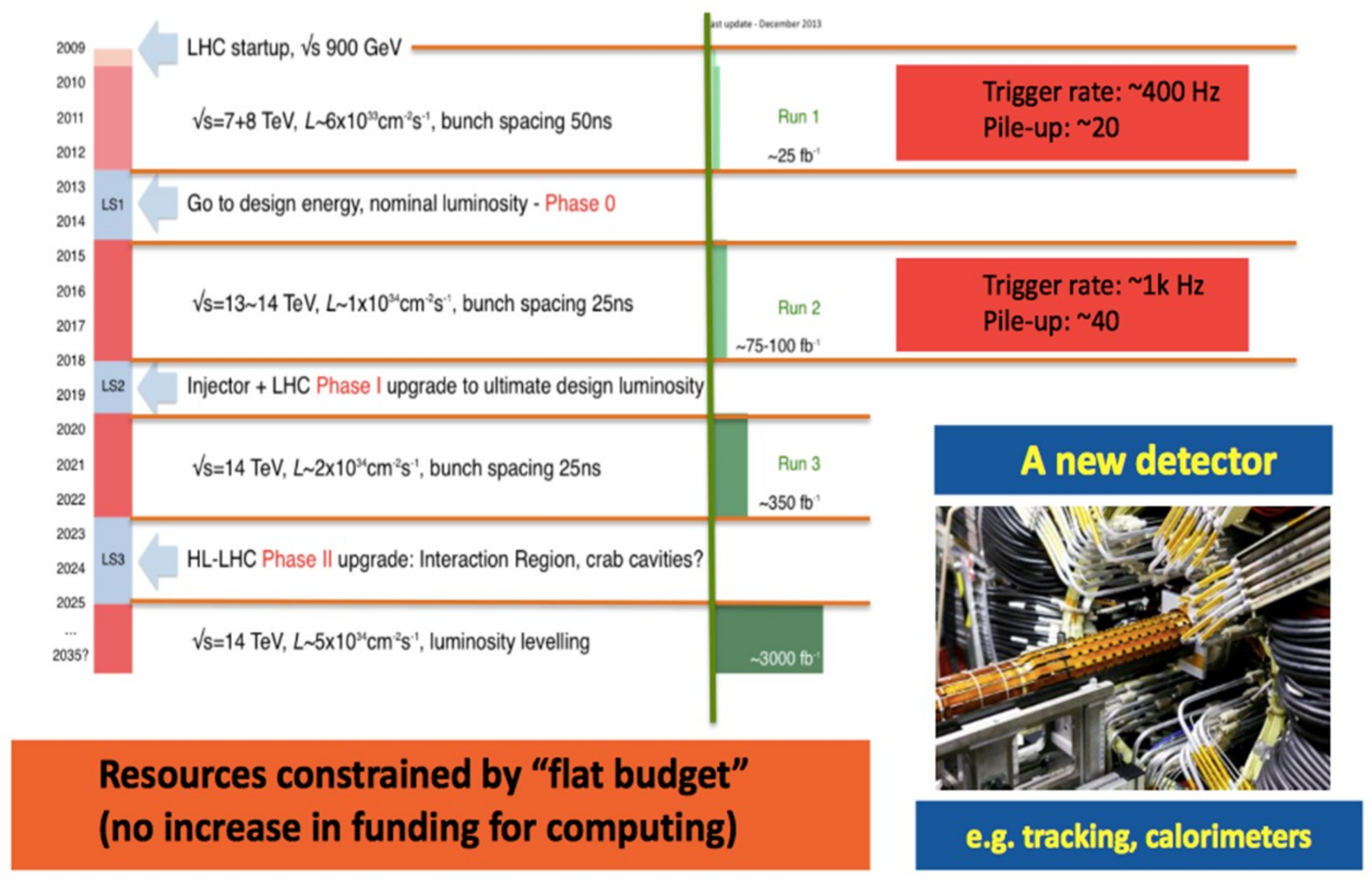
Federated data storage system prototype for LHC experiments and data intensive science

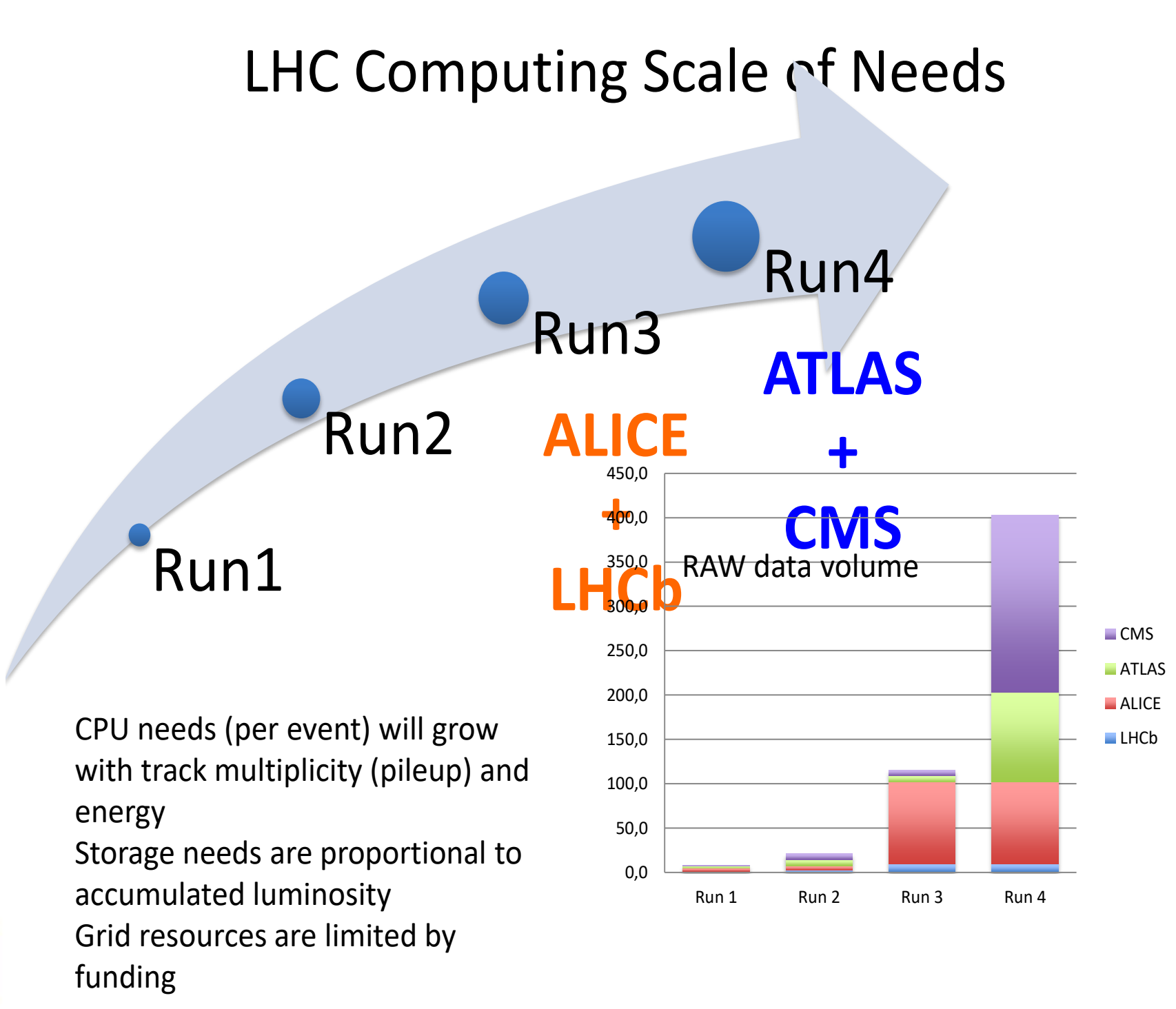
Andrey Kiryanov (PNPI, NRC “KI”) , Alexei Klimentov (NRC “KI”), Dimitrii Krasnopevtsev (NRNU “MEPhI”, NRC “KI”),

Artem Petrosyan (JINR, NRC “KI”), Eygene Ryabinkin(NRC “KI”), Andrey Zarochentsev (SPbSU, NRC “KI”)

Project motivation


ast update - December 2013
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
...
2035?
LS1
LS2
LS3
LHC startup, √s 900 GeV
√s=7+8 TeV, L~6x10³³cm⁻²s⁻¹, bunch spacing 50ns
Run 1
~25 fb⁻¹
Trigger rate: ~400 Hz
Pile-up: ~20
Go to design energy, nominal luminosity - Phase 0
√s=13~14 TeV, L~1x10³⁴cm⁻²s⁻¹, bunch spacing 25ns
Run 2
~75-100 fb⁻¹
Trigger rate: ~1k Hz
Pile-up: ~40
Injector + LHC Phase I upgrade to ultimate design luminosity
√s=14 TeV, L~2x10³⁴cm⁻²s⁻¹, bunch spacing 25ns
Run 3
~350 fb⁻¹
HL-LHC Phase II upgrade: Interaction Region, crab cavities?
√s=14 TeV, L~5x10³⁴cm⁻²s⁻¹, luminosity levelling
~3000 fb⁻¹
A new detector
e.g. tracking, calorimeters

Resources constrained by “flat budget” (no increase in funding for computing)



CPU needs (per event) will grow with track multiplicity (pileup) and energy
Storage needs are proportional to accumulated luminosity
Grid resources are limited by funding

- Computing models for the LHC Run3 and High Luminosity era anticipate a growth of storage needs of at least two orders of magnitude;
- The reliable operation of large scale data facilities need a clear economy of scale;
- A distributed heterogeneous system of independent storage systems is difficult to be used efficiently by user communities and couples the application level software stacks with the provisioning technology at sites;
- Small institutions have not enough people to support a fully-fledged software stack. Distributed stuff like ATLAS FAX, ALICE xrootd, EOS@CERN, CMS AAA, dCache, etc (mostly) works;
- Federating the data centers provides a logical homogeneous and consistent reliable resource for the end users;
- In our R&D project we try to analyze how to set up a distributed storage within national region and how it can be used from Grid sites, from HPC, academic and commercial clouds, etc.
  - Also part of WLCG Federated storage demonstrator.

Basic Requirements for a Federated Storage

- Single entry point;
- Should be usable by at least two major LHC experiments;
- Scalability and integrity: it should be easy to add new resources;
- Data transfer optimization: transfers should be routed directly to the disk servers avoiding intermediate gateways and other bottlenecks;
- Stability and fault tolerance: redundancy of core components;
- Built-in virtual namespace, no dependency on external catalogues.

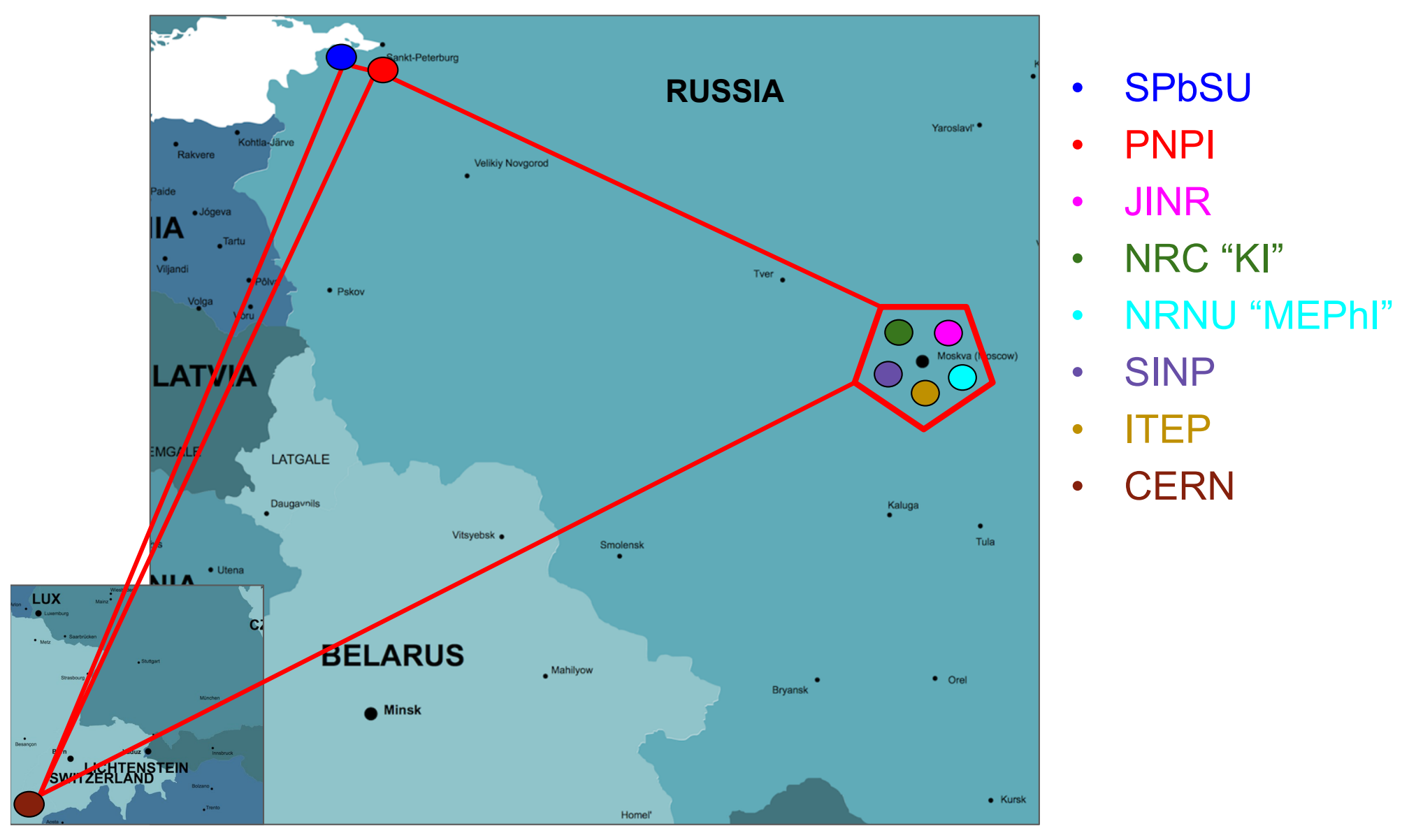
Technology choice

We had to find a software solution that is capable of federating distributed storage resources. This very much depends on a transfer protocol support for redirection. Three protocols that are capable of it are xroot, HTTP and pNFS. We have looked through various storage solutions used in WLCG and selected three of them for thorough testing:

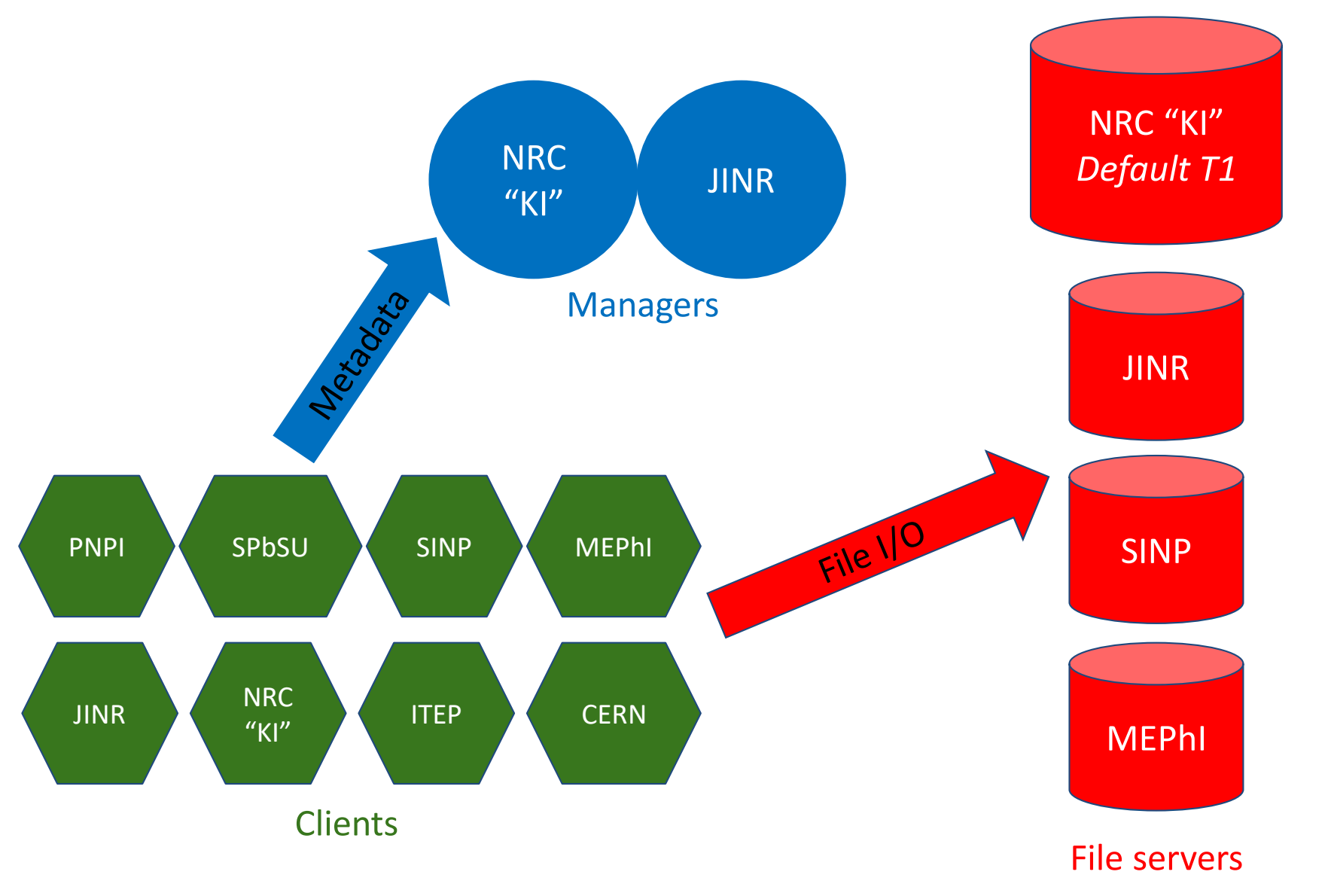
1. EOS: xroot-based solution that is developed at CERN (we know where to ask for help), has characteristics closely matching our requirements, and is already used by all major LHC experiments. EOS is used for our first tests.
2. dCache: dCap/pNFS-based storage system developed at DESY. Depending on the Persistency Model, dCache provides methods for exchanging data with backend (tertiary) storage systems as well as space management, pool attraction, dataset replication, hot spot determination and recovery from disk or node failures. Planned for the next series of tests.
3. DynaFed: HTTP-based federator developed at CERN. This software is highly modular but only provides a federation frontend while storage backend(s) have to be chosen separately. While we were looking for more all-in-one solution it would be interesting to try it out eventually, because it can enable mixed EOS/dCache federation.

Prototype structure

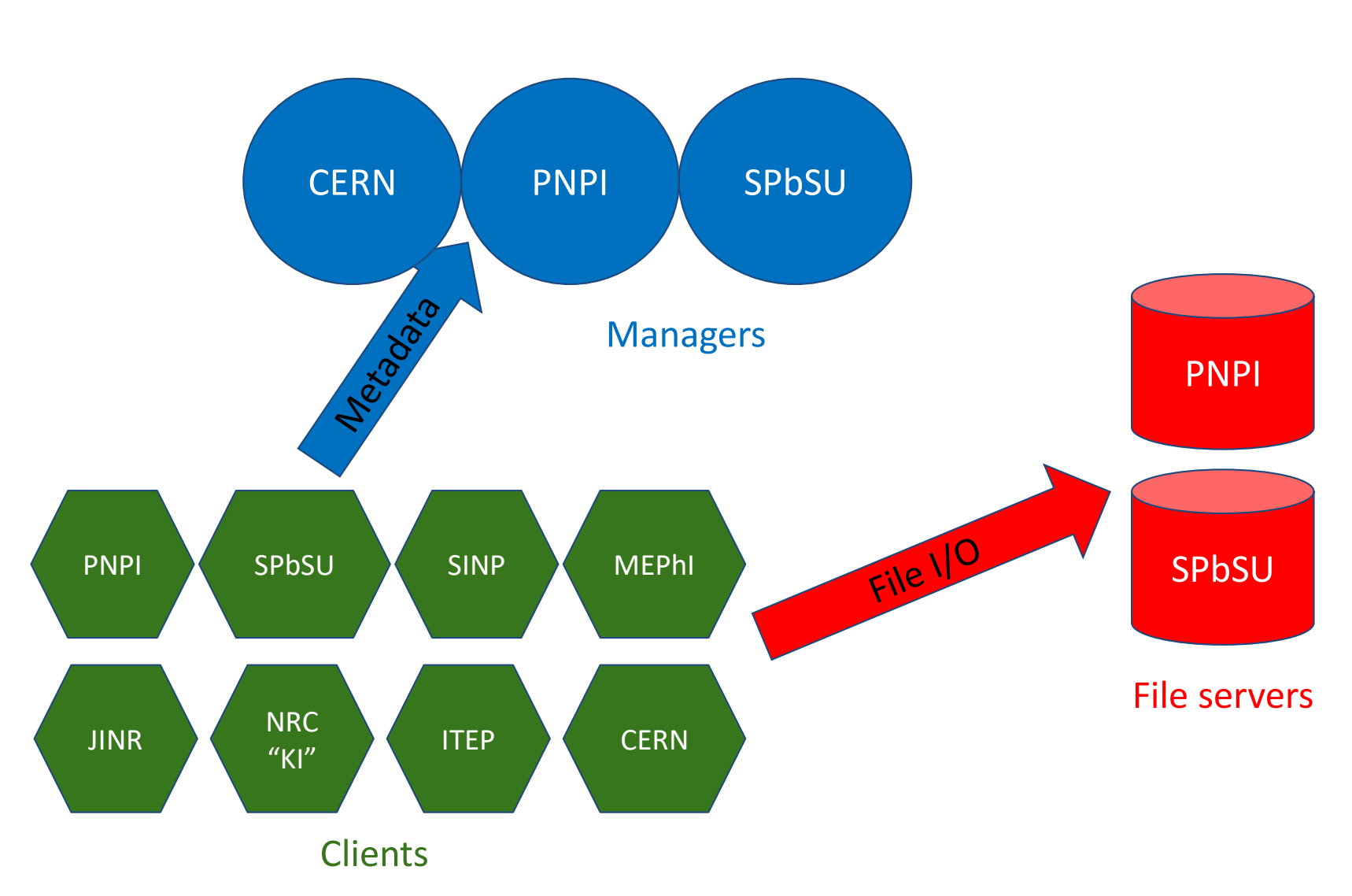
Participating sites



Main testbed



Secondary testbed



Testbeds

Our project joins resources from seven major Russian research institutes.

We had to set up two separate testbeds because we were doing functional tests on the small secondary testbed in parallel with EOS deployment and configuration on the main testbed.

EOS tests

One-shot tests

- Proof-of-concept test: install and configure distributed EOS, hook up GSI authentication, test basic functionality (file/directory create/delete, FUSE mount, access permissions);
- Redirection impact test: check if there’s performance degradation with remote “head” node;
- Reliability test: MGM master-slave migration.

Continuous tests

- Performance tests: file and metadata I/O, network;
- Data locality test: evaluate EOS geo-tags role in data distribution;
- Real-life tests using experiment software and real data.

Base OS: SL6 64bit

Storage system: EOS Aquamarine

Authentication scheme: GSI

Network monitoring: perfSONAR

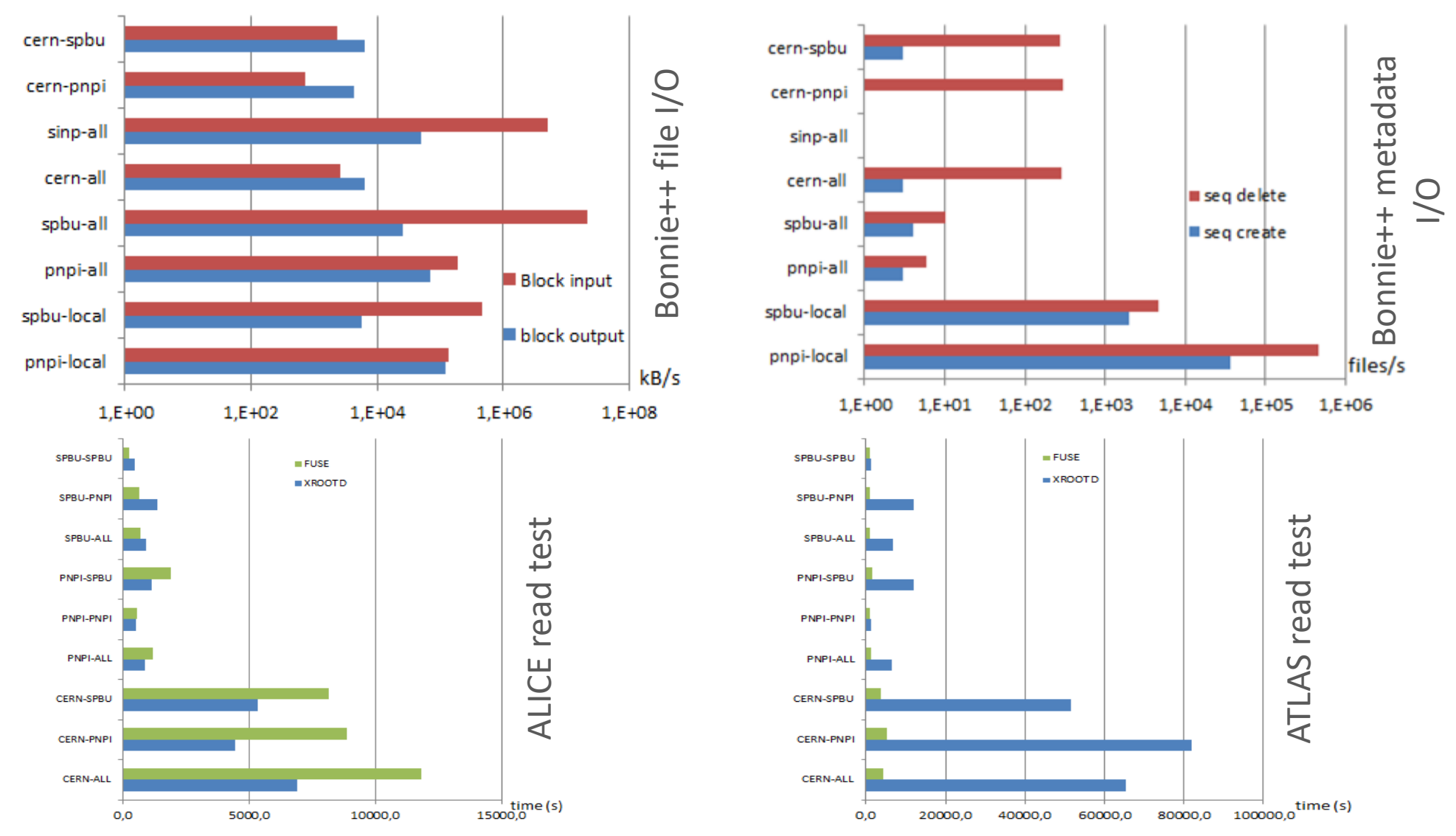
Synthetic tests

- Bonnie++: file and metadata I/O test for mounted file systems (FUSE)
- xrdstress: EOS file I/O stress test via xroot protocol

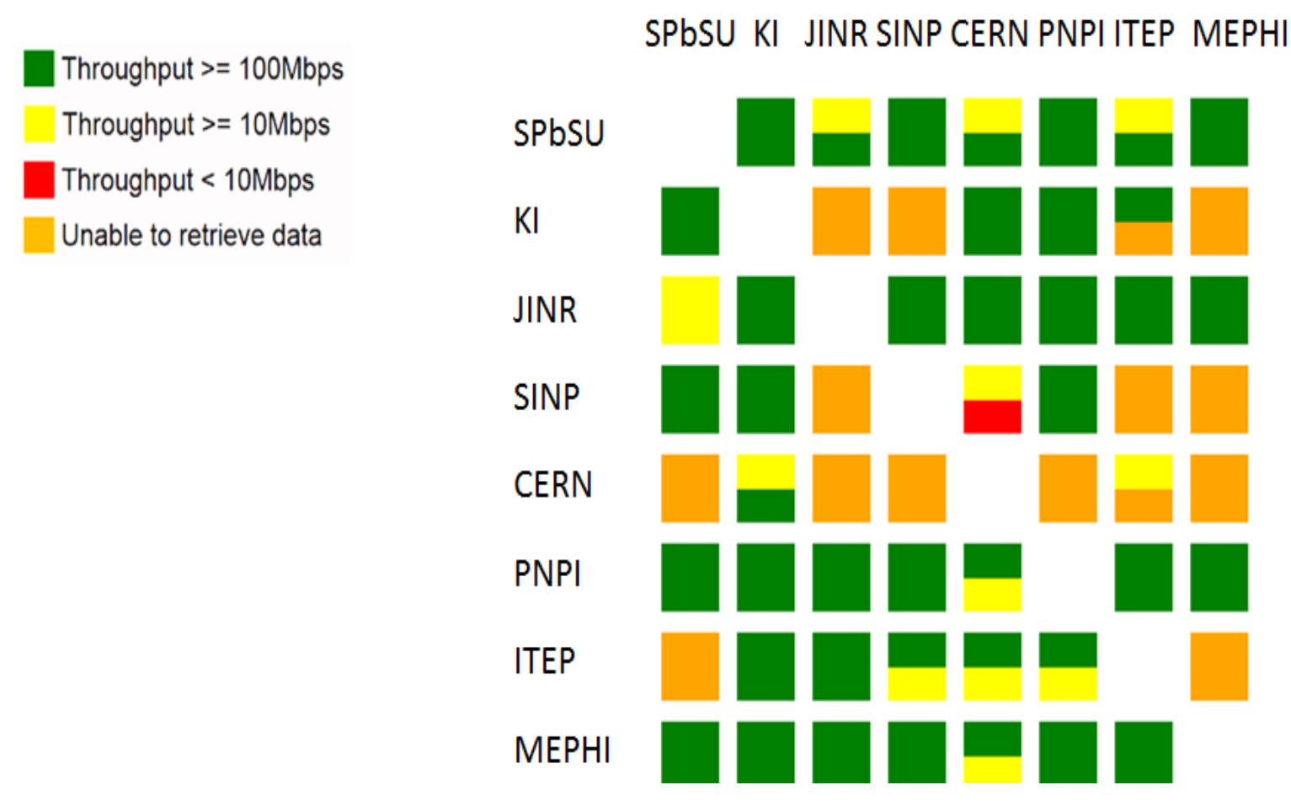
Real-life experiment tests:

- ATLAS test: standard ATLAS TRT reconstruction workflow with Athena
- ALICE test: sequential ROOT event processing

The first results with EOS. PNPI and SPbSU.



Network performance measurements



Desired policy:

Read the closest replica.

Write two replicas: first one on a closest SE (2nd or 1st level), second one on a configured 1st level SE.

EOS implementation:

Hybrid placement policy allows first replica to be placed on a closest FST and the second one “scattered” to a random FST. EOS developers think that our desired policy makes sense and may be implemented in the future.

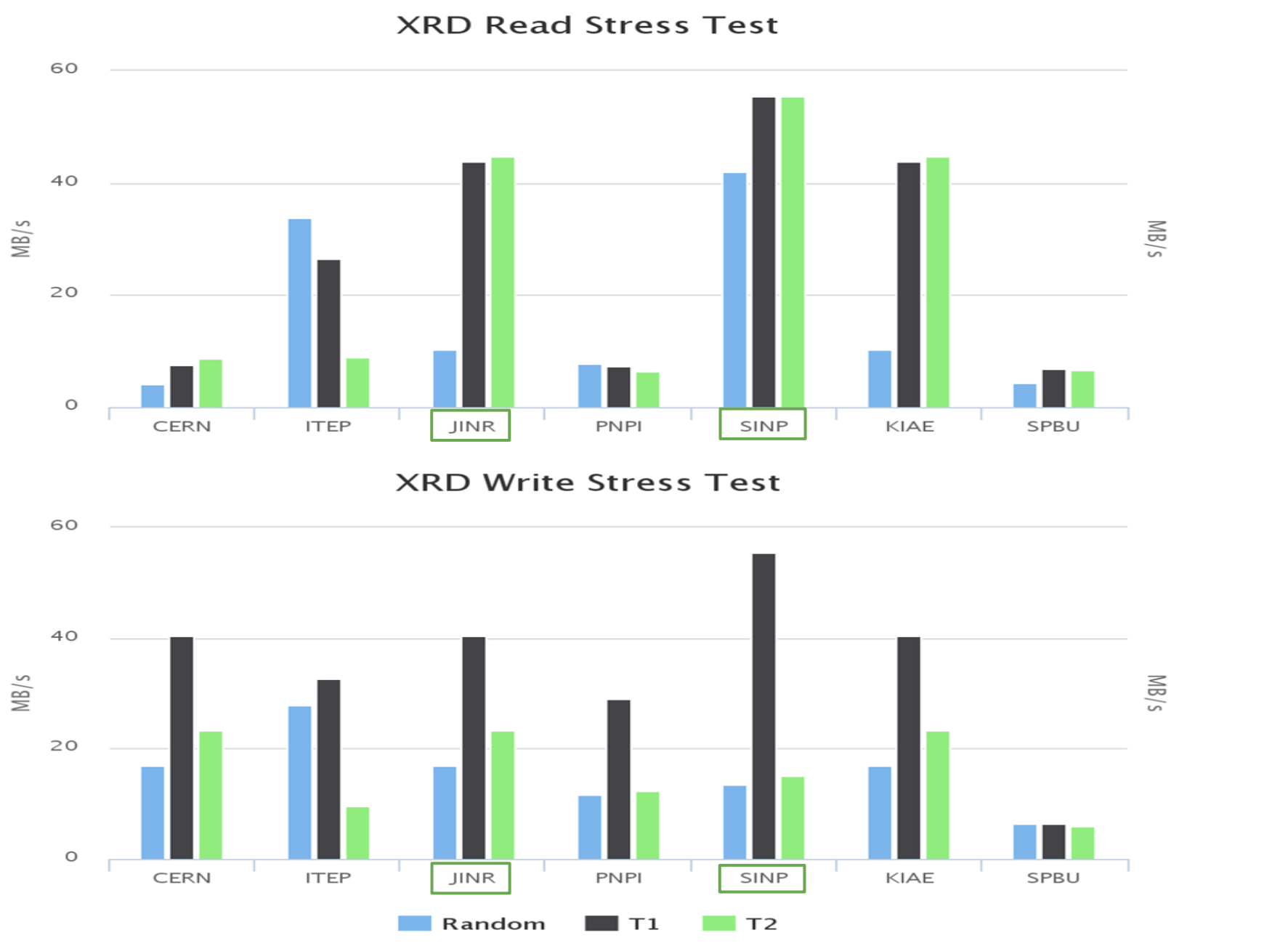
Currently we have three placement policies:

Random: Single replica without geotags: random placement;

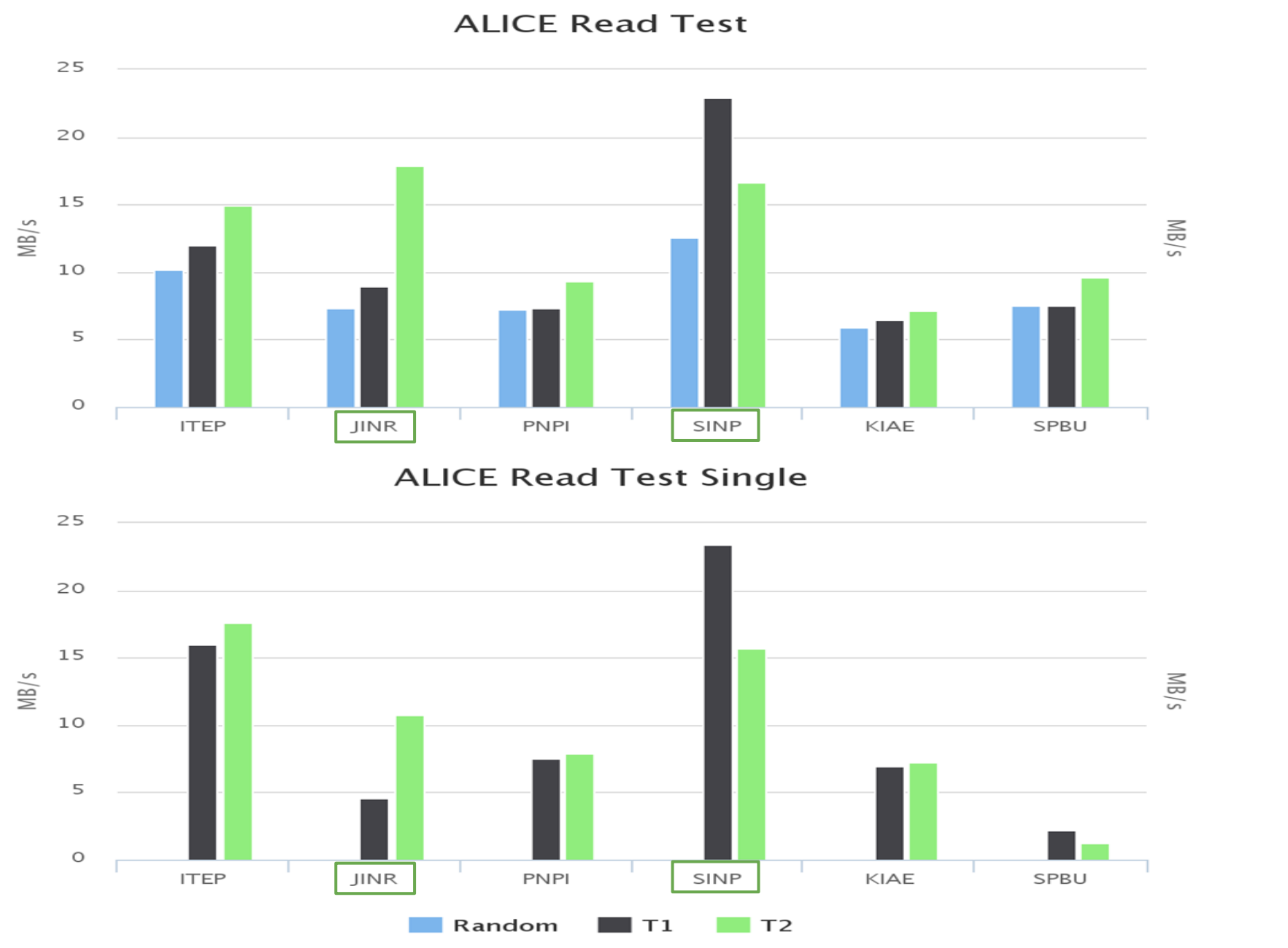
T1: Single replica with geotags: write to FST with geotag matching the UI, if there’s no match default FST is used (KIAE);

T2: Two replicas with geotags: 1st replica on a closest FST, 2nd on a random FST.

Synthetic tests with geo-tags on the Main testbed



ALICE read test with geo-tags on the Main testbed



Acknowledgements

This work was funded in part in part by the Russian Fund of Fundamental Research under contract “15-29-07942 офи_м” and U. S. DOE, Office of Science, High Energy Physics and ASCR under Contract No. DE-AC02-98CH10886 and SPbSU under research grant 11.38.242.2015. 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.