Improving Federated Access for ALICE

Costin.Grigoras@cern.ch

Outline

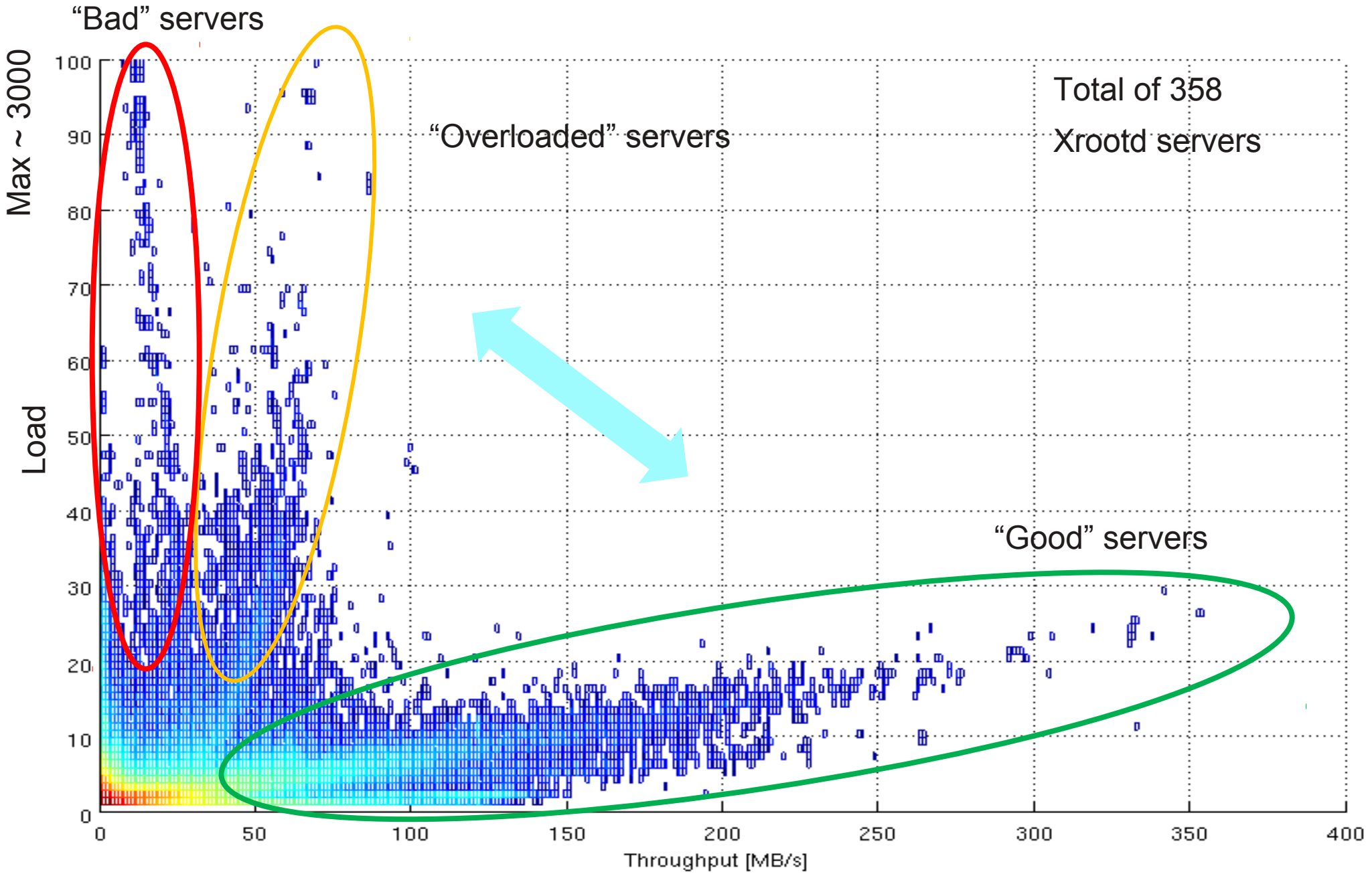
- Xrootd monitoring results
- Possible server improvements
- Client feature requests

Monitoring information

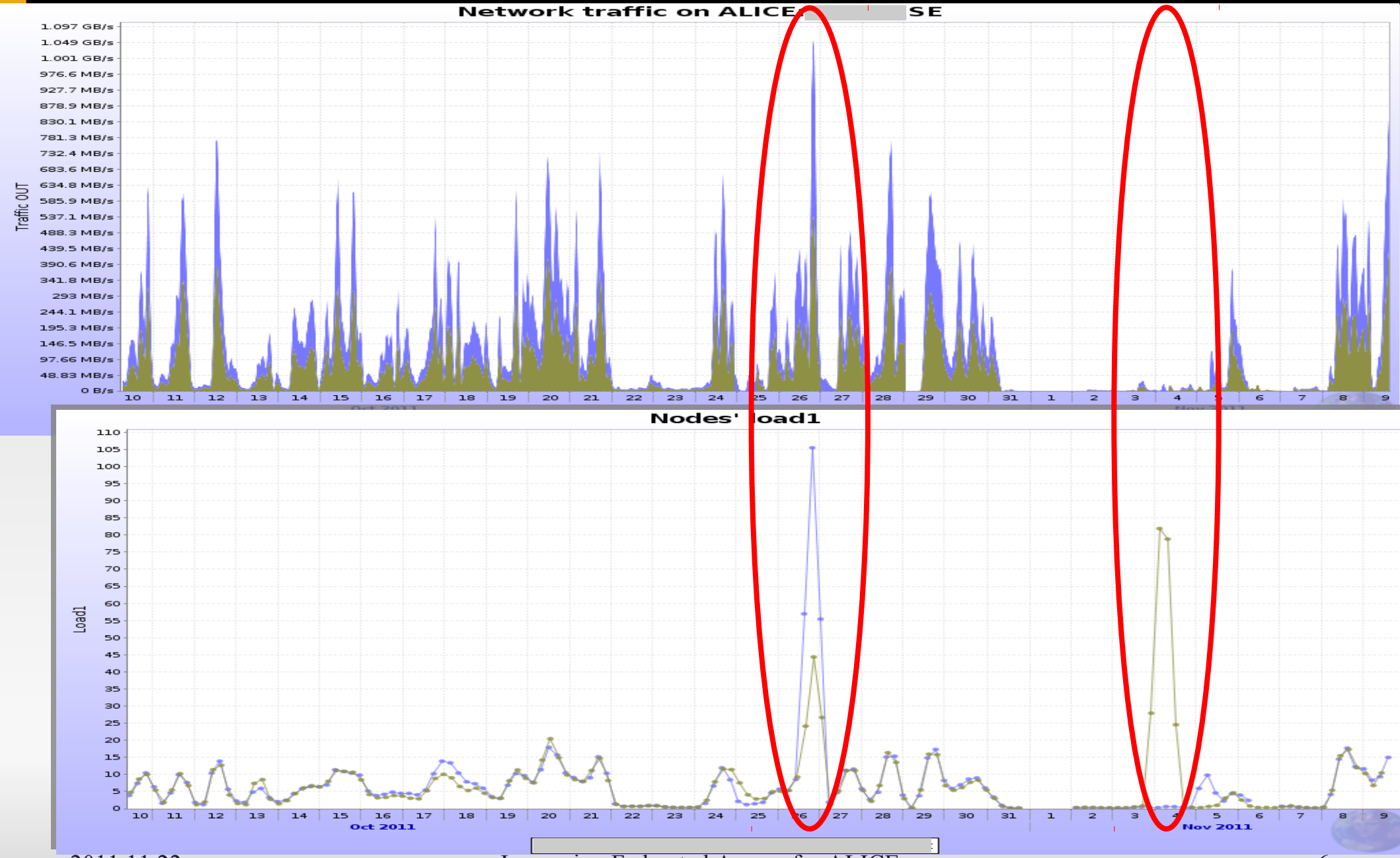
- Host and Xrootd server monitoring daemon packaged with the Xrootd plugin
  - CPU and memory usage
  - Load, processes, sockets
  - Network and disk IO
- Running on each server (and redirector)
  - Sending data to the closest MonALISA service (on the same site)
- Collected centrally and aggregated per SE

Job efficiency vs load

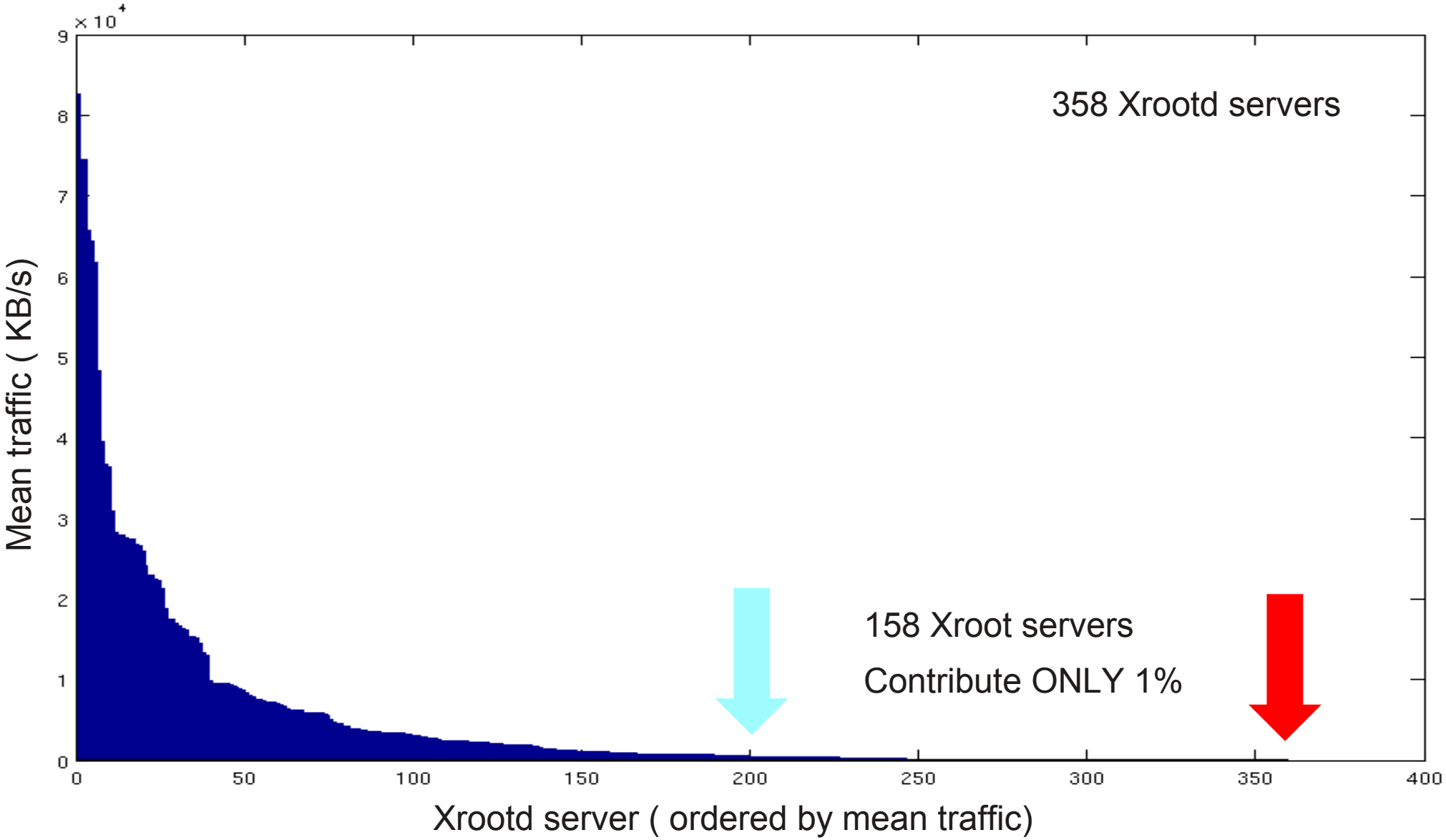
Throughput vs Load



Good Servers ... have problems



Mean traffic distribution



What to do next ?

- Understand the excessive load on the xrootd servers
  - alarms & help from the sys admins
- More internal parameters exposed for better understanding and debugging
  - Request queue length, number of (active) clients, IO threads' status (Apache server-status-like)
  - Avg time to serve a request, avg request length
  - Anything else related to server's health

Suggestions

- Server autotuning
  - Based on the load and/or the above stop accepting clients when they cannot be reasonably served
    - Toaster's approach on how not to burn your bread
  - Use the new IO Utilization kernel parameter of each device for load balancing of block devices based on their performance (`*iostat -x*`, a decently new kernel)
- Cluster autotuning
  - Same but at the redirector level
  - Default load balancing based on the IO capabilities of the machines

Client feature requests

- Native third-party copying capabilities
  - Currently using Andreas' xrd3cp tool
    - All T0 to T1 and wherever else we need to copy data
  - With progress report from the source
    - Now implemented by opening the target twice, but some implementations don't like it so much
- Separate timeouts
  - Connect
  - Request
  - overall transfer timeout

Client feature requests (2)

- Min & max bandwidth limits
- Use all available replicas for dw / transfer
  - Maybe only if the above min bw limit cannot be reached from the first source
  - As trasparent fallback mechanism
    - We will pass the sorted list
    - Now it is transparent only for Open()
  - Or aggresively ask the same block from everywhere
- Directly download a member of an zip archive
  - ROOT has this and is very helpful

Summary

- We need extensive monitoring
  - And the means to correlate and understand it
- Self-protecting servers
  - Cannot serve it, don't accept it
- Smart clients
  - Transparent fallback, optimizations, third-party …