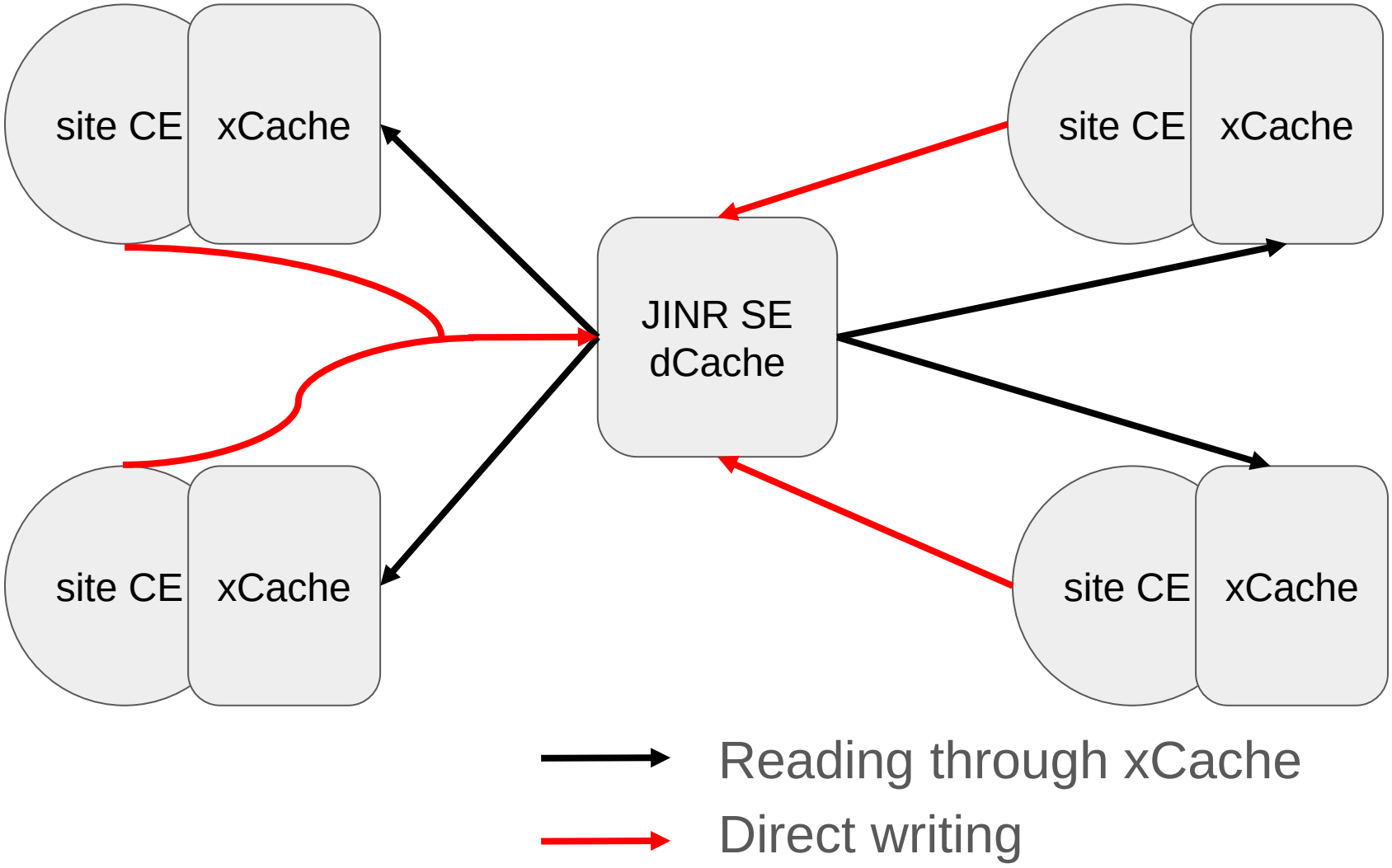
# xCache on RU-DataLake deployment and testing with ATLAS tools

Andrey Zarochentsev, Aleksandr Alekseev, Stephane Jezequel, Andrey Kiryanov

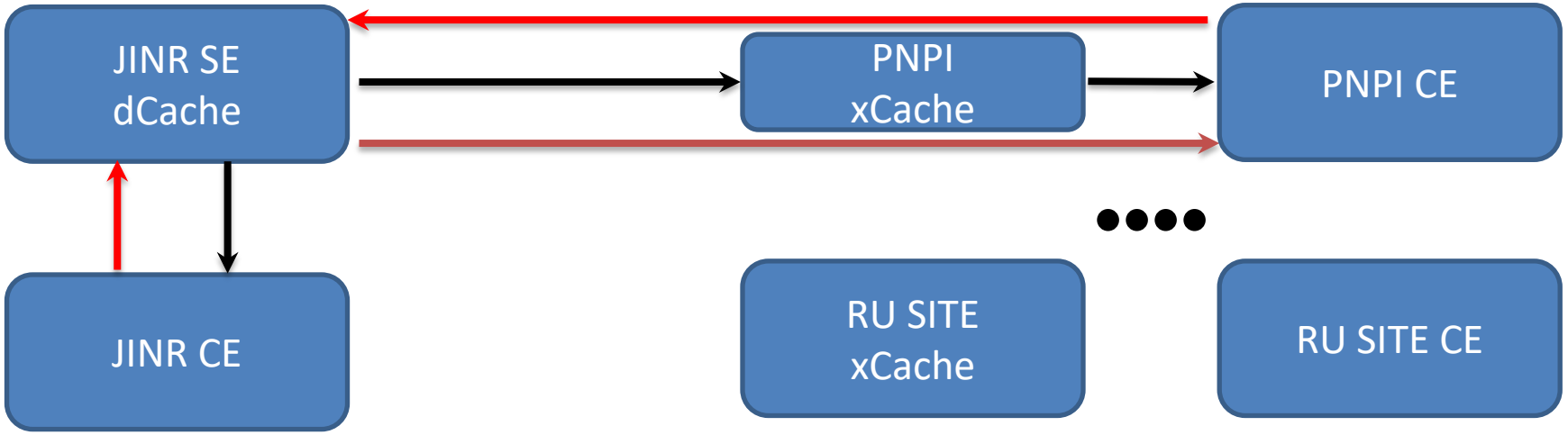
# Russian DataLake 2019



# Goal of tests

1. To check the potential gain by including xCache for a 'small' diskless sites which are 5-10 ms away from SE
2. To check HC instrumentation for testing local DataLikes on Russian DataLikes example
3. To check monitoring systems for DataLake tests

# Plan of tests




Submitted tests:

1. Synthetic tests from Worker Nodes by hand and through Cream-CE
2. Two types of standard ATLAS tests through HammerCloud:
   a. Copy2Scratch
   b. Directaccess

Reading through xCache

Direct reading

Direct writing

# Authorization

- PNPI xCache - > JINR SE: GSI authorization by local gridmapfile on JINR SE. “DN=/C=RU/O=RDIG/OU=hosts/OU=pnpi.nw.ru/CN=v008.pnpi.nw.ru” associated with ATLAS Production role
- PNPI WN -> PNPI xCache: GSI authorization by VOMS (ATLAS,ALICE)
- PNPI UI -> JINR CE,SE; PNPI CE (for local tests): GSI authorization by VOMS ALICE & ATLAS Production roles
- PNPI,JINR WN -> JINR SE: GSI authorization by VOMS ALICE & ATLAS Production roles
- Hammer Cloud -> ALL: GSI authorization by VOMS (ATLAS)

# xCache settings

/etc/xrootd/xrootd-cache.cfg

```
{
ofs.osslib libXrdPss.so
all.export /
all.role server
oss.localroot /data/namespace
oss.cache public /data/xrootd
pss.origin =
pss.cachelib libXrdFileCache.so
pss.config streams 256
pfc.ram 6g
pfc.blocksize 512k
pfc.prefetch 0
pfc.trace info
xrootd.seclib /usr/lib64/libXrdSec.so
sec.protocol unix
sec.protocol /usr/lib64 gsi -certdir:/etc/grid-security/certificates -cert:/etc/grid-security/hostcert.pem \
                -key:/etc/grid-security/hostkey.pem  \
                -crl:1 \
                -authzfun:/usr/local/lib/libXrdLcmaps.so\
                -authzfunparms:lcmapscfg=/etc/lcmaps/lcmaps.db,loglevel=4\
                -gmapopt:10 \
                -gmapto:0\
                -d:1

http.secxtractor /usr/local/lib/libXrdLcmaps.so
sec.protbind localhost unix
sec.protbind * gsi
}
```

Add an external library for VOMS authorization:

https://github.com/opensciencegrid/xrootd-lcmaps

# xCache settings (LCMAPS)

/etc/lcmaps/lcmaps.db:

```
{
# where to look for modules
path = /usr/lib64/lcmaps
good = "lcmaps_dummy_good.mod"
bad  = "lcmaps_dummy_bad.mod"
verifyproxy = "lcmaps_verify_proxy.mod"
        "--allow-limited-proxy"
        " -certdir /etc/grid-security/certificates"
        "--discard_private_key_absence"
vomslocalaccount = "lcmaps_voms_localaccount.mod"
        " -gridmapfile /etc/grid-security/grid-mapfile"
xrootd_policy:
verifyproxy -> vomslocalaccount
vomslocalaccount -> good | bad
}
```

xCache (and probably any xrootd service) does not actually switch UNIX users. Therefore we use *nobody* user as a stub.

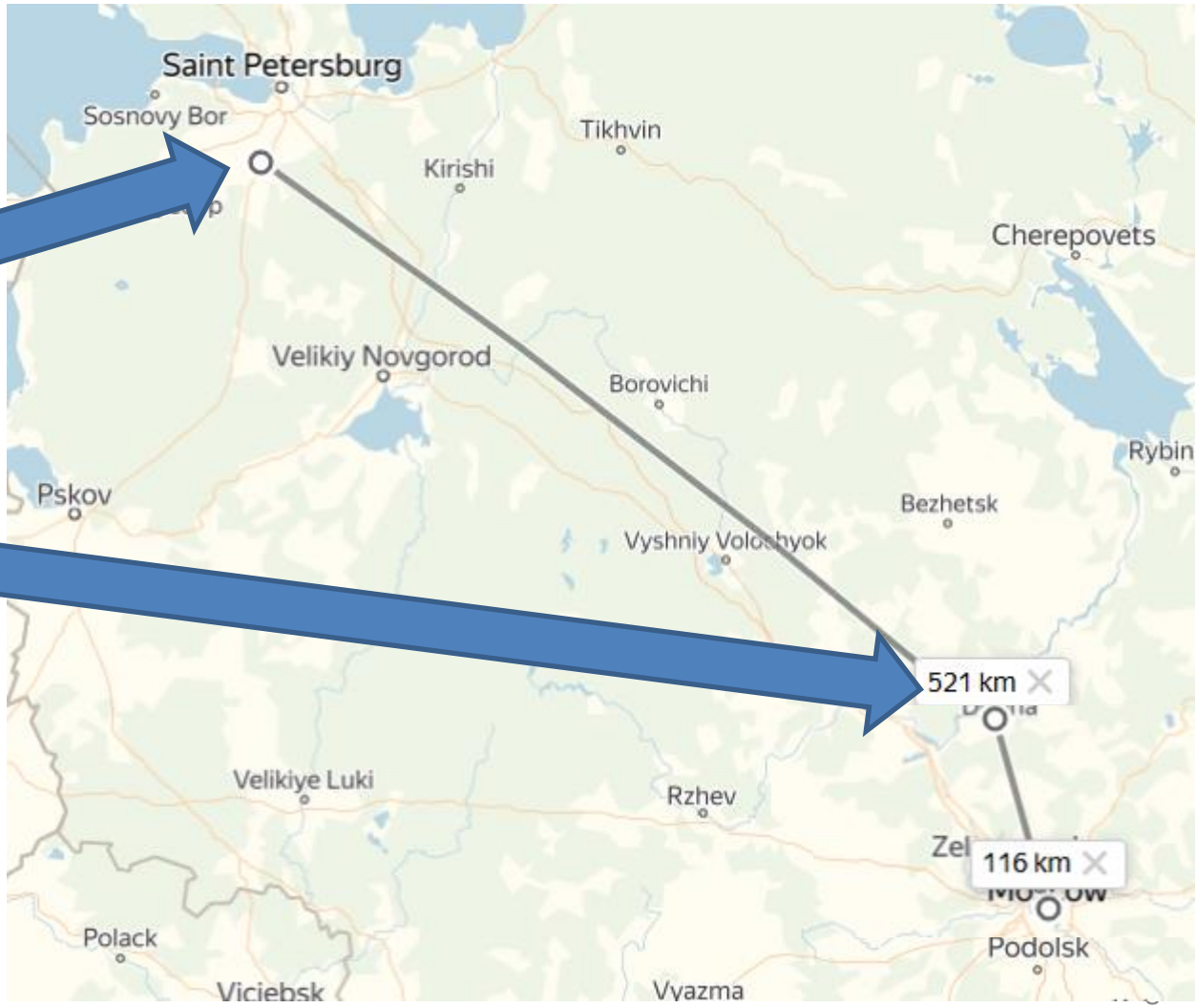
/etc/grid-security/grid-mapfile

```
{
"/dteam/Role=NULL/Capability=NULL" nobody
"/atlas/Role=production/Capability=NULL" nobody
"/atlas/Role=production" nobody
"/atlas/Role=lcgadmin/Capability=NULL" nobody
"/atlas/Role=lcgadmin" nobody
"/atlas/Role=NULL/Capability=NULL" nobody
"/atlas/Role=NULL" nobody
"/atlas" nobody
"/alice/Role=production/Capability=NULL" nobody
"/alice/Role=production" nobody
"/alice/Role=lcgadmin/Capability=NULL" nobody
"/alice/Role=lcgadmin" nobody
"/alice/Role=NULL/Capability=NULL" nobody
"/alice/Role=NULL" nobody
"/alice" nobody
}
```

# Main points

Petersburg Nuclear Physics Institute

JINR



# Technical characteristics

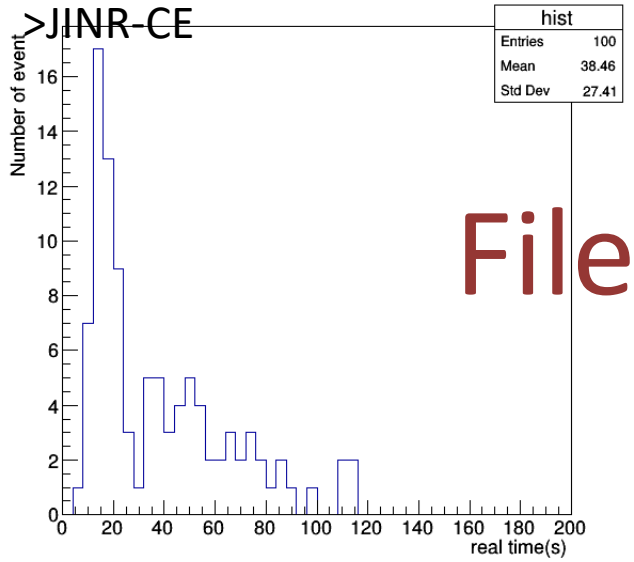
- Work node JINR: 8 cores, Xeon E5420, 16GB RAM, 8.74 HEP-SPEC06 per-Core
- Work node PNPI: 8 cores, Xeon E5-2680, 32GB RAM (VM)
- Local network JINR (SE<->WN) 1Gb/s
- Local network PNPI (SE<->WN) 10Gb/s
- Network IPv4,6 JINR->PNPI: Latency ~5ms
- Network IPv4,6 PNPI->JINR: Latency ~10ms
- Network IPv4,6 JINR->PNPI: Throughput ~1Gb/s
- Network IPv4,6 PNPI->JINR: Throughput ~1,5Gb/s

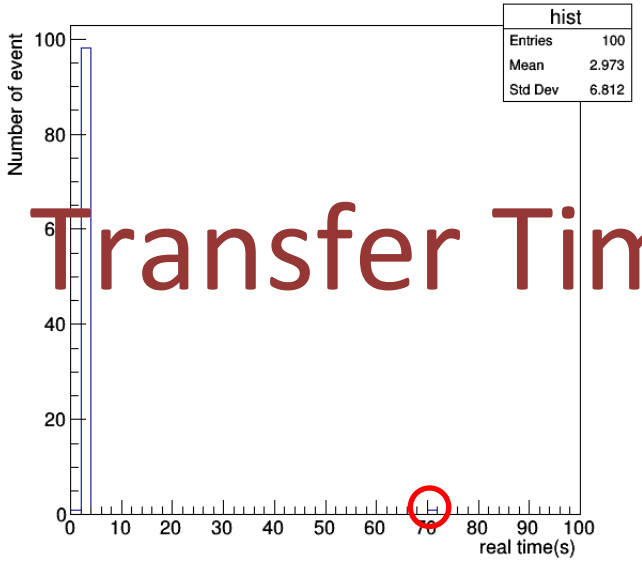
# Local tests result: copy from JINR-SE 1.9 GB root file (100 times)

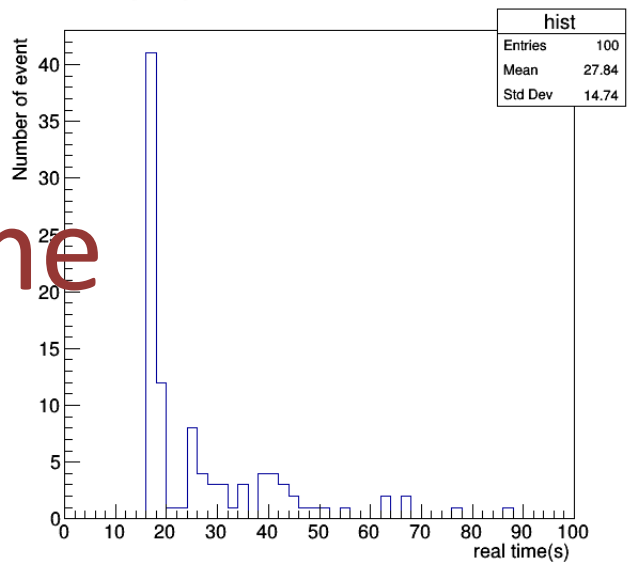
JINR-SE->PNPI-CE

JINR-SE->xCache->PNPI-CE

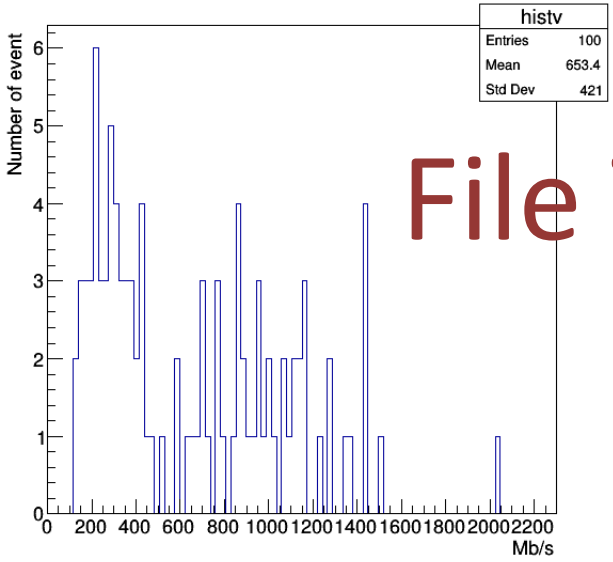
JINR-SE->JINR-CE

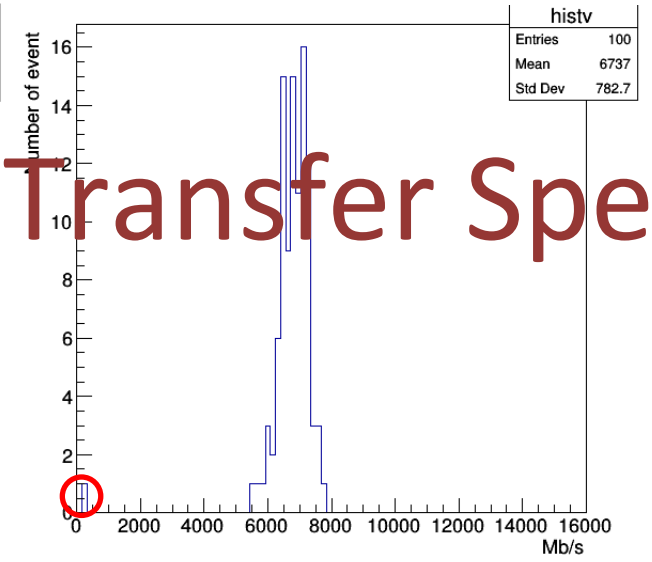


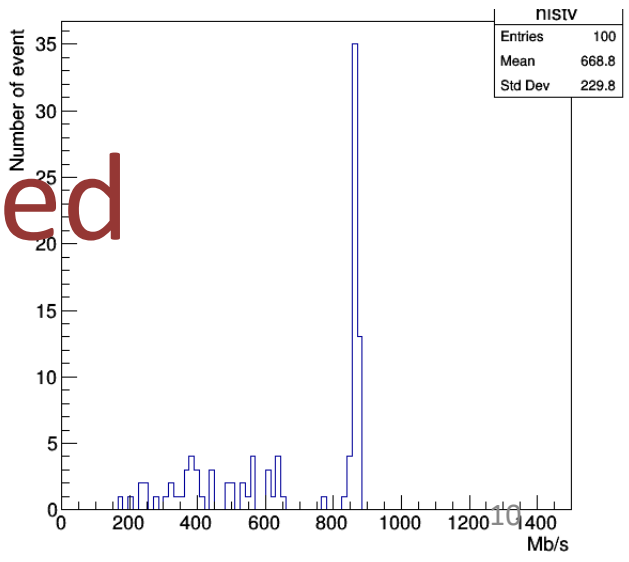




File Transfer Time




File Transfer Speed

# Local tests result: copy from JINR-SE 1.9 GB root file (100 times)

- Mean FTS PNPI – Direct-SE: 650±40 Mb/s
  - < 1Gb/s
  - Time 38s
- Mean FTS PNPI – xCache-SE: 6700±700 Mb/s
  - One hit on 219 Mb/s, other hits with minimal deviation
  - Time 2s – We have 95% gain in time
- Mean FTS JINR – SE: 660±220 Mb/s
  - < 1Gb/s, large deviation

# Hammer cloud settings for tests

- HC templates for tests:
  - 1099 (copy2scratch), 1100 (direct_access_lan)
  - Category: stress
  - Jobtemplate: CeleryProd/ProdTrans/digi-reco_Athena.21.0.53
  - ATLAS derivation jobs with high IO access
  - NUM DATASETS PER BULK: 1, MIN QUEUE DEPTH: 4 MAX RUNNING JOBS: 8 (for JINR_UCORE-TEST we changed tests)
  - Duration time of tests: 2 days
- Panda Queues:
  - JINR_UCORE-TEST (JINR-LCG2)
  - PNPI-TEST (ru-pnpi_NoXCACHE)
  - PNPI_XCACHE-TEST (ru-PNPI_XCACHE)

# HammerCloud tests

| state | id | host | clouds | start time (CET) | end time (CET) | total jobs |
|---|---|---|---|---|---|---|
| completed | 20146370 | hammercloud-ai-11 | RU_PROD | 24/9/2019 15:00 | 26/9/2019 15:00 | 411 |

| Site | S | R | C | F | Eff | T |
|---|---|---|---|---|---|---|
| PNPI_XCACHE-TEST | 4 | 4 | 204 | 0 | 1.00 | 212 |
| PNPI-TEST | 5 | 1 | 186 | 2 | 0.99 | 194 |
| JINR_UCORE-TEST | 1 | 0 | 4 | 0 | 1.00 | 5 |
| **Site** | **S** | **R** | **C** | **F** | **Eff** | **T** |

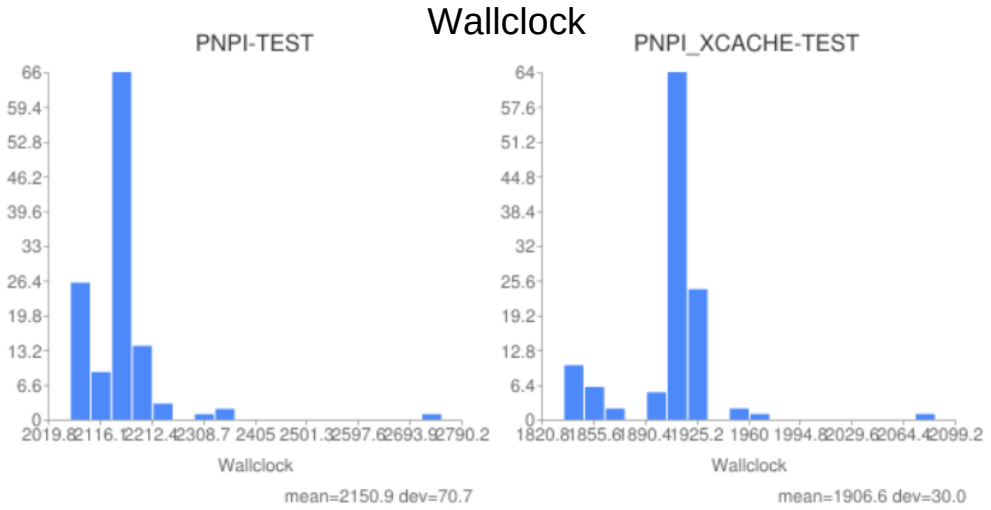
- Test number 20146370 from Template 1099 (copy2scratch)

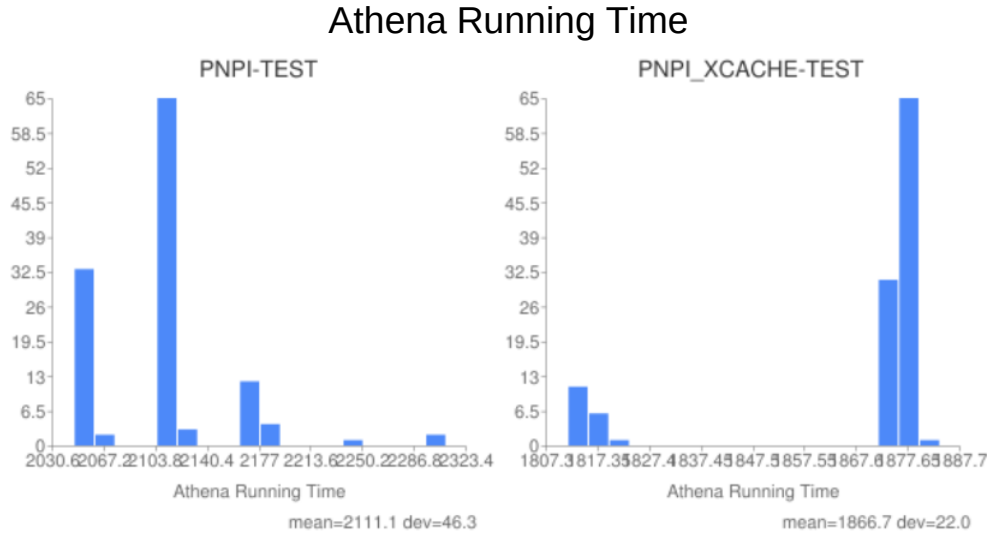
| state | id | host | clouds | start time (CET) | end time (CET) | total jobs |
|---|---|---|---|---|---|---|
| completed | 20146182 | hammercloud-ai-11 | RU_PROD | 19/9/2019 12:00 | 21/9/2019 12:00 | 254 |

| Site | S | R | C | F | Eff | T |
|---|---|---|---|---|---|---|
| PNPI_XCACHE-TEST | 5 | 0 | 115 | 0 | 1.00 | 120 |
| PNPI-TEST | 5 | 0 | 122 | 2 | 0.98 | 129 |
| JINR_UCORE-TEST | 0 | 0 | 4 | 1 | 0.80 | 5 |
| **Site** | **S** | **R** | **C** | **F** | **Eff** | **T** |

- Test number 20146182 from Template 1100 (direct access)
- Weak statistics from JINR-CE for both tests (local problem with JINR-TEST-CE)

# HammerCloud test results - N20146182 from Template 1100 (direct access)



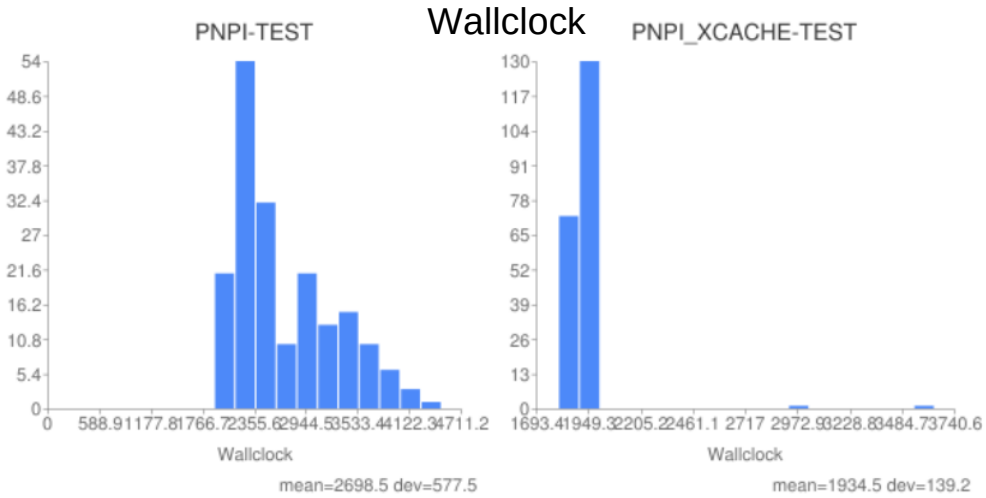




**Wallclock**:
Direct mean time = 2150s ± 70s
xCache mean time = 1906s ± 30s
Difference ~ 250s, ~12%

**Download of input files time:**
Direct mean time = 12s
xCache mean time = 13s

**Athena Run Time:**
Direct mean time = 2111±46s
xCache mean time = 1856±22s
Difference ~ 255s, ~12%

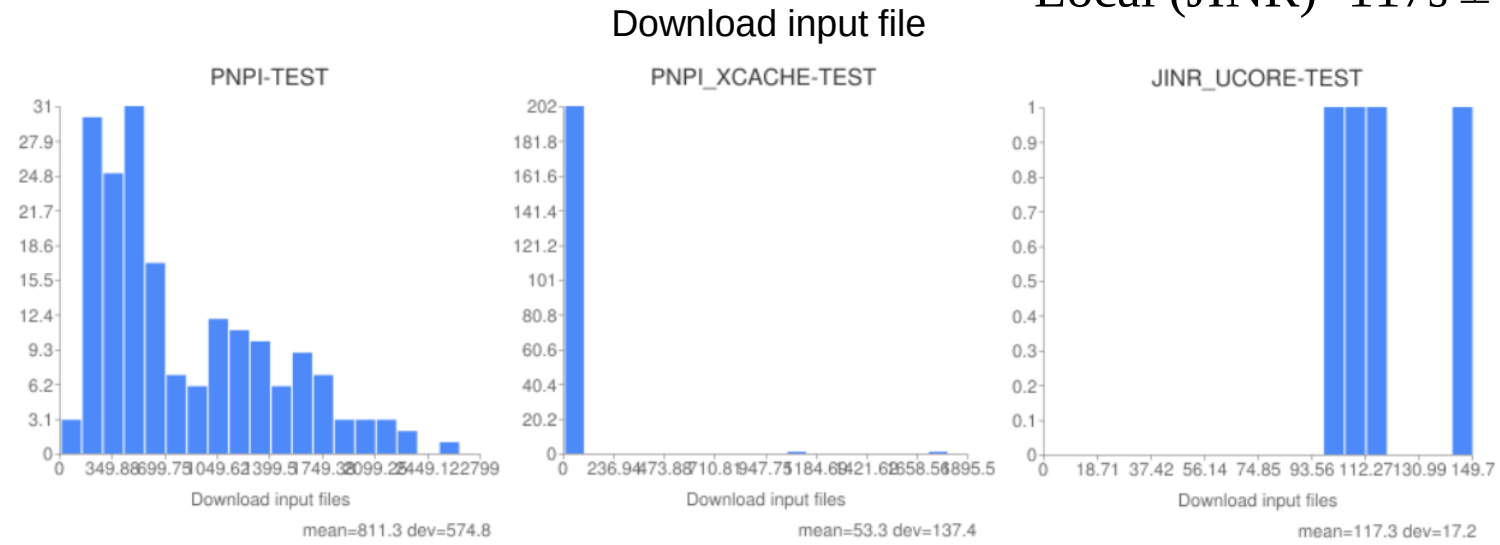
# HammerCloud test results - N20146370 from Template 1099 (copy2scratch)




**Wallclock**:
Direct mean time = 2698s ± 577s
xCache mean time = 1934s ± 139s
Difference ~ 770s, ~30%

**Download input files time:**
Direct mean time = 811s ± 574s
xCache mean time = 53s ± 137s
Difference ~ 770s, ~95%

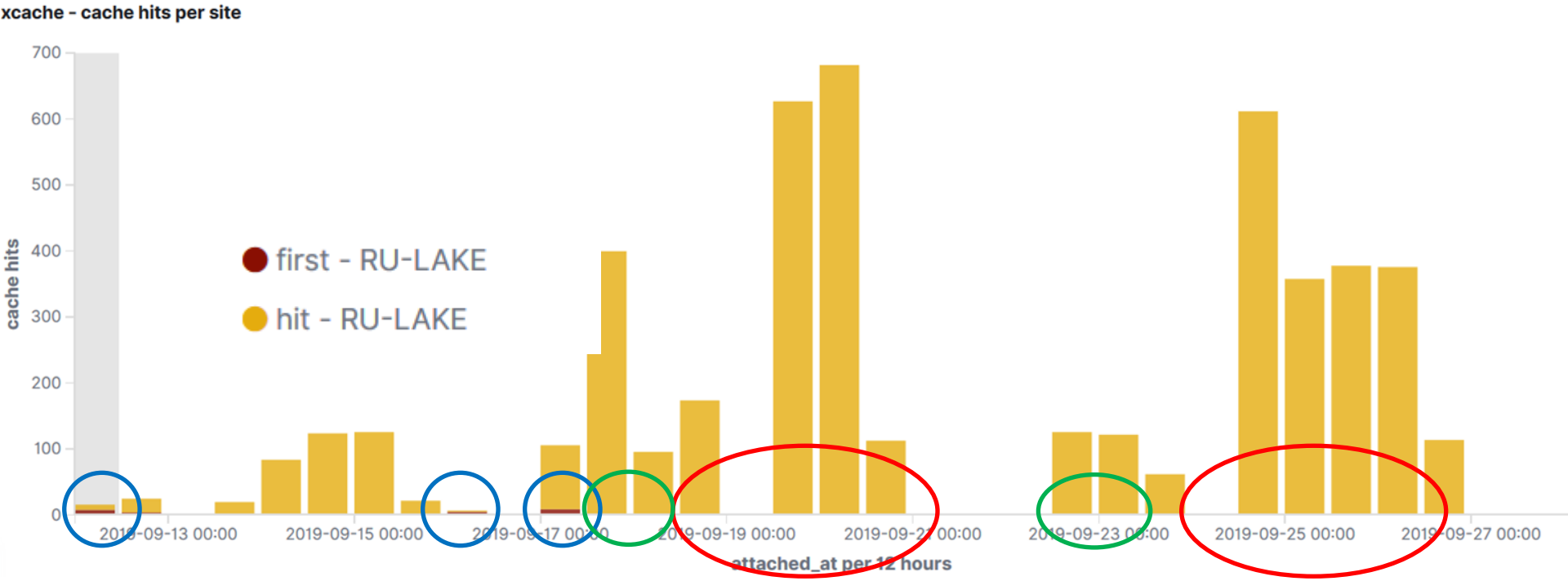
Local (JINR)=117s ± 17s



# HammerCloud test results

- Size of input data file from BigPanda monitoring: **Dataset summary:** input: 3, size: 5923.35(MB); log: 1; output: 1; pseudo_input: 1) ~ 6GB (5923.35 (MB)) ~ 48Gb. For difference of speed 6.7Gb/s and 0.7 Gb/s (see sl. 9) we can expect time (if not look on Latency) $\Delta t = 48 / 0.7\text{-}48 / 6.7 \sim 61s$
- For direct access tests “HC input file” does not include a data input file and we can check the time of download only by difference time of Athena running only for the same WN - for PNPI, we cannot rate JINR-CE. $\Delta t = 250 + -50s$ 12% gain for wallclock.
- For copy2scratch tests, we can use “Download input files time” for all CE. And we estimate FST by this volume but with a very big deviation. It will: PNPI direct ~ 0.06Gb / s, PNPI xcache ~ 0.9Gb / s, JINR local = 0.04Gb / s. $\Delta t$ for PNPI = 770 ~ 30% of wallclock

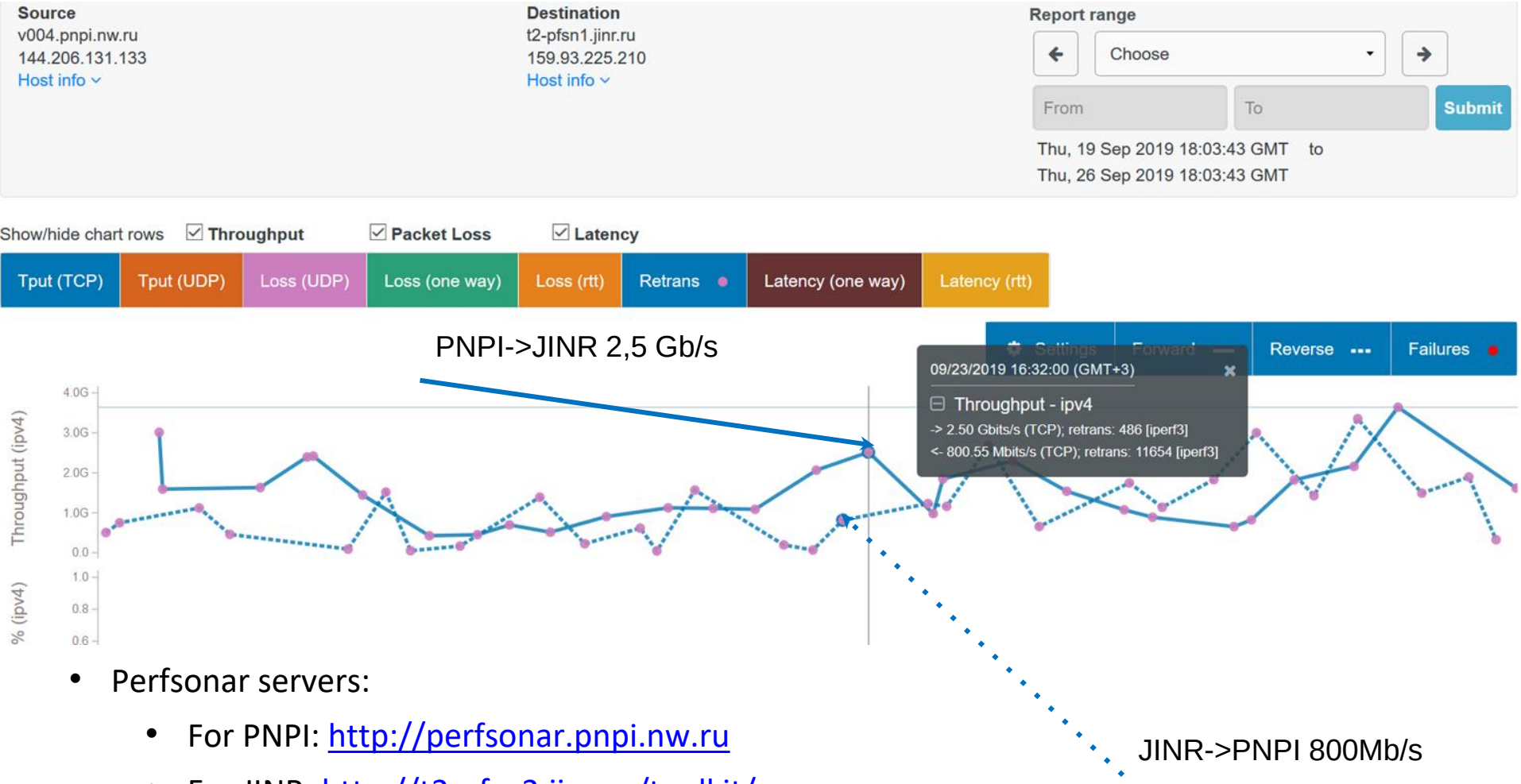
# xCache Monitoring in Kibana




- First access to files
- Activity of HC tests
- Activity of synthetic tests

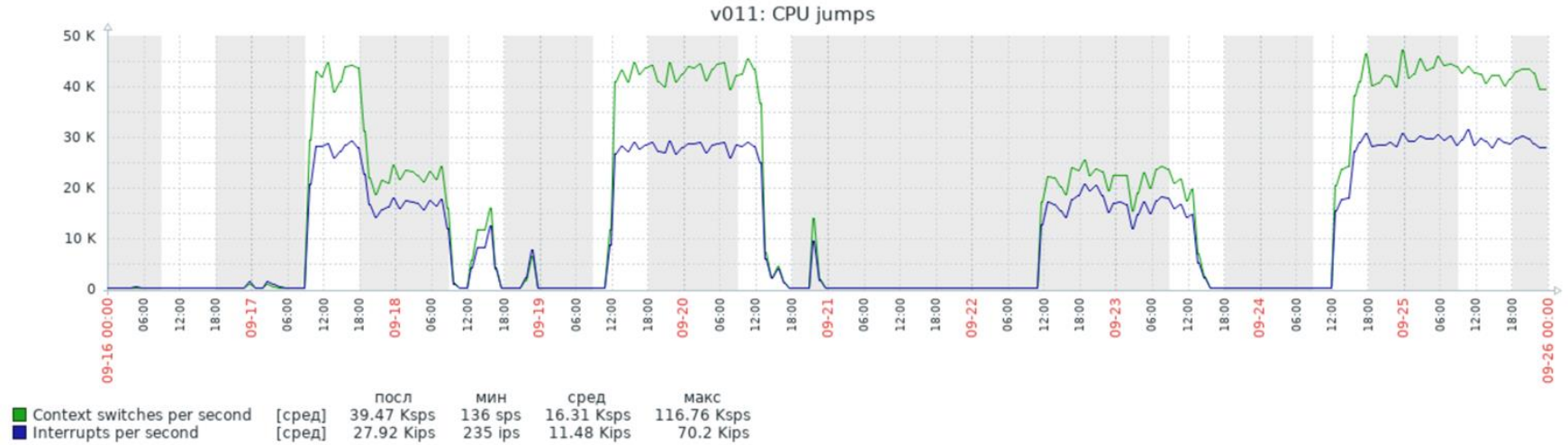
https://atlas-kibana.mwt2.org:5601/s/xcache/app/kibana

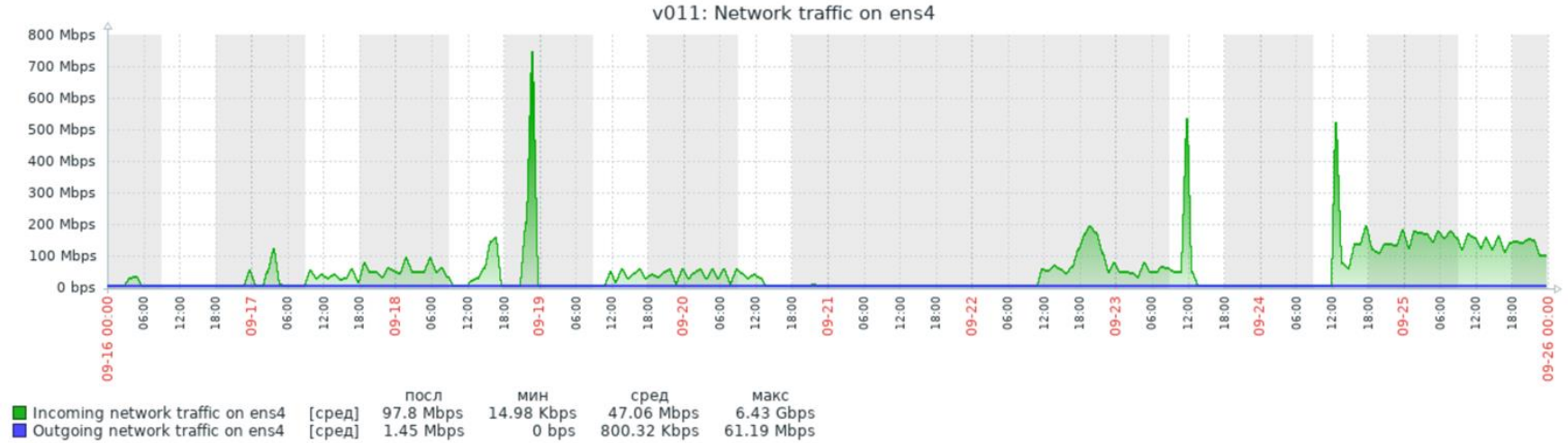
# PerfSonar 19.09-25.09



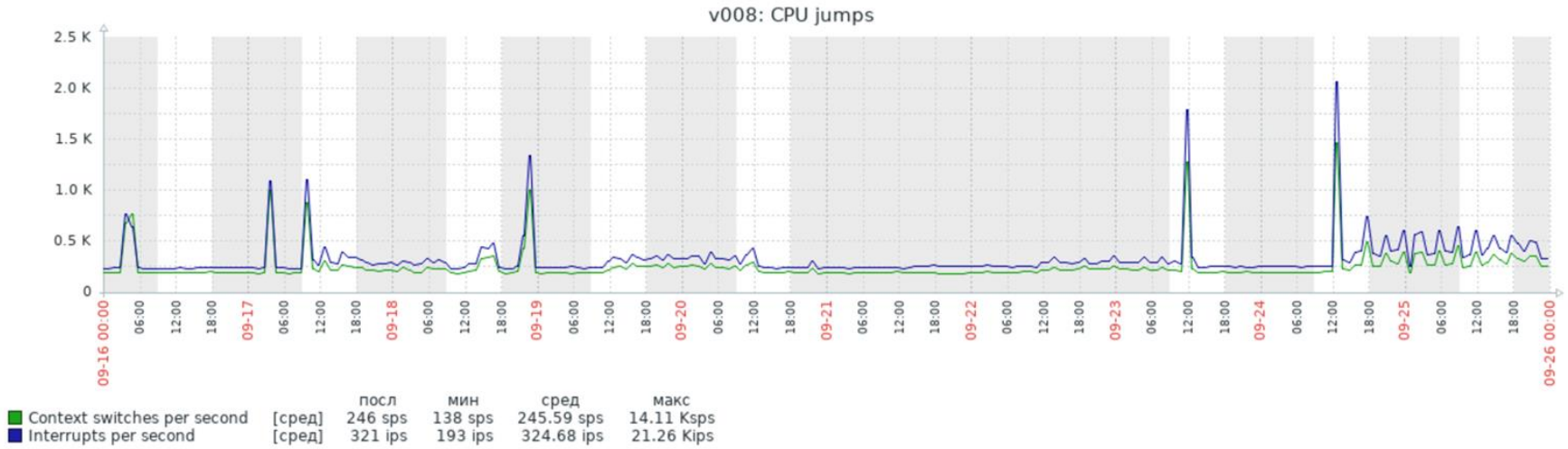

- Perfsonar servers:
  - For PNPI: http://perfsonar.pnpi.nw.ru
  - For JINR: http://t2-pfsn2.jinr.ru/toolkit/
    http://t2-pfsn1.jinr.ru/toolkit/

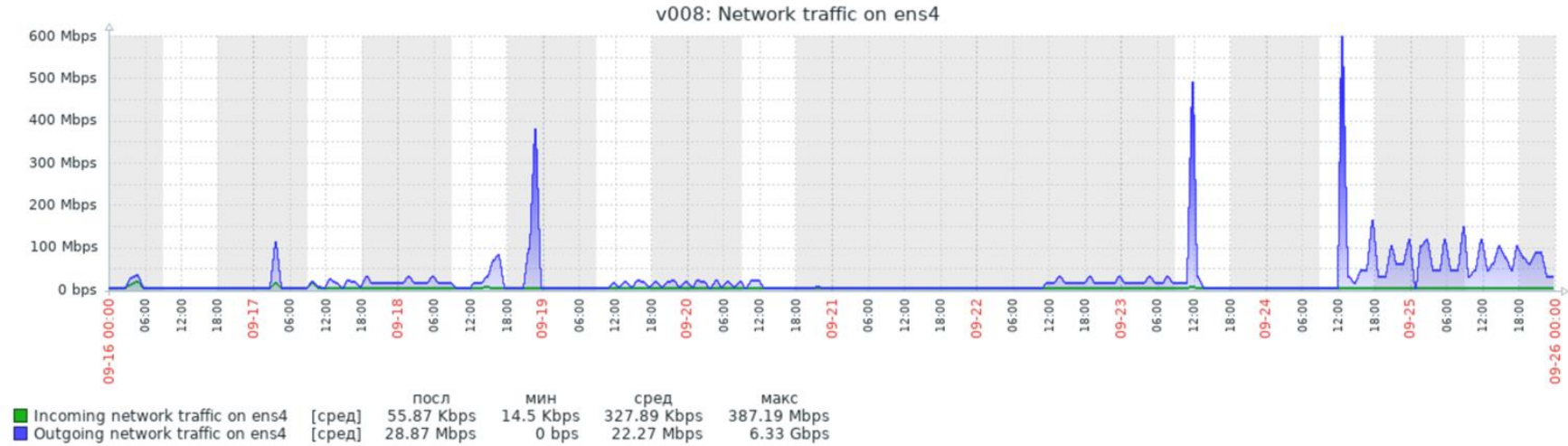
# ZABBIX monitoring of WN PNPI



# ZABBIX monitoring of xCache PNPI



# Test details

Before each test we clean xCache by hand:

{

rm -f /data/xrootd/%data%namespace%pnfs%jinr.ru%data%atlas%atlasdatadisk%rucio%mc16_13TeV%eb%3c%HITS.10701335._005865.pool.root.1*

rm -fr /data/namespace//pnfs/jinr.ru/data/atlas/atlasdatadisk/rucio/mc16_13TeV/eb/3c/HITS.10701335._005865.pool.root.1*

}

This is not a “standard method” and maybe this is the reason why Kibana does not see our “first” access

# Summary

**Done:**

- xCache is configured and works well
- The automatic configuration in AGIS with xCache is checked
- Only PNPI tests are informative now
- Result of synthetic tests show 95% gain in time for 100 repetitions of a file copy
- Result of HC tests show 30% gain in time for “copy2scratch” and 12% gain in time for “Direct access” tests
- We can see details on xCache monitoring and on PerfSonar

# Summary 2 (plans)

**Not done yet:**

- Scaling tests to other Russian sites
- Understanding control of xCache (cleaning, etc)
- Unified monitoring from all sources - Perfsonar, kibana, BigPanda monitoring, etc.

# Thanks!

This work was funded in part by the Russian Science Foundation under contract No. 19-71-30008 (research is conducted in the Plekhanov Russian University of Economics)

# Backups

# Russian DataLake Phase 2 (2020-2021)