Status PUNCH4NFDI Deliverable TA6/WP5-2

Jan Knedlik

28.03.2023

Dynamic Disk Caching for Opportunistic Resources

- Goal: Efficiently include opportunistic resources(HPC Clusters, Clouds, Grids, . . .)
- Solution: Introduce a caching service with direct cache access for smart *re-use of data*
  - Mitigate remote data latencies
  - Increase client scalability by lowering bandwith requirements

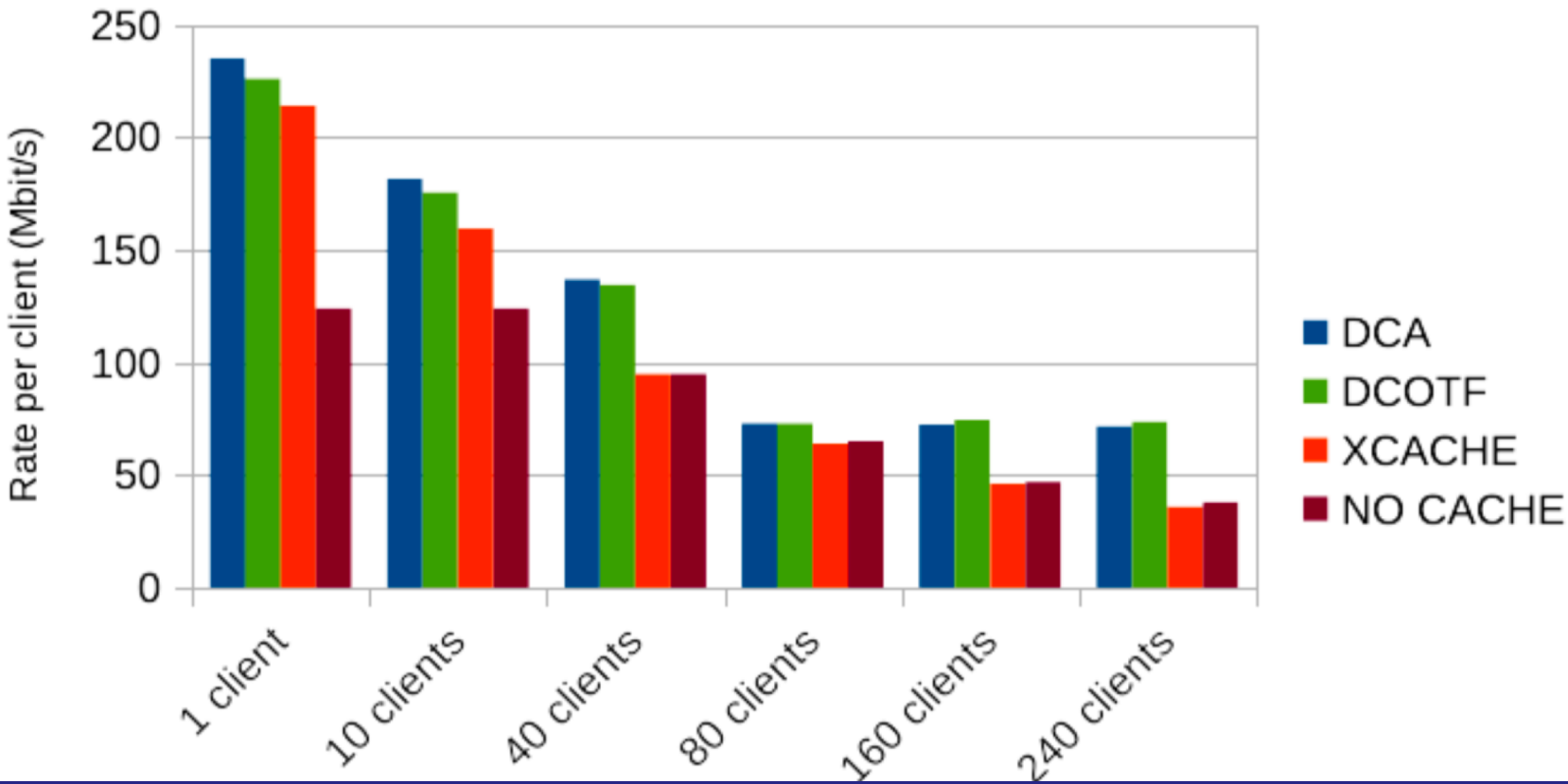
Development of caching solutions for XRootD

- Prior work on caching systems @ GSI: “Disk Cache on the Fly” prototype based on XRootD plug-ins
- Idea layed groundwork for high popularity of caching solutions in the XRootD Community
- Implementation of “XRootD Cache with direct cache access” XCache_dca integrating earlier developments @ GSI now part of base XRootD

-> Which is better?

Comparison of caching solutions

- Preliminary comparisons conducted in close coordination by FIAS Group in context of FIDIUM.



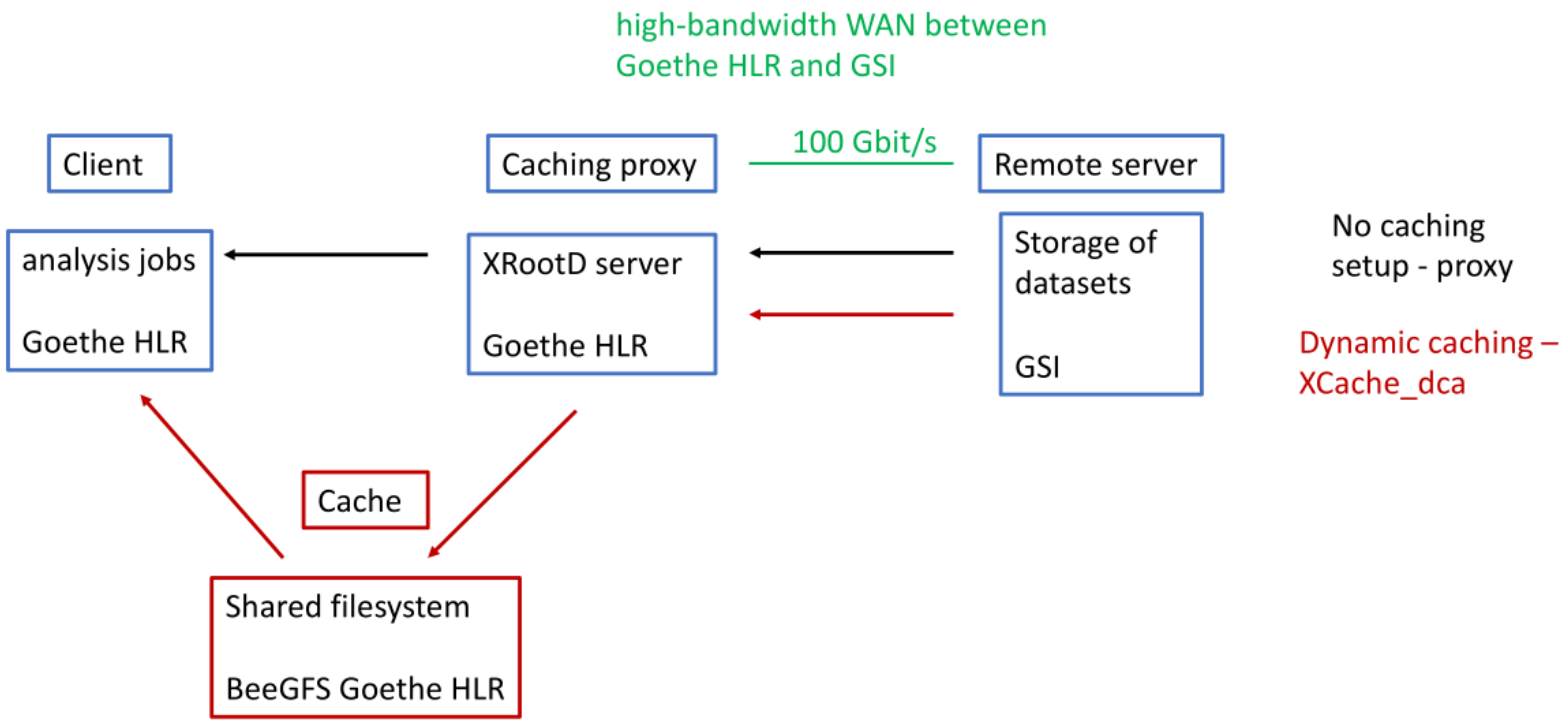
Decision for XCache_dca

Comparable performance, key problems using DCOTF:

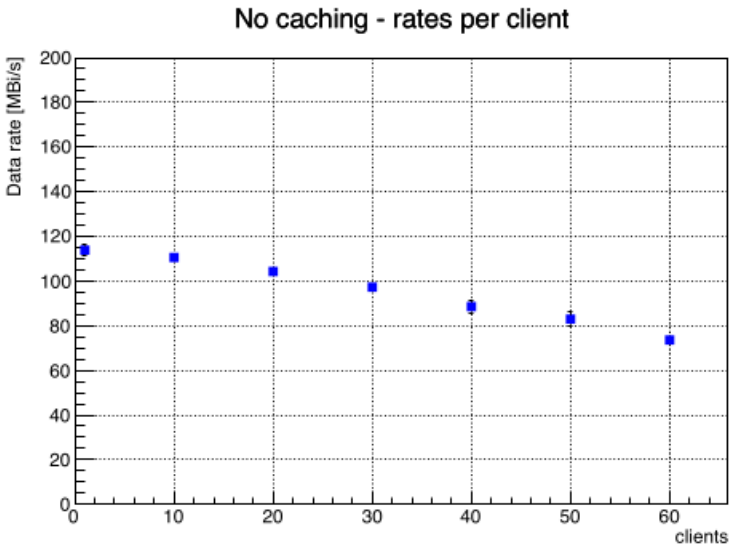
- Needing a dedicated redirector
- Maintainance/Development for DCOTF plugin
- Wrong file permissions in cache (always saved by xrootd user)
- No access records -> client errors when missing that while purging cold cache files
- Does not check for file coherence

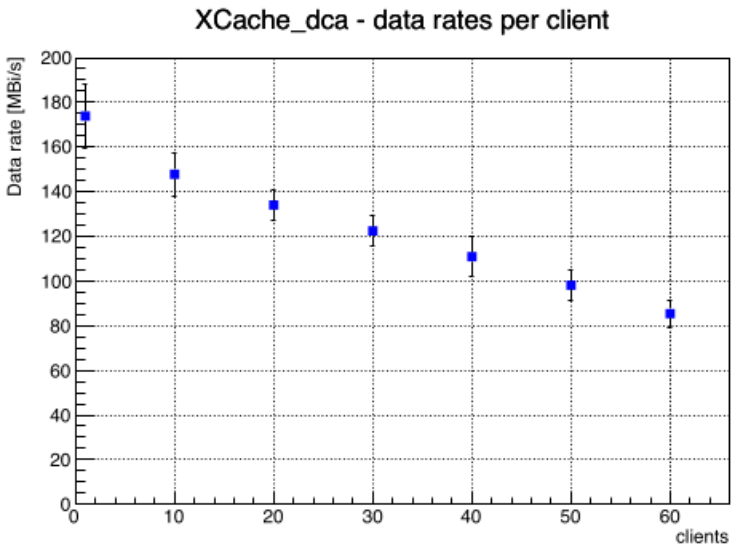
-> XCache_dca

Test setup of XCache_dca @Goethe HLR and GSI



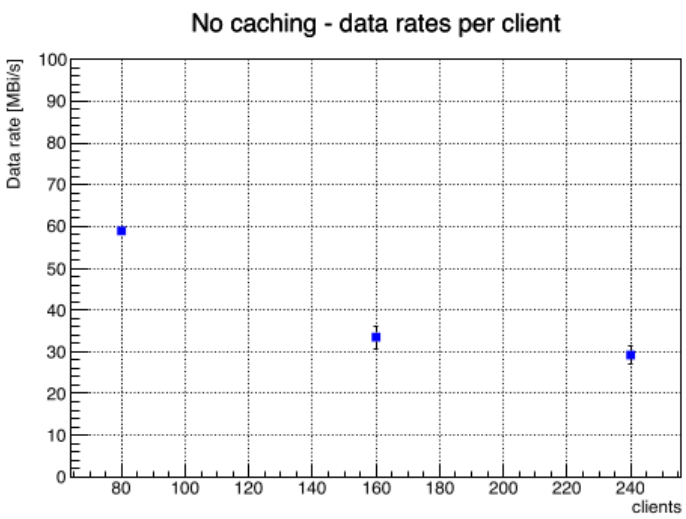
Performance evaluation (uninode)

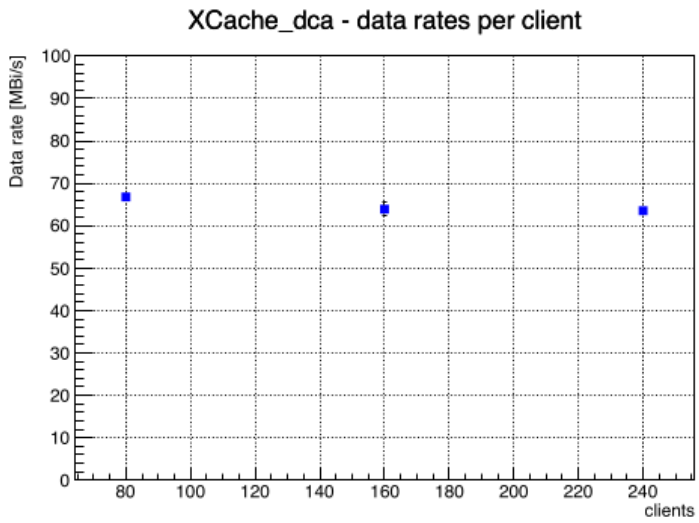




- 1 Node, 60 clients => faster in all cases using XCache_dca

Performance evaluation (multinode)





- 3 Node, 60x3 clients => much better scalability using XCache_dca

Results

- Efficiently include opportunistic resources(HPC Clusters, Clouds, Grids, . . .)
- Efficent *reuse* of data when using XCache_dca as caching service:
  - Mitigate remote data latencies
  - Increase client scalability by lowering bandwith requirements
  - Lower Client runtime when utilizing resource local distributed filesystems
  - Promising: Mature, performant, maintained by XRootD Devs

fin