Distributed data storage and management for SPD experiment

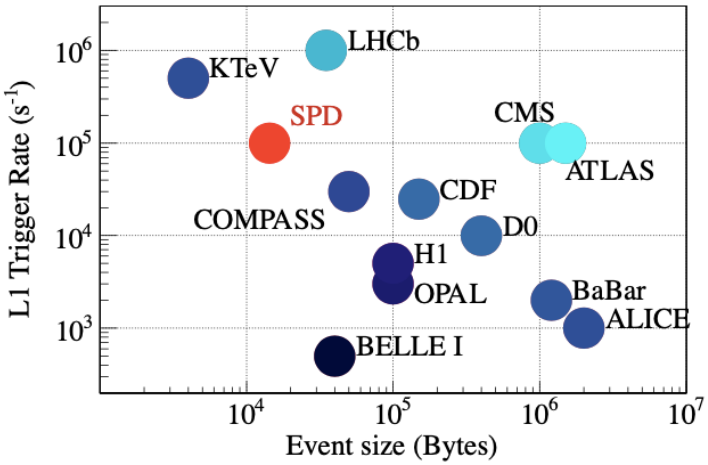
Andrey Kiryanov, NRC KI – PNPI

SPD Collaboration Meeting, 24-27 April 2023

Introduction

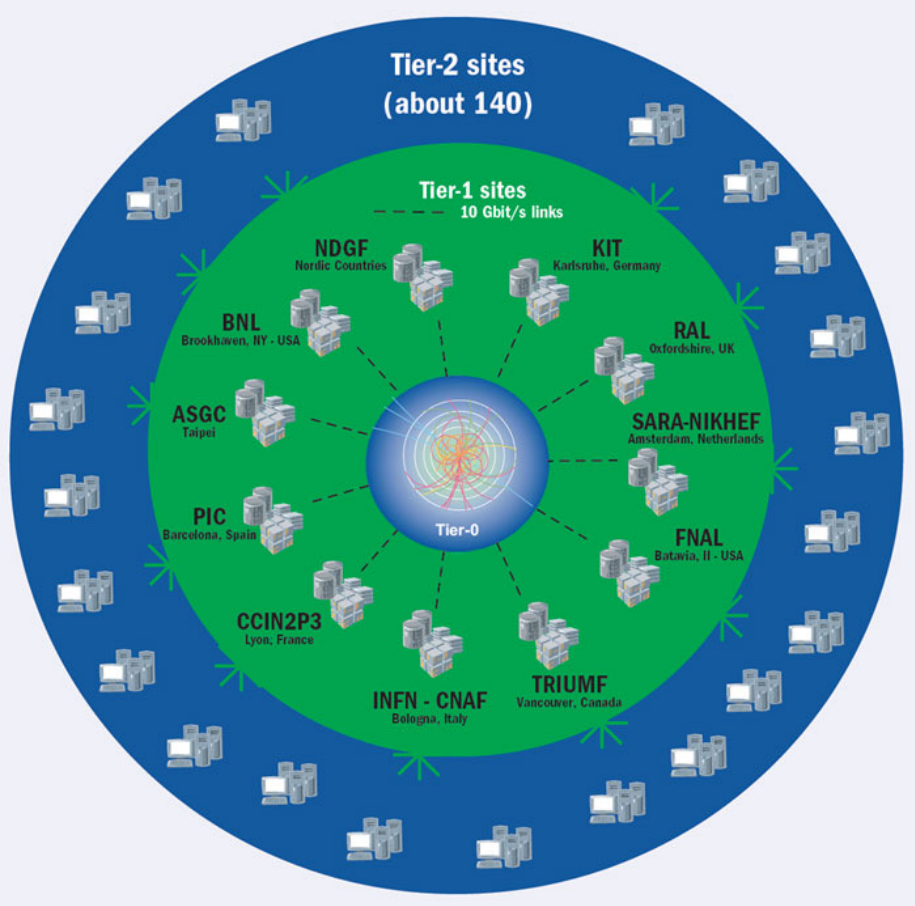
The expected event rate of the SPD experiment is about 3 MHz (pp collisions at $\sqrt{s}$ = 27 GeV and 10^{32} cm^{-2}s^{-1} design luminosity). This is equivalent to a raw data rate of 20 GB/s or 200 PB/year, assuming a detector duty cycle is 0.3, while the signal-to-background ratio is expected to be on the order of 10^{-5}. Taking into account the bunch-crossing rate of 12.5 MHz, one may conclude that pile-up probability cannot be neglected.

SPD TDR



The goal of the online filter is at least to decrease the data rate by a factor of 20, so that the annual growth of data, including the simulated samples, stays within 10 PB. Then, data are transferred to the Tier-1 facility, where a full reconstruction takes place and the data is stored permanently. The data analysis and Monte-Carlo simulation will likely run at the remote computing centers (Tier-2s). Given the large data volume, a thorough optimization of the event model and performance of the reconstruction and simulation algorithms are necessary.

Ever seen this picture?



The MONARC Project
is an acronym for
"Models of Networked Analysis at Regional Centres for LHC Experiments"

Its website is still alive!

https://monarc.web.cern.ch/MONARC/

But dates back 13th January 2000

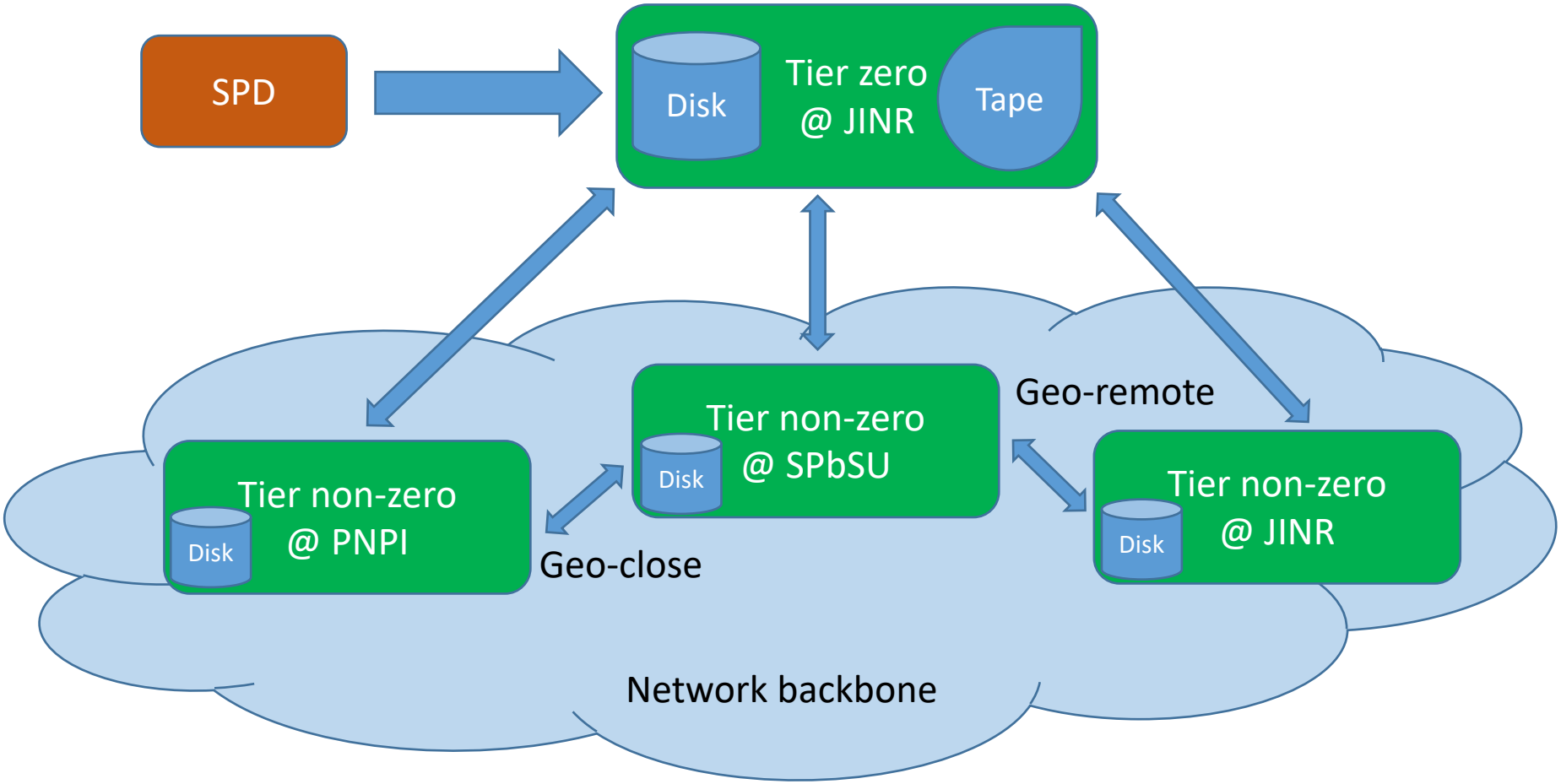
Three-tier model is 20 years old

- Started as strictly hierarchical: Tier-0 only talks to Tier-1's, etc.
- Initial implementation in WLCG followed this hierarchy precisely
  - When FTS first appeared, it was designed around a hard-defined "transfer channels" between endpoints
- Data placement defined the CPU resource allocation
- Lessons learned / changes proposed after the LHC Run 1
  - Network performance growth was significantly underestimated
  - Integration of disparate storage resources into storage federations
  - Decoupling of where the data is and where the jobs run
  - Shift from a hierarchical model towards a mesh-like

Distributed data storage

- Many experiments in various fields of science do not need this. E.g. nuclear physics experiments usually do not have enough data volume to justify for any kind of a distributed infrastructure.
- Not just a load distribution, but also redundancy and security
  - Storing hundreds of petabytes of precious data at one physical location is not the brightest idea
- Resources are to be provided by collaboration participants
- Needs a stable set of software tools, protocols, etc. in the long run
  - Authentication, authorization and accounting: who can do what (and keep the traces!)
  - Data placement policies: what goes where, how many copies
  - Data transfer protocols: how do we ship data from A to B
  - Data location and popularity: how do we find out where it is and what is the access cost

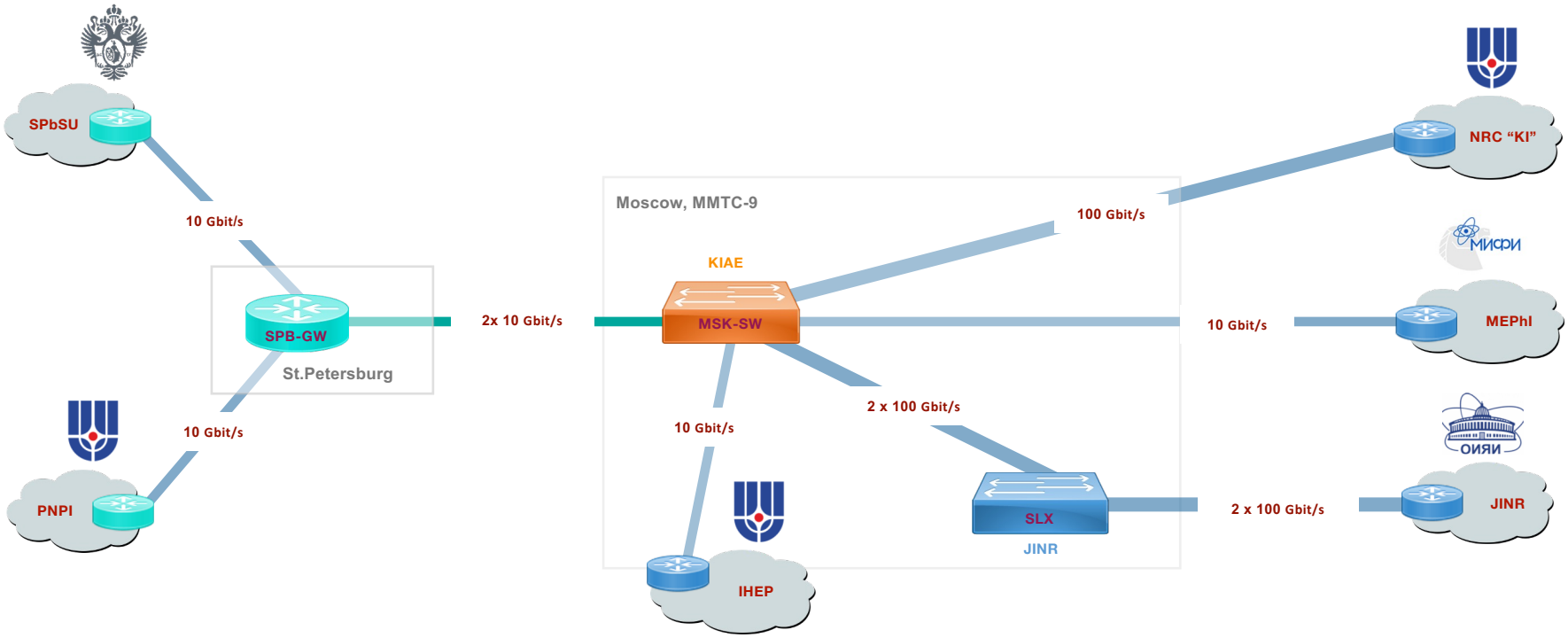
Data transfer mesh



Where do we start from

- Data volume mandates some baselines
  - >10 Gbps network per site (from TDR)
  - >500 TB storage capacity per site (not from TDR)
- Try to use existing free software as much as possible
  - Experience comes form large LCG experiments
- Minimize management effort
  - Do not deploy home-grown solutions that are different from site to site
  - Provide a reasonable guidelines for interfacing physical resources with central data management services

Russian WLCG network backbone



Data storage building blocks

- Site level
  - Storage system interface that talks necessary protocols (xroot or http/webdav): EOS
  - Local disk resource management: EOS, Ceph
- Collaboration level
  - Data location and placement service (catalog): Rucio
  - Data transfer engine: FTS
  - Authentication (not directly related to storage, but all storage-related components must be compatible with it)

Storage software (1) - Ceph

- Ceph is a very popular open-source software-defined storage platform
  - https://ceph.io/
  - Fault-tolerant
  - Manages block storage devices (HDDs and SSDs) directly
  - Runs on commodity hardware
  - Provides block- and file-level storage interfaces
  - Scales from gigabytes to exabytes
  - Maintains configured redundancy level
  - Minimizes administration time

Storage software (2) - EOS

- EOS is an open-source storage technology used at the LHC
  - https://eos-web.web.cern.ch/eos-web/
  - “EOS provides a service for storing large amounts of physics data and user files, with a focus on interactive and batch analysis”
  - Based on XRootD storage software
  - Designed to manage individual storage devices (HDDs and SSDs), requires POSIX filesystem
  - Implements a lot of bells and whistles for physics data storage
    - Geo-location awareness
    - Per-directory placement policies
    - Automatic replication
    - ACLs
    - Works with tape libraries via a CTA interface

Storage software (3) - FTS

- FTS is an open-source software for reliable and large-scale data transfers
  - https://fts.web.cern.ch/fts/
  - Automates massive transfers of file-organized datasets between storages
  - Talks multiple protocols (xroot, gridftp, http, etc.)
  - Adapts automatically to available network bandwidth
  - Scales linearly

Storage software (4) - Rucio

- Rucio is an open source data management solution
  - https://rucio.cern.ch/
  - “Rucio installation for the ATLAS Experiment is responsible for more than 450 Petabytes of data, stored in a billion files, distributed over 120 data centres globally, and orchestrating an Exabyte of data access and transfer per year.”
  - Declarative data management
  - Smart namespace with datasets and containers
  - Supports multiple storage types
  - Insights and analytics
    - Data popularity
    - Space accounting

Storage software (5)

- Aforementioned building blocks are well-tested and have shown very good reliability and performance in the WLCG
- There’s an overlap in functionality
  - Some added flexibility for the sites
  - Domains of control (site vs central)
- These blocks are enough to build a stable distributed data management system for a large scientific collaboration

Storage hardware

- We do not need expensive hardware RAIDs anymore, even in enterprise
  - Software-based redundancy is robust and much more cost-effective
  - Ceph is designed to deal with direct-attached hard drives
  - ZFS provides reliable local file system with features like snapshots
  - EOS deals easily with individual drives
- Solid-state drives get cheaper and cheaper
  - Fast buffers for disks and tapes
- Tape robots are the only remaining mammoths
  - Still cannot get rid of them in the long-term scientific data storage
  - Open source interface (CTA) exists for EOS

What’s missing?

- Specific pipelines for data processing are still missing
  - How much goes out, how much in per site
  - What are the data types and how precious they are (how many copies?)
- Pledged resource structure (initial amount, yearly growth, etc.) needs to be defined
- Participating sites will need to sign an SLA which is also missing at this point
- Funding – grants for resource providers

Thank you!