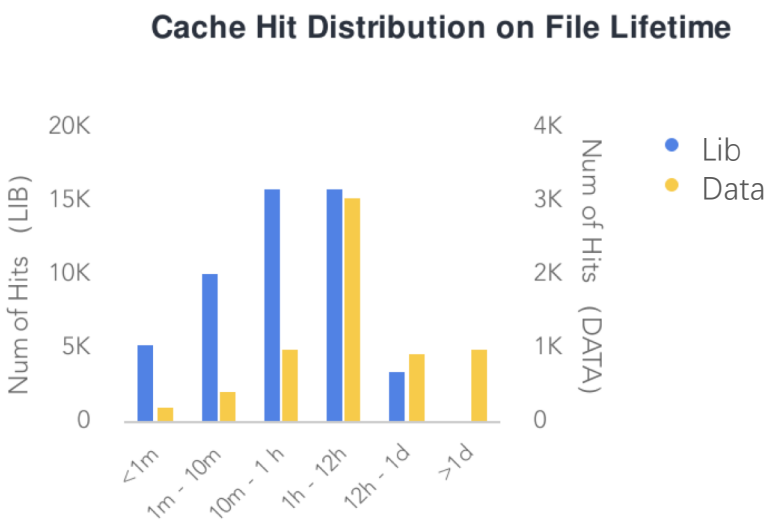


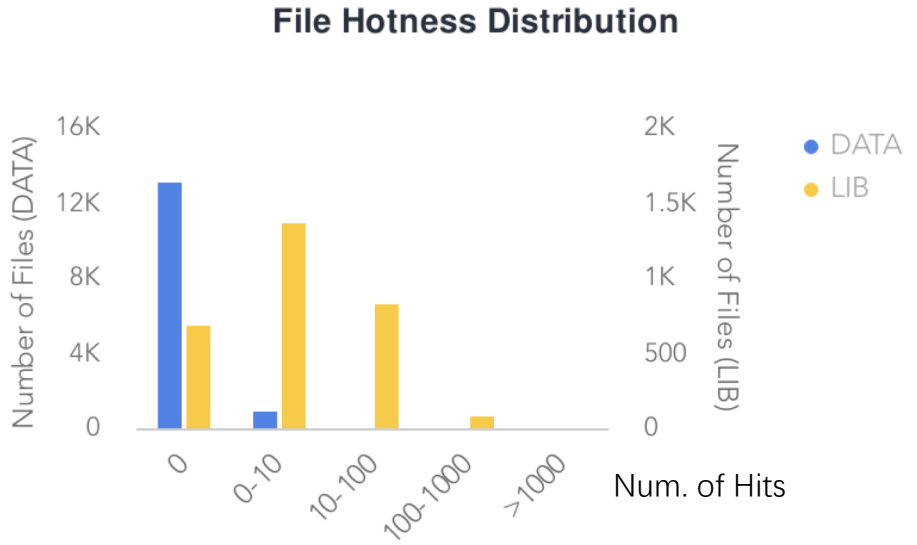
Closer look

4 kinds of files are cached:

- **input**: input data files (AOD, DAOD, ⋯)
- **output**: user output
- **library**: user library files (dispatched by panda)
- **log**: job log files

type	portion in disk	hit contribution
Input	92.1%	70.6%
library	1.3%	29.1%
log	0.05%	0.27%
output	6.5%	~0



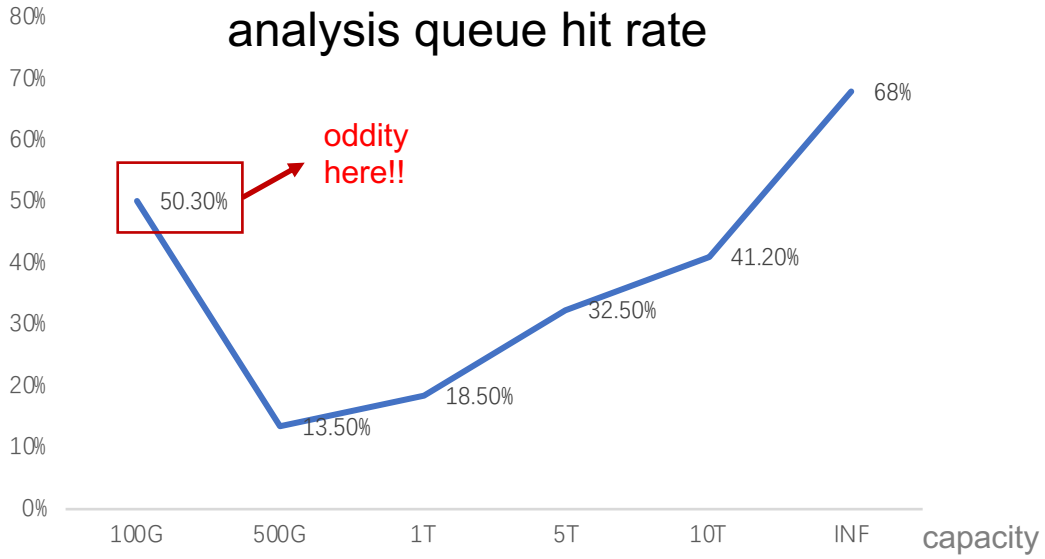


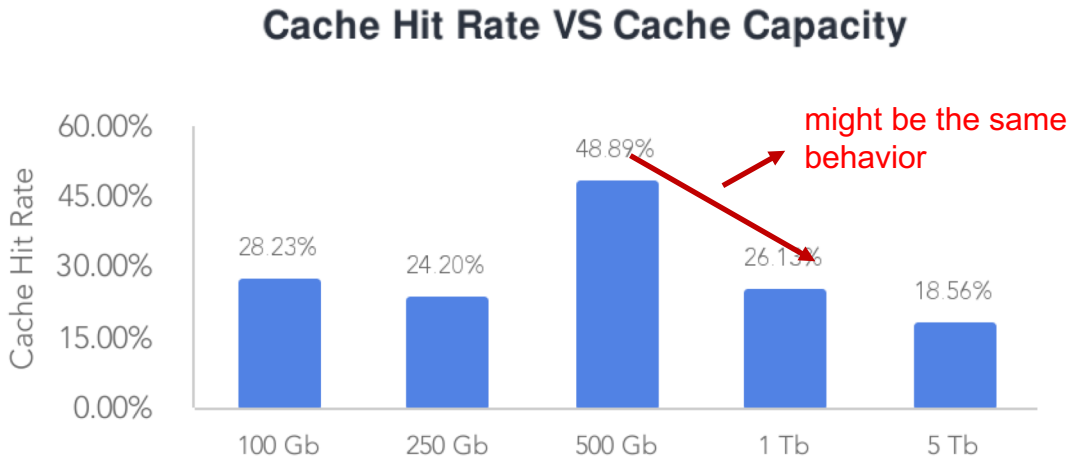
Summary

- Library files are extremely hot
- Most AOD input files are cold (minority makes most contribution)
- Files are usually hot for the first 12 hours

Recent updates

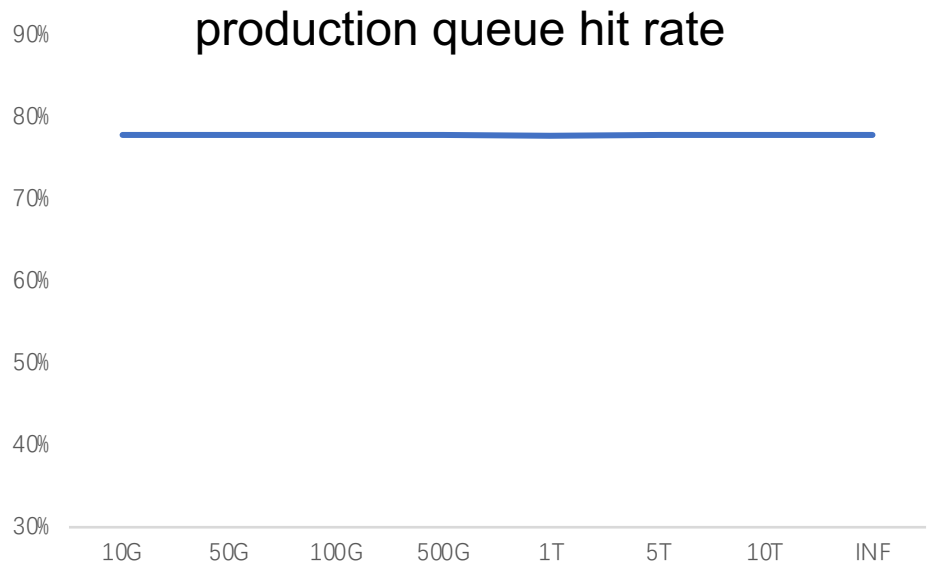
- Results of Ilija's XCache simulation code
  - ECDF analysis queue (~100 cores) and production queue (~1k cores) are tested
  - 2~6 months of data, cache disk usage: 85%-95%, cleanup policy: largest access_time





Recent updates

- Results of Ilija's XCache simulation code
  - ECDF analysis queue (~100 cores) and production queue (~1k cores) are tested
  - 2~6 months of data, cache disk usage: 85%-95%, purge alg: access_time



Brief summary

- Cache hit rate varies greatly with capacity for analysis job
- Oddity with small capacity (might agree with real data)
- Cache hit rate for production jobs is higher and doesn't change with capacity
- This needs more investigation

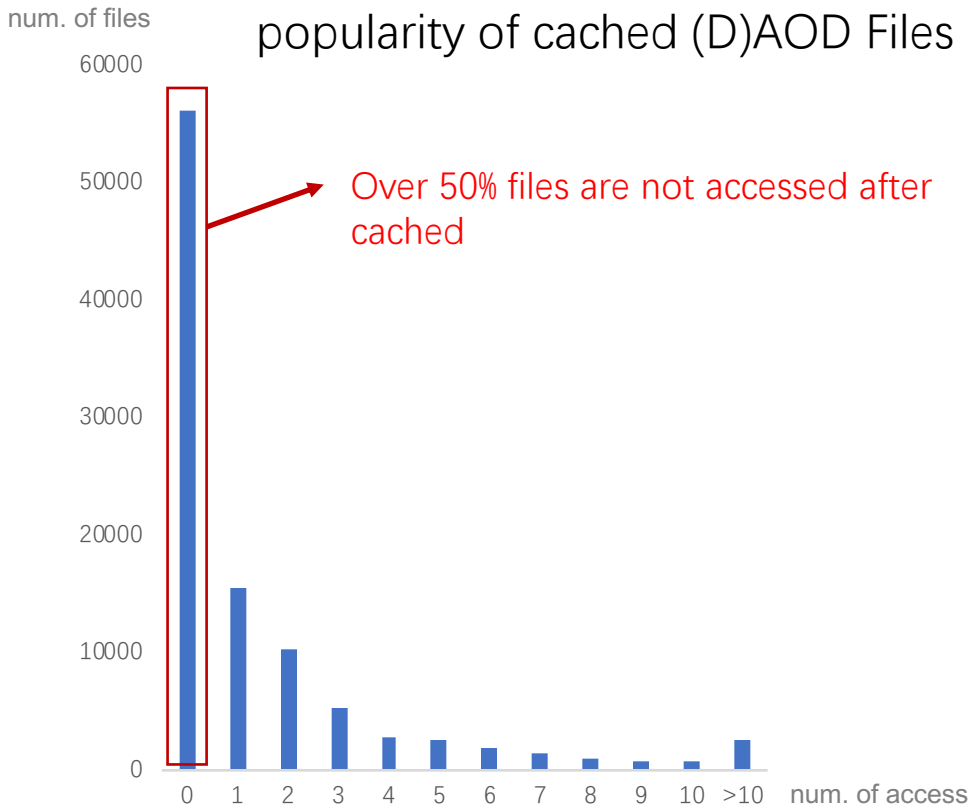
Closer look on the analysis queue

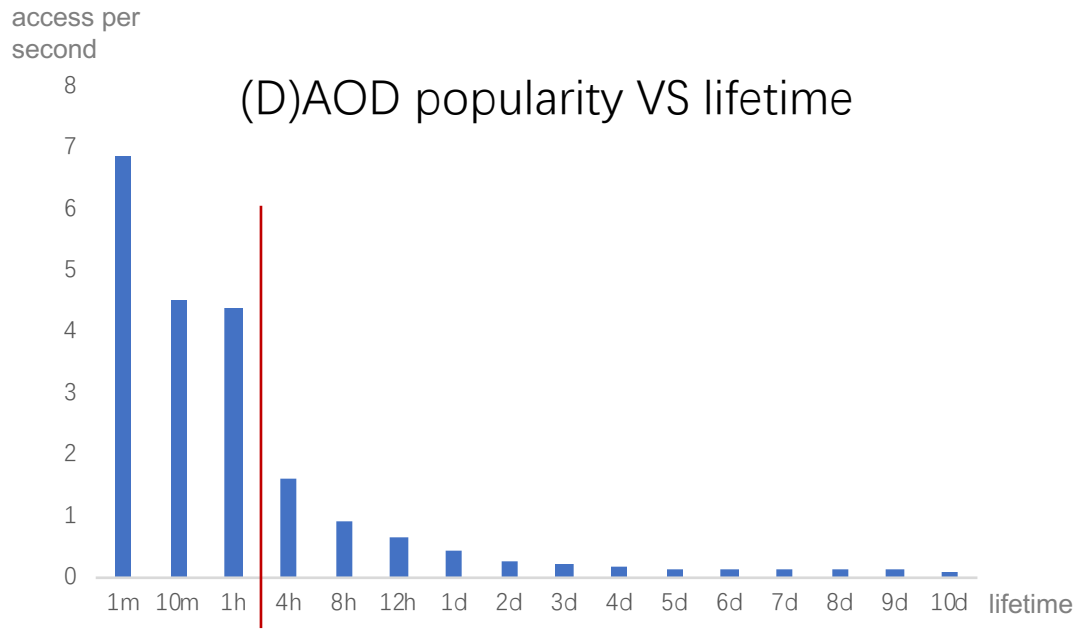
- Cached files (Gb)
  - (D)AOD contributes most of the traffic and cache hit (optimization should focus on them)

	Write into cache	Read from cache	Cache hit	Cache hit rate
AOD	110957	343629	232671	67.7%
library	173	5052	4879	96.6%
log	7.7	7.8	0.07	~0
output	1275	1371.7	96.6	7%

  - One oddity: libraries contribute more cache hit in previous study (~30%, only 2% in simulation)

Closer look on the analysis queue





Data is obviously hotter within 4 hours after cached, but remains constant after days

AOD.05536542._000001.pool.root.1 is read 19,452 times over 2 months???

Summary & Plans

- Summary
  - Caching behaviors is different for analysis/production jobs
  - Some quick investigation of access pattern of cached data shows great potential for optimization
    - Smart decision library/ cleanup policy
- Issues found during study
  - Integrate XCache into ATLAS workflow (without FAX) is tricky
  - XCache for multiple-VO usage is missing
- Plans
  - Continue unfinished study to figure out oddities
  - Look into optimization methods (desired to be VO/workflow agnostic)
  - Study partial file cache performance on ECDF analysis queue