Load balancing XRootD

Jyothish Thomas, James Walder, Thomas Byrne

jyothish.thomas@stfc.ac.uk,
james.walder@stfc.ac.uk, tom.byrne@stfc.ac.uk

Science and
Technology
Facilities Council

The RAL Tier-1 Setup

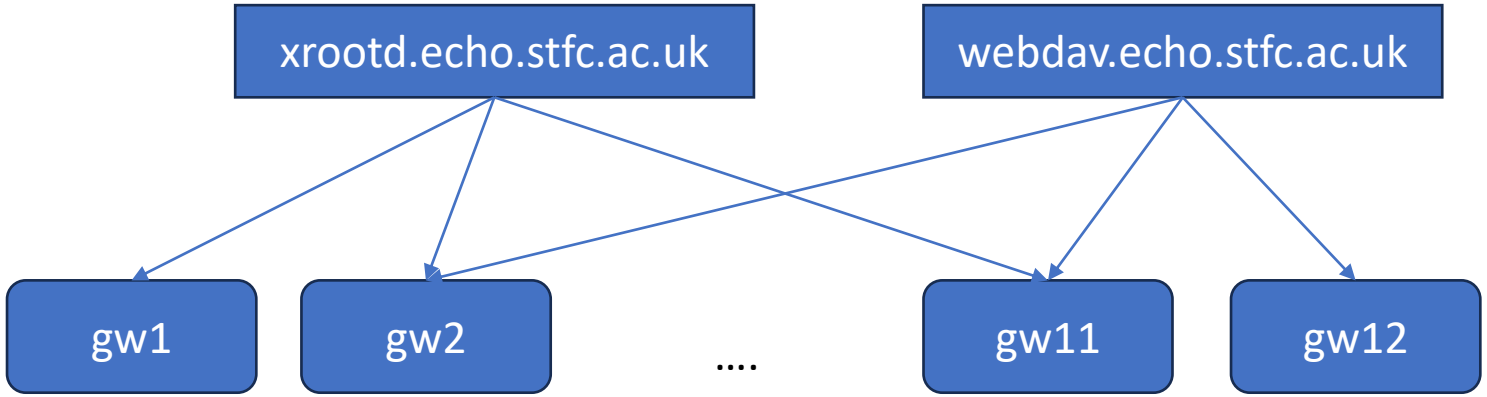
The RAL tier 1 setup - Q1 2023

- RAL has a large Ceph object store disk pool (Echo)
- Which is accessed through a set of XRootD server gateways
- The gateways access the Ceph storage trough the XrdCeph plugin

Science and Technology Facilities Council

The RAL tier 1 setup - Q1 2023

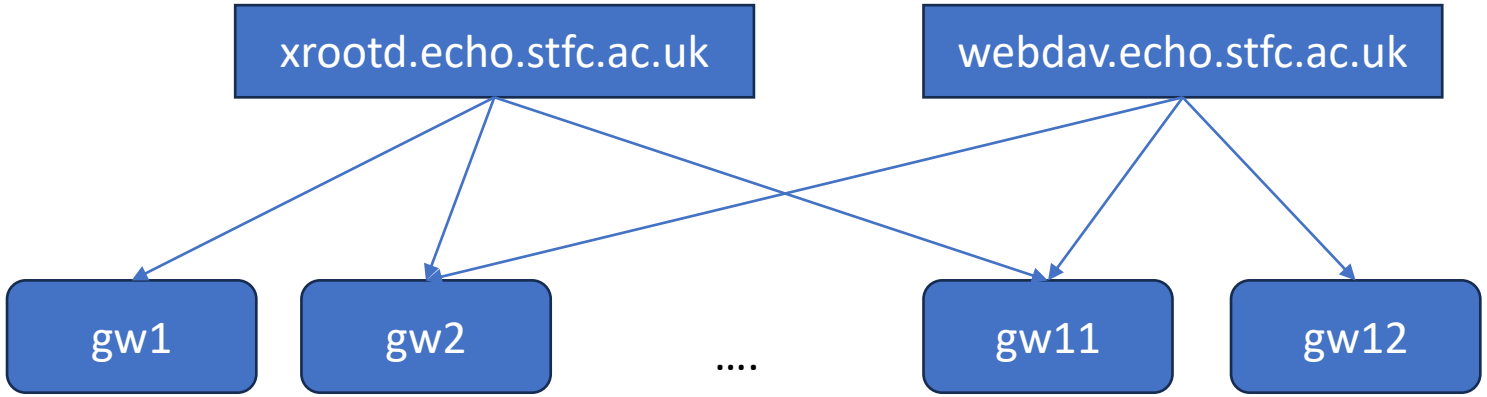
- We had 12 external gateways that did write operations from the Worker Nodes, as well as all external traffic.
- These were managed under a DNS round robin under the alias xrootd.echo.stfc.ac.uk and webdav.echo.stfc.ac.uk



Science and Technology Facilities Council

The RAL tier 1 setup - Q1 2023

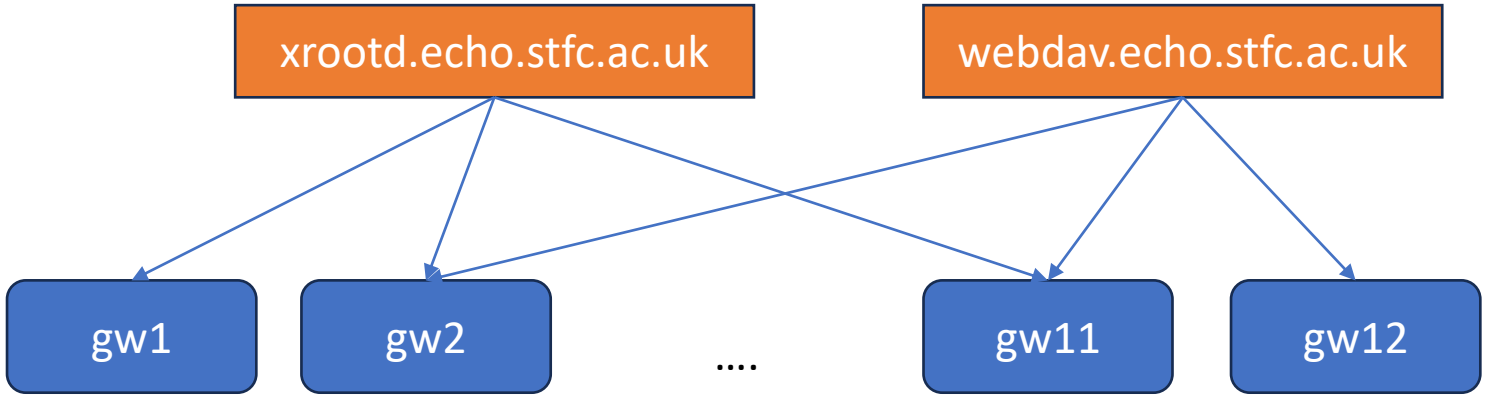
- You may notice that there is one gateway that exclusively belonged to one alias, but not both
- This was used as a control set in case of issues to detect if it affected a single protocol or both



Science and Technology Facilities Council

The RAL tier 1 setup - Q1 2023

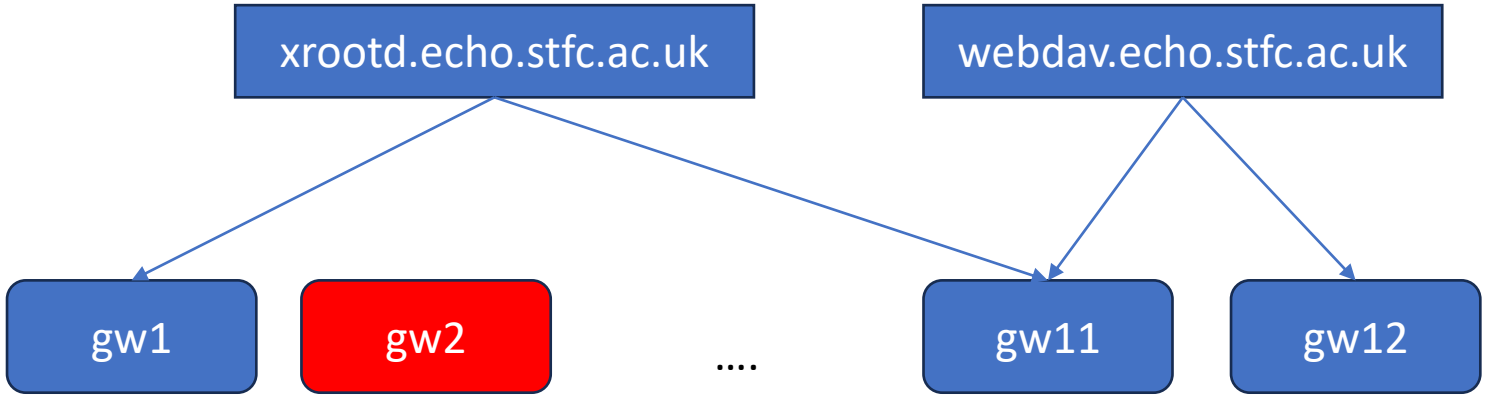
- The aliases are not directly managed by the Tier-1
- Meaning any change would have to go through a ticketing system to take effect, and are not as quick as we'd like for addressing immediate operational issues.



Science and Technology Facilities Council

The RAL tier 1 setup - Q1 2023

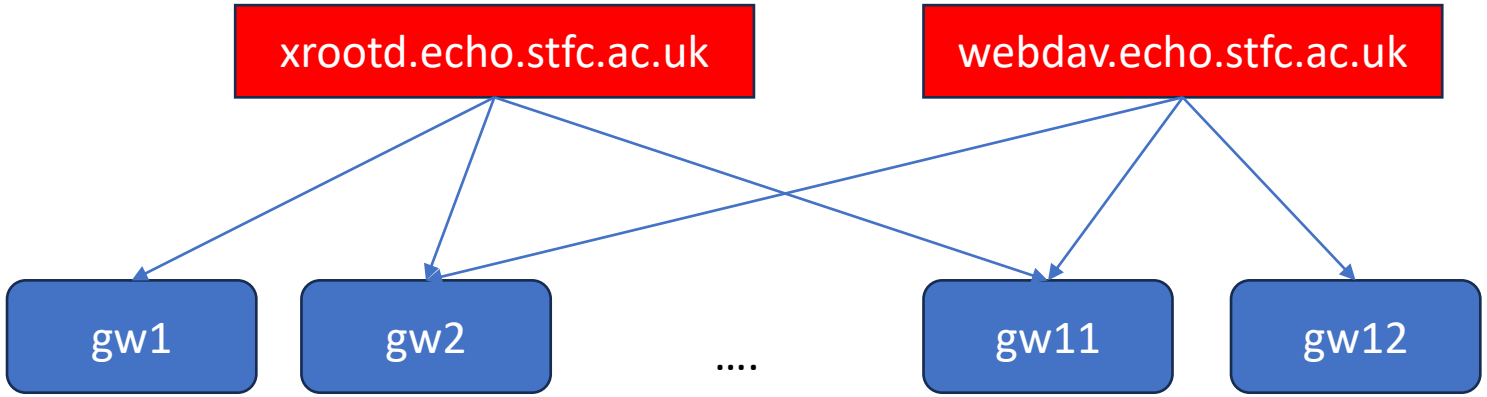
- For example, if one gateway stops responding or has issues not immediately solved on restart
- It needs a manual intervention to send a ticket to change the round robin



Science and Technology Facilities Council

The RAL tier 1 setup - Q1 2023

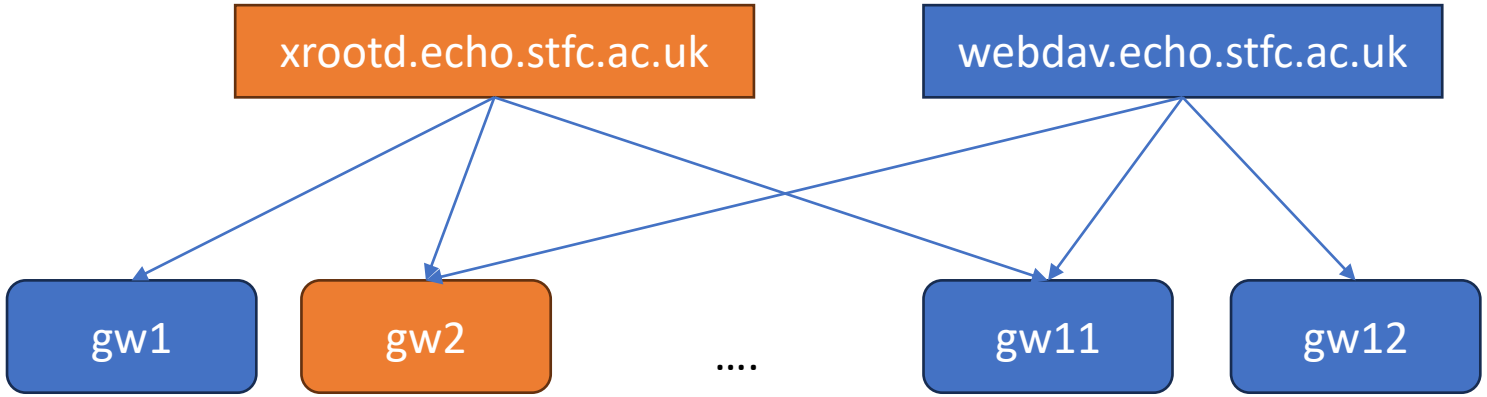
- And if the DNS servers had issues, the short TTL would result in an immediate impact on the service



Science and Technology Facilities Council

The RAL tier 1 setup - Q1 2023

- If a client caches the alias to a particular host, it bypasses the load balancing and focuses the load onto particular hosts

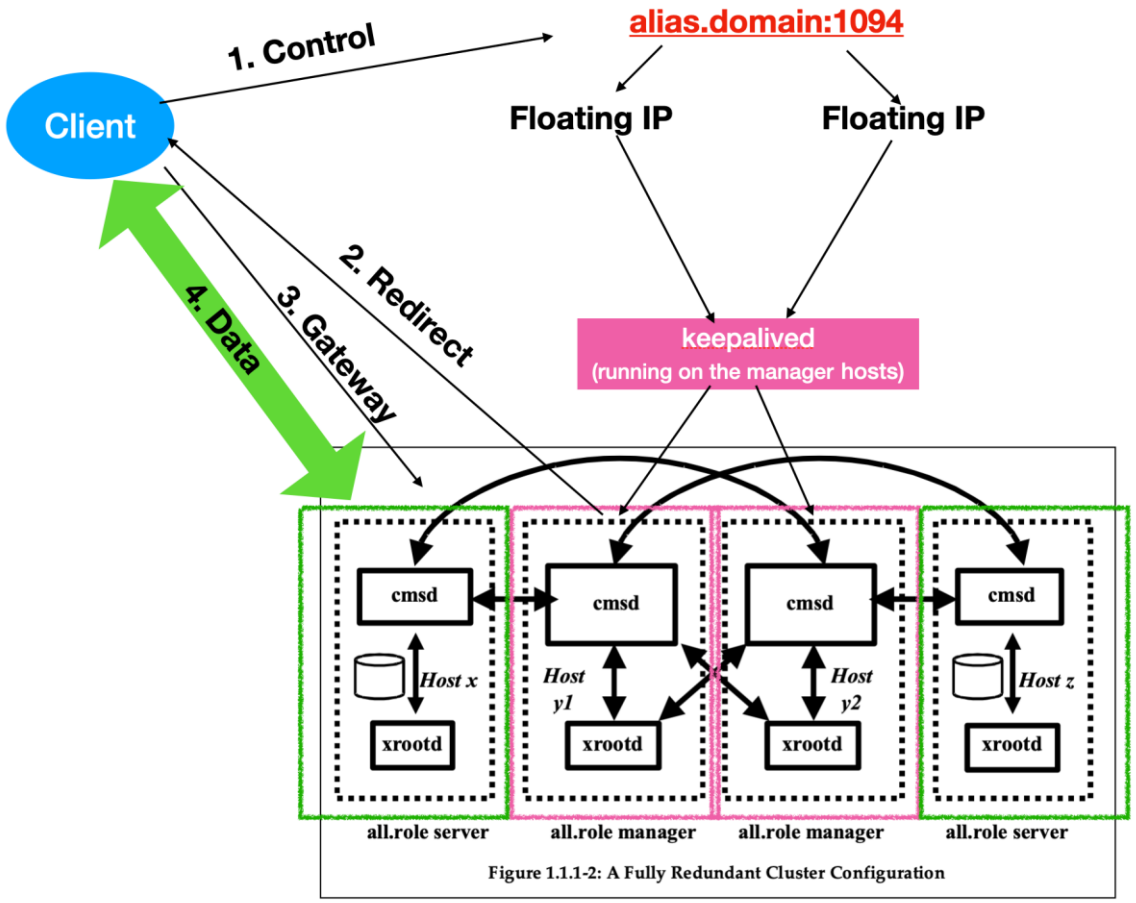


Science and Technology Facilities Council

CMSD setup

CMSD

- Seamlessly deal with a failed gateway or intervene on individual gateways.
- Evenly spread the load between the gateway hosts and automatically mitigate the pattern of ‘hotspotting’
- Reduce our dependence on the DNS provider.
- Allow us to use a much longer TTL for our Echo alias, and so make Echo more resilient against any DNS issues

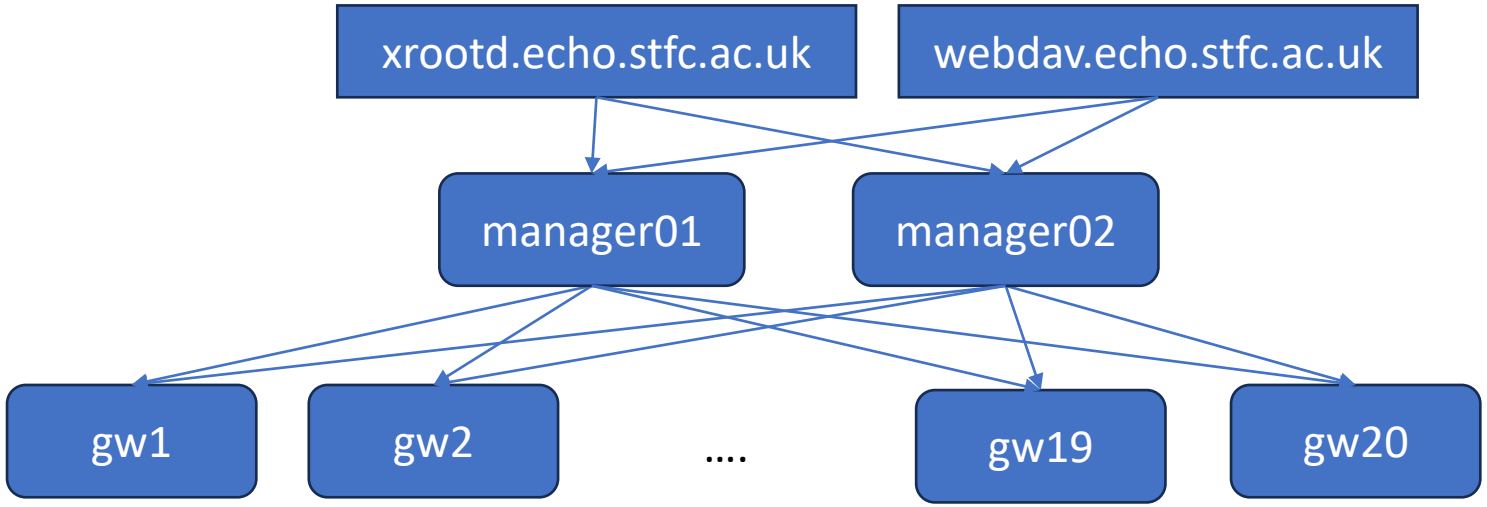


Initial CMSD setup diagram, by James Walder and Tom Byrne

Science and Technology Facilities Council

The RAL tier 1 setup - Redirector

- This made the service much easier to manage, and the XRootD cluster management load balancing worked well under normal conditions
- Adding and removing hosts can be done by editing the file specified in the cms.blacklist configuration option



Science and Technology Facilities Council

Science and Technology Facilities Council

Data Challenge ‘24

DC24

The WLCG ran a Data Challenge in February ’24 which stress tested the WLCG infrastructure and helped in identifying bottlenecks.

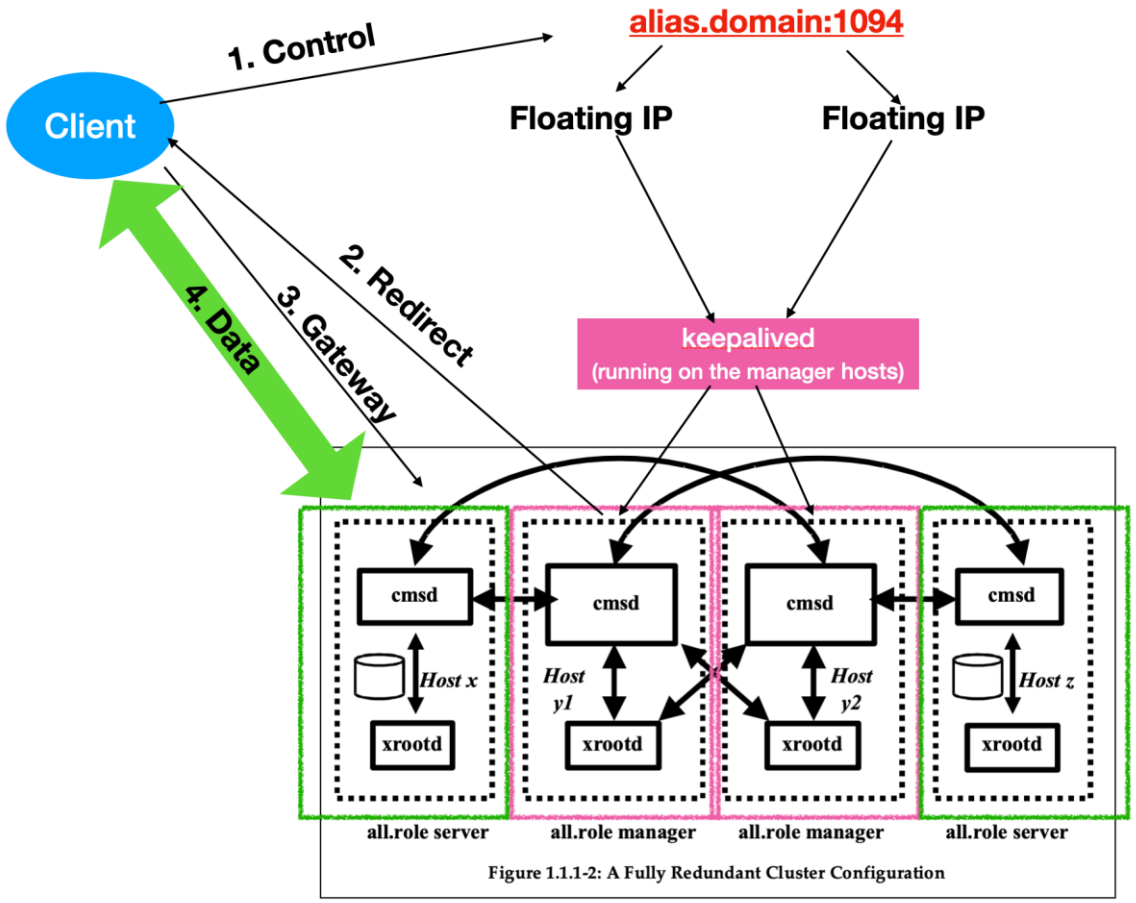
At the time the challenge started, we had more gateways (26 total) and were using the redirector setup

Science and Technology Facilities Council

CMSD setup

CMSD

- Seamlessly deal with a failed gateway or intervene on individual gateways.
- Evenly spread the load between the gateway hosts and automatically mitigate the pattern of ‘hotspotting’
- Reduce our dependence on the DNS provider.
- Allow us to use a much longer TTL for our Echo alias, and so make Echo more resilient against any DNS issues



Initial CMSD setup diagram, by James Walder and Tom Byrne

Science and Technology Facilities Council

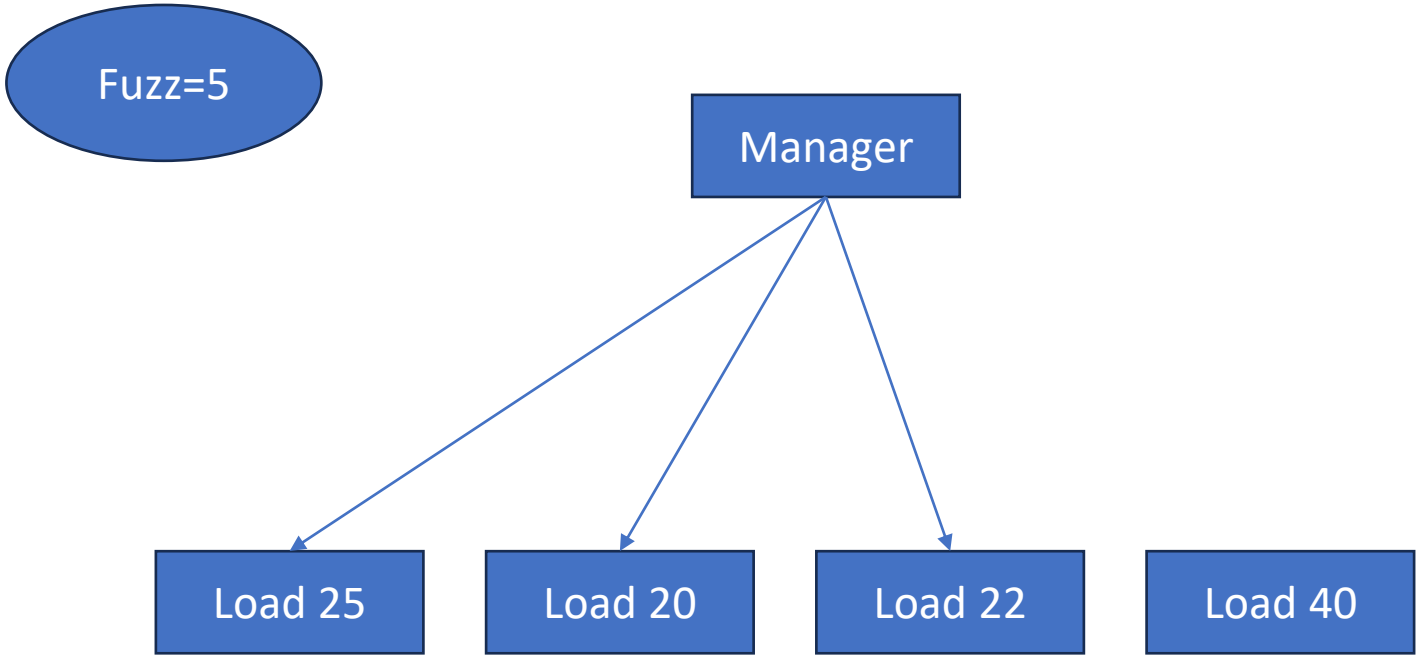
The XRootD load balancing algorithm

How it's intended to work:

1. Generate an overall load score based on a weighted sum of the different load metrics reported (network, cpu load, system load, memory usage, disk space)
2. Skip unusable nodes (not responding, over the configured max load, etc..)
3. Assign incoming transfers by round robinin between the least loaded gateway and other gateways within a set window (fuzz) around it

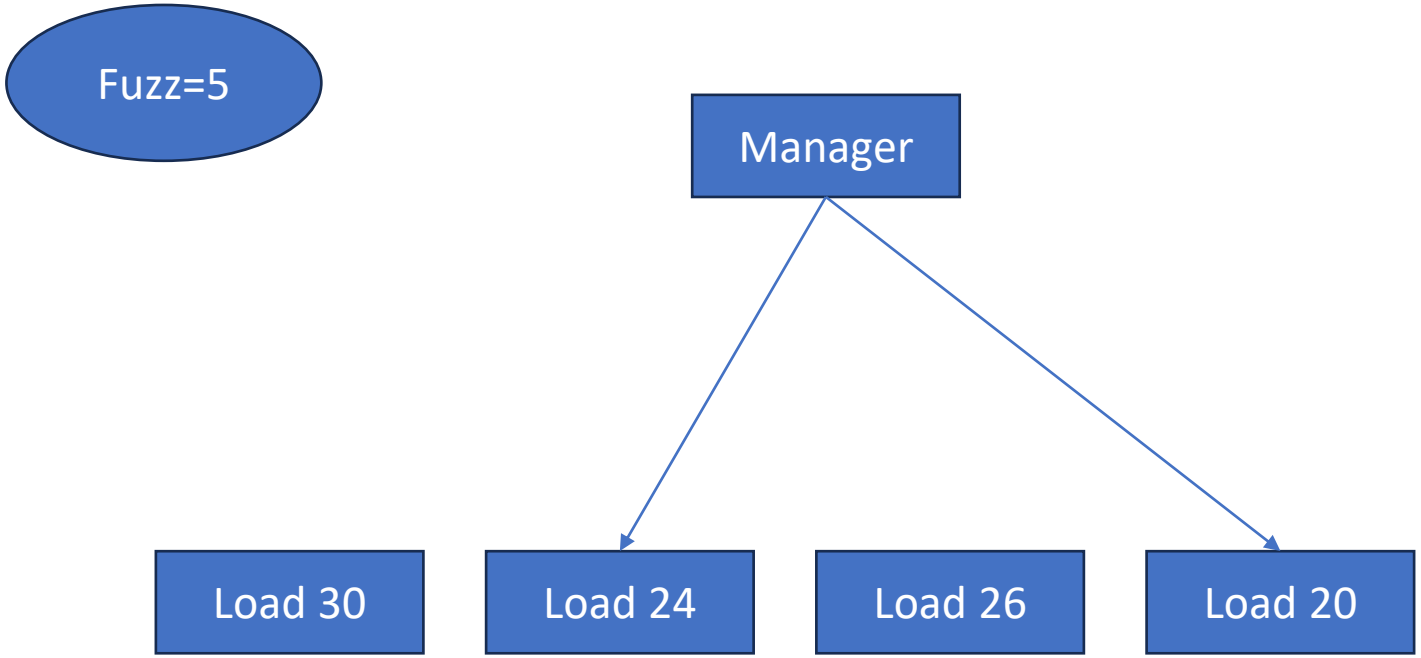
Science and Technology Facilities Council

The XRootD load balancing algorithm



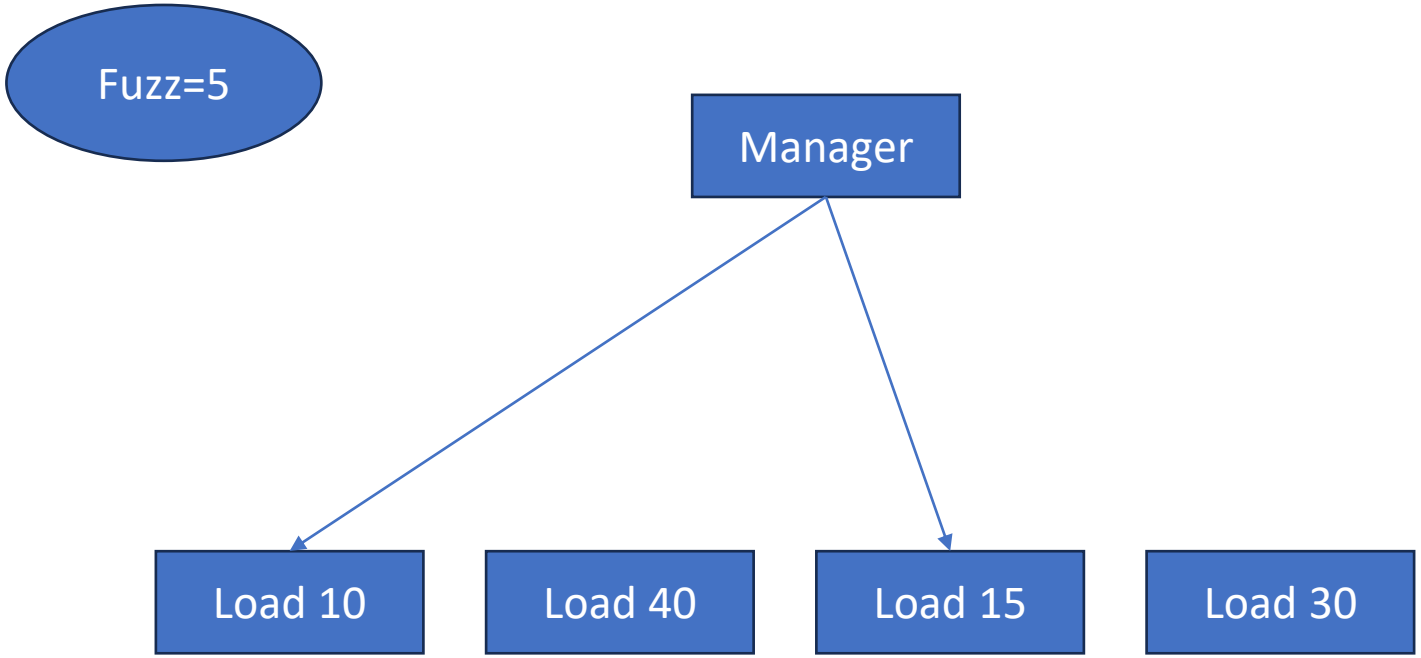
Science and Technology Facilities Council

The XRootD load balancing algorithm



Science and Technology Facilities Council

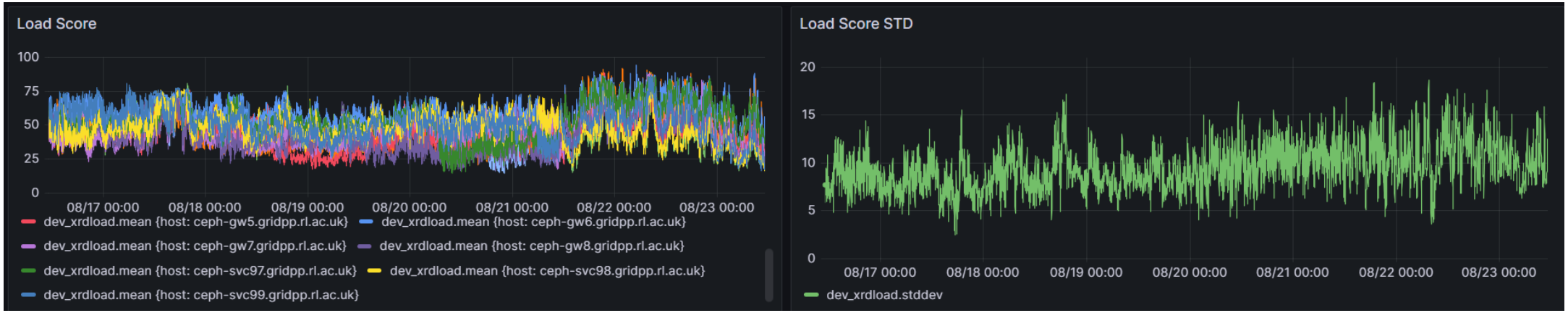
The XRootD load balancing algorithm



Science and Technology Facilities Council

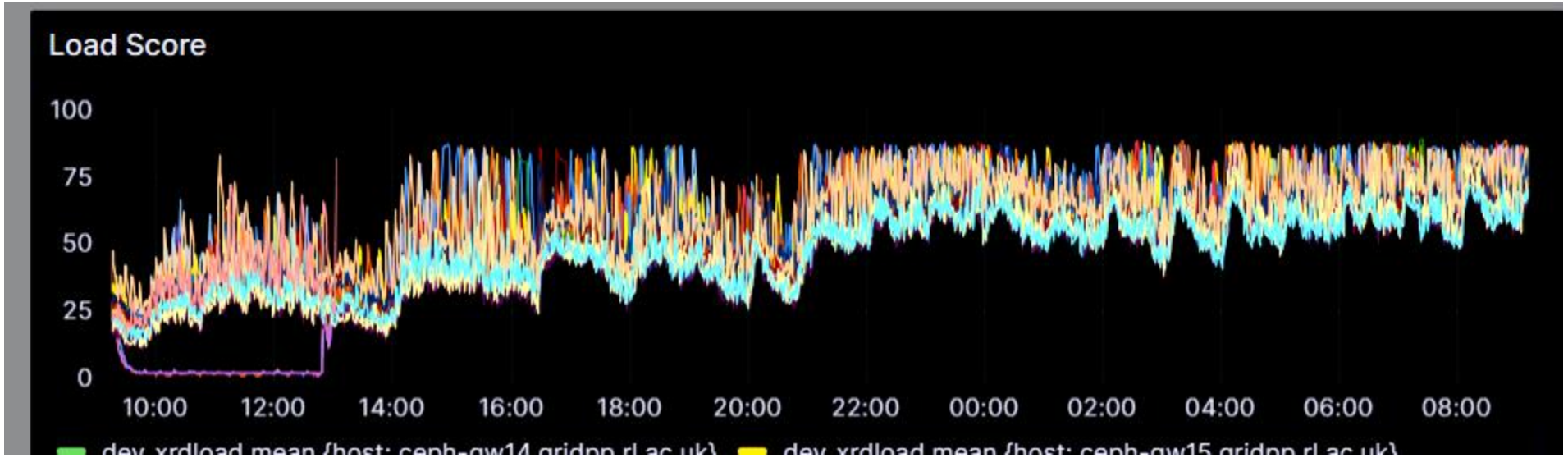
The XRootD load balancing algorithm

- This works fine under normal conditions as the load fluctuations over time make balance it out, although resulting in a ‘bouncing’ load pattern



Science and Technology Facilities Council

XRootD load balancing under pressure



Load keeps capping out and transfers go to whichever gateways go below the maximum load first, Pushing it back over the load limit

Science and Technology Facilities Council

The XRootD load balancing algorithm

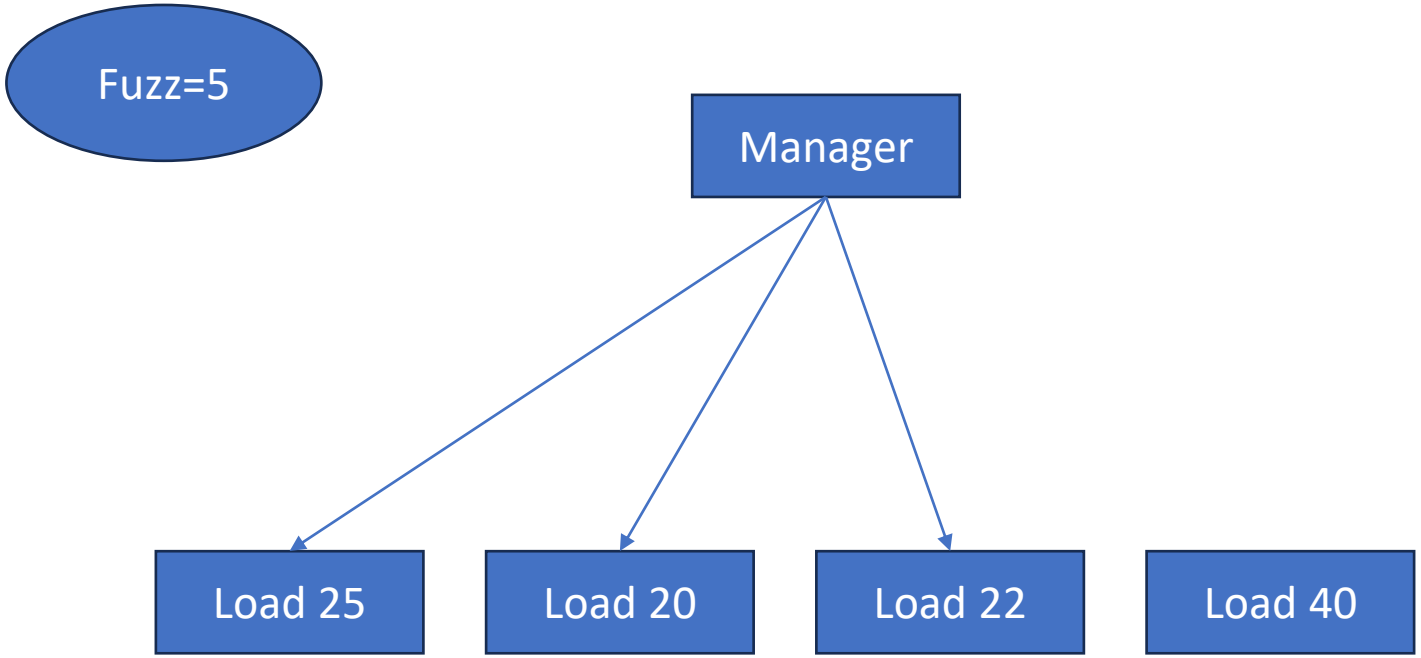
How it works[1]:

1. Generate an overall load score based on a weighted sum of the different load metrics reported (network, cpu load, system load, memory usage, disk space)
2. Skip unusable nodes (not responding, over the configured max load, etc..)
3. go through the gateways in order of first appearance in the cluster, switching the selected gateway to the next one if it's significantly less loaded or within the fuzz and had received less transfers than the currently selected one

Science and Technology Facilities Council

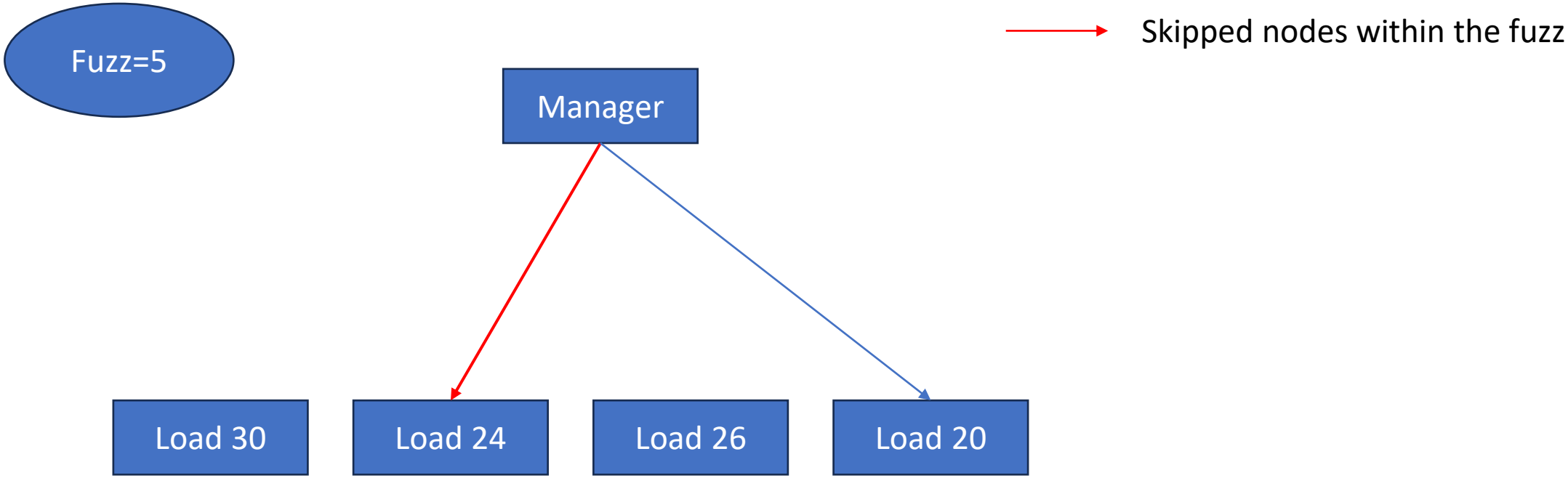
[1] XRootD CMSD SelByLoad Analysis, Thomas Byrne 2024

The XRootD load balancing algorithm



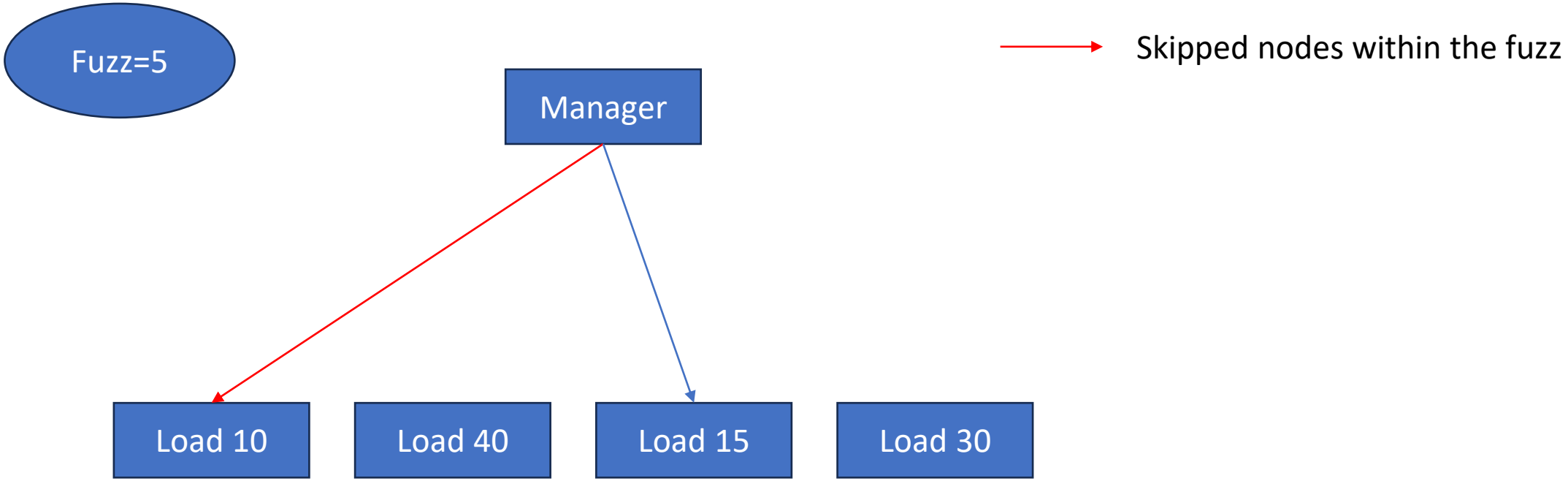
Science and Technology Facilities Council

The XRootD load balancing algorithm



Science and Technology Facilities Council

The XRootD load balancing algorithm



Science and Technology Facilities Council

The XRootD load balancing algorithm

- This is not an issue under normal operations, as focused load results in increasing load on the underloaded gateways, removing it from the selection pool the next time round
- But it’s an issue when every gateways is overloaded or close to it.
- Some load patterns are particularly problematic (shown later)

Science and Technology Facilities Council

The 5 phases of XRootD load balancing

1. Bargain
2. Explore
3. Nostalgia
4. Vintage
5. Innovate

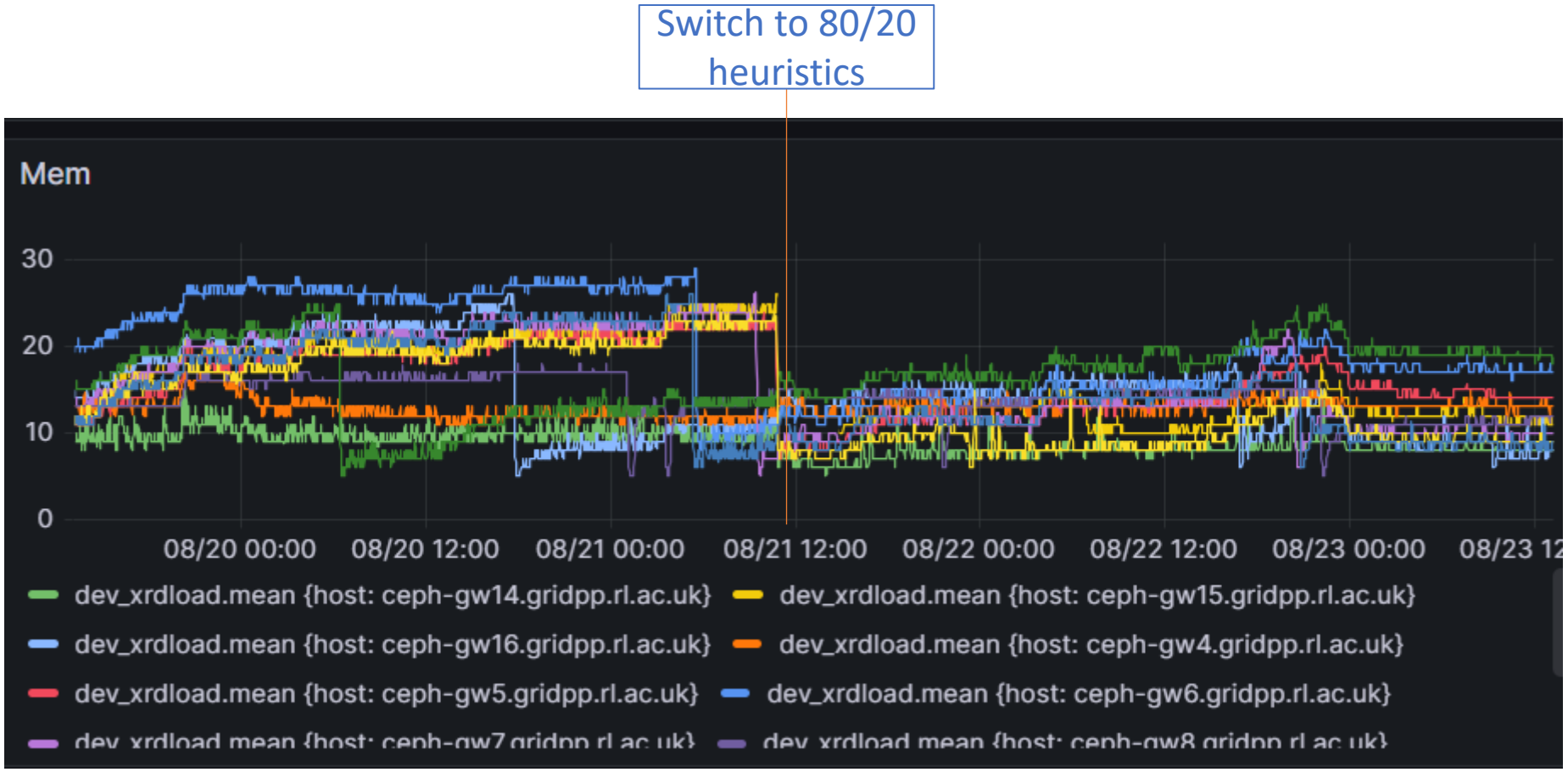
Science and Technology Facilities Council

Phase 1 – Bargain

Tune the existing load balancing to distribute load very evenly

- 80/20 split of system load/cpu, with fuzz 3 provided the best load balancing we could get
- Generally ok but performance degrades under heavy load

Science and Technology Facilities Council



Following the heuristic switch, load has been balanced better between the gateways

Science and Technology Facilities Council

Phase 2 – Exploration

Explore the space for better alternatives under heavy load

- 50/50 split of system network/cpu
  - no significant difference. Some improvement in performance for newer hardware at the expense of the older ones
- Non-standard metrics
  - Number of active connections, heartbeat time
  - Not very consistent and hard to tune equally among gateways under heavy load

Science and Technology Facilities Council

Phase 3 - Nostalgia

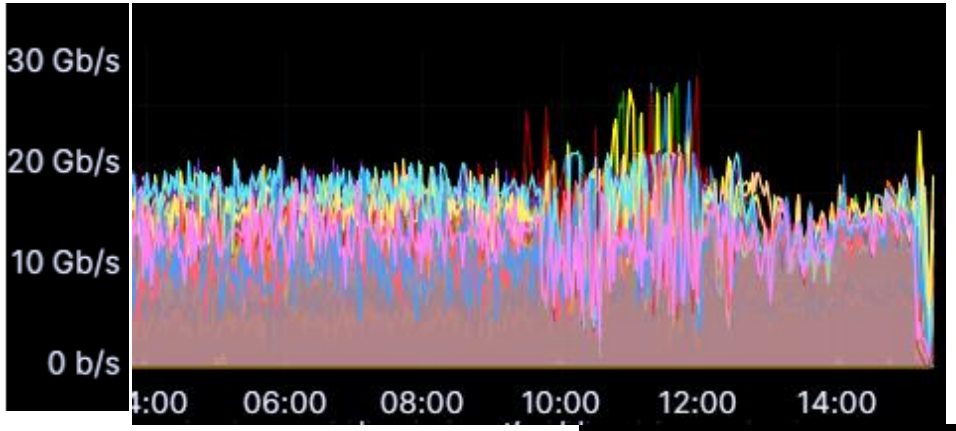
Simulate Round Robin

- All gateways report the same loads artificially (passive load balancing)
- A lot more stable, low error rate and better throughput even under heavy load
- If an individual gateway starts to get loaded, it will keep getting loaded until it breaks

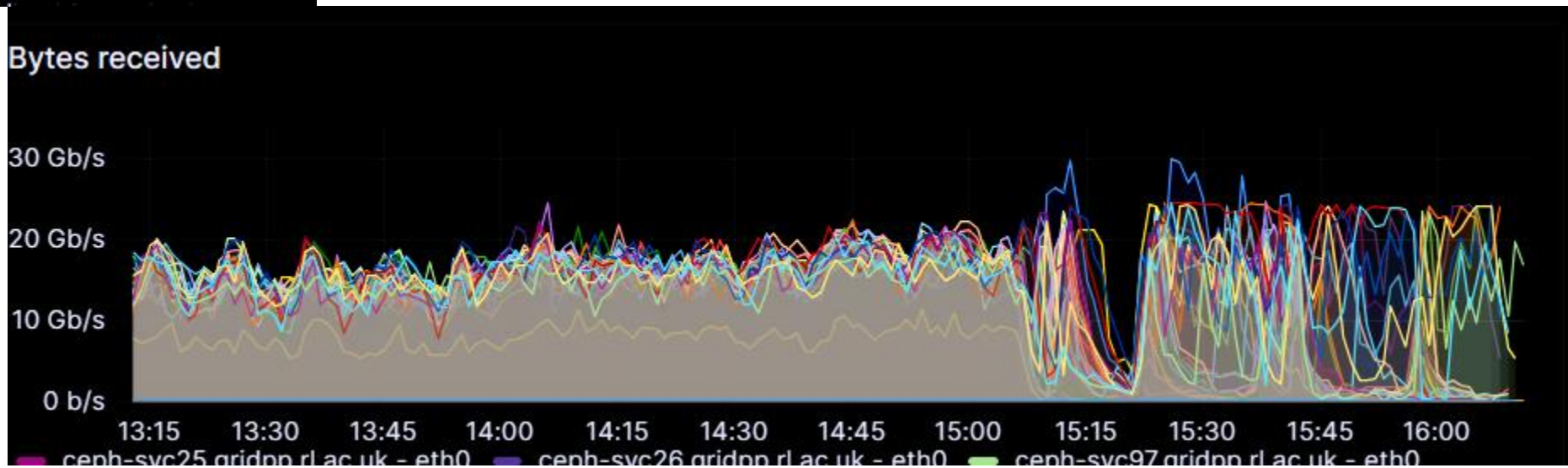
Science and Technology Facilities Council

XRootD load based balancing vs Round Robin

XRootD load balancing to RR (bytes received)



RR to XRootD load balancing based on nonstandard loads

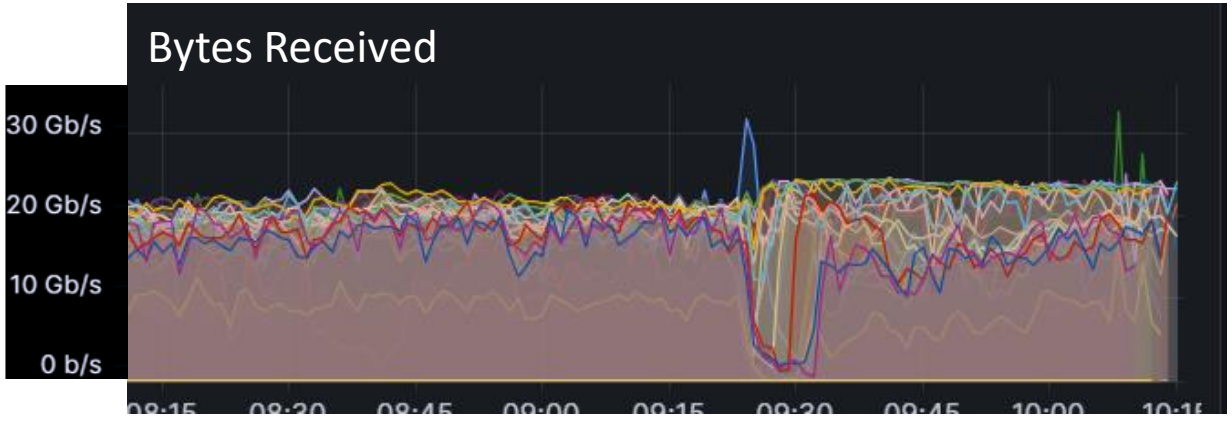


Science and Technology Facilities Council

Phase 4 – Vintage

Gateways report the same load unless it's nearing problematic levels (80% system load) at which point the reported load is set higher to remove it from the Round Robin

- Similar benefits to Round Robin approach, but would usually keep gateways from getting overloaded
- It's easier to fall into the pitfalls of the existing algorithm (seen later) and some states cannot be gotten out of without manual intervention

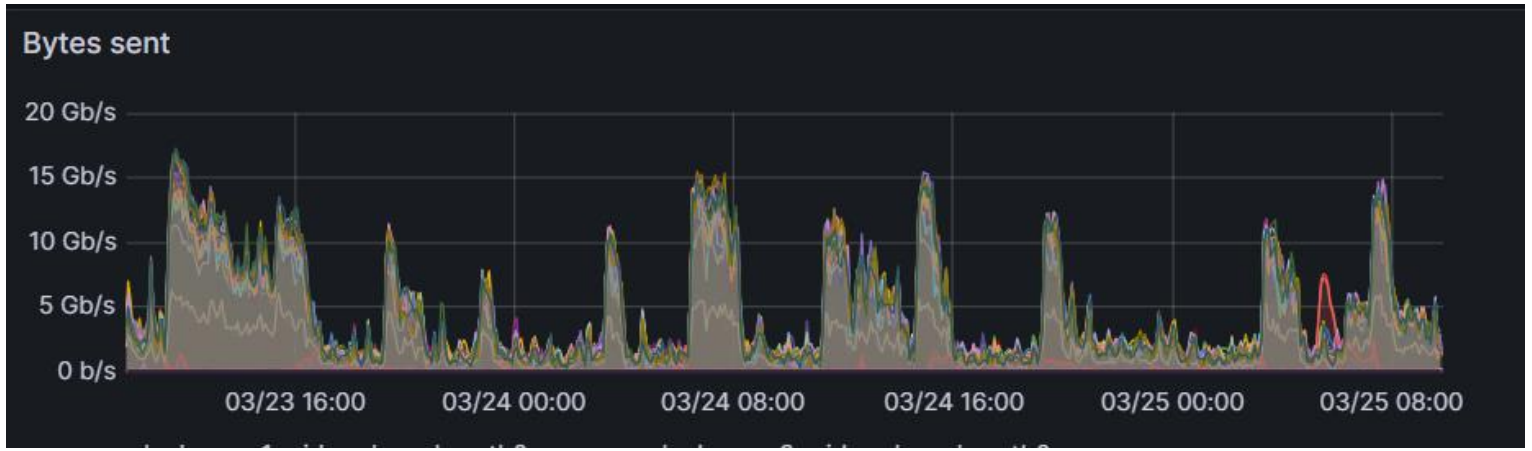


Science and Technology Facilities Council

Phase 5 – Innovate

We decided to make our own load balancing algorithm

- Variant of weighted random load balancing
- More likely to send transfers to less loaded gateways
- A gateway will only be excluded when it goes over the allowed maximum load or is unreachable



Science and Technology Facilities Council

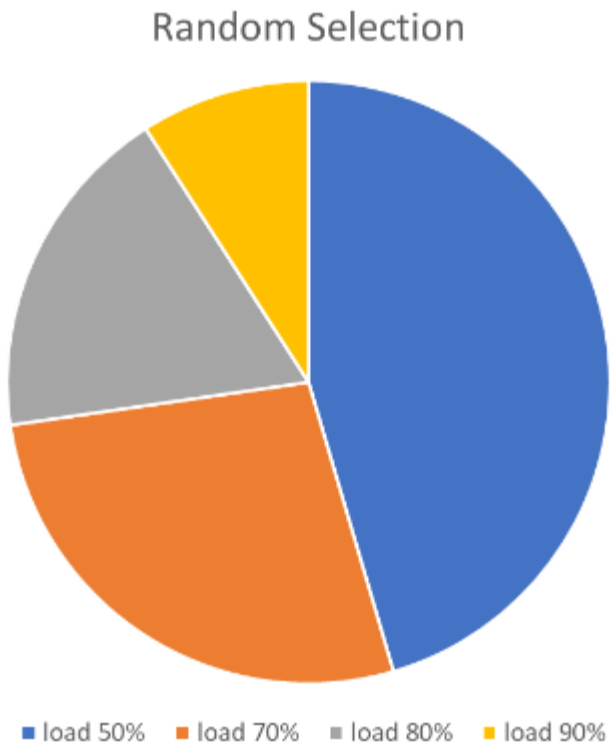
How the new algorithm works

The basic principle of this algorithm is a random weighted selection. An easy way to picture it is as follows:

Imagine a spinning wheel divided in slices. The less loaded gateways will have a larger slice of the pie. Now throw a dart at the board. Whichever slice the dart landed on is going to be selected for the transfer.

Science and Technology Facilities Council

How the new algorithm works

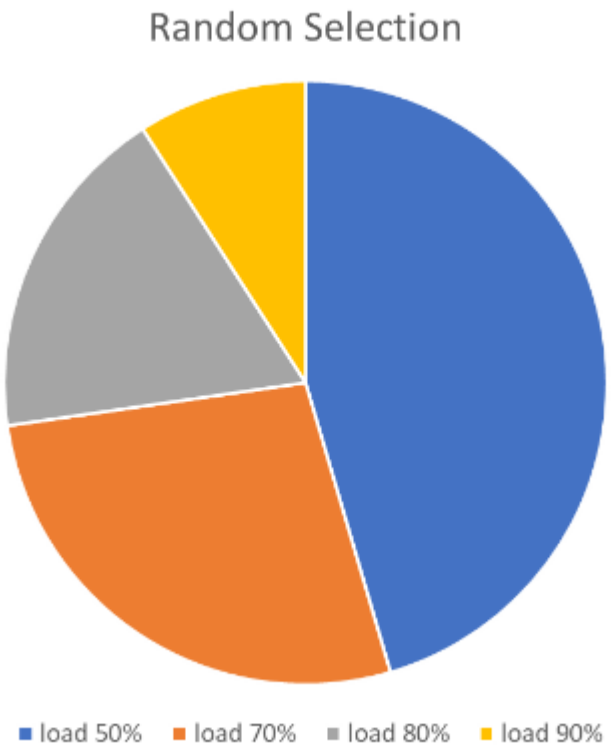


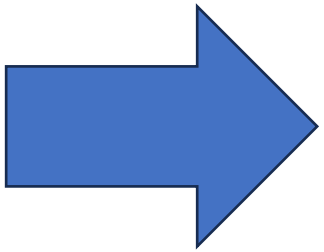
Science and Technology Facilities Council

How the new algorithm works

- If a node is unavailable or has issues, it has an effective slice size of 0, meaning it will never be selected.
- A fuzz value provides a baseline for each slice size, providing some tuning adjustments for a more even distribution. e.g. a fuzz of 20 on the same values above will result in this wheel:
- Thanks to Guilherme Amadio for helping optimize this section of the algorithm!

How the new algorithm works





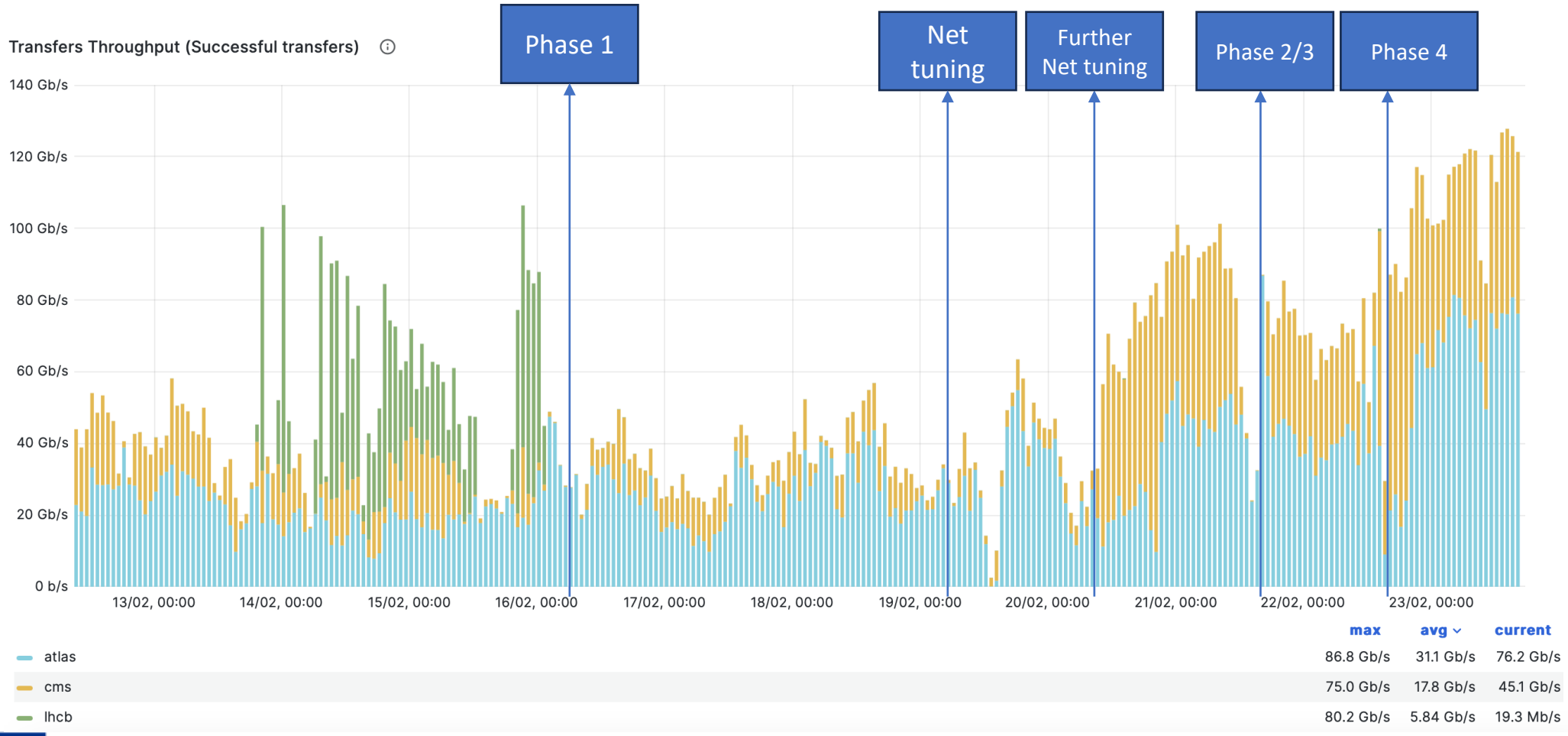
Science and Technology Facilities Council

How the new algorithm works

- How resilient is it?
  - Due to each transfer adding load unto a gateway until the transfer is complete, each selection results in reducing the size of the slice if it's selected consecutively
  - 10M simulated repetitions resulted in a maximum consecutive selection of the same node = 10
- How to switch to using it?
  - Available since XRootD 5.7.0
  - cms.sched affinity randomized
  - cms.sched cpu 50 io 50 mem 0 pag 0 runq 0 space 0 fuzz N
  - N>0

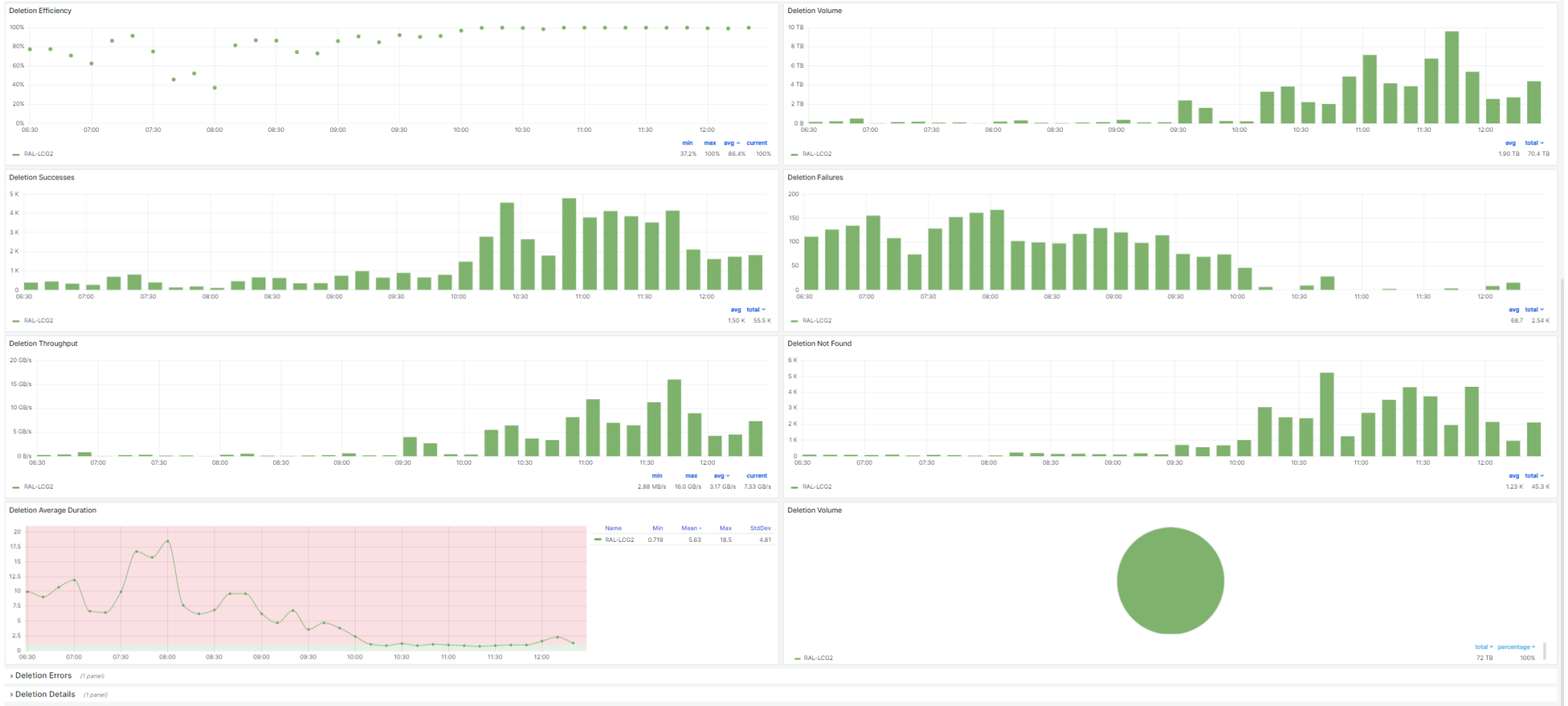
Science and Technology Facilities Council

Transfer throughput over DC24



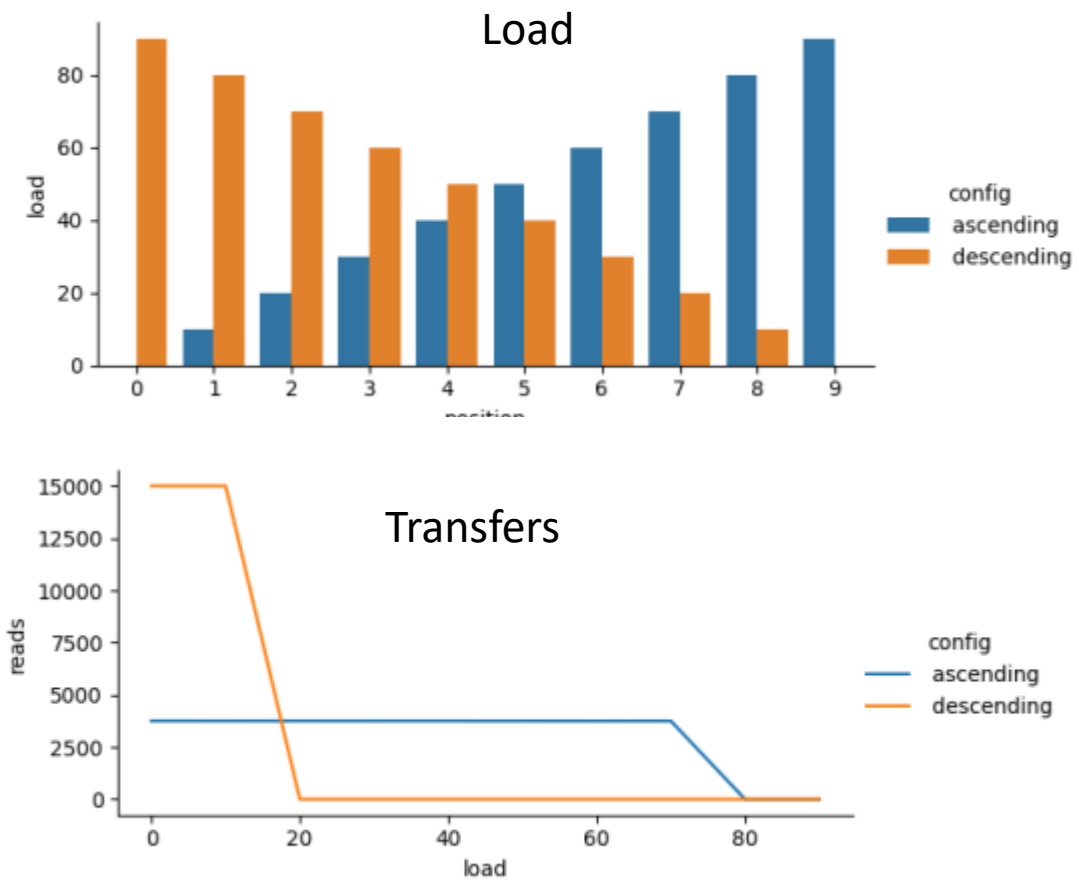
Science and Technology Facilities Council

Deletion efficiency default vs new algorithm



Science and Technology Facilities Council

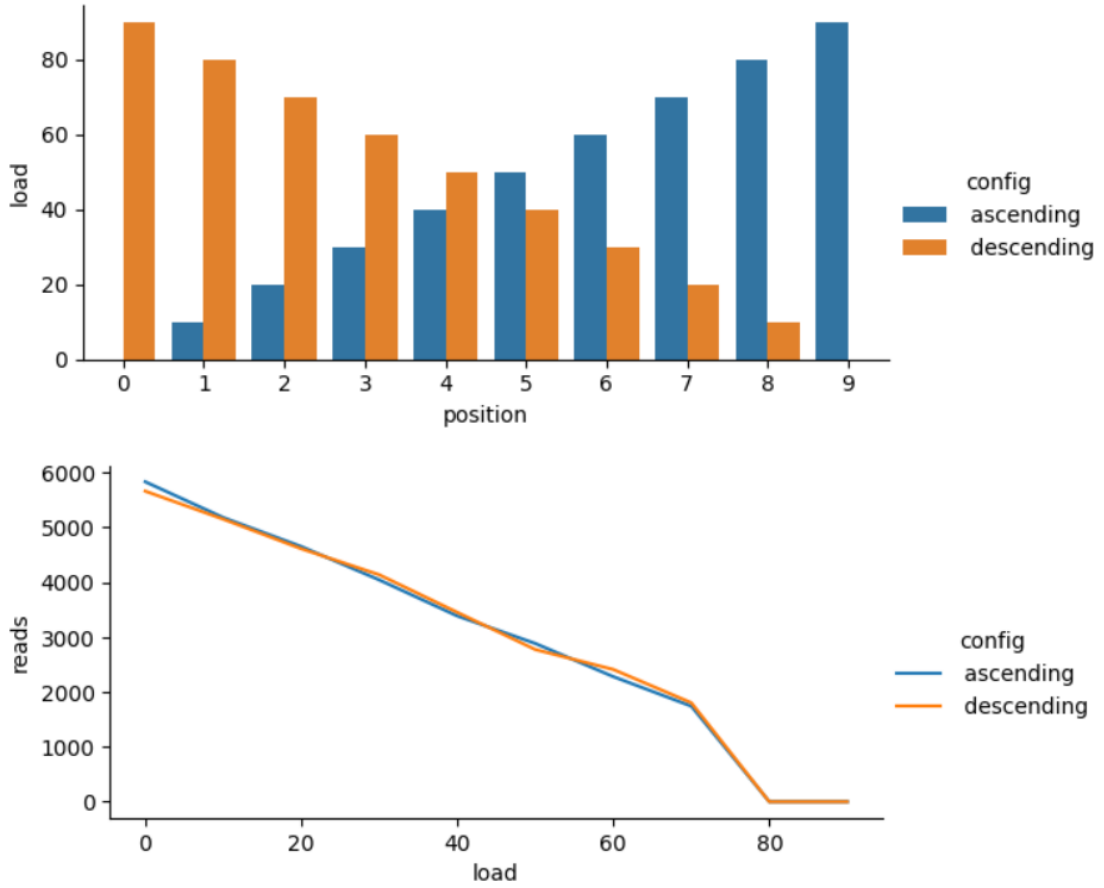
Problematic load patterns - Ascending/descending order of load



Science and Technology Facilities Council

XRootD CMSD SelByLoad Analysis, Thomas Byrne 2024

Problematic pattern- Ascending/descending order of load - new algorithm

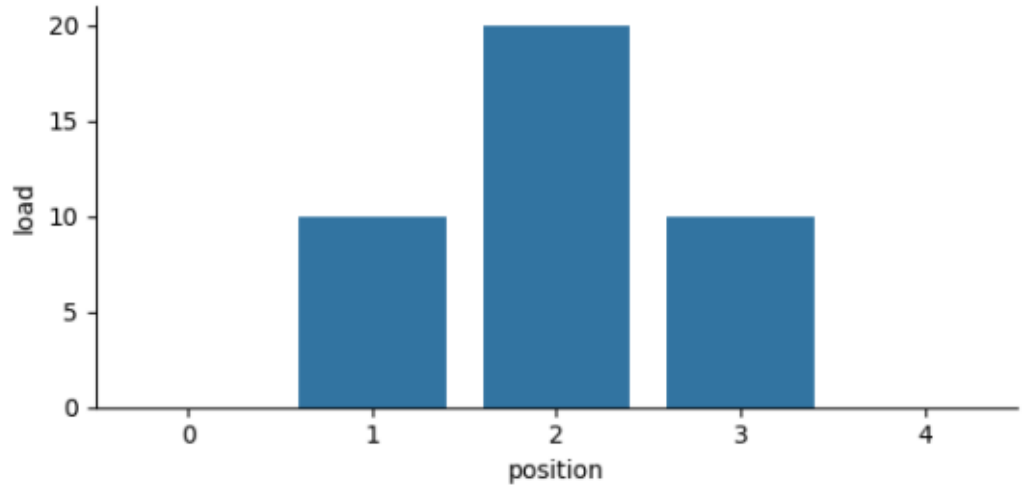


Science and Technology Facilities Council

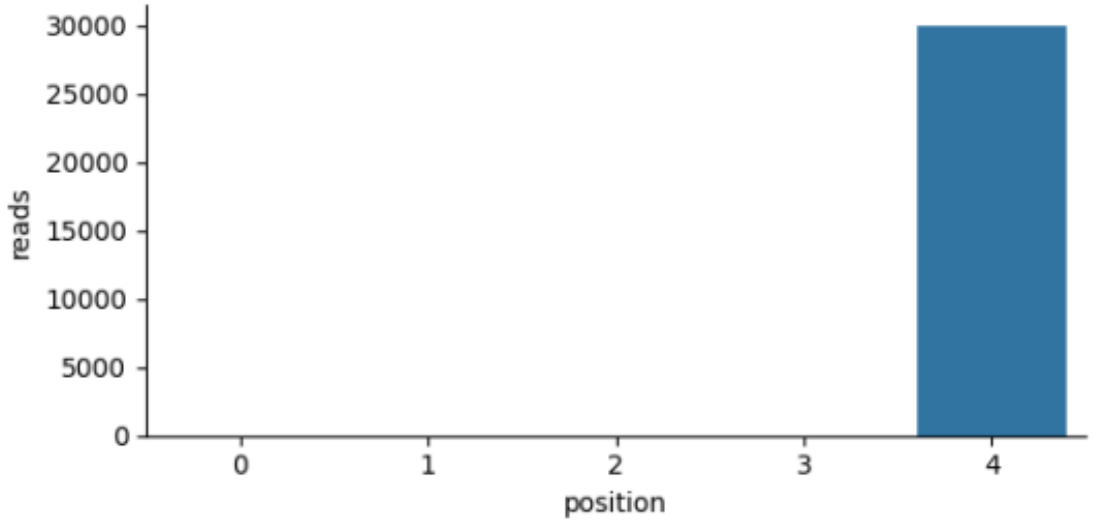
XRootD CMSD SelByLoad Analysis, Thomas Byrne 2024

Problematic load patterns- Artificial hotspotting

	name	load	reads
0	gw1	0	2
1	gw2	10	1
2	gw3	20	0
3	gw4	10	1
4	gw5	0	29996



Load

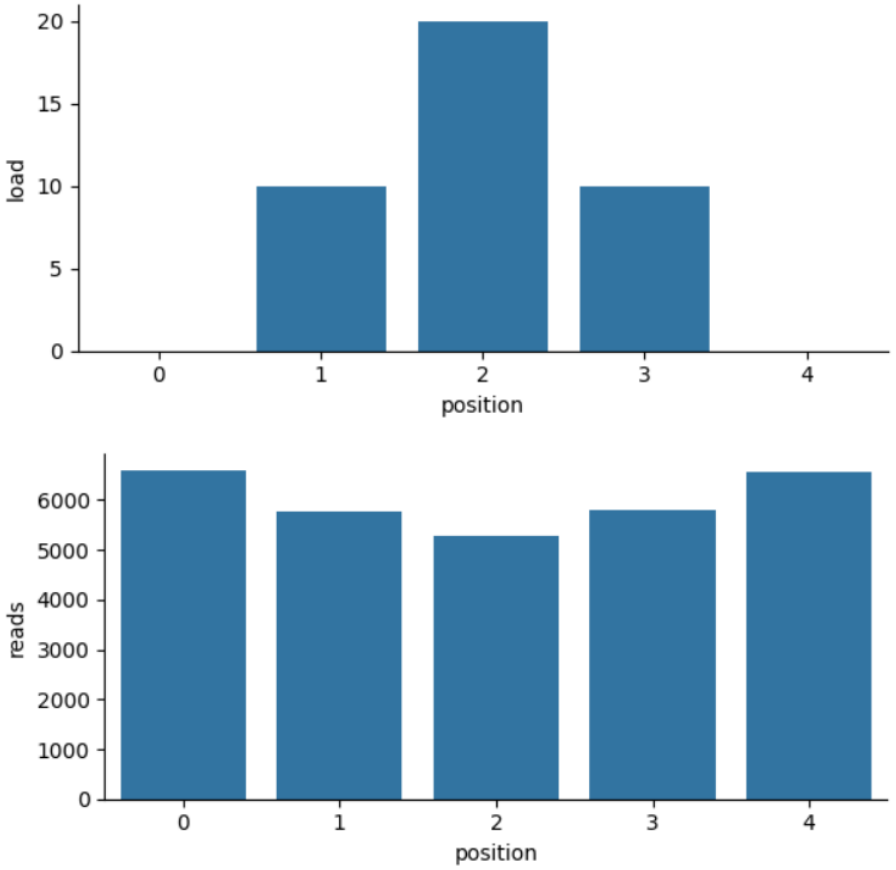


Transfers

Science and Technology Facilities Council

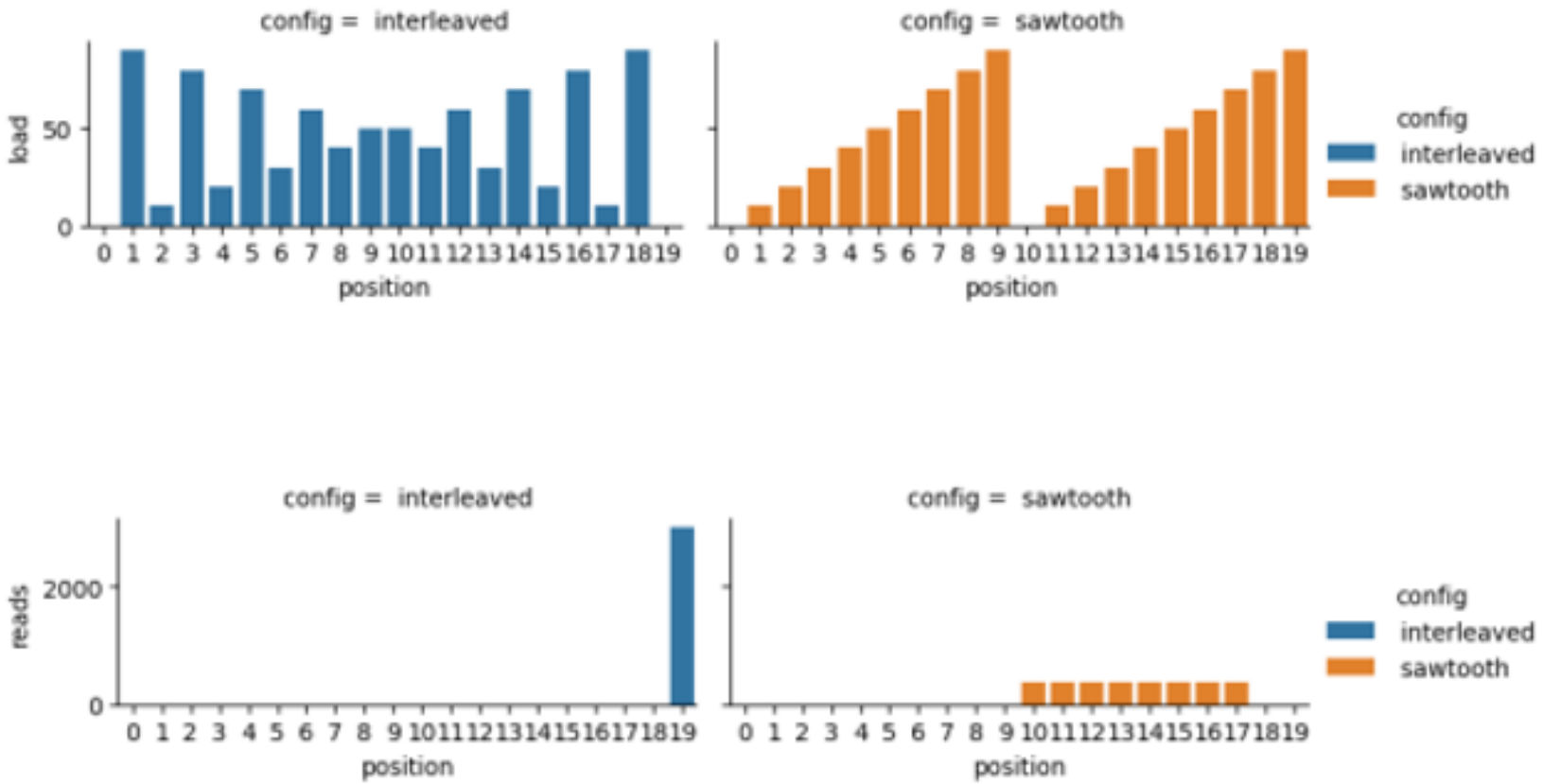
XRootD CMSD SelByLoad Analysis, Thomas Byrne 2024

Problematic load patterns- Artificial hotspotting – new algorithm



Science and Technology Facilities Council

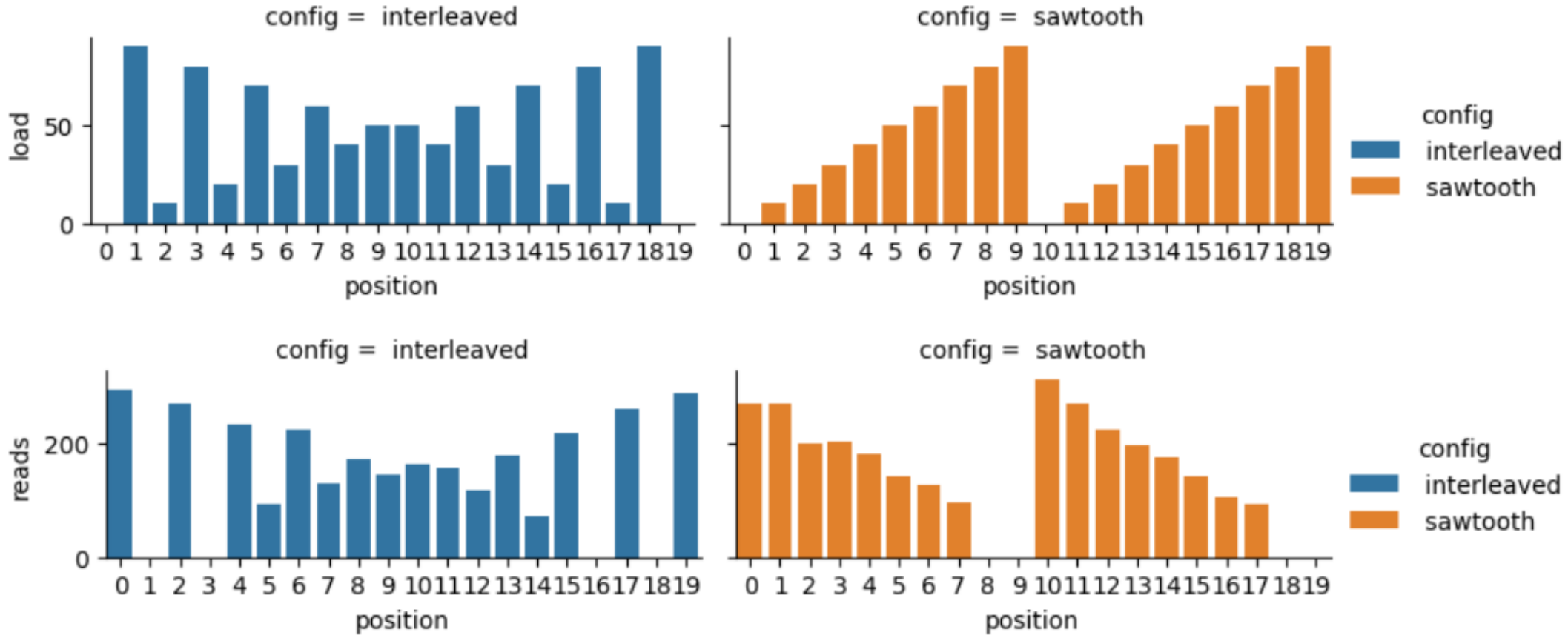
Other problematic patterns



Science and Technology Facilities Council

XRootD CMSD SelByLoad Analysis, Thomas Byrne 2024

Other problematic patterns – new algorithm



Facilities Council

XRootD CMSD SelByLoad Analysis, Thomas Byrne 2024

Thank you

Science and Technology Facilities Council

Randomized Algorithm - detail

Variables used

- **sp** - selected node
- **np** - current node
- **np->load** - load reported by the node
- **TotWeight** - the current sum of inverse load, adjusted with a fuzz factor for tuning
- **NodeWeight** - array of total weights at the current node

Science and Technology Facilities Council

Randomized Algorithm - detail

Initialization

1. **TotWeight** is set to 0
2. **NodeWeight** for the current node is set to 0
3. **sp** is set to the first valid node

Science and Technology Facilities Council

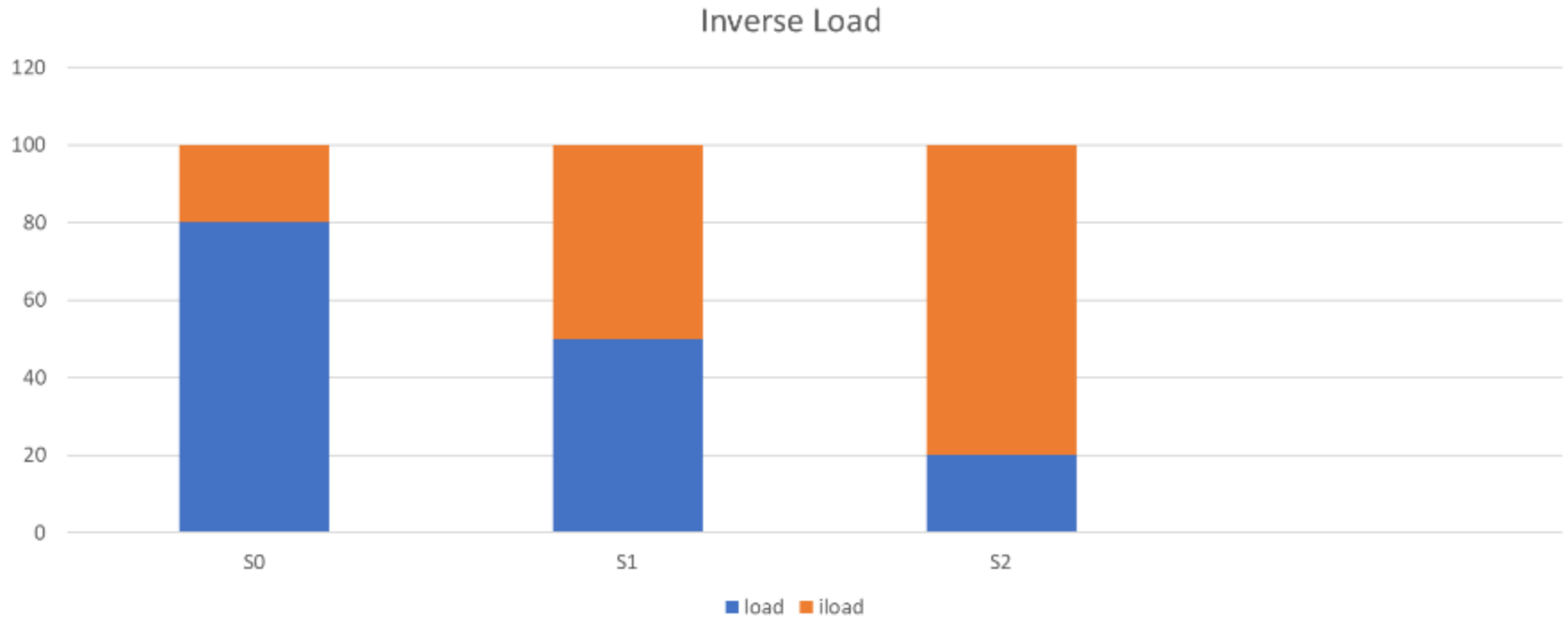
Randomized Algorithm - detail

Looping on each node (including the first **sp**)

1. **TotWeight** is incremented by **fuzz** + (100 - **np->load**)
    1. **fuzz** prevents inverse load being 0 for a gateway at 100 load. this provides even load balancing in cases where every node is at 100 load
    2. higher **fuzz** values reduce the importance assigned to the load, helping to tune the algorithm.
2. **NodeWeight** for the current node is set to the current **TotWeight**.

Science and Technology Facilities Council

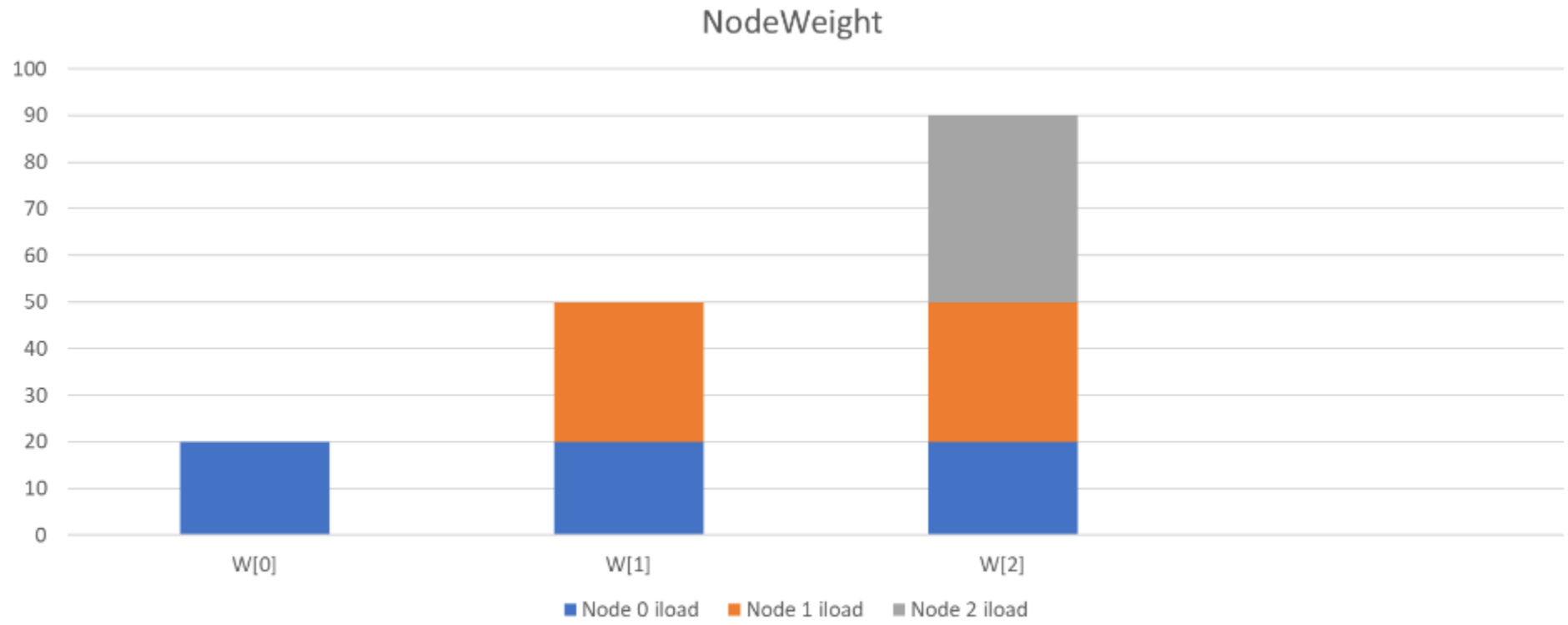
Randomized Algorithm - detail



inverse load results in larger values for less loaded nodes

Science and Technology Facilities Council

Randomized Algorithm - detail



At the end of the loop, the NodeWeight array will look like this.

Science and Technology Facilities Council

Randomized Algorithm - detail

After the loop has gone through all available nodes

1. Generate a random number from 1 to the final **TotWeight**
2. Select the first node where the value in **NodeWeight** is greater than the random number