Science and
Technology
Facilities Council

Optimising Vector Read requests for RAL disk storage

Alexander Rogovskiy
STFC, RAL

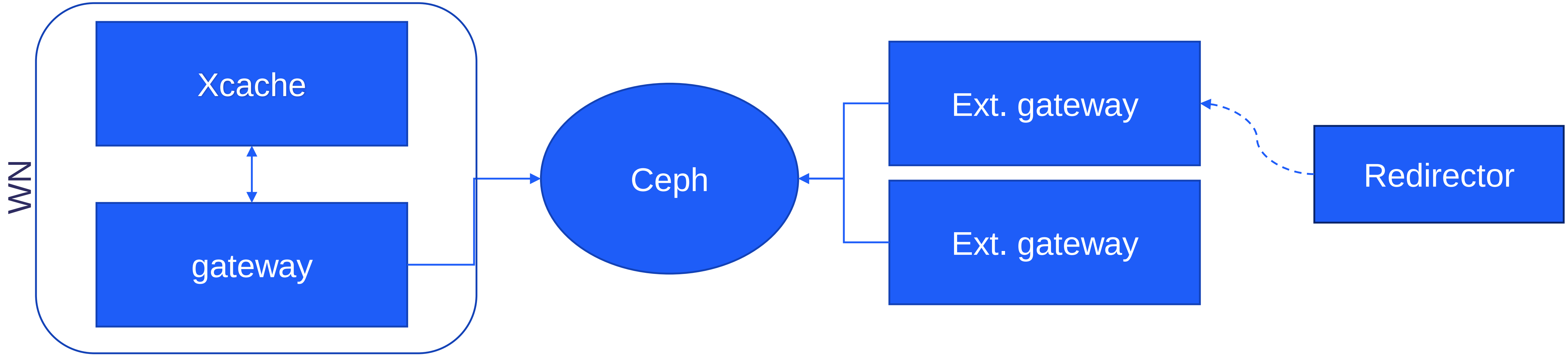
XRootD+FTS workshop @ STFC

RAL Disk storage

- RAL Disk storage (Echo) is a Ceph-based object store
  - Name comes from its key properties: Erasure coded, Ceph, High throughput and Object store
- Used by LHC VOs and other communities
- Use special nodes – gateways – to allow clients to access it
- Gateways are running XRootD ~~and GridFTP~~ servers to allow clients to access the storage via XRootD, https ~~and GridFTP~~ protocols
  - For XRootD we use XrdCeph plugin
  - Worker Nodes have their own gateways

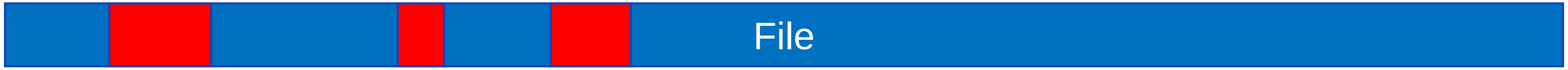
Gateway Configuration

- Ceph does not like small reads. To deal with it, we need some caching
  - So that data is read in big blocks instead of small chunks
    - WNs are using XRootD proxy for this, external gateways – buffering layer
      - Buffering layer developed by James Walder



Vector Reads

- Vector read (aka readv) is an XRootD request that reads multiple chunks of file identified by their offset and length



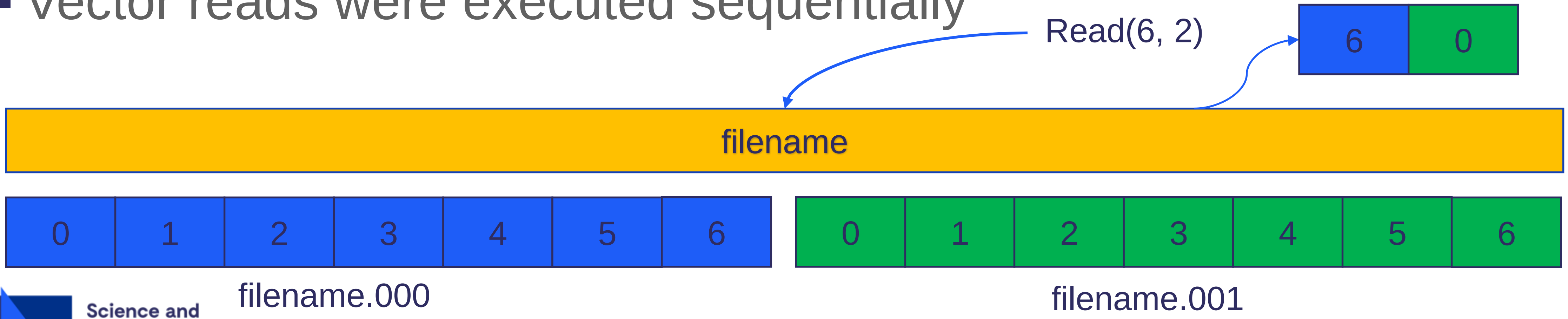
- VOs use these requests to execute “Direct Access” jobs
  - i.e jobs that do not download input data, but access it directly from the storage
- These requests were problematic for ECHO for a long time
  - Causing lots of problems for VOs, especially LHCb

Vector Reads: problem

- The error happened (most of the time) when ReadV operations took too long to execute
  - Ceph in general does not like when it is hit with lots of small reads
- There is a “stream timeout” in XRootD – if nothing is transferred in the data channel for the given amount of time, failure will be declared
- It is possible to increase this timeout via environment variable
  - It was tried multiple times, without success

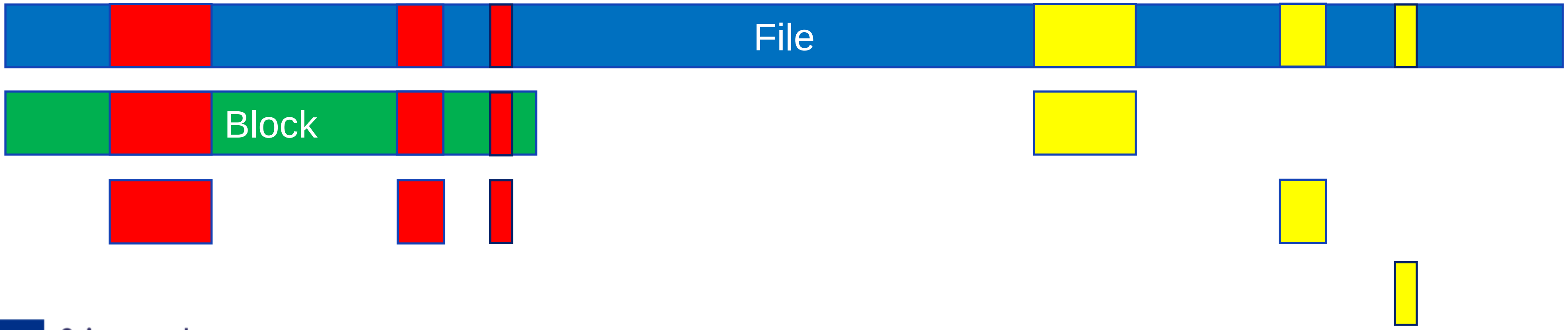
XrdCeph

- As said above, XrdCeph plugin is used to access Ceph backend
  - Each file is split into objects, each object is <= 64MiB in size
  - To do this split, Rados striper library is used
  - The library is designed to handle simultaneous reads and writes to the same file
    - This is not used in our case
- Vector reads were executed sequentially



Gateway Configuration: Xcache

- Why Xcache?
  - Memory proxy seems more suitable
    - Removes unnecessary copying to local disk
  - Unfortunately, memory proxy executes vector reads sequentially
    - I.e. requests each chunk using ordinary Read requests one by one
  - While Xcache can extract necessary chunks from blocks



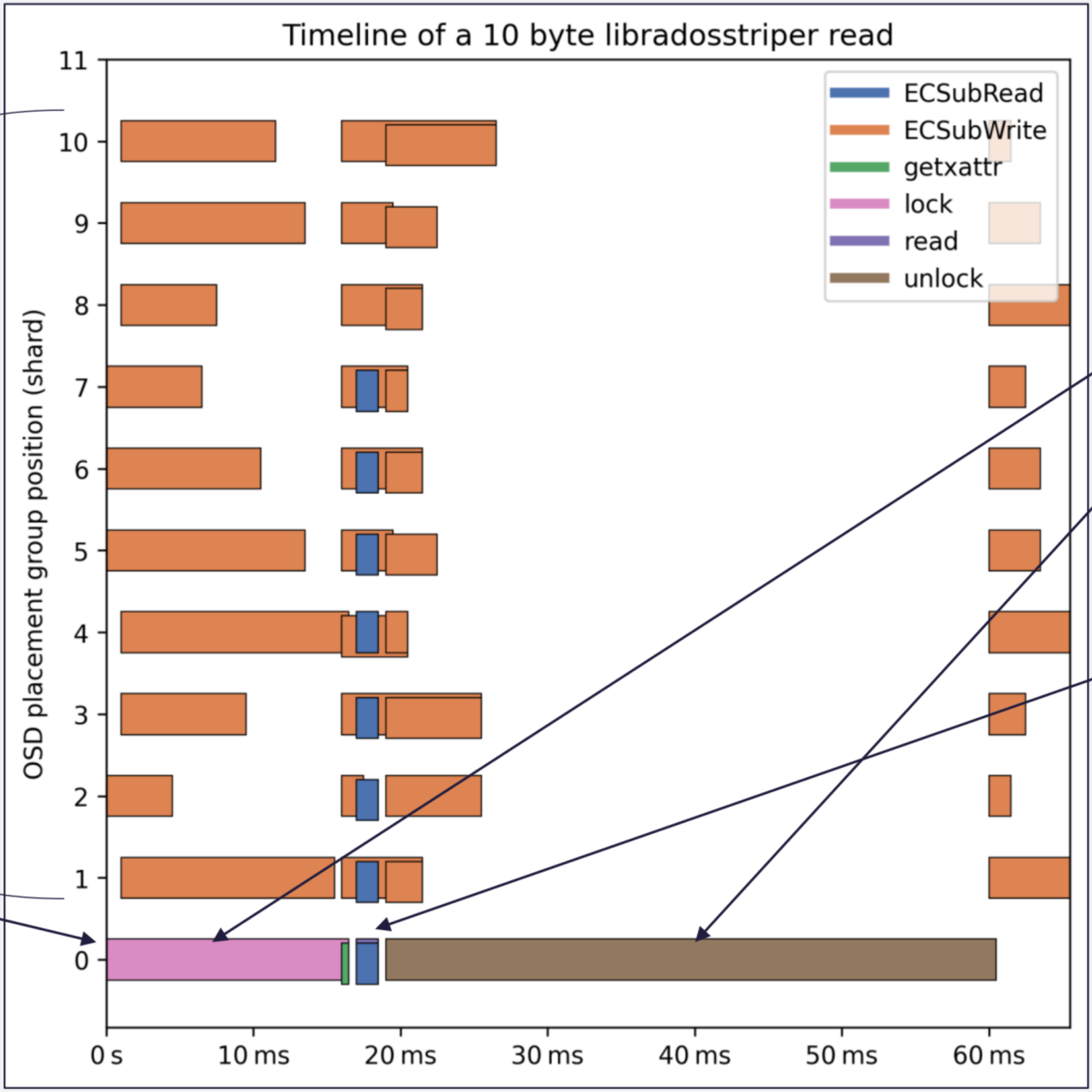
Issue: when a Read is not read

- Rados striper is designed to handle complex access scenarios
  - Which is not useful for us, since in the Grid files are considered immutable
- Because of this, every read operation involves writing of the locks
  - Thanks to Tom Byrne for the investigation! See here for details

Small libradosstriper reads

Non-primary OSDs are just dealing with reading and writing to disk

Primary OSD handles requests from the client and sends “sub requests” to the rest of the OSDs in the PG



Lock and unlock require expensive updates on all OSDs in PG

The actual read is comparatively quick

(information derived from OSD messenger debug logs)

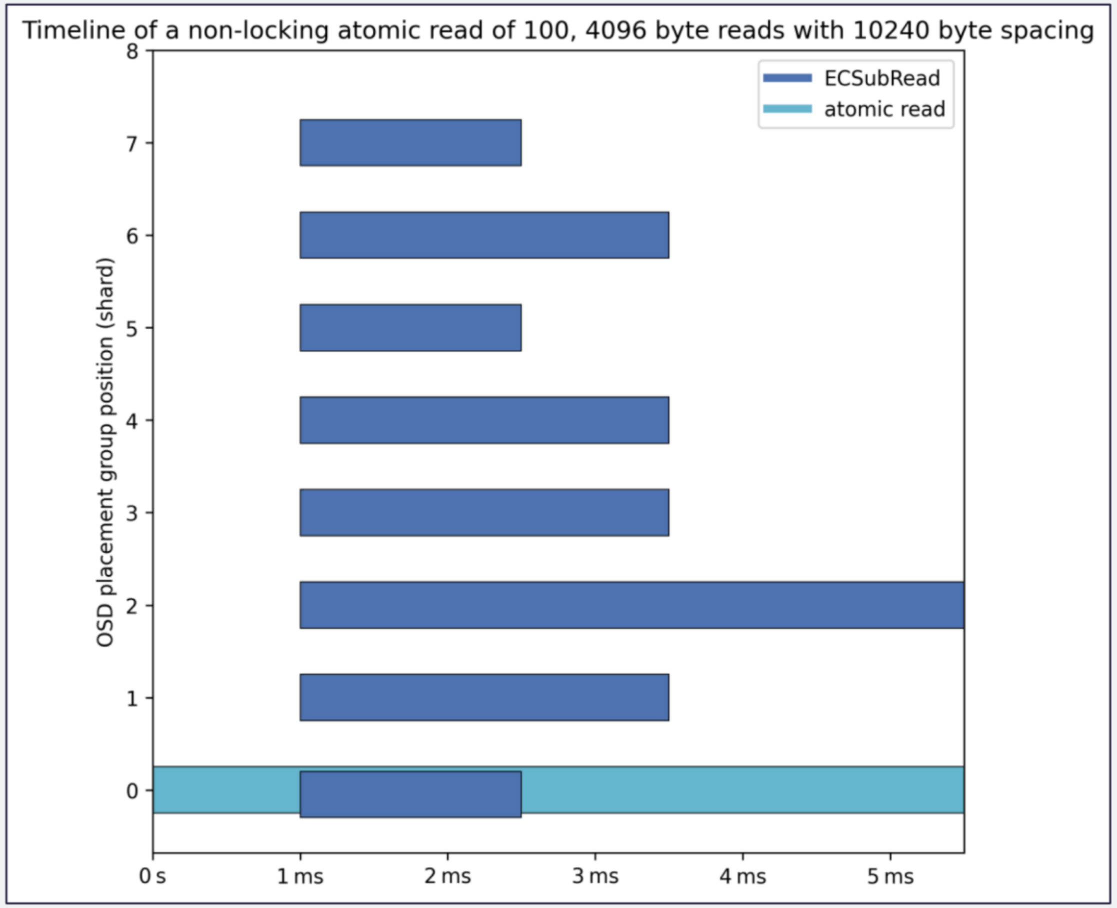
Science and Technology Facilities Council

XrdCeph improvement (1)

- So, we had two problems:
  - Vector Reads were executed sequentially by XrdCeph
  - All reads were relying on Rados striper and therefore were slow
- A new version of XrdCeph was developed
  - Does not use Rados striper for synchronous reads
  - Uses Rados atomic reads
  - Does not write any locks

Atomic librados read operations

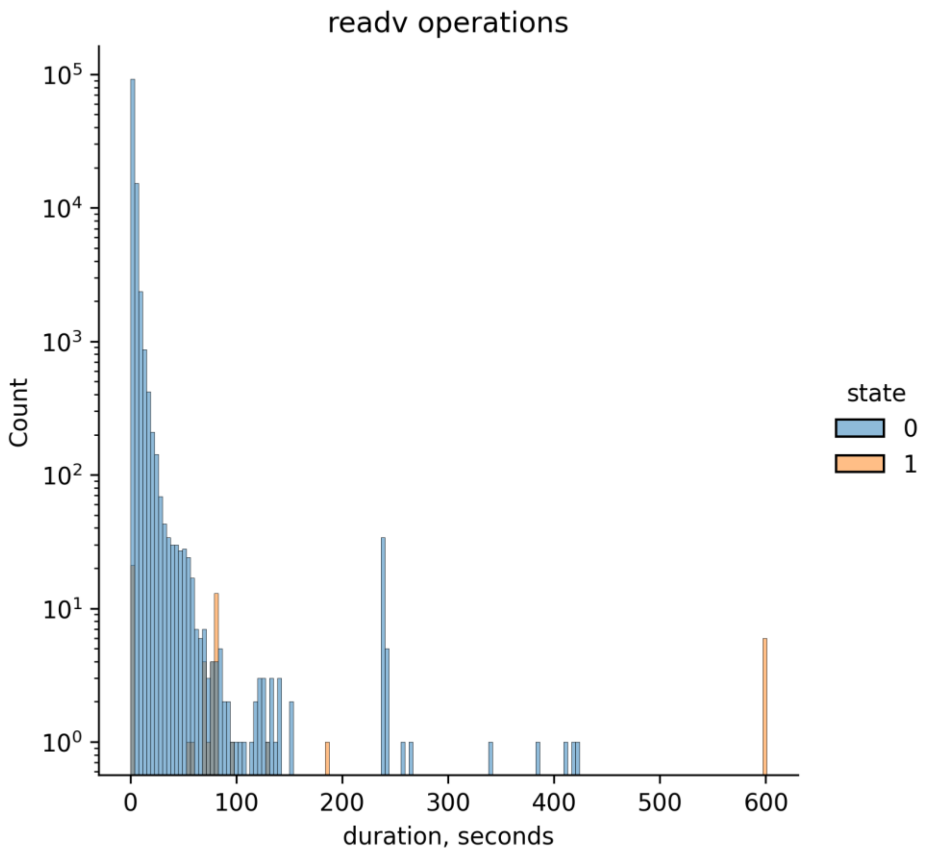
- Librados supports atomic operations – multiple operations on an object batched up by the client and then sent to the PG. e.g.
  1. *rados_create_read_op*
  2. *rados_read_op_read (x100)*
  3. *rados_read_op_operate*
- This seems to be analogous to our XRootD vector reads, and results in promising efficiency gains in testing
- One round trip within the placement group OSDs, and no excessive queuing on the primary

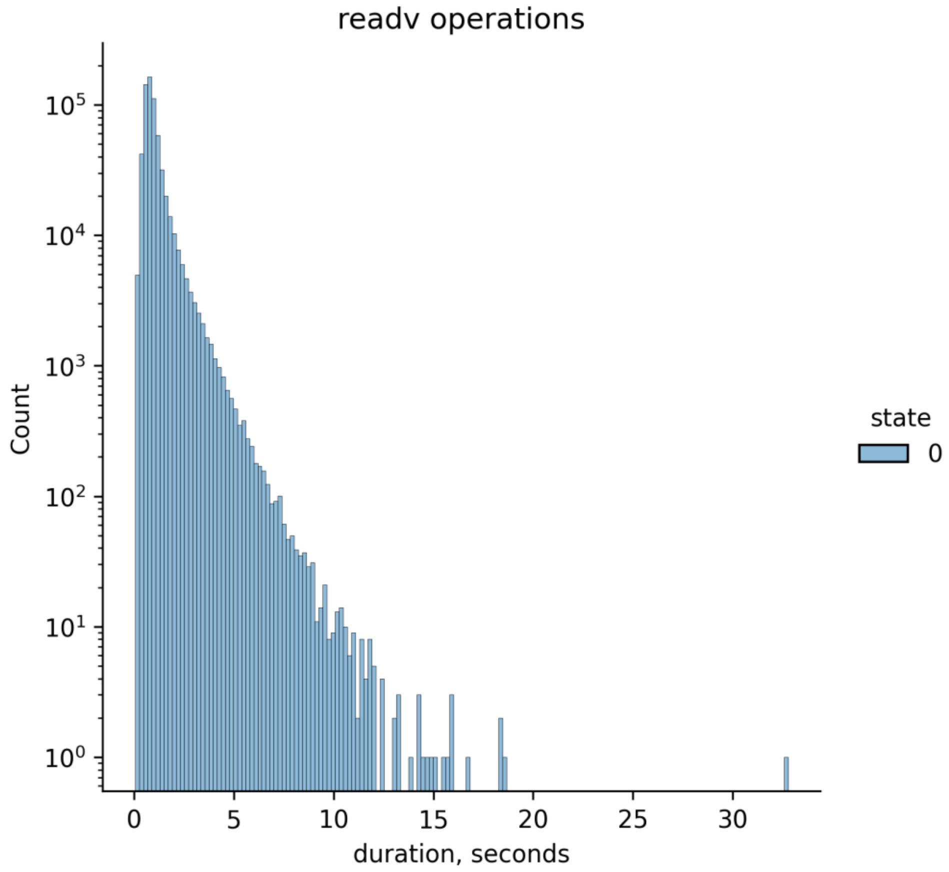


Science and Technology Facilities Council

XrdCeph improvement (2)

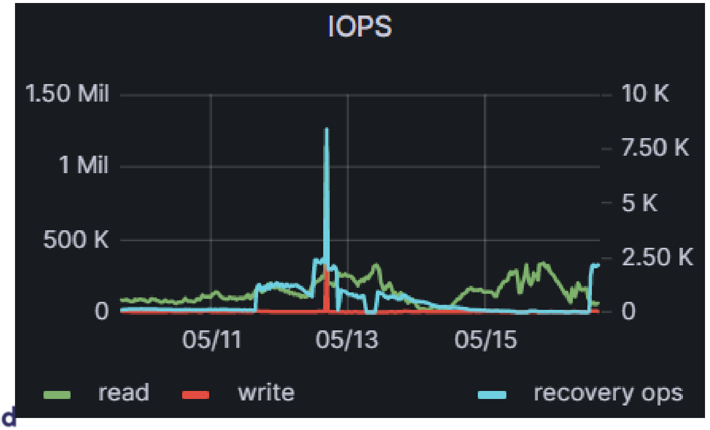
- Tests shown that the new XrdCeph version was better

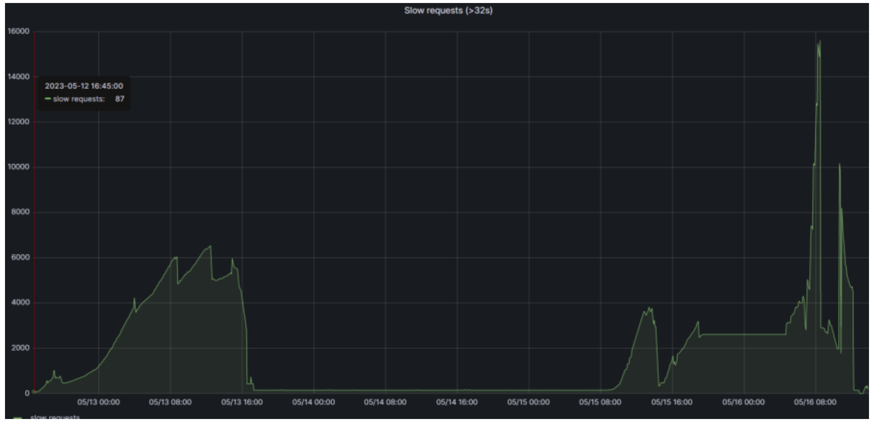

readv operations


readv operations

XrdCeph improvement: deployment (1)

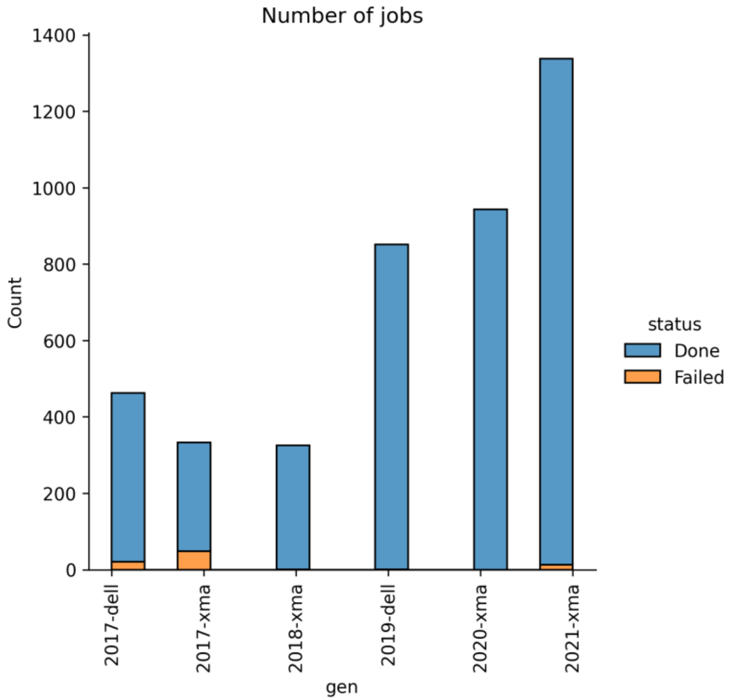
- It took quite some time to deploy the fix, find an optimal configuration and fully resolve the problem
- At first, we tried to remove proxy from the WNs at all
  - That worked fine for a couple of days, then ECHO got overloaded with IOps and crashed
    - Unfortunately, buffering layer (developed by James Walder) was not present in the deployed version
- The proxy then was added again
  - Annoyingly enough, Vector Read errors reappeared as well

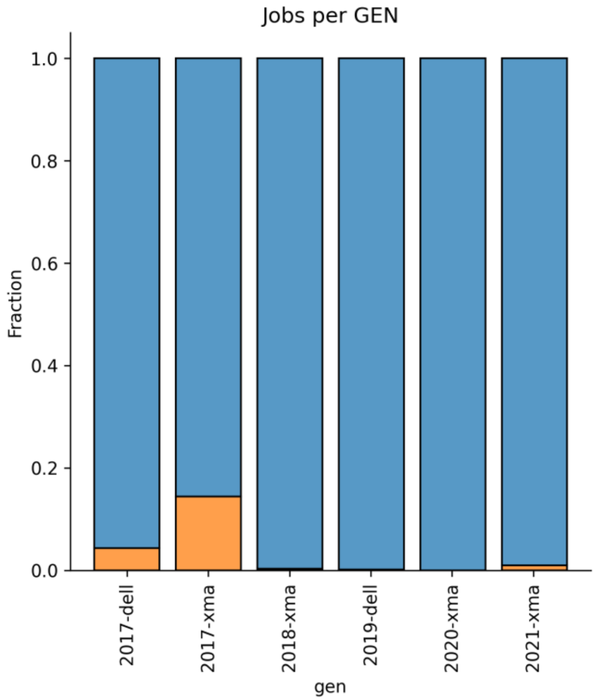


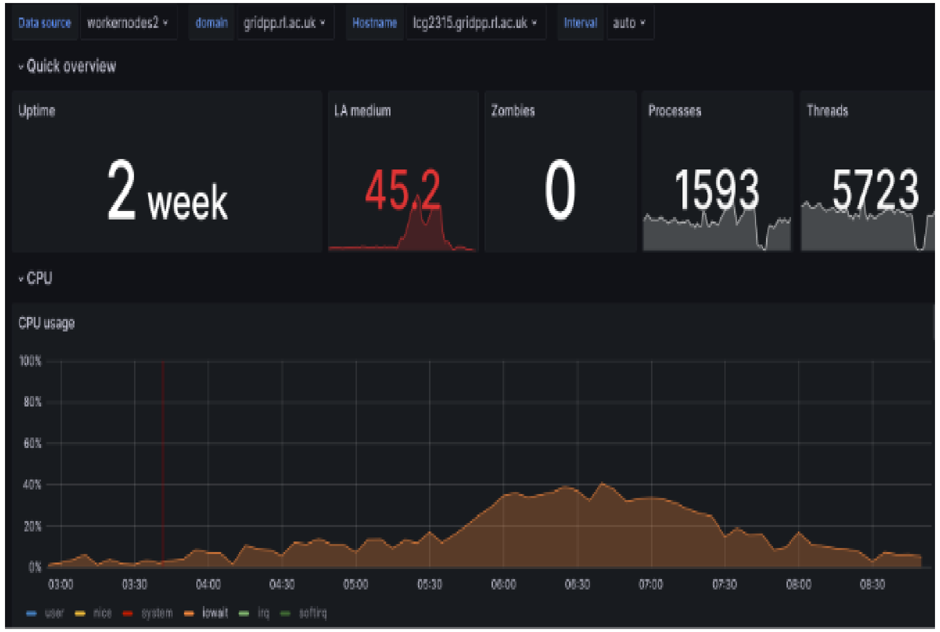


XrdCeph improvement: deployment (2)

- It was then found out that most of the errors were coming from two particular generations of WNs
  - These generations were the only ones with HDDs, others were using SSDs
  - LHCb (the most affected VO) was them removed from this Gen

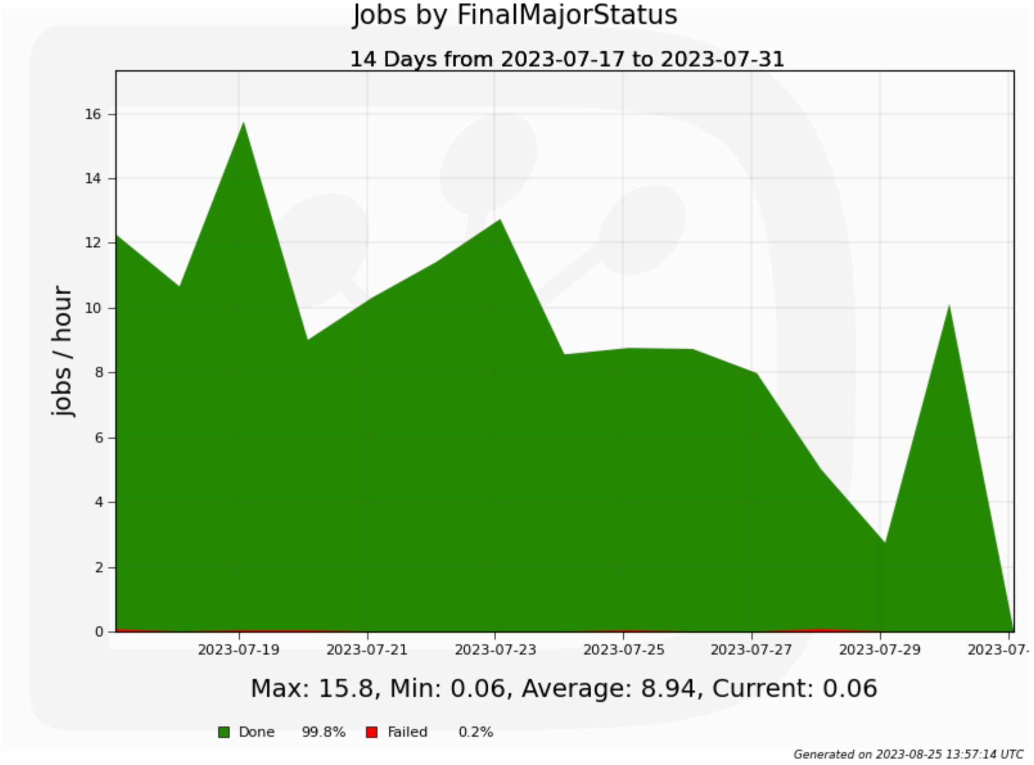


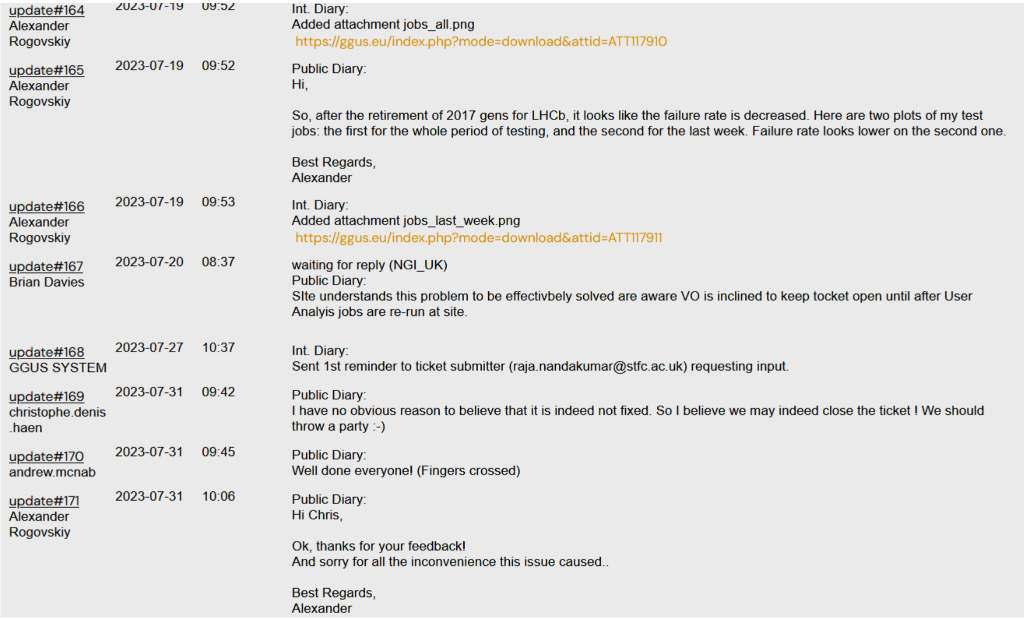



Data source workernodes2 · domain gridpp.rl.ac.uk · Hostname lcg2315.gridpp.rl.ac.uk · Interval auto

XrdCeph improvement: deployment (3)

- That brought failure rate to tolerable levels, and the infamous Vector Read Ticket was closed (after 4 years!)
- That was not the end of the story, though…





update#164 Alexander Rogovskiy	2023-07-19	09:52	Int. Diary: Added attachment jobs_all.png https://ggus.eu/index.php?mode=download&attid=ATT117910
update#165 Alexander Rogovskiy	2023-07-19	09:52	Public Diary: Hi, So, after the retirement of 2017 gens for LHCb, it looks like the failure rate is decreased. Here are two plots of my test jobs: the first for the whole period of testing, and the second for the last week. Failure rate looks lower on the second one. Best Regards, Alexander
update#166 Alexander Rogovskiy	2023-07-19	09:53	Int. Diary: Added attachment jobs_last_week.png https://ggus.eu/index.php?mode=download&attid=ATT117911
update#167 Brian Davies	2023-07-20	08:37	waiting for reply (NGI_UK) Public Diary: SIte understands this problem to be effectivbely solved are aware VO is inclined to keep tocket open until after User Analyis jobs are re-run at site.
update#168 GGUS SYSTEM	2023-07-27	10:37	Int. Diary: Sent 1st reminder to ticket submitter (raja.nandakumar@stfc.ac.uk) requesting input.
update#169 christophe.denis.haen	2023-07-31	09:42	Public Diary: I have no obvious reason to believe that it is indeed not fixed. So I believe we may indeed close the ticket ! We should throw a party :-)
update#170 andrew.mcnab	2023-07-31	09:45	Public Diary: Well done everyone! (Fingers crossed)
update#171 Alexander Rogovskiy	2023-07-31	10:06	Public Diary: Hi Chris, Ok, thanks for your feedback! And sorry for all the inconvenience this issue caused.. Best Regards, Alexander

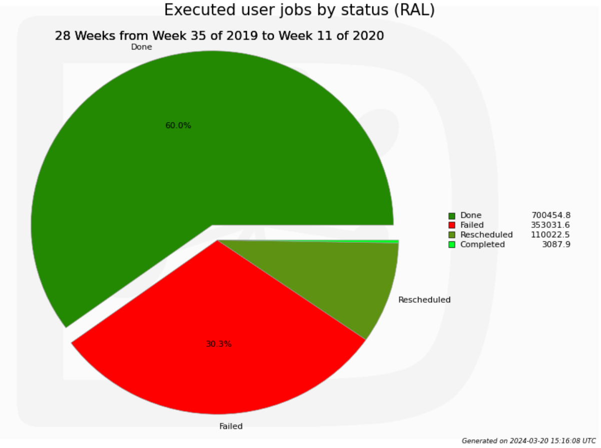
Subsequent improvements

- It was found almost immediately after the closure of the Ticket that errors still happen occasionally
    - Namely, when lots of Direct Access jobs are submitted
- Other issues were found, namely
    - Limit inconsistency between the proxy and the gateway
    - Lack of memory for the proxy
        - Memory limits were not proportional to the number of cores
        - Though proxies lack of memory should not be a big problem
    - Lack of memory for the gateway
        - For gateways this is a big problem
    - Bug in our restart scripts
- The issues were addressed

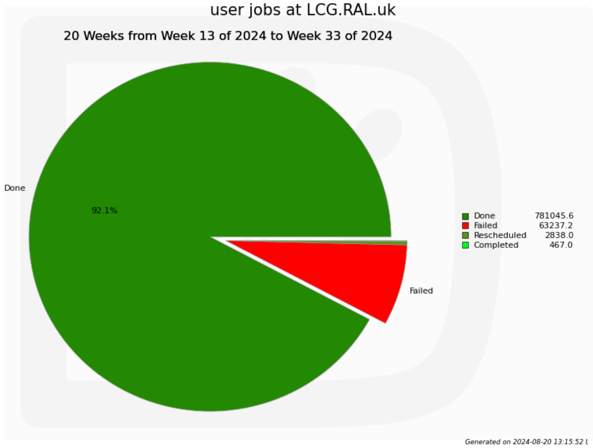
Summary

- The issue turned out to be complicated, and it took quite some time to solve it
  - Thanks to everyone involved: Alastair Dewhurst, James Walder, Jyothish Thomas, Raja Nandakumar, Robert Currie, Steven Simpson, Tom Birkett, Tom Byrne et al.

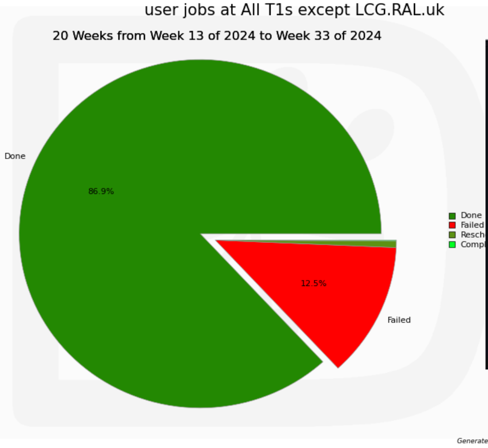
RAL 2019



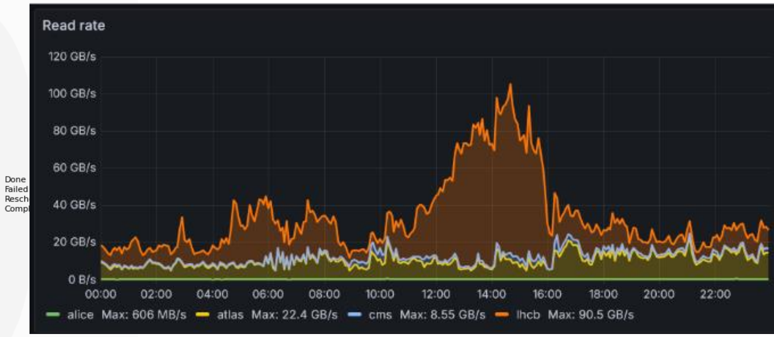
RAL 2024



Other T1 2024



RAL Aug24



UKRI Science and Technology Facilities Council

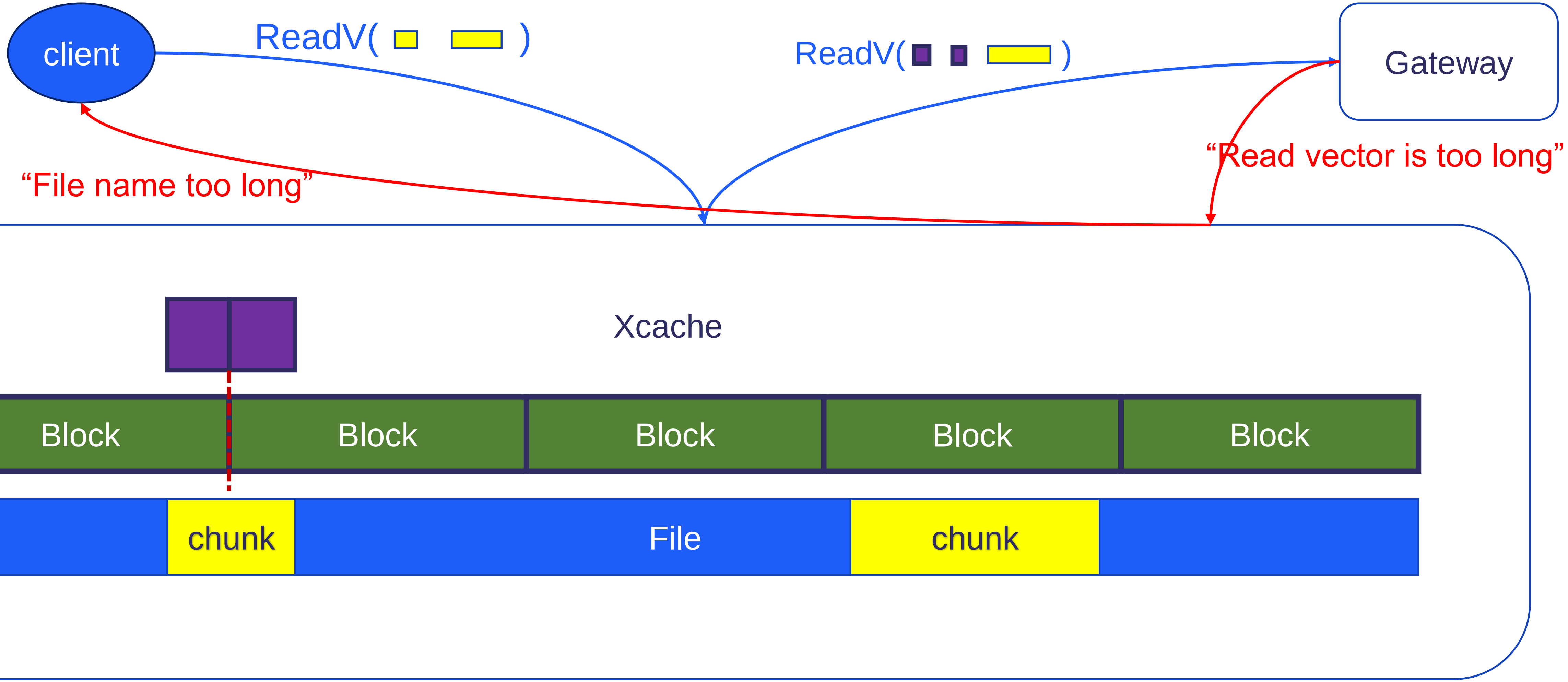
Thank you!

Any questions?

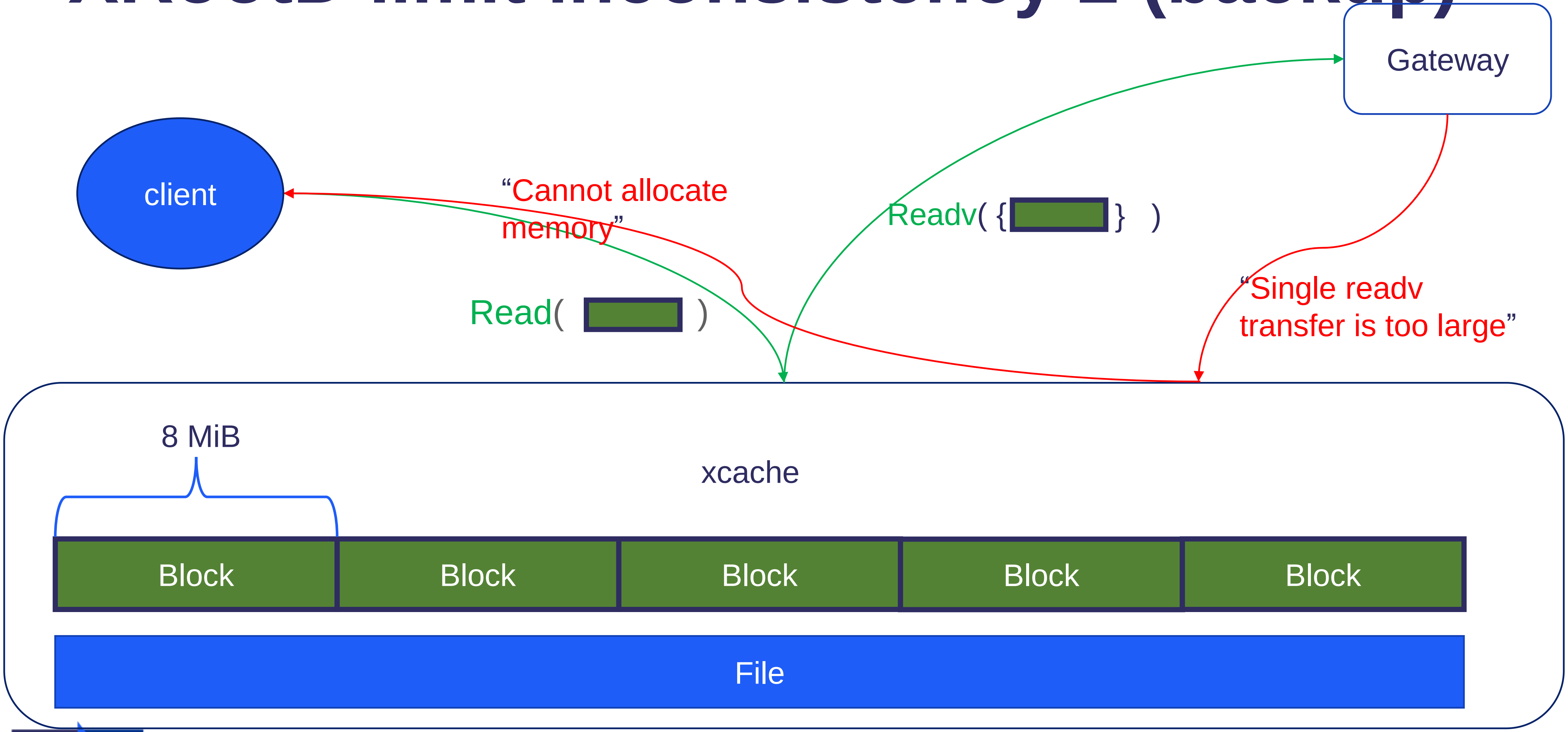
XRootD limit inconsistency

- We found that under certain circumstances Xcache can send an incorrect ReadV request to the gateway
    - Namely, request exceeding server’s limit either on number of chunks in read vector or size of individual chunks
- Unfortunately, the error messages reported to the client were very misleading
    - The first limit excess was reported as “File name too long”, while the second one as “Cannot allocate memory”
    - Can we improve it? Looks like error message mangling is happening here
- Bugs were fixed in version 5.5.x (or earlier) and 5.7.1 respectively

XRootD limit inconsistency 1 (backup)



XRootD limit inconsistency 2 (backup)


Gateway
client
"Cannot allocate memory"
Readv({ })
Read( )
"Single readv transfer is too large"
8 MiB
xcache
Block
Block
Block
Block
Block
File