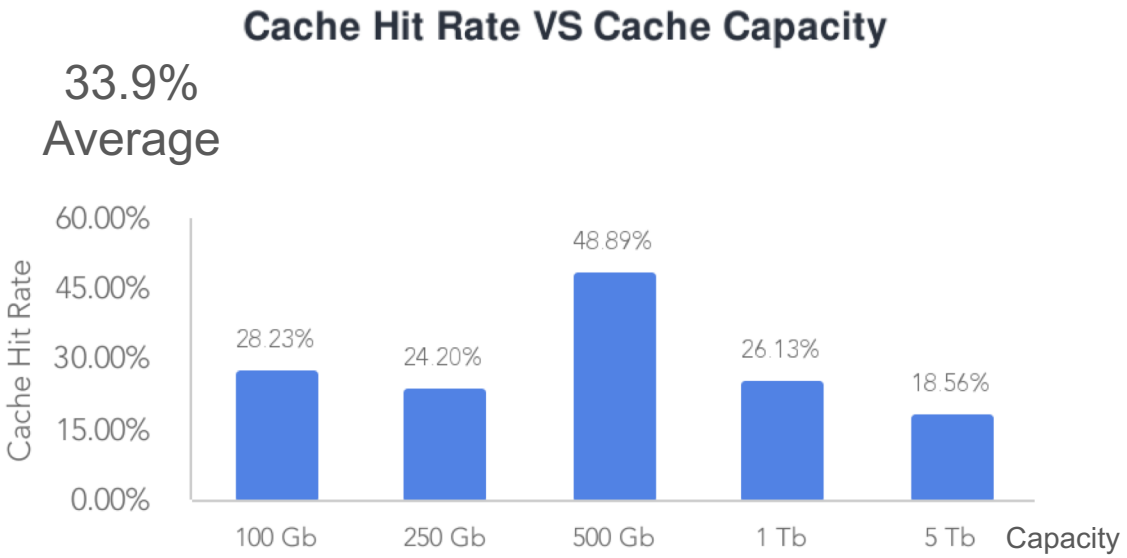
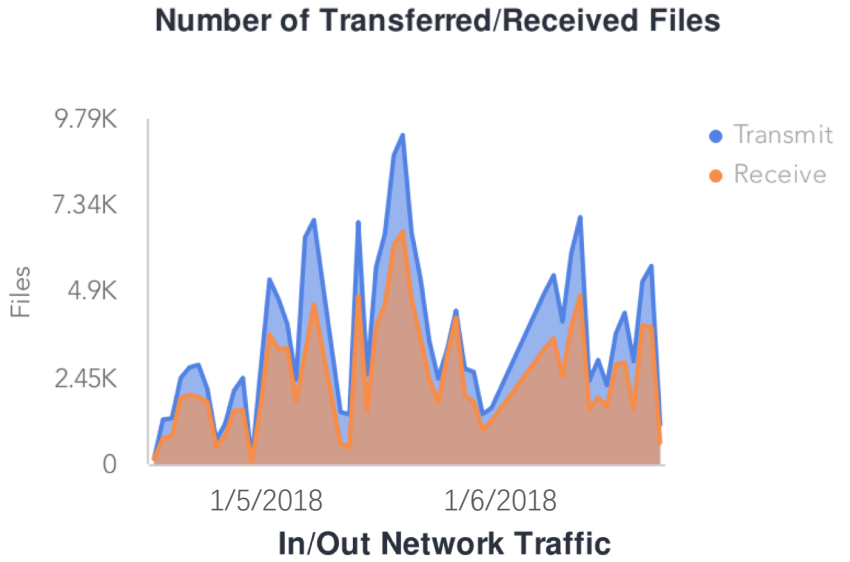
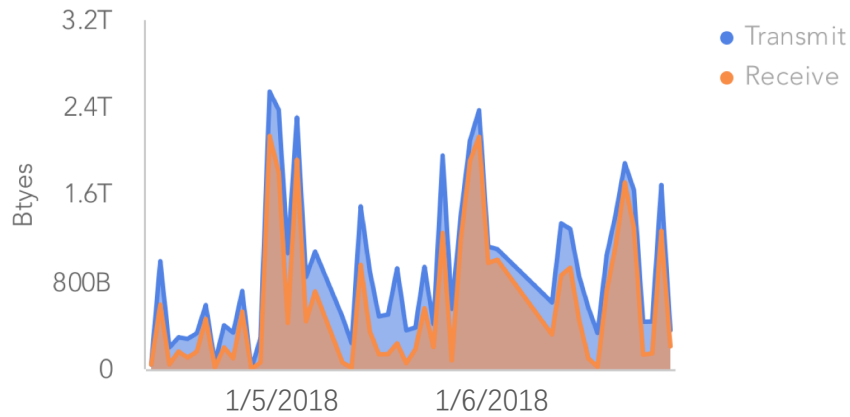
Experiences with XCache (ATLAS)

- We run XCache (for ATLAS) @ECDF for 4 months for feasibility study & performance study
- XCache sits between the analysis queue and DPM storage
- Transparent to ATLAS





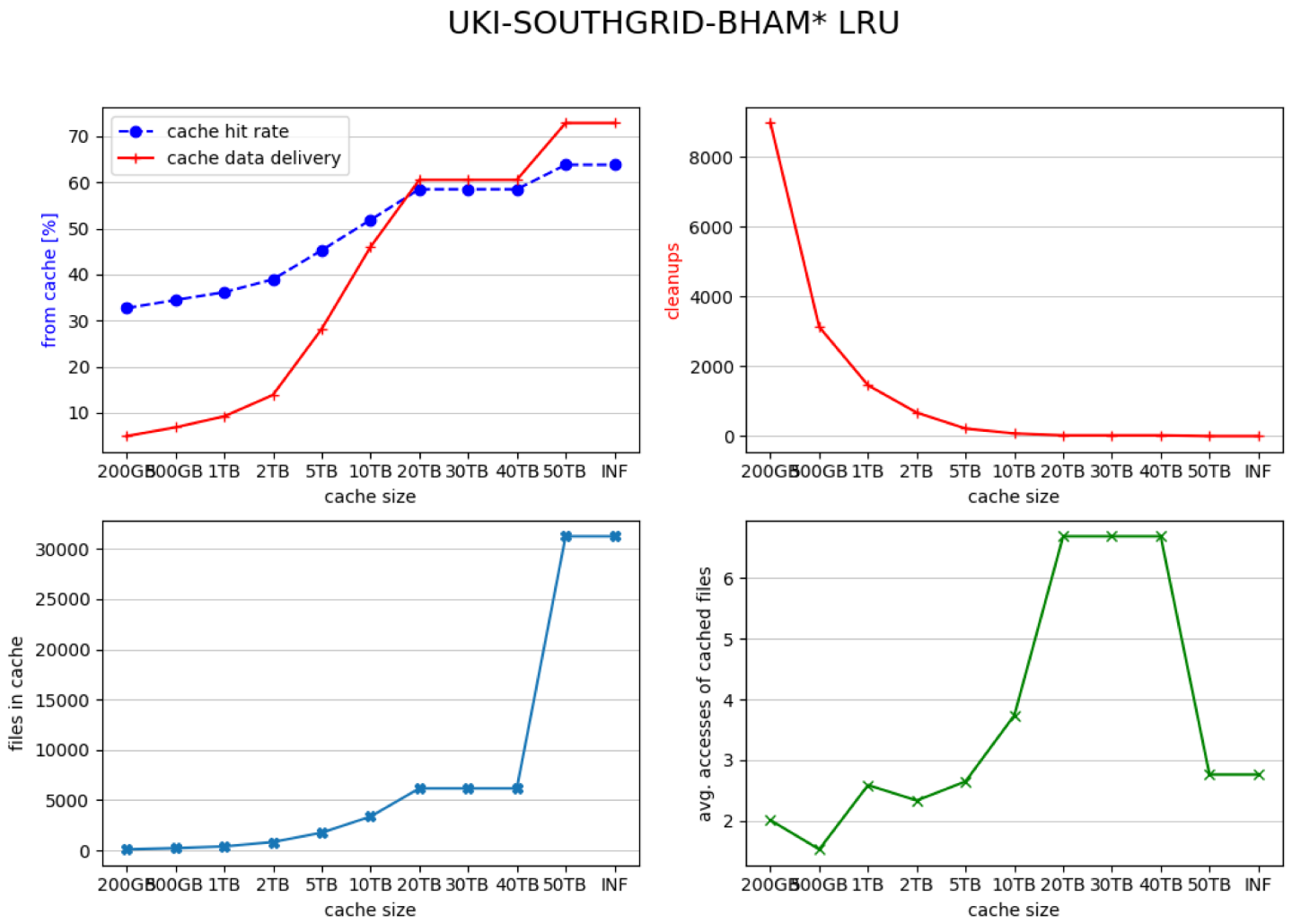


Notes

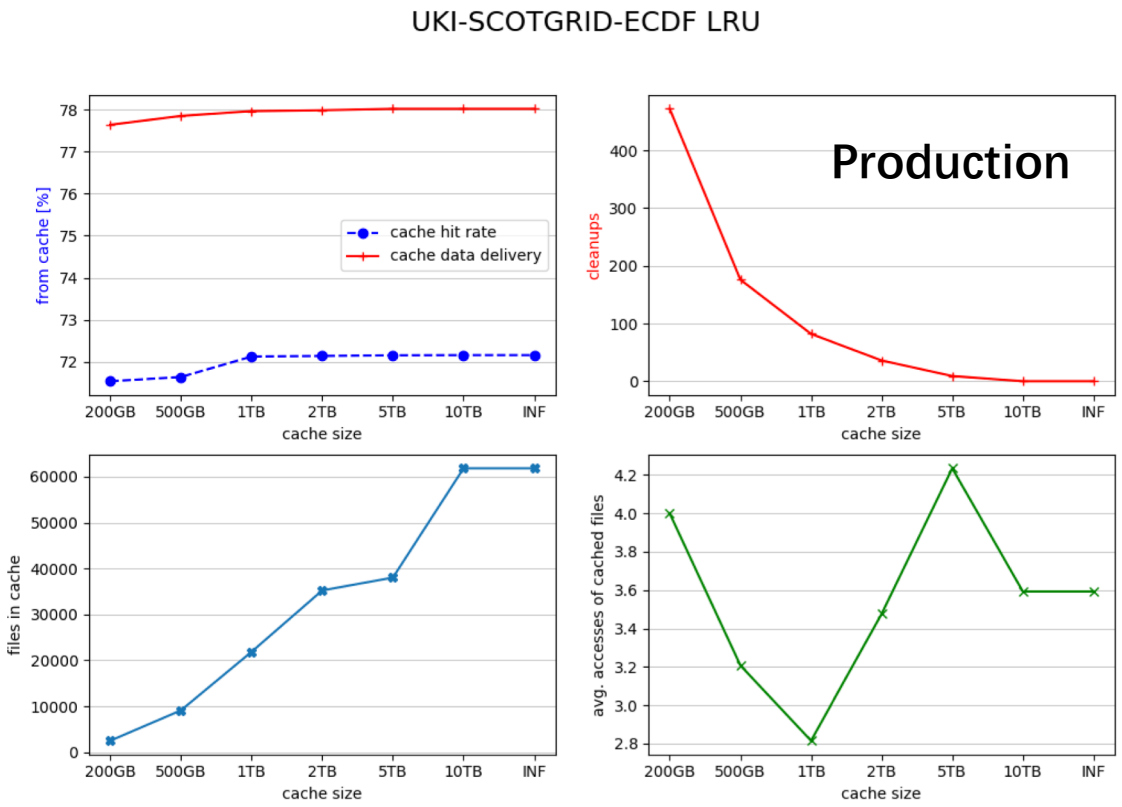
- A robot certificate is needed. Contact Ilija (Ilija.Vukotic@cern.ch) if you want a copy
- Integrating XCache to ATLAS workflow is tricky
  - We used a xrootd client plugin, https://github.com/feipengsy/XrdCLROProxyPlugin
- Deploy-able via container. Refer to
  - https://github.com/wyang007/rucioN2N-for-Xcache/wiki/Deploy-Xcache-via-a-Singularity-Container
- A config file is all needed for XCache setup, where you need to specify
  - Security: client → XCache → Storage
  - Disk cleanup watermark: 85%--95% in the simulation
  - Where to put cached data
- XCache is relatively stable. Several cronjobs would solve 90% issues:
  - Certificate renewal
  - Log backup (cleanup)
  - Fail-and-restart

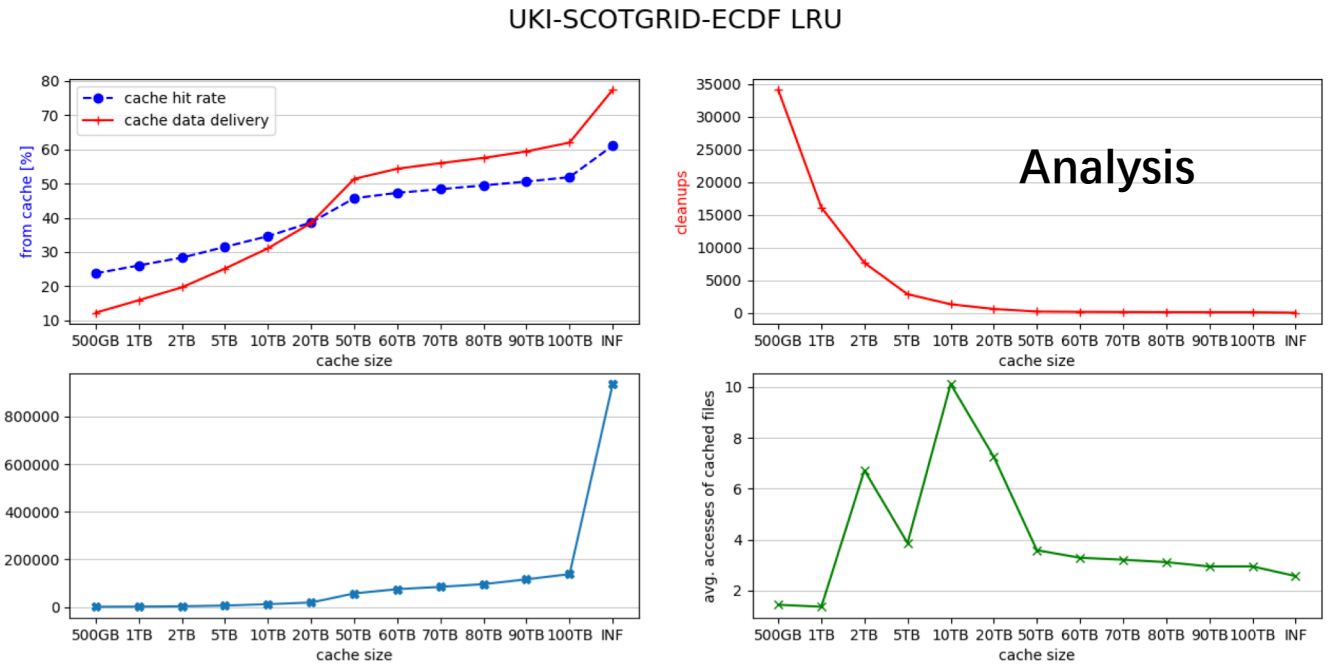
Cache simulation for BHAM

- Based on the ATLAS elasticsearch data from 4/18 – 8/18
- All file access requests from the target queue are recorded (UKI-SOUTHGRID-BHAM-HEP_MCOREVAC)
- Code to simulate the behavior of XCache
- Here's what you'll see if you run a cache @ BHAM
  - Could reach ~60% "cache hit rate" with ~20T (bytes delivered by the cache / bytes read by the queue)



Cache simulation for ECDF





Backup

Simulation @ ECDF

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

Analysis

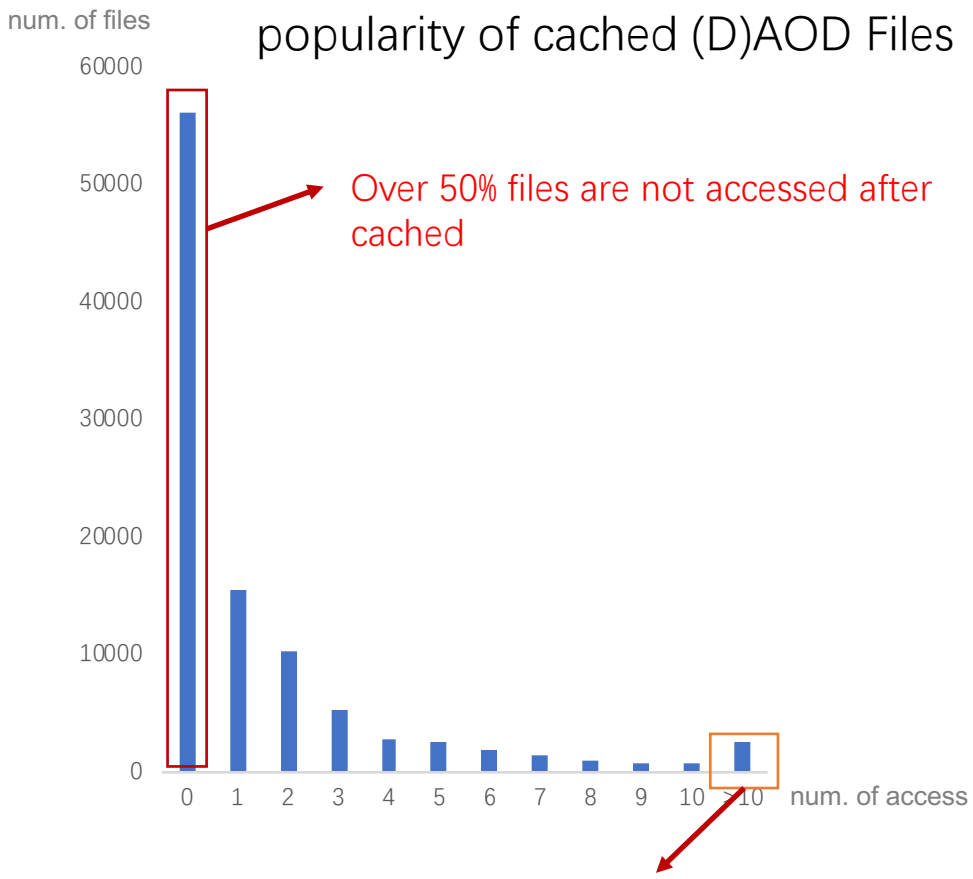
	Write into cache	Read from cache	Cache hit	Cache hit rate
AOD	110957	343629	232671	67.7%
library	173	5052	4879	96.6%
log	7.7	7.8	0.07	~0
output	1275	1371.7	96.6	7%

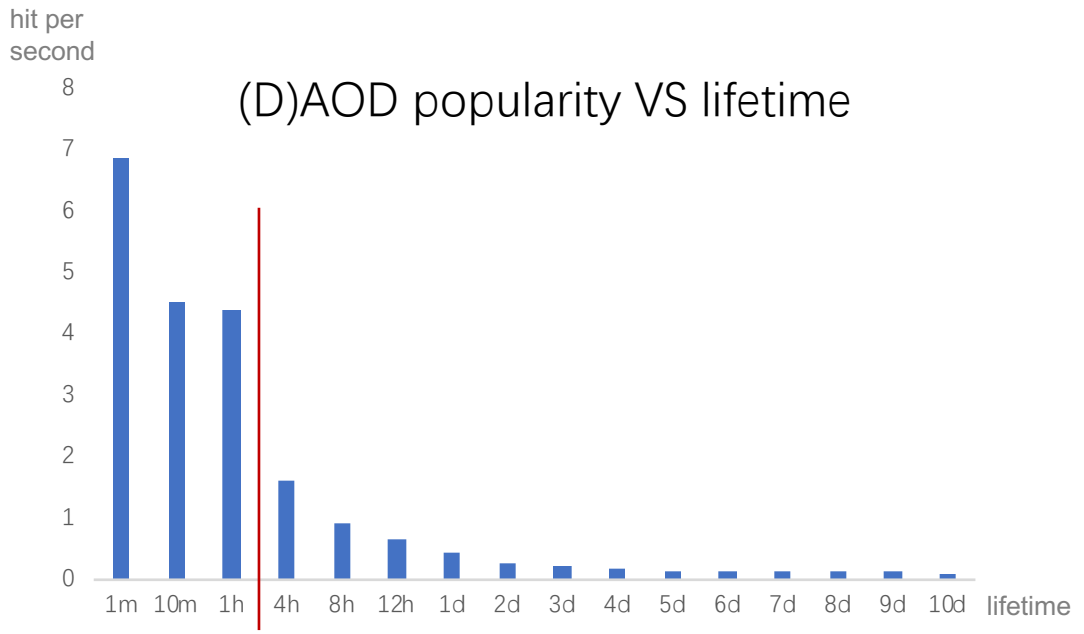
Production

	Write into cache	Read from cache	Cache hit	Cache hit rate
AOD	4205.6	11490.9	7285.2	63.4%
DRAW_*	576	576	0	0
HIT	9.34	5047.8	5038.5	99.8%
TXT*	761.6	762.2	0.6	0
GEN*	1.39	448.6	447.2	99.7%
EVNT	2908.1	20060.4	17152.2	85.5%

* Gb as unit

Simulation @ ECDF





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

A large space for optimization!!