National Research Centre (NRC)
"Kurchatov Institute"

Big Data Technologies Laboratory
http://bigdatalab.nrcki.ru/






Federated Data Storage System Prototype based on dCache

Andrey Kiryanov, Alexei Klimentov,
Artem Petrosyan, Andrey Zarochentsev

on behalf of BigData lab @ NRC "KI" and
Russian Federated Data Storage Project

Russian federated data storage project

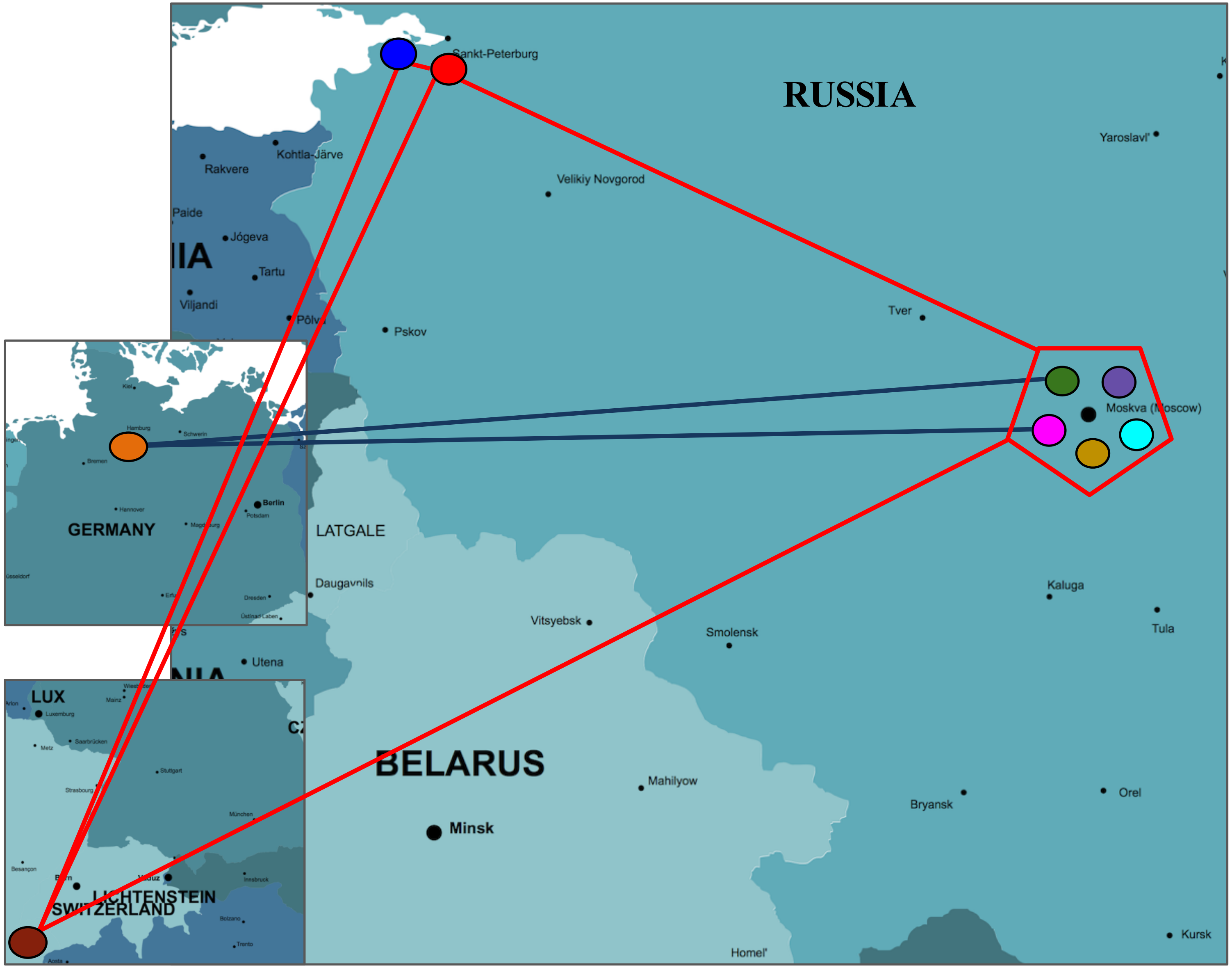
1. In the fall of 2015 the "Big Data Technologies for Mega-Science Class Projects" laboratory at NRC "KI" has received a Russian Fund for Basic Research (RFBR) grant to evaluate federated data storage technologies.
2. This work has been started with creation of a storage federation for geographically distributed data centres located in Moscow, Dubna, St. Petersburg, and Gatchina (all are members of Russian Data Intensive Grid and WLCG).
3. This project aims at providing a usable and homogeneous service with low requirements for manpower and resource level at sites for both LHC and non-LHC experiments.

Basic Considerations

1. Single entry point
2. Scalability and integrity: it should be easy to add new resources
3. Data transfer and logistics optimisation: transfers should be routed directly to the closest disk servers avoiding intermediate gateways and other bottlenecks
4. Stability and fault tolerance: redundancy of core components
5. Built-in virtual namespace, no dependency on external catalogues.

EOS and dCache seemed to satisfy these requirements.

Federation topology

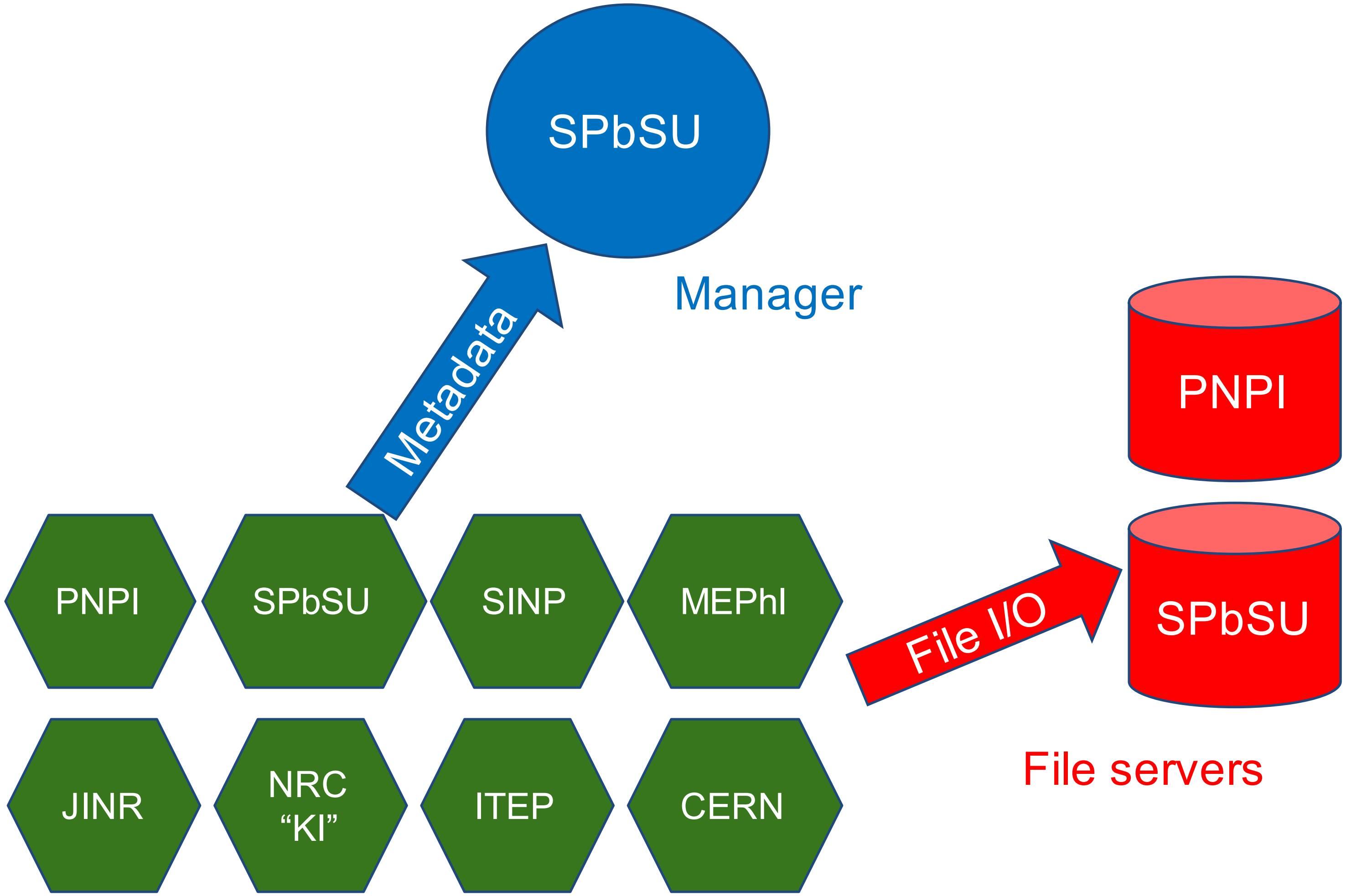


– EOS & dCache
– dCache 3 (plan)

- SPbSU
- PNPI
- JINR
- NRC «KI»
- SINP MSU
- MEPhI
- ITEP
- CERN
- DESY

Initial testbed

proof of concept tests and optimal settings evaluation



Clients: xrootd client, voms client, test and expriment tools

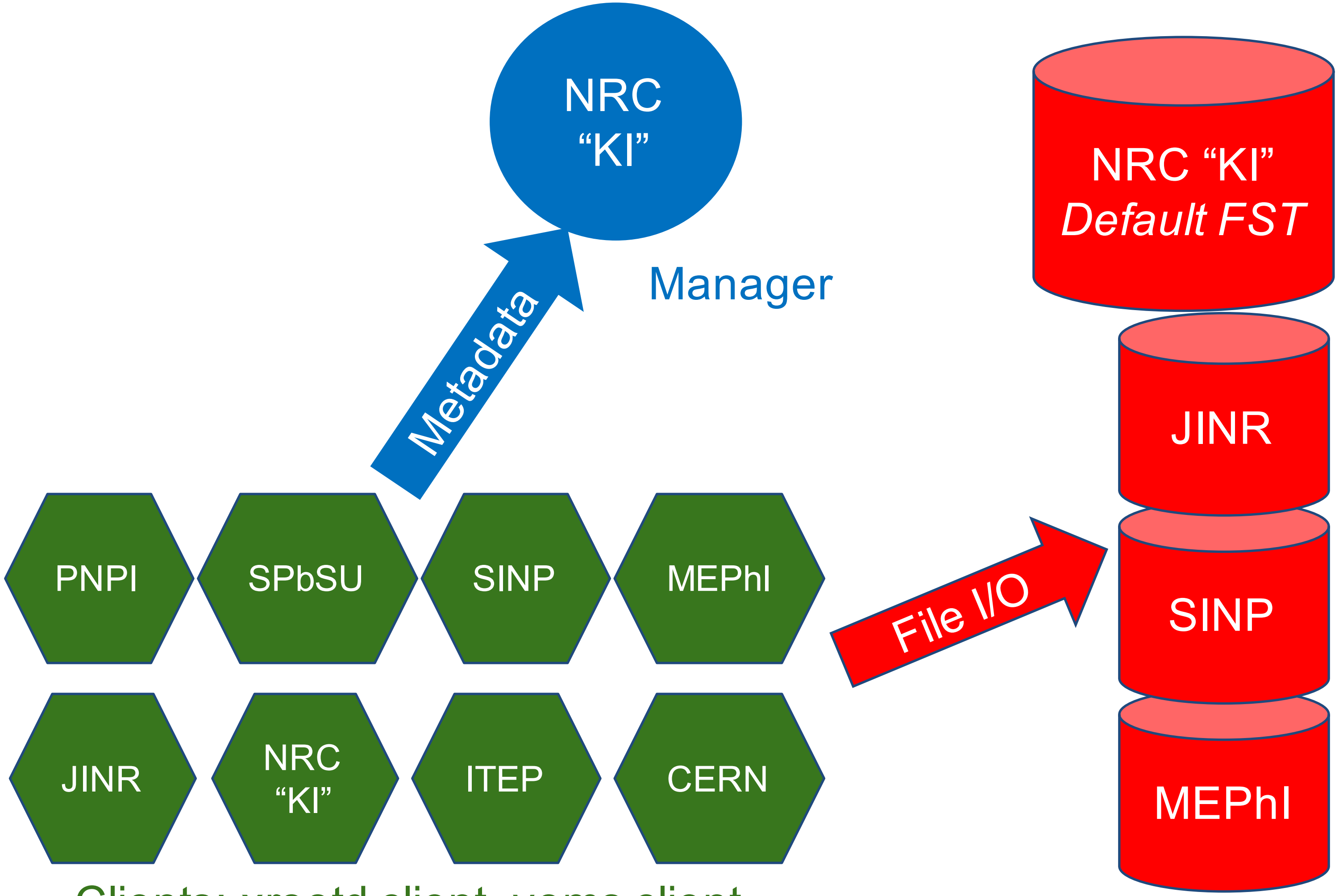
Managers: entry point, virtual name space.

File servers: Data storage servers.

Clients: user interface for synthetic and real-life tests. Same software for EOS and dCache tests.

Extended testbed

for full-scale testing



Clients: xrootd client, voms client, test and expriment tools

File servers

Managers: entry point, virtual name space.

File servers: data storage servers. Two types of storages – highly reliable and normal.

Clients: user interface for synthetic and real-life tests. Same software for EOS and dCache tests.

Test goals, methodology and tools

Goals:

- Set up a distributed storage, verify basic properties, evaluate performance and robustness
  - Data access, reliability, replication

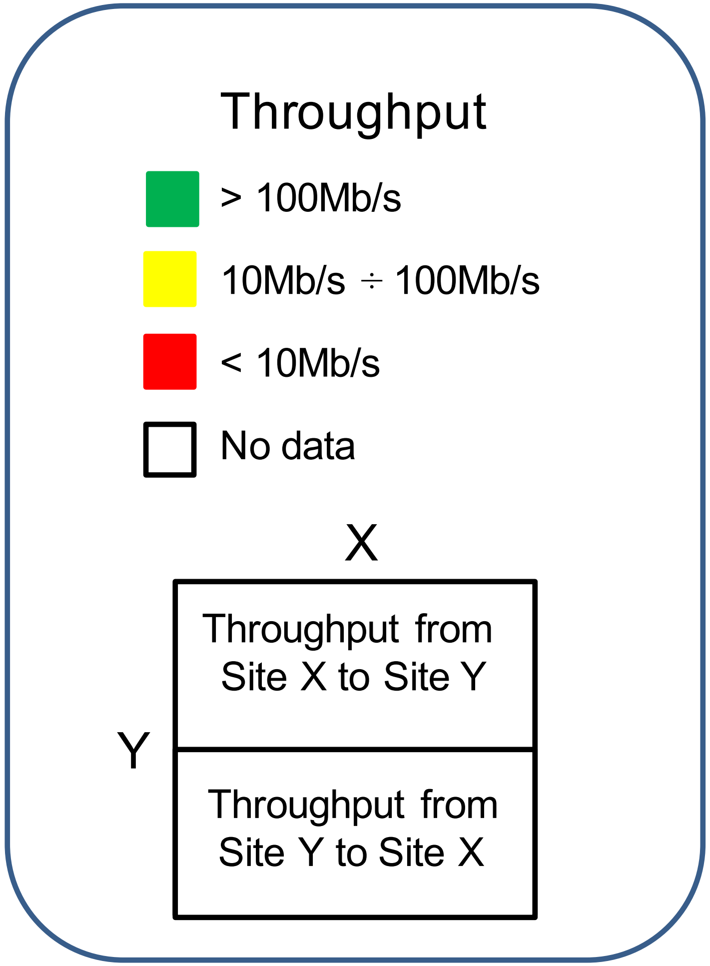
Tools:

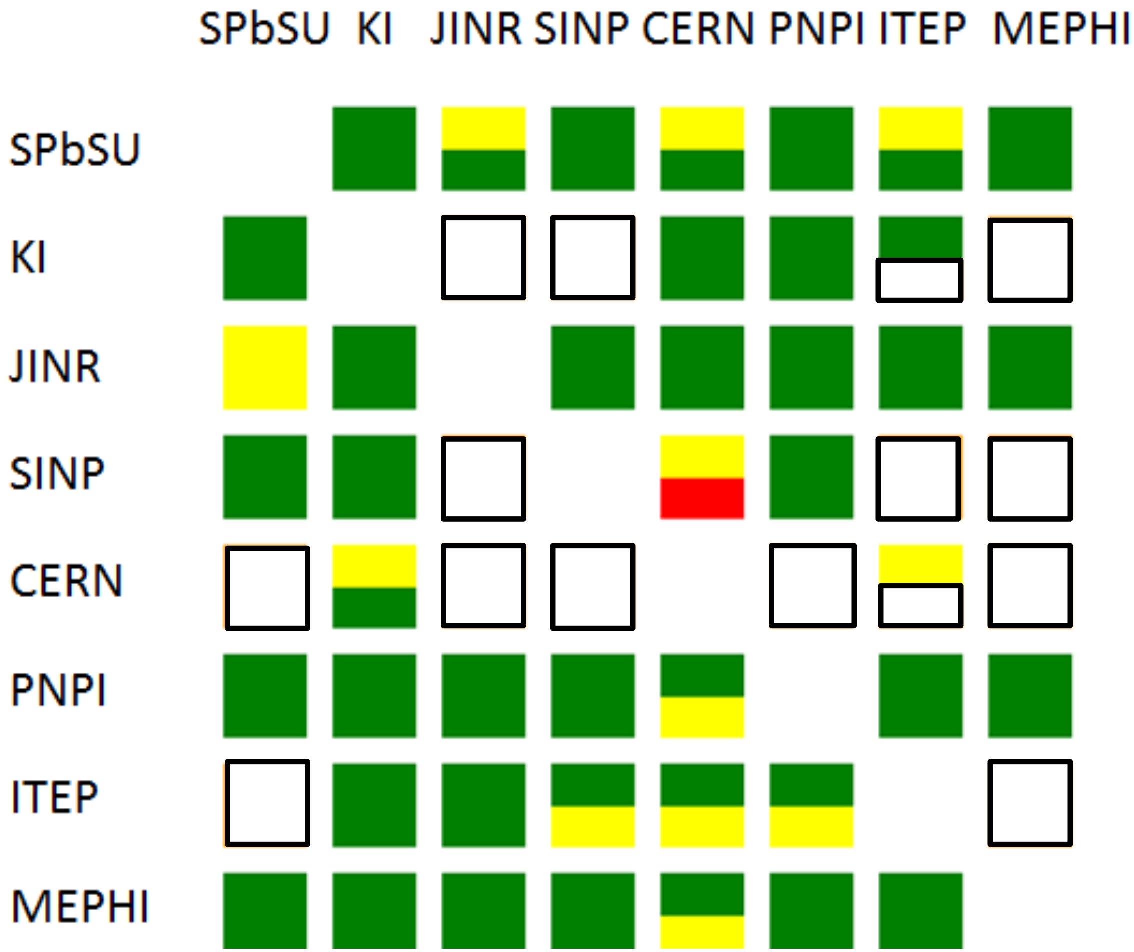
- Synthetic tests:
  - Bonnie++: file and metadata I/O test for mounted file systems (FUSE only)
  - xrdstress: EOS-bundled file I/O stress test for xrootd protocol
  - Experiment-specific tests:
    - ATLAS test: standard ATLAS TRT reconstruction workflow with Athena
    - ALICE test: sequential ROOT event processing (thanks to Peter Hristov)
- Network monitoring:
  - Perfsonar: a widely-deployed and recognized tool for network performance measurements

Software components:

- Base OS: CentOS 6, 64bit
- Storage system: EOS Citrine, dCache 2.16
- Authentication scheme: GSI / X.509
- Access protocol: xrootd

Network performance measurements

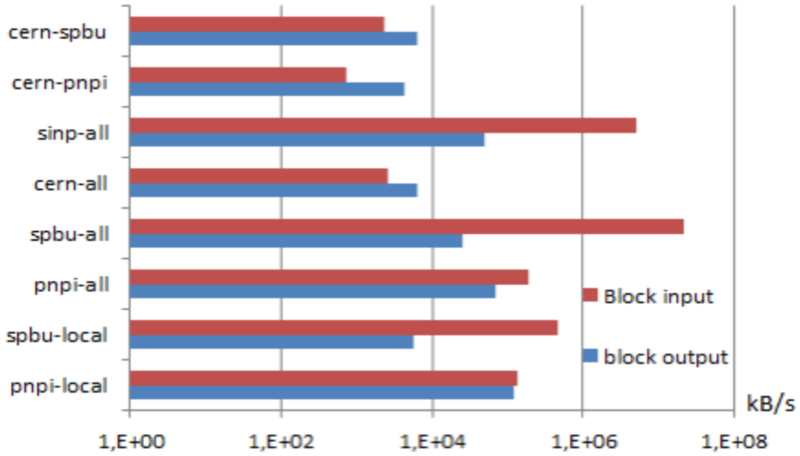




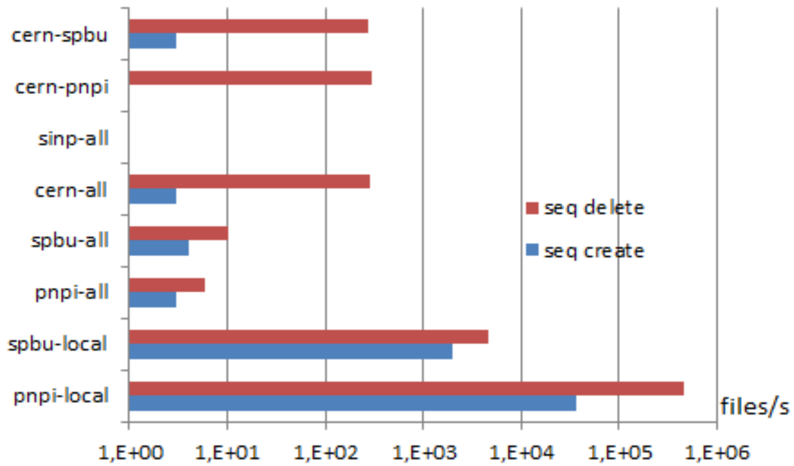
The first results with EOS on initial testbed (2015-2016)

Bonnie++ tests

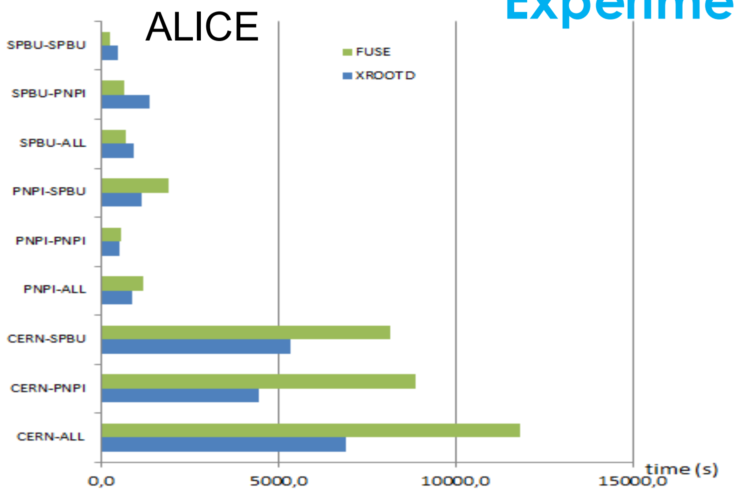
Data read-write



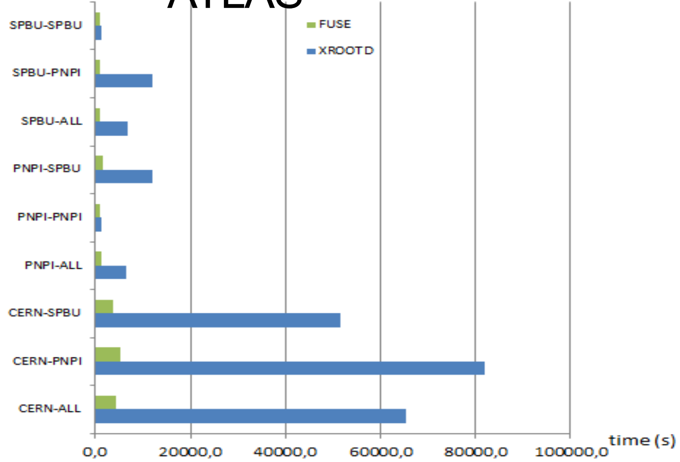
Metadata create-delete



Experiment-specific tests



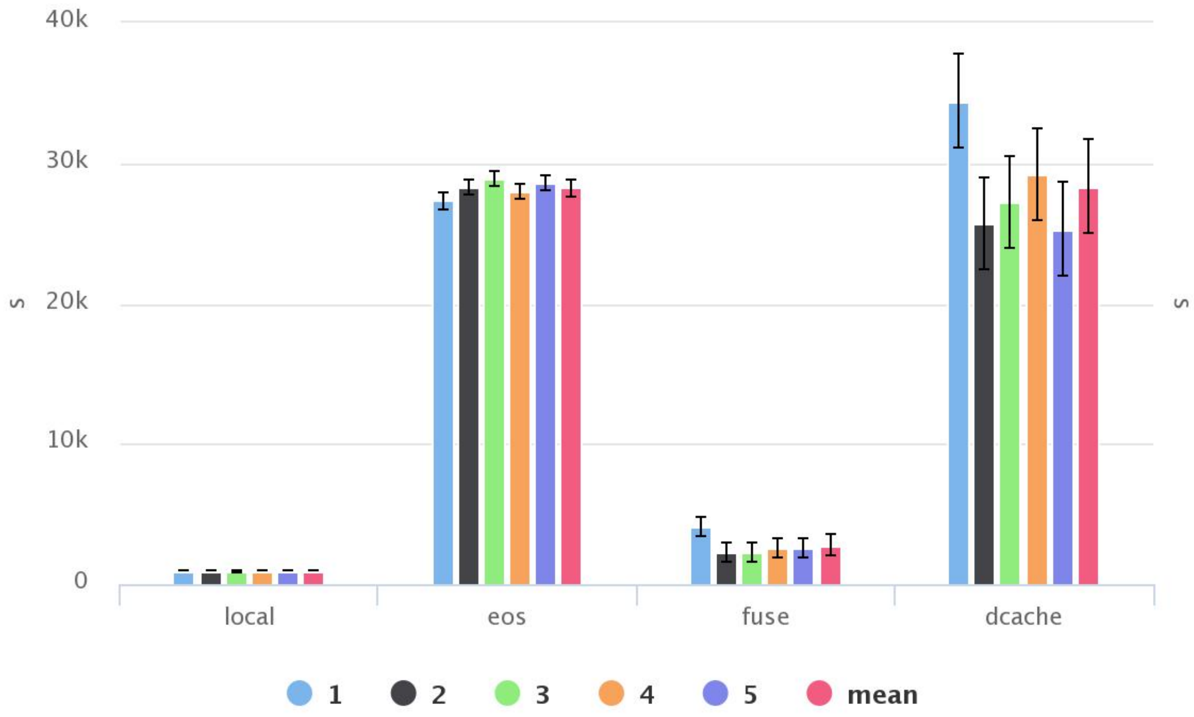
ATLAS



Our First experience with EOS and intermediate conclusion

1. Basic stuff works as expected
2. Some issues were discovered and communicated to developers
3. Metadata I/O performance depends solely on a link between client and manager while data I/O performance does not depend on it
4. *Experiment-specific tests for different data access patterns have contradictory preferences with respect to data access protocol (pure xrootd vs. FUSE-mounted filesystem)*

ATLAS test on initial testbed for different protocols.



Tests were run from PNPI UI

Times of five repetitions of the same test along with a mean value.

Four test conditions on the same federation: **local** data, data on **EOS** accessed via xrootd, data on EOS accessed via **FUSE** mount, data on **dCache** accessed via xrootd.

As we can see, first FUSE test with a cold cache takes a bit more time than the rest, but it's still much faster than accessing data directly via xrootd.

First run with dCache also shows this pattern (internal cache warm-up?)

Data placement policy

1. Number of data replicas depends from data family (replication policy has to be defined by experiments / user community);
2. Federated storage may include reliable sites("T1s") and less reliable sites ("Tns");
3. Taking aforementioned into account we have three data placement scenarios which can be individually configured per dataset:

Scenario 0: Dataset is randomly distributed among several sites

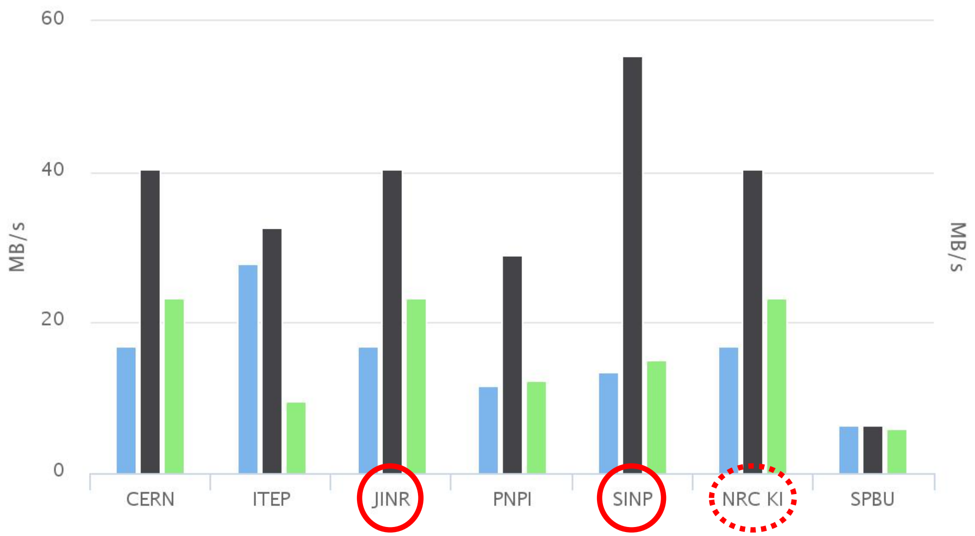
Scenario 1: Dataset is located as close as possible to the client. If there's no close storage, the default reliable one is used (NRC "KI" in our tests) (EOS)

Scenario 2.1: Dataset is located as in scenario 1 with secondary copies as in scenario 0 (EOS)

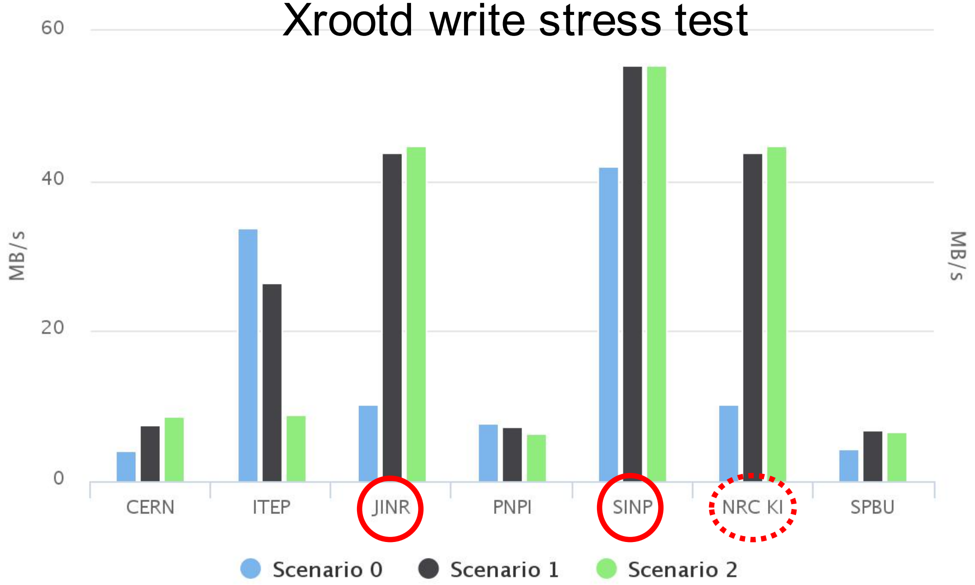
Scenario 2.2: Dataset is located as in scenario 0 with secondary copies as in scenario 0 (dCache)

These policies can be achieved with EOS geotags and dCache replica/pool managers.

Synthetic data placement stress test for EOS



Xrootd write stress test



Xrootd read stress test

Stress test procedure is as follows:

Scenario 0: Files are written to and read from random file servers

Scenario 1: Files are written to and read from a closest file server if there is one or the default file server at NRC "KI"

Scenario 2.1: Primary replicas are written as in Scenario 1, secondary replicas as in Scenario 0. Reads are redirected to a closest file server if there is one or to the default file server at NRC "KI"

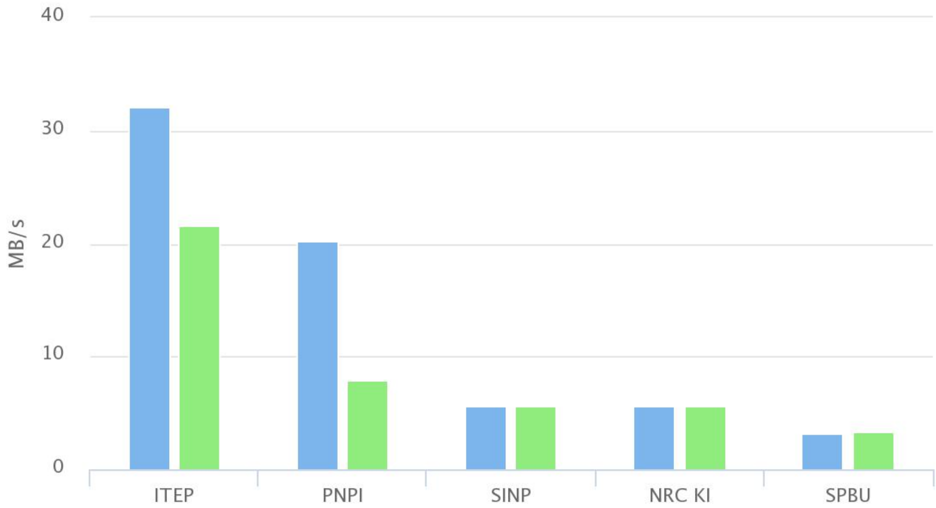
○ – this client can find data on a closest file server

◌ – closest and default file server is the same

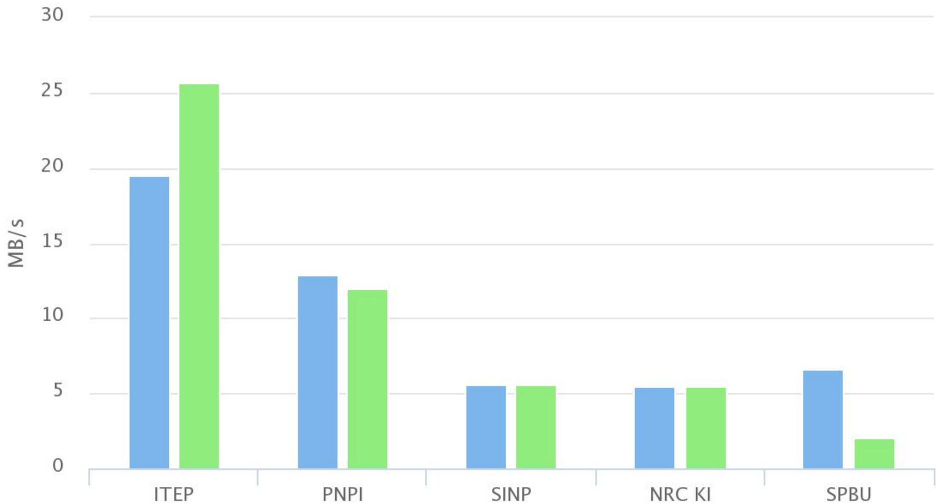
On both plots clients are shown on X axis

Presence of the closest server does not bring much of improvement, but presence of the high-performance default one does. Replication slows down data placement because EOS creates all replicas during the transfer.

Synthetic data placement stress test for dCache



Xrootd write stress test



Xrootd read stress test

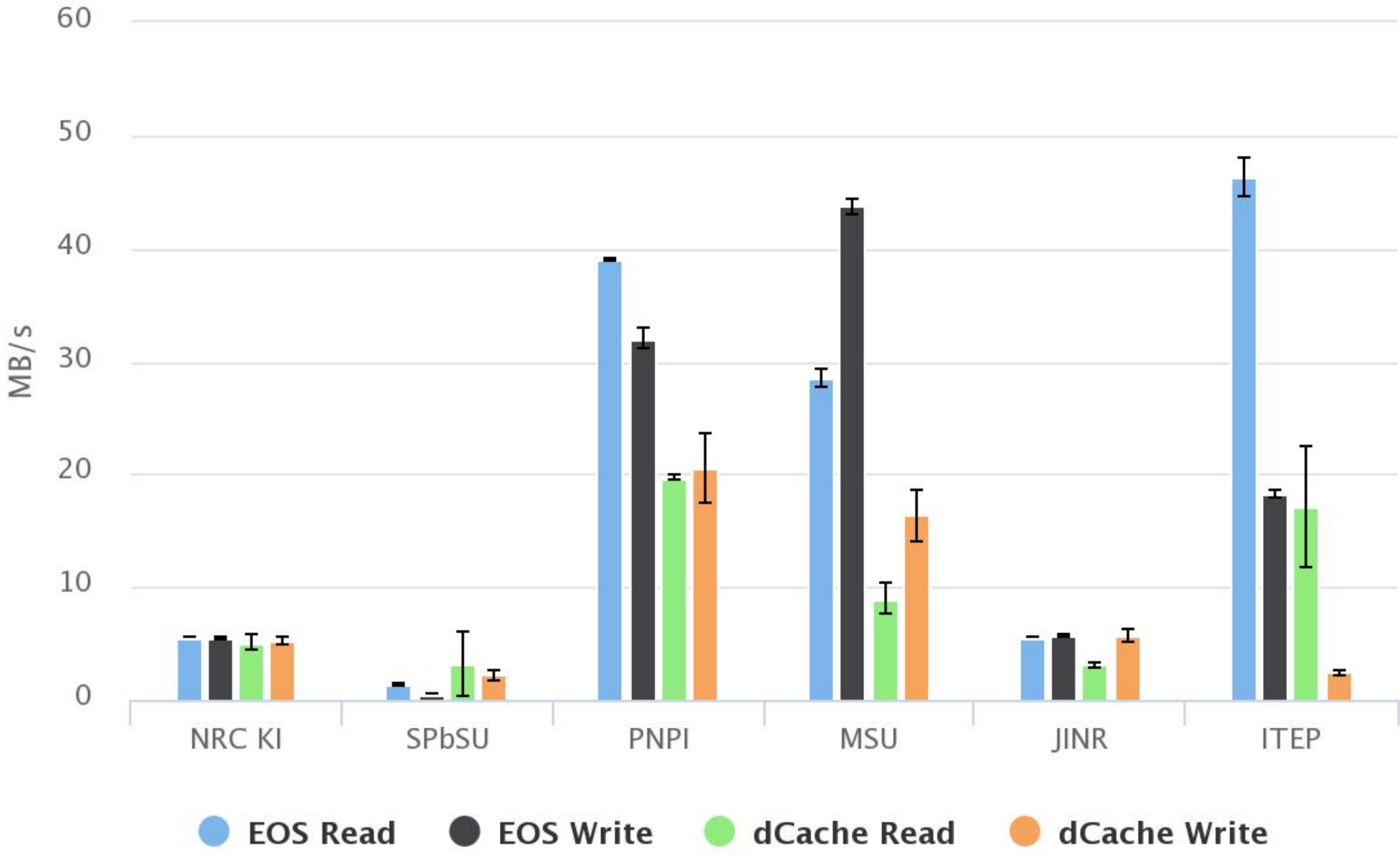
1. **Scenario 1:** Files are scattered among several file servers
2. **Scenario 2.2:** Files have two replicas both of which are scattered among several file servers

In this test closest file servers were not configured. On two sites replication slowed down the write test, despite that dCache performs background replication after the transfer.
Read test does not show any specific pattern because of the random data placement in both scenarios.

First experience with dCache

- Well-known and reliable software used on many T1s
- Different software architecture, platform (Java) and protocol implementations
- dCache xrootd implementation does not support FUSE mounts
  - This is supposed to be fixed in dCache 3 within a collaborate project between NRC "KI", JINR and DESY (thanks to Ivan Kadochnikov)
- No built-in security for control channel between Manager and File servers.
  - This is solved by stunnel in dCache 3, but we didn't have a chance to test it yet
- No built-in Manager redundancy
  - Also part of dCache 3 feature set
- We have just started testing replication policies with dCache

Performance comparison
EOS and dCache read/write stress test



Summary

- We have set up a working prototype of federated storage:
  - Seven Russian WLCG sites organized as one homogeneous storage with single entry point
  - All basic properties of federated storage are respected
- We have conducted an extensive validation of the infrastructure using:
  - Synthetic tests
  - Experiment-specific tests
  - Network monitoring
- We have exploited EOS as our first technological choice and we have enough confidence to say that it behaves well and has all the features we need
- We're in the middle of extensive testing with dCache, but our first results look very promising
- Yet we have other software solutions to exploit (HTTP-based federation and DynaFed)
  - Might be used as an inter-federation solution, as both EOS and dCache are going to stick around for at least another decade

Acknowledgements

This talk drew on presentations, discussions, comments and input from many. Thanks to all, including those we've missed.

P. Hristov and D. Krasnopevtsev for help with experiment-specific tests.

A. Peters for help with understanding EOS internals.

P. Fuhrmann and dCache core team for help with dCache.

I. Kadochnikov for his work on dCache xrootd improvement.

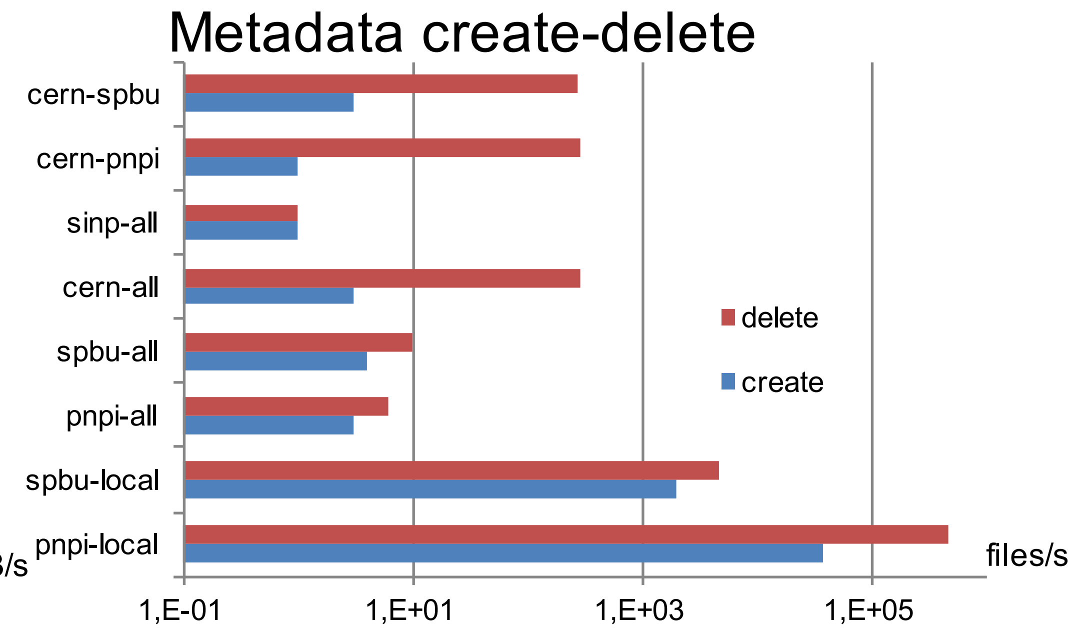
This work was funded in part by the Russian Ministry of Science and Education under Contract No. 14.Z50.31.0024 and by the Russian Fund for Fundamental Research under contract "15-29-0794_2 офи_м"

Authors express appreciation to SPbSU Computing Center, PNPI ITAD Computing Center, NRC "KI" Computing Center, JINR Cloud Services, ITEP, SINP and MEPhI for provided resources.

Thank you!

BACKUP

Bonnie++ test with EOS on initial testbed: MGM at CERN, FSTs at SPbSU and PNPI

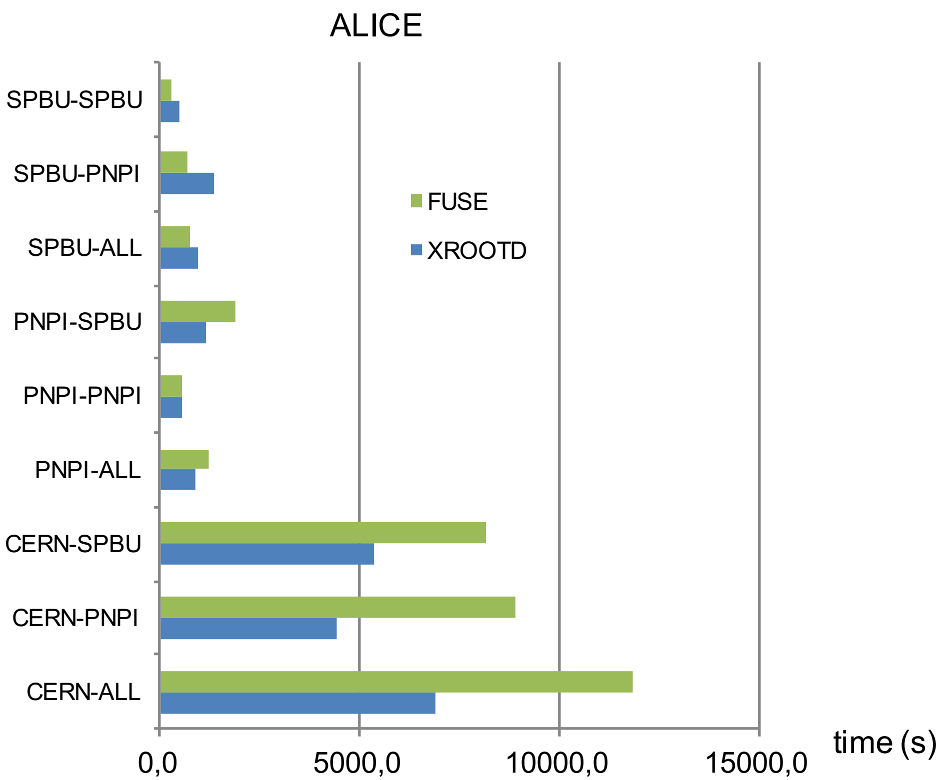


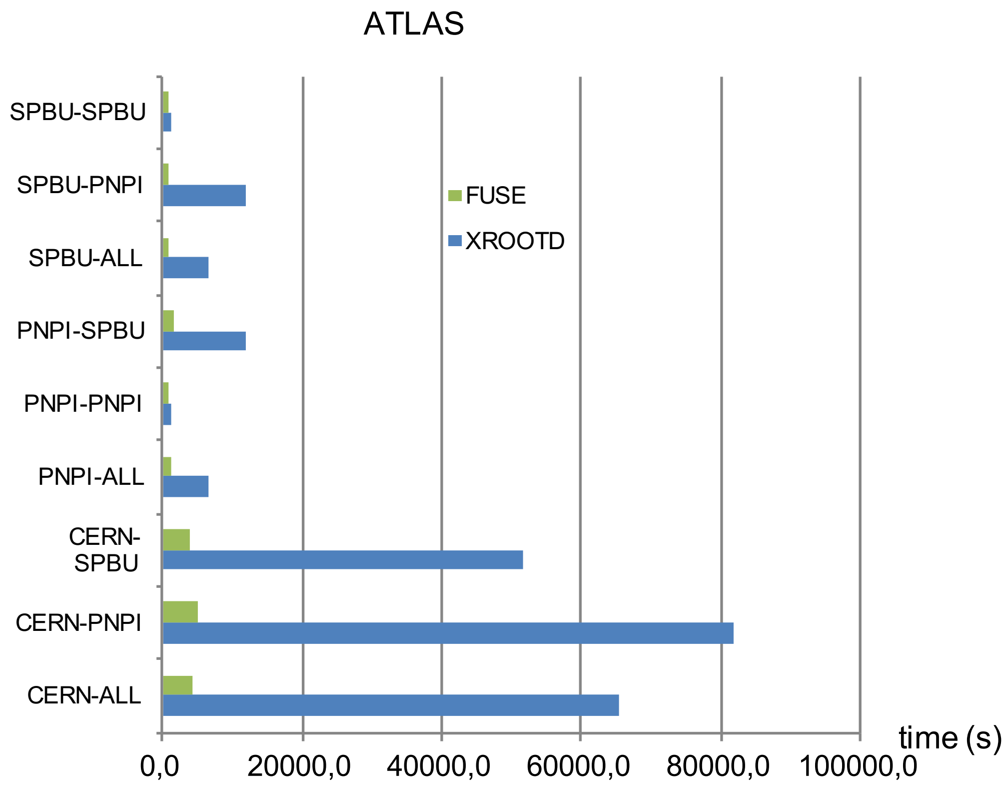
pnpi-local – local test on PNPI FST
spbu-local – local test on SPbSU FST
pnpi-all – UI at PNPI, MGM at CERN, Federated FST
spbu-all – UI at SPbSU, MGM at CERN, Federated FST

cern-all – UI at CERN, MGM at CERN, Federated FST
sinp-all – UI at SINP, MGM at CERN, Federated FST
cern-pnpi – UI at CERN, MGM at CERN, FST at PNPI
cern-spbu – UI at CERN, MGM at CERN, FST at SPbSU

- metadata I/O performance depends solely on a link between client and manager
- data I/O performance does not depend on a link between client and manager

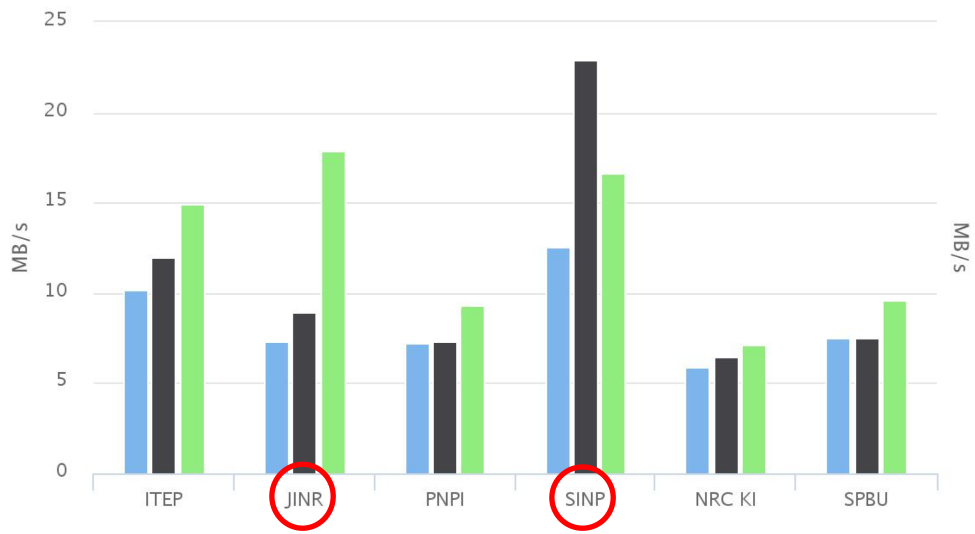
Experiment-specific tests with EOS for two protocols: pure xrootd and locally-mounted file system (FUSE)



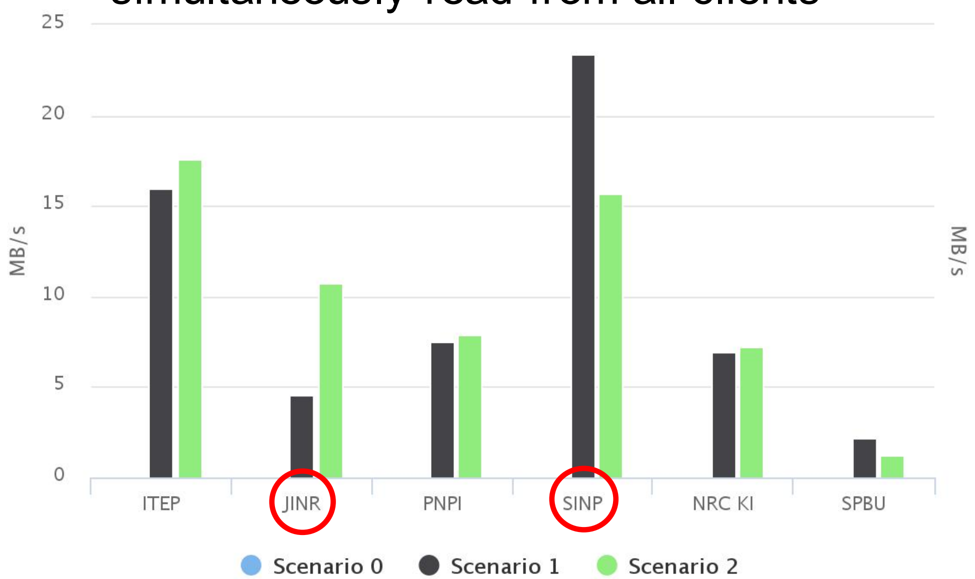


Experiment's applications are optimized for different protocols (remote vs. local)

ALICE read test for EOS 3 Scenarios



simultaneously read from all clients



only one client reads data at any given time

Read procedure is as follows:

Scenario 0: Files are scattered among several file servers

Scenario 1: All files are on the default file server at NRC "KI"

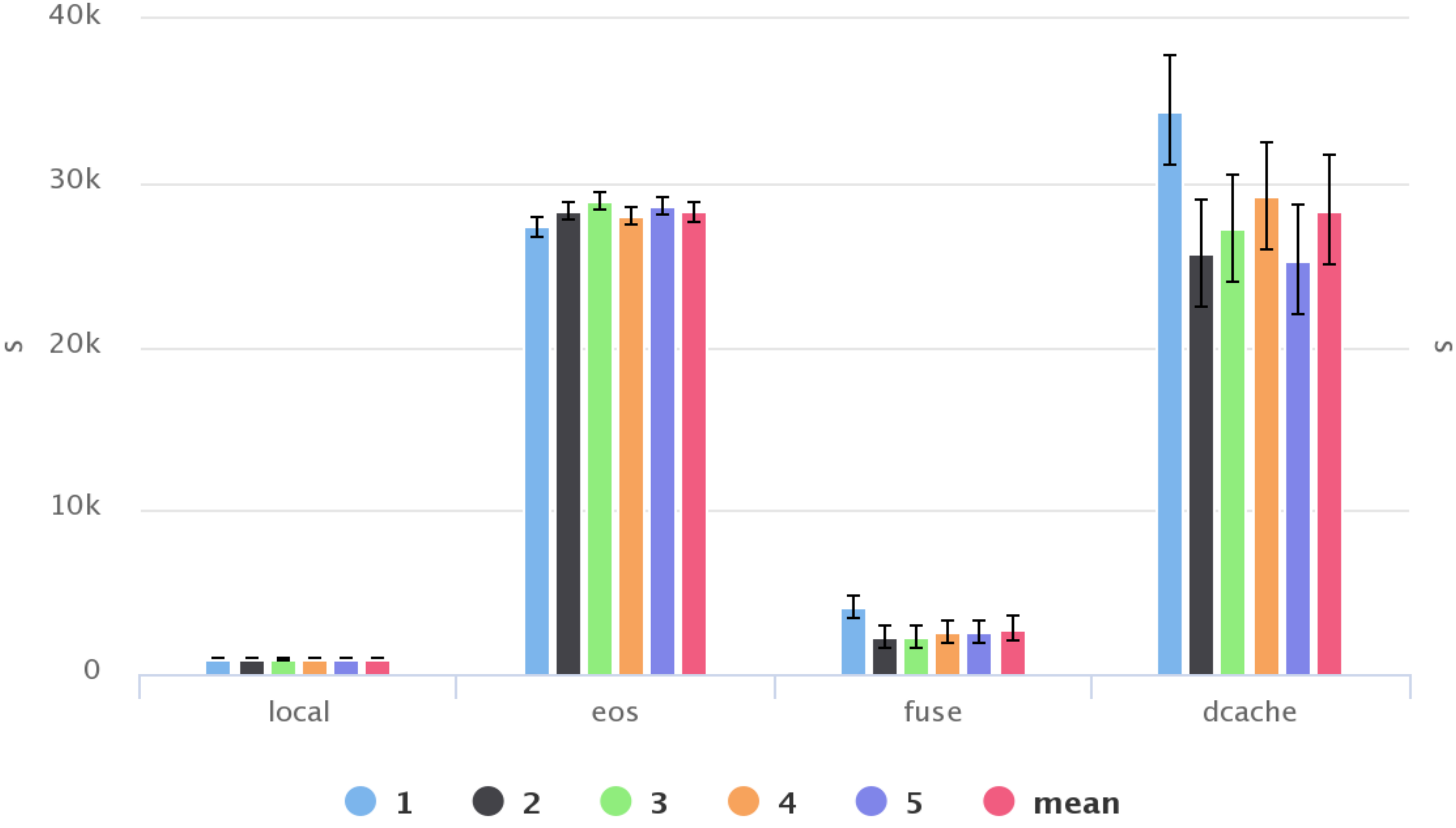
Scenario 2: All files are on the default file server at NRC "KI" with replicas that may end up on a closest file server

◯ – this client can find data on a closest file server

On both plots clients are shown on X axis

Impact of a system load is negligible at this scale.
Logistics optimization only makes sense for sites with a proper infrastructure.

ATLAS test with EOS and dCache



1-5 – five independent test runs