CESNET

FZÚ

ACAT Conference 2013

Influence of Distributing a Tier-2 Data Storage on Physics Analysis

Jiří Horký[1,2]
(horky@fzu.cz)
Miloš Lokajíček[1], Jakub Peisar[2]
[1]Institute of Physics ASCR, [2]CESNET

17th of May, 2013

Background and Motivation

- FZU is a Tier-2 site, mainly used for ATLAS, ALICE and D0 experiments
  - based in Prague, Czech Republic
  - 4000 CPU cores, 2.5PB of usable disk space
    - DPM storage element for ATLAS and xrootd for ALICE
- decrease of financial support from grants
- increasing demand for capacity from CERN experiments foreseen

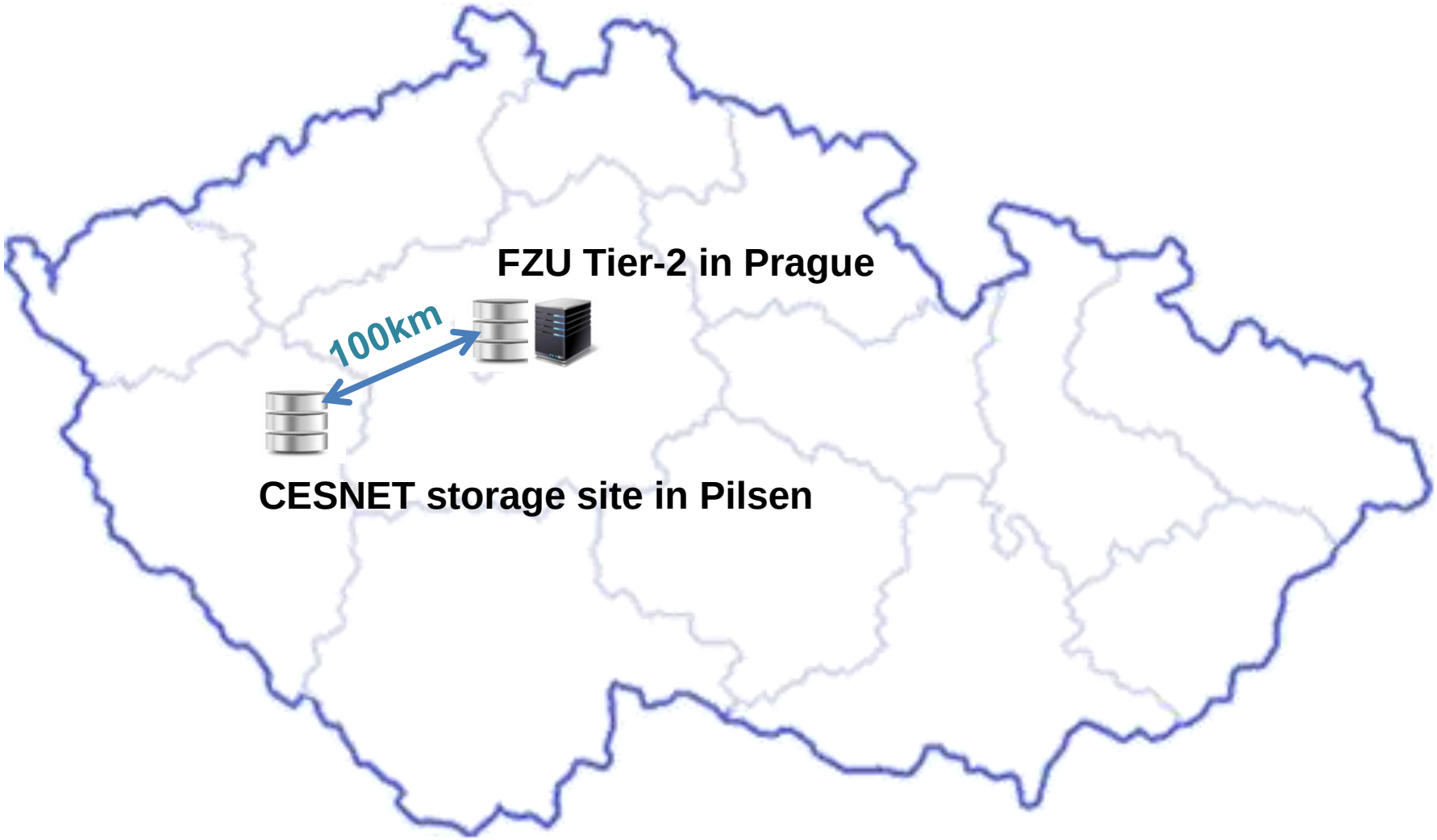
→ novel resource providers must be looked for

Background and Motivation

- New e-infrastructure projects in the Czech Republic
  - CESNET
    - Czech NREN, but not only a plain network provider
    - NGI CZ – computing infrastructure, but with limited resources for HEP experiments
    - new service: data storage facility
      - three distributed HSM based storage sites
      - designed for research and science community

      → opportunity for collaboration

Site Locations



Implementation – Design Choices

- Under which site to publish the storage resources?
  - ATLAS nor ALICE experiments supported on CESNET's computing infrastructure (prague_cesnet_lcg2)

  → another SE under FZU's Tier-2 site (praguelcg2)
  - part of the site operated by someone else - concerns about influence on reliability, monitoring etc.
- Which SE implementation?
  - HSM system (DMF) with 500TB of disk and 3PB of tape space in Pilsen
  - only ~35TB of disk space could be devoted to grid services – SE that could handle tapes needed

  →dCache chosen, gsidcap and gsiftp protocols

Implementation – Details

- FZU<->Pilsen - 10Gbit link with ~3.5ms latency
  - public Internet connection – shared with other institutes within the area
  - dedicated link from FZU to the CESNET's backbone to be installed soon
- Concerns about chaotic use of the HSM system from users (migrations, recalls from/to tape)
  - disk-only spacetoken (ATLASLOCALGROUPDISK) provided for user analysis data
  - tape-only spacetoken (ATLASTAPE) as an "archive" of users' datasets
  - similar setup for Auger

Implementation – ATLAS

- New DDM (ATLAS data management system) endpoint created (PRAGUELCG2_PPSLOCALGROUPDISK)
  - a user selects which endpoint to send the data in ATLAS DaTRI/DQ2 system
- The same Panda (ATLAS job submission system) queue as for the rest of pragulcg2 site
  - transparent job submission for data on the local and the remote SE

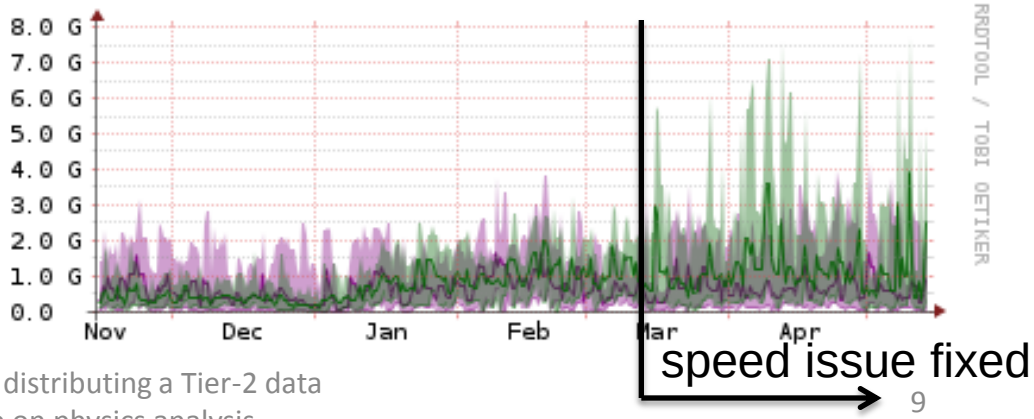
Operational Challenges

- 3rd party service provider to FZU
  - problems with dCache affect FZU's reliability
  - SAM Nagios messages regarding dCache go to FZU's team instead of CESNET, same for GGUS

    – Nagios reconfigured, GGUS still need to be reposted (or receive all the unrelated tickets)
  - CESNET's members were added the possibility to add scheduled downtimes in GOCDB (but for the whole site)
- Some trust necessary

Impact on User Analysis

- Initial user analysis tests very slow
  - 14% job efficiency in comparison with 71% against local storage with a single job
  - manual iperf showed 30Mbps throughput only
- Cisco FWSM module identified to be the issue
  - even with CPU load close to 0 – HW filtering limit!
  - only in effect on a public Internet link – the one to dCache
  - 2.5Gbit hard limit in one direction, much less on a single connection
  - moved away from FWSM to ACLs



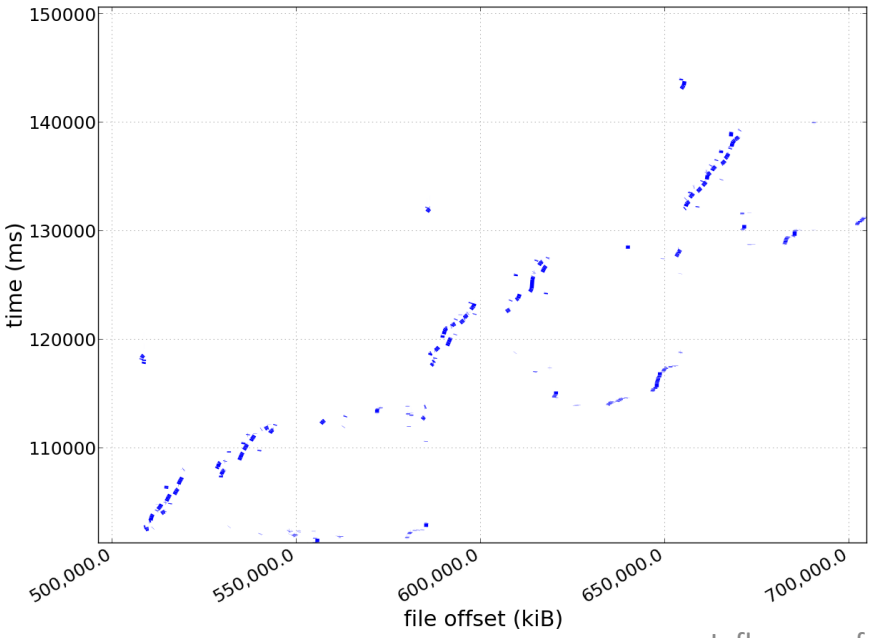
Impact on User Analysis

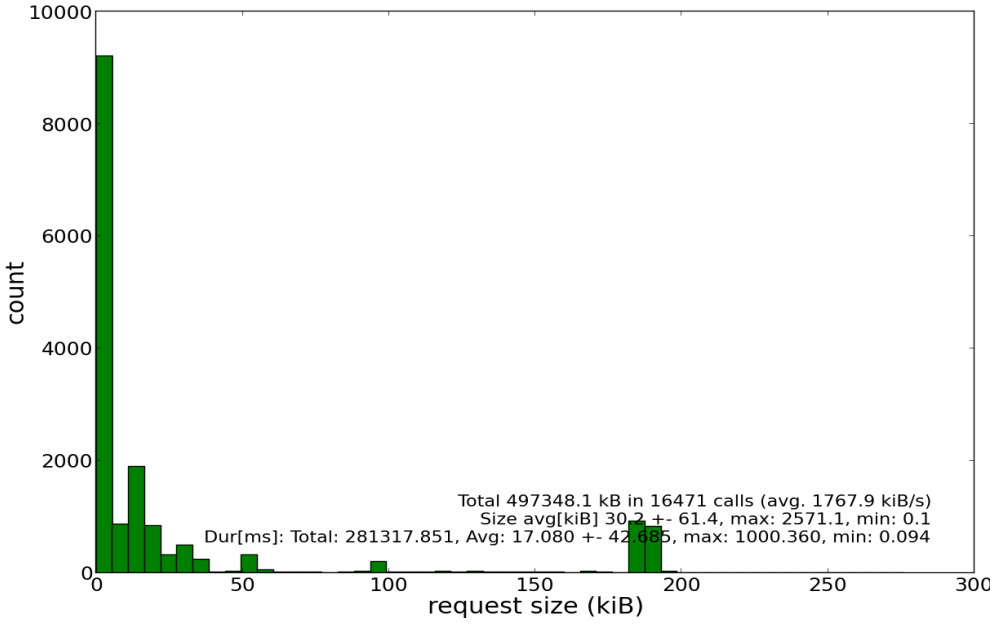
- Still concerned about job efficiency due to network latency
  - ~3.5 ms instead of 0.2 ms locally
  - line bandwidth obvious limitation as well
- Several factors to be tested
  - TTreeCache on/off
  - dCap read ahead (DCACHE_RAHEAD) on/off
  - number of roundtrips & network bandwidth used

Impact on User Analysis

- An example ATLAS analysis job selected and examined
  - IO access pattern
  - number of read requests, size of the requests, sequential/random manner

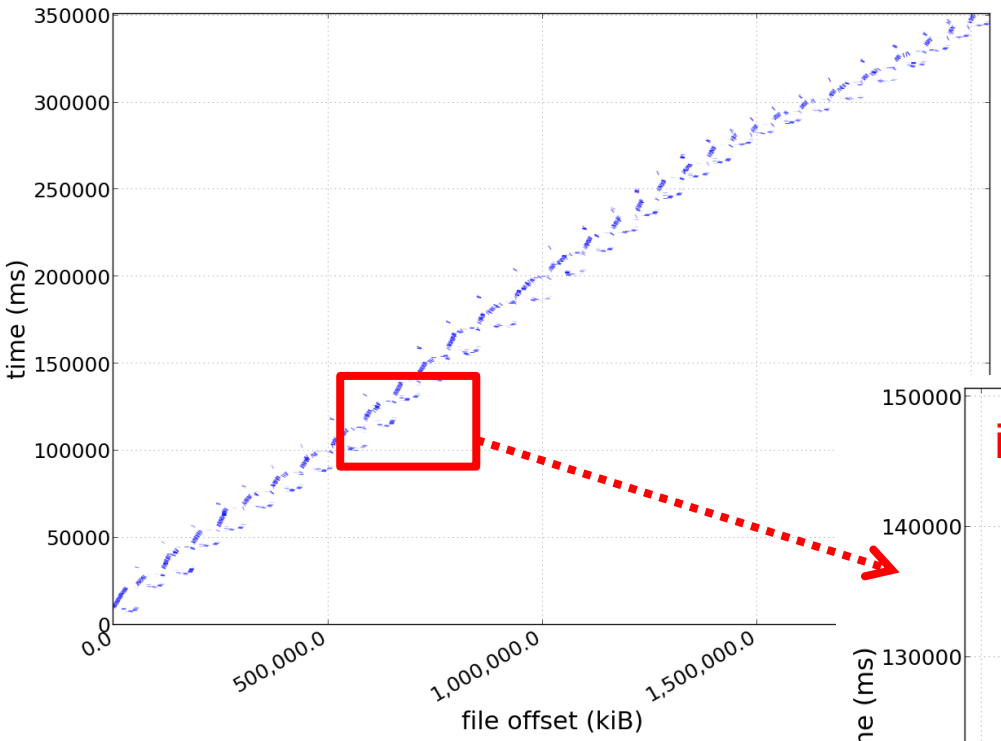
  -> access pattern diagram, size histogram of the requests

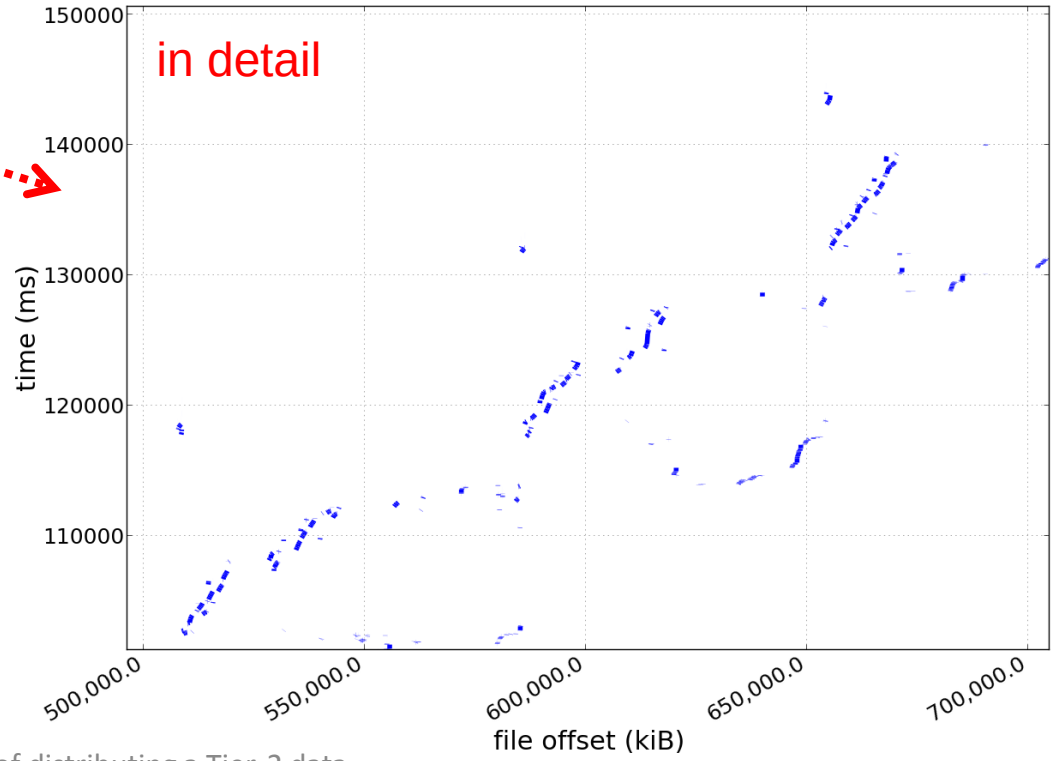




Impact on User Analysis

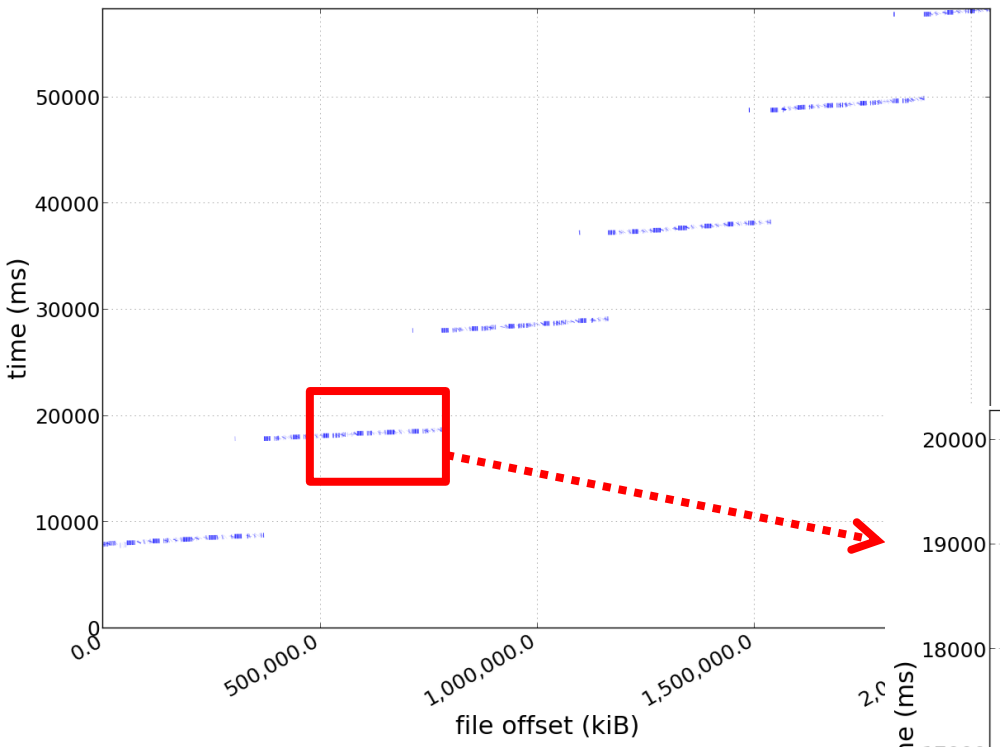
TTreeCache off



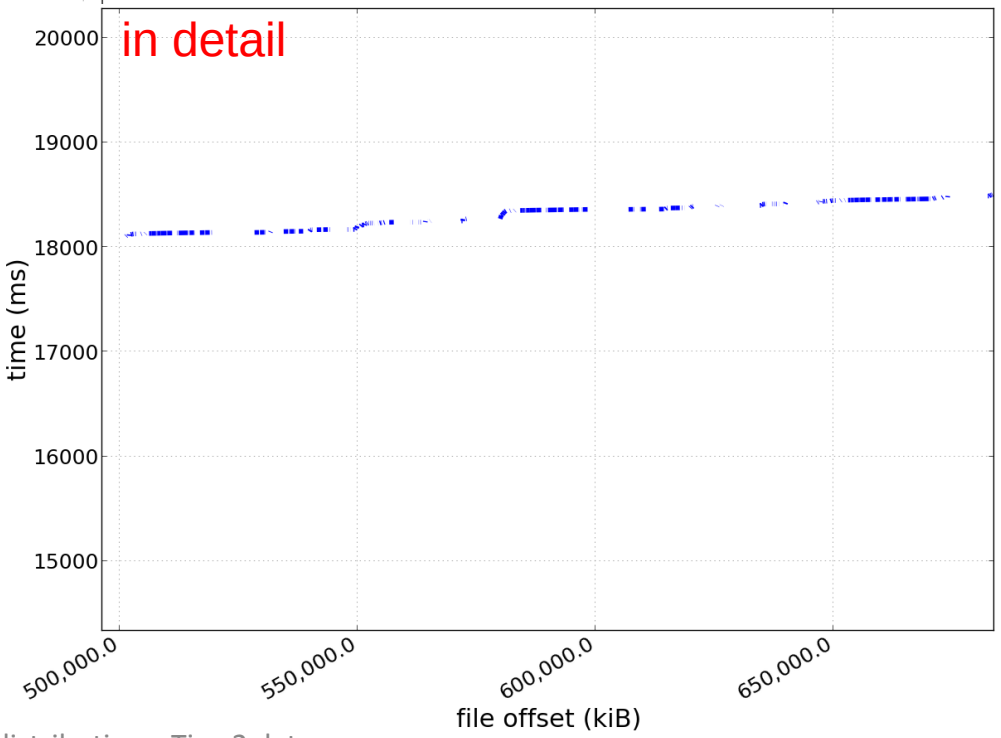


485MiB out of 2GB read
15471 IO calls
median size 2.6KiB, avg size 30KiB

Impact on User Analysis



TTreeCache on

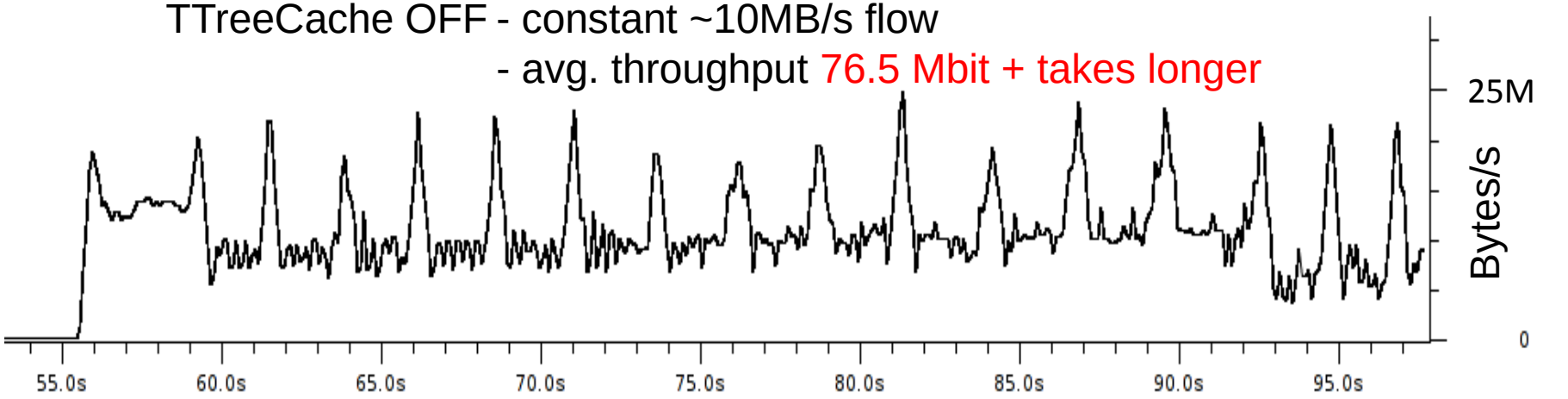


566MiB out of 2GB read – 16% more
2311 IO calls
median size 90KiB, avg size 251KiB

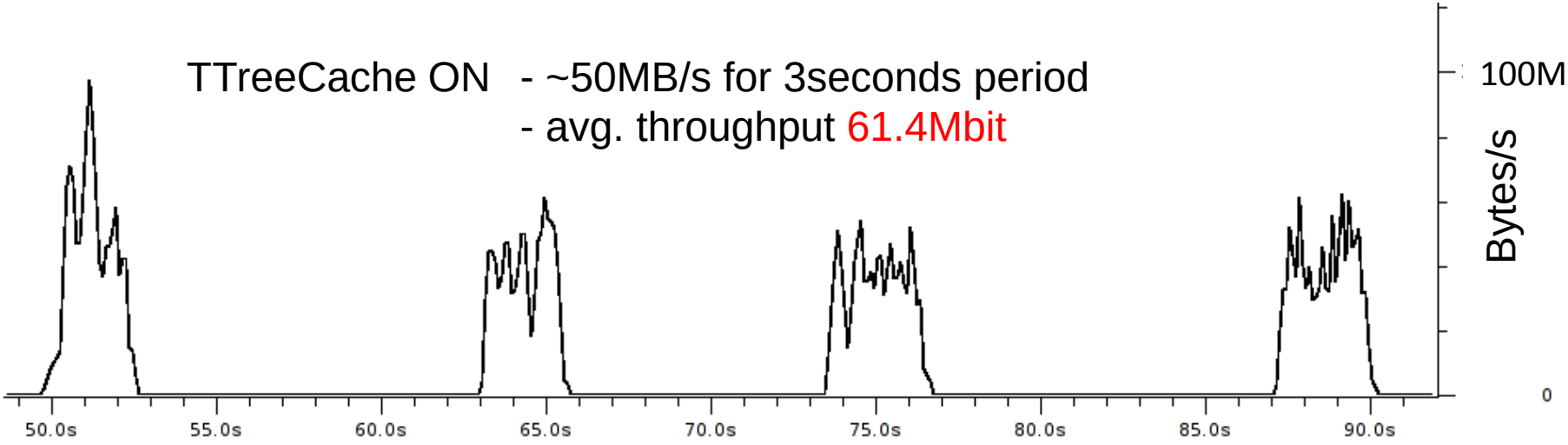
Impact on User Analysis

- Network utilization using dCap, no RA cache tuning

TTreeCache OFF - constant ~10MB/s flow
- avg. throughput 76.5 Mbit + takes longer



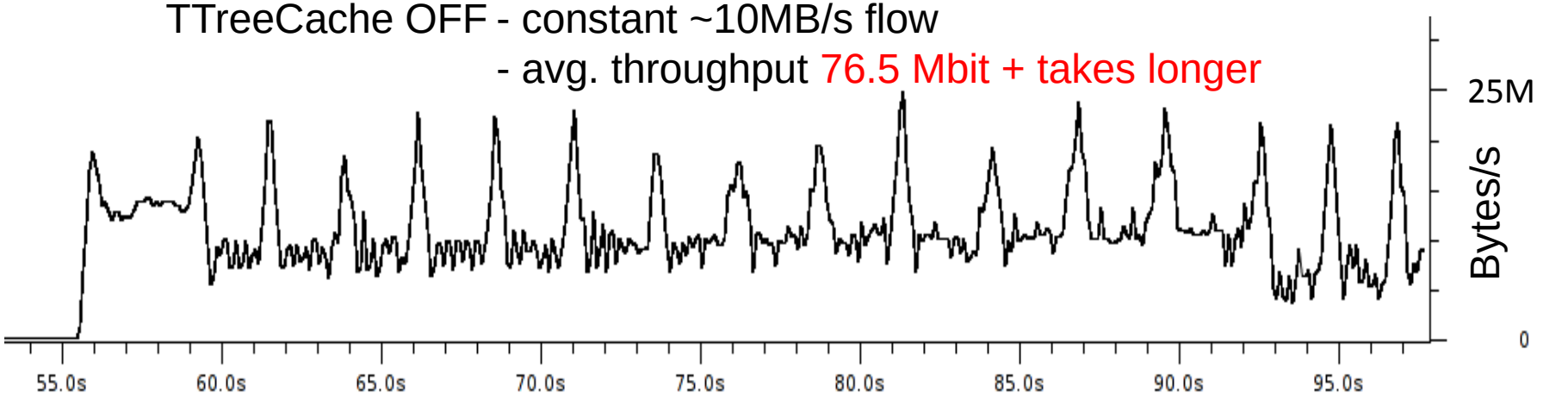
TTreeCache ON - ~50MB/s for 3seconds period
- avg. throughput 61.4Mbit



Impact on User Analysis

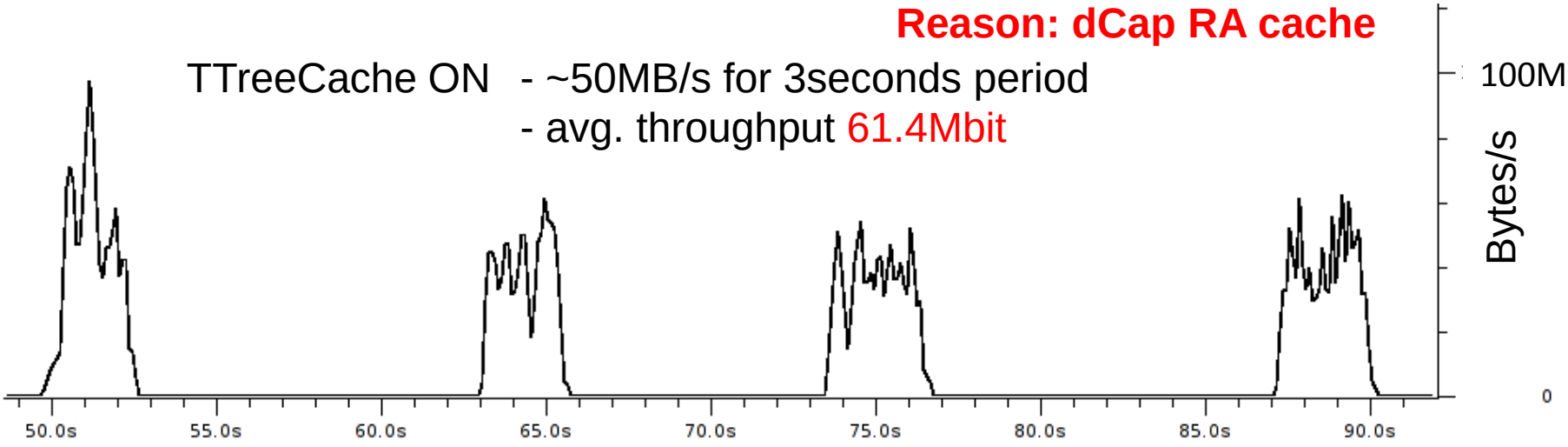
- Network utilization using dCap, no RA cache tuning

TTreeCache OFF - constant ~10MB/s flow
- avg. throughput 76.5 Mbit + takes longer



Reason: dCap RA cache

TTreeCache ON - ~50MB/s for 3seconds period
- avg. throughput 61.4Mbit



Impact on User Analysis

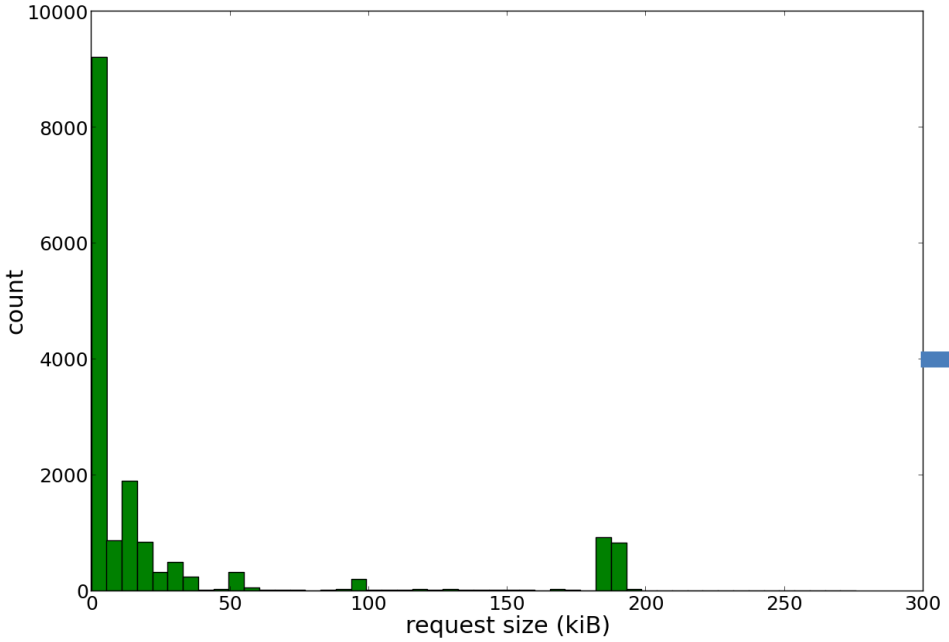
dCap ReadAhead cache in ROOT:

- enabled by default, 128KiB, cache size the same
  - actually, can not be really disabled!
  - but can be set to really small value - DCACHE_RA_BUFFER env. variable
- quite strict behavior
  - for every IO not within 128KiB from last read, at least 128KiB transferred
  - bigger requests split to 256KiB
  - performance killer for small random reads – TTreeCache OFF case

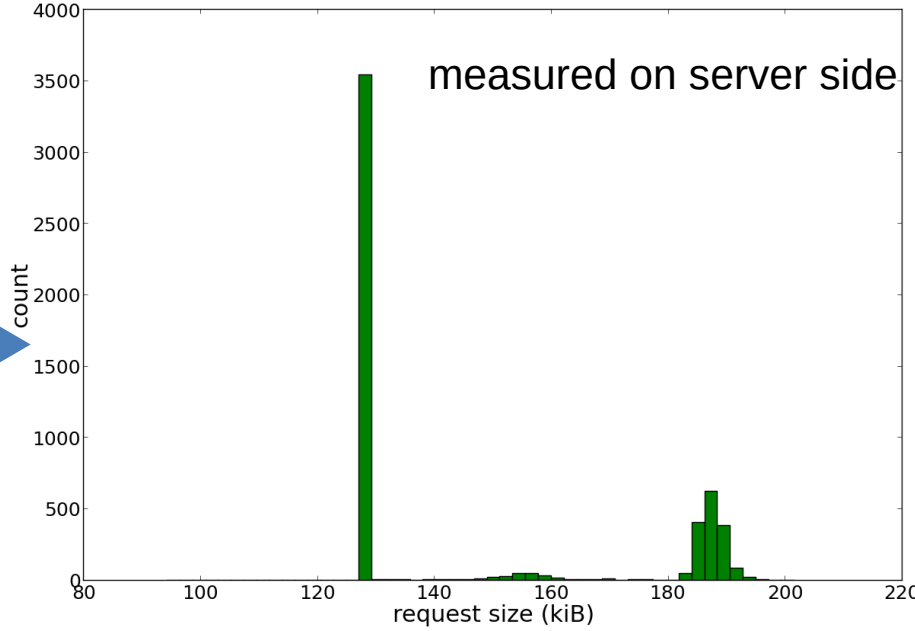
Impact on User Analysis

dCap ReadAhead cache in ROOT – TTreeCache OFF

Local access (no RA)



dCap with default 128KiB RA



485MiB in 16471 calls

770MiB in 5376 calls

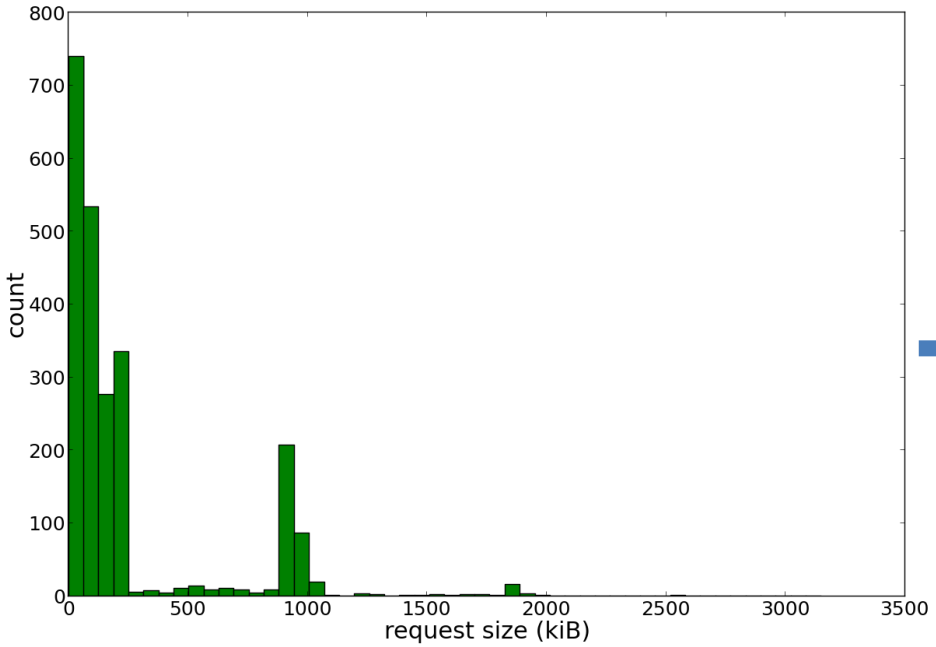
→ 3x less IO reqs, no IO vectorization

→ 58% more data read

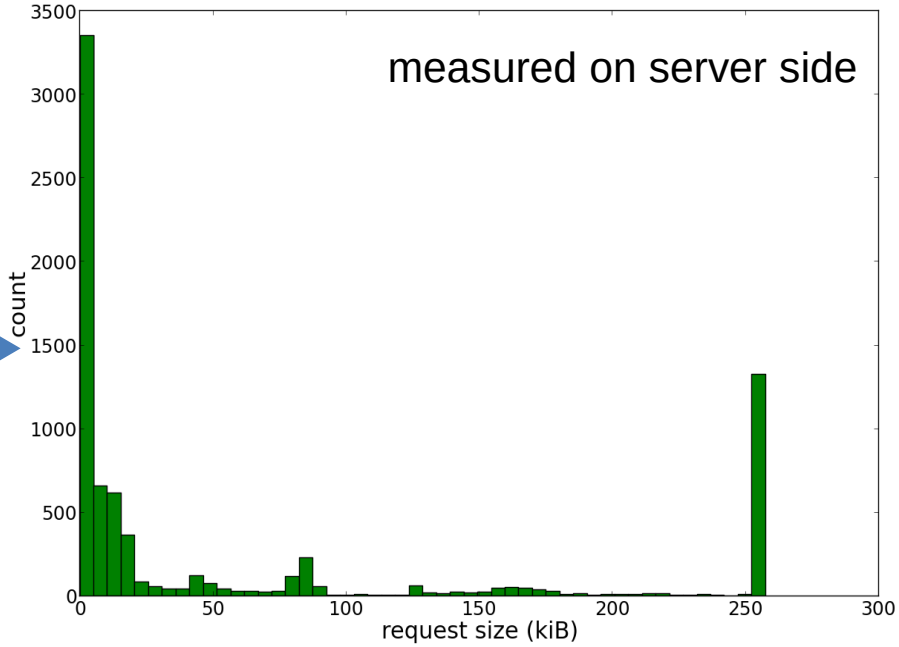
Impact on User Analysis

dCap ReadAhead cache in ROOT - TTreeCache ON

Local access (no RA)



dCap with default 128KiB RA



566MiB in 2311 calls

490MiB in 7772 calls

– non cache reads are vectorized

Side by Side Comparison

The same analysis job run under different conditions – reading 100 ~2GB files

remote/local	Method	TTreeCache	dCap RA	events/s (%)	Bytes transferred %	CPU Efficiency
local	rfio	ON	N/A	100%	117%	98,9%
local	rfio	OFF	N/A	74%	100%	72,7%
remote	dCap	ON	off	76%	117%	75,0%
remote	dCap	ON	128KiB	75%	101%	73,5%
remote	dCap	OFF	off	46%	100%	46,9%
remote	dCap	OFF	128KiB	54%	159%	59,7%

Side by Side Comparison

remote/local	Method	TTreeCache	dCap RA	events/s (%)	Bytes transferred %	CPU Efficiency
local	rfio	ON	N/A	100%	117%	98,9%
local	rfio	OFF	N/A	74%	100%	72,7%
remote	dCap	ON	off	76%	117%	75,0%
remote	dCap	ON	128KiB	75%	101%	73,5%
remote	dCap	OFF	off	46%	100%	46,9%
remote	dCap	OFF	128KiB	54%	159%	59,7%

- TTreeCache helps a lot – both for local and for remote transfers
  - efficient coupling with dCap RA mechanism – almost no extra bandwidth overhead

Side by Side Comparison

remote/local	Method	TTreeCache	dCap RA	events/s (%)	Bytes transferred %	CPU Efficiency
local	rfio	ON	N/A	100%	117%	98,9%
local	rfio	OFF	N/A	74%	100%	72,7%
remote	dCap	ON	off	76%	117%	75,0%
remote	dCap	ON	128KiB	75%	101%	73,5%
remote	dCap	OFF	off	46%	100%	46,9%
remote	dCap	OFF	128KiB	54%	159%	59,7%

- TTreeCache helps a lot – both for local and for remote transfers
  - efficient coupling with dCap RA mechanism – almost no extra bandwidth overhead
- dCap RA can cause considerable bandwidth overhead without TTreeCache

Side by Side Comparison

remote/local	Method	TTreeCache	dCap RA	events/s (%)	Bytes transferred %	CPU Efficiency
local	rfio	ON	N/A	100%	117%	98,9%
local	rfio	OFF	N/A	74%	100%	72,7%
remote	dCap	ON	off	76%	117%	75,0%
remote	dCap	ON	128KiB	75%	101%	73,5%
remote	dCap	OFF	off	46%	100%	46,9%
remote	dCap	OFF	128KiB	54%	159%	59,7%

- TTreeCache helps a lot – both for local and for remote transfers
  - efficient coupling with dCap RA mechanism – almost no extra bandwidth overhead
- dCap RA can cause considerable bandwidth overhead without TTreeCache
- TTreeCached remote jobs faster than local ones without the cache

Conclusion

- Operating CESNET's dCache SE under FZU's Tier-2 site works well
- Several issues identified and fixed (FW, SW issues)
- Proper job settings needed (TTreeCache) to ensure reasonable performance and link utilization

FZÚ

Thank you for your attention.
Questions?

Acknowledgements:

This work was supported by the CESNET Storage group members as well as by the computing center team at FZU.

The work was supported by grant INGO LG13031.

Jiří Horký

(horky@fzu.cz)