CERN

Data Storage, Management and Access evolution:

a glimpse into the future

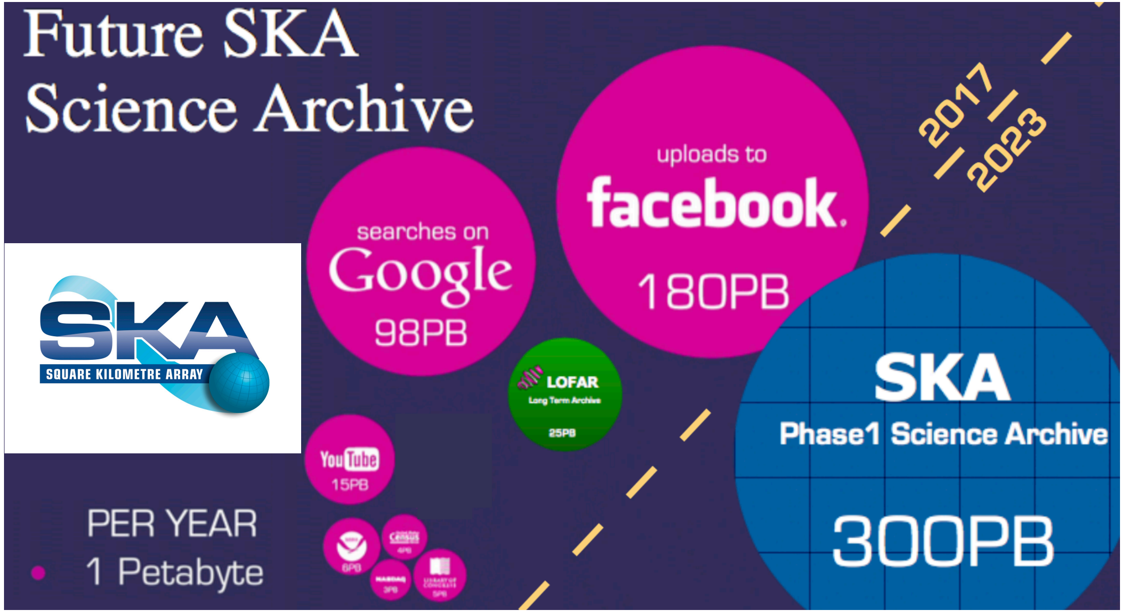
Xavier Espinal (CERN)

Outline

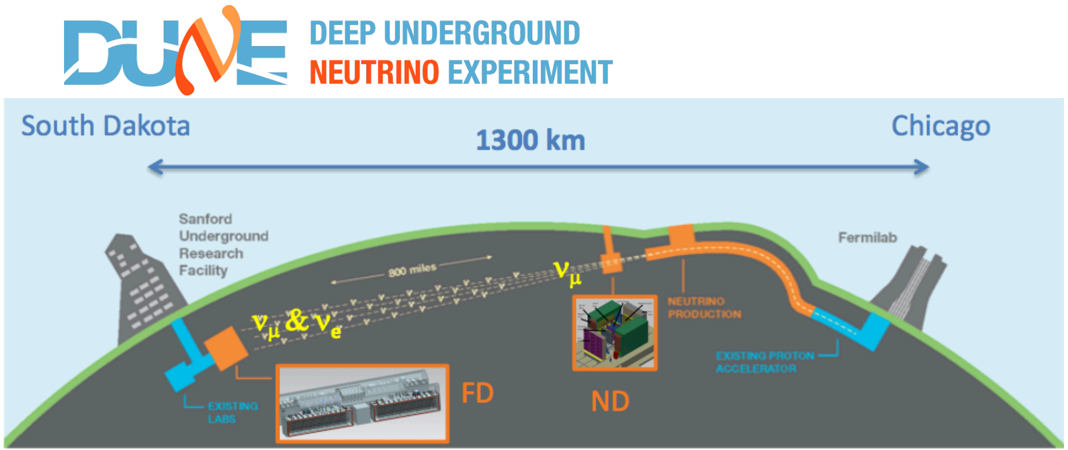
- Large scale disk storage challenges
  - Namespace (metadata) scaling
  - Space scaling
  - Cost scaling
  - Understanding applications and access patterns
- Storage paradigm shifting
  - Data federations evolution
  - Backup and archive
  - Flexibility and adaptability: IaaS and HPC
- Analysis, re-analysis and knowledge preservation
  - Analysis preservation
  - New analysis trends

The motivation

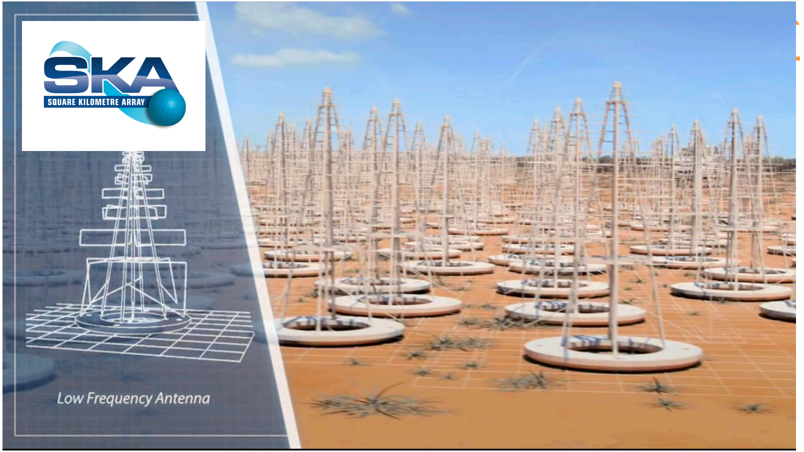
- A future change of scale in data volumes is common to all scientific communities: physics, astrophysics, cosmology
- More data not only means more bytes. Classic scaling solutions do not apply anymore


Future SKA
Science Archive
SKA
SQUARE KILOMETRE ARRAY
PER YEAR
● 1 Petabyte
searches on
Google
98PB
uploads to
facebook.
180PB
2017
2023
LOFAR
Long Term Archive
25PB
YouTube
15PB
6PB
Census
4PB
NASDAQ
3PB
LIBRARY OF CONGRESS
5PB
SKA
Phase1 Science Archive
300PB

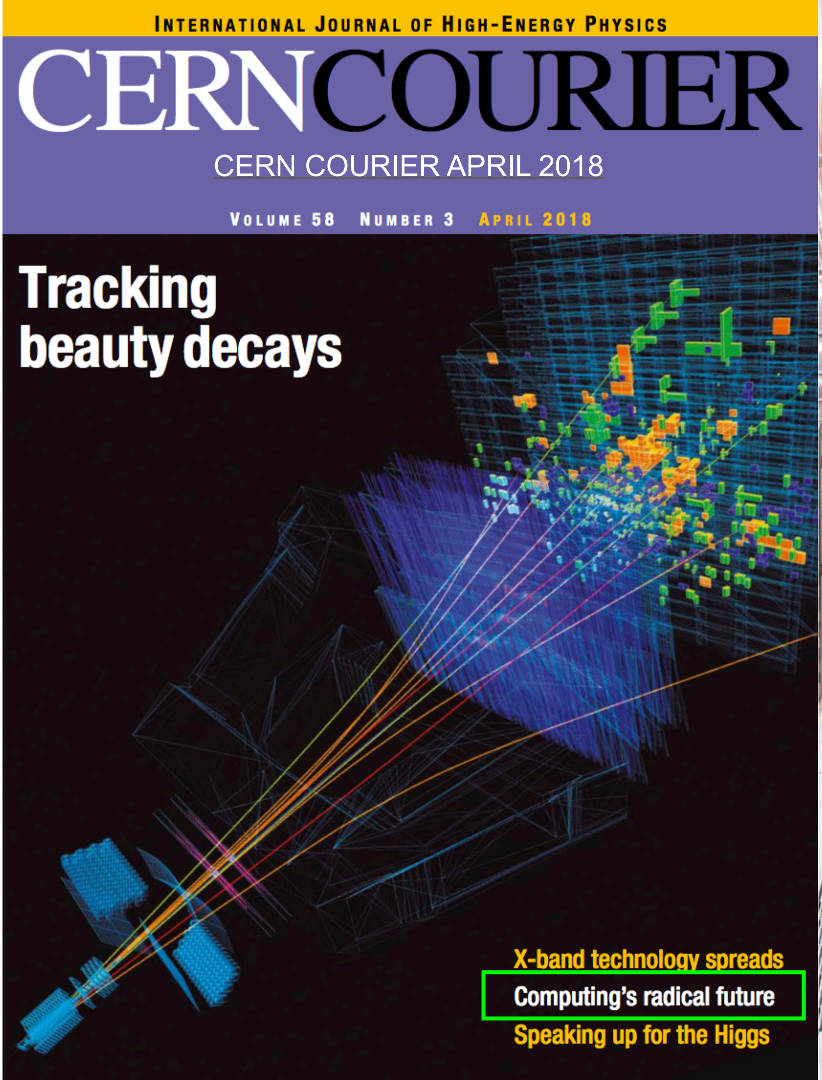
LSST
20PB/night


DUNE
DEEP UNDERGROUND
NEUTRINO EXPERIMENT
South Dakota
1300 km
Chicago
Sanford
Underground
Research
Facility
800 miles
ν_μ & ν_e
ν_μ
NEUTRINO
PRODUCTION
Fermilab
EXISTING PROTON
ACCELERATOR
EXISTING
LABS
FD
ND

Nati


SKA
SQUARE KILOMETRE ARRAY
Low Frequency Antenna

tri



INTERNATIONAL JOURNAL OF HIGH-ENERGY PHYSICS

CERNCOURIER

CERN COURIER APRIL 2018

VOLUME 58 NUMBER 3 APRIL 2018

Tracking beauty decays

X-band technology spreads

Computing's radical future

Speaking up for the Higgs

Time to adapt for big data

Radical changes in computing and software are required to ensure the success of the LHC and other high-energy physics experiments into the 2020s, argues a new report.

It would be impossible for anyone to conceive of carrying out a particle-physics experiment today without the use of computers and software. Since the 1960s, high-energy physicists have pioneered the use of computers for data acquisition, simulation and analysis. This hasn't just accelerated progress in the field, but driven computing technology generally – from the development of the World Wide Web at CERN to the massive distributed resources of the Worldwide LHC Computing Grid (WLCG) that supports the LHC experiments. For many years these developments and the increasing complexity of data analysis rode a wave of hardware improvements that saw computers get faster every year. However, those blissful days of relying on Moore's law are now well behind us (see panel overleaf), and this has major ramifications for our field.

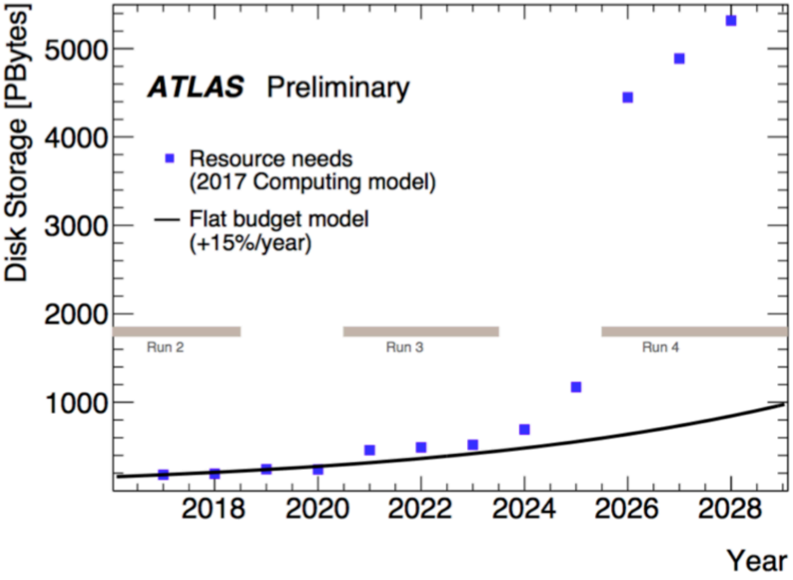
The high-luminosity upgrade of the LHC (HL-LHC), due to enter operation in the mid-2020s, will push the frontiers of accelerator and detector technology, bringing enormous challenges to software and computing (*CERN Courier* October 2017 p5). The scale of the HL-LHC data challenge is staggering: the machine will collect almost 25 times more data than the LHC has produced up to now, and the total LHC dataset (which already stands at almost 1 exabyte) will grow many times larger. If the LHC's ATLAS and CMS experiments project their current computing models to Run 4 of the LHC in 2026, the CPU and disk space required will jump by between a factor of 20 to 40 (figures 1 and 2).

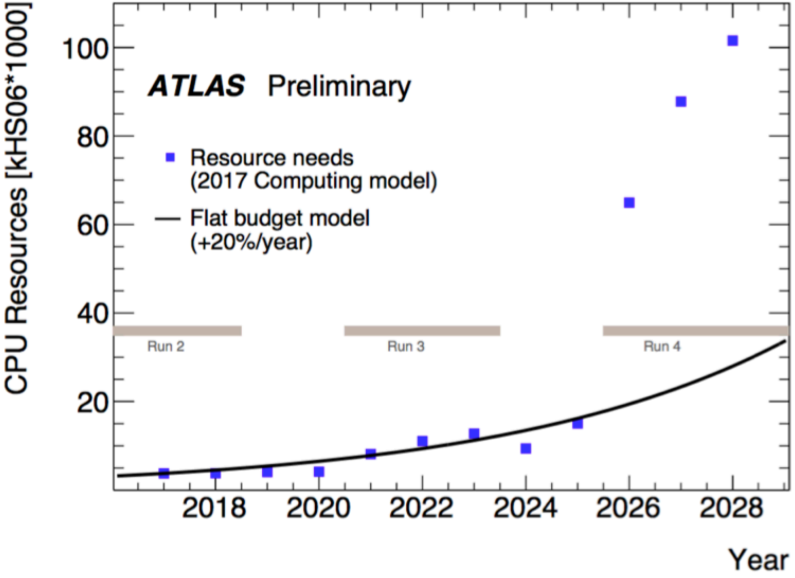
Even with optimistic projections of technological improvements there would be a huge shortfall in computing resources. The WLCG hardware budget is already around 100 million Swiss francs per year and, given the changing nature of computing hardware and slowing technological gains, it is out of the question to simply throw

Inside the CERN computer centre in 2017. (Image credit: J Ordan/CERN.)

The motivation

- Future storage needs are above the expected technology evolution (15%/yr) and funding (flat)





Document Classification: Restricted

Large scale disk storage

The namespace
(metadata) challenge

- Namespace scaling
  - Metadata is the first interface with the storage system
    - User experience is largely dominated by its coherence and speed
  - O(Billions) files
    - Data *catalogues* in future HEP, Astro and Cosmo and user data (~0.5B files/yr increase at CERN only for users)
  - Architecture: central vs distributed metadata storage?
    - DDBB, KV store, *CRUSH*-like maps,…
- Namespace accessibility
  - Single global namespace? metadata federations?
  - Data easily browsable/findable by experiments and users: AnydataAnytimeAnywhere
    - Ideally: de-localization, no privileged places for metadata ops

Large scale disk storage

- Space scaling
  - Data volumes moving towards the **EB** scale
  - Disks getting **big** (20TB+) and diskserver market favouring high density servers (1PB+/4U)
  - Adding **capacity** is part of day-by-day operations: should not be a scalability limit in the number of diskservers.
    - Lightweight namespace-diskserver orchestration (messaging, notification, journaling,…)
  - Hardware **lifecycle** is aggressive: space density (TB/m^2) and power efficiency (TB/kW) keep increasing
    - Diskerver replacements as standard operations and transparent to users: keeping data available with efficient draining and rebalancing mechanisms
- Space access: scaling-up guaranteeing accessibility
  - Protocols and interoperability, data maintenance (draining/balancing), data healing and caches

Large scale disk storage

- Cost Scaling
  - Redundancy:
    - RAIDs are dead (disks too big for rebuilt time) and redundancy on a single server pose bandwidth limitiations
    - Duplication solves the single-location problem but cost increases
    - **E**rasure **C**oding (RAIN) could be a potential solution. But at which cost?
      - *Fat* diskservers and increased LAN traffic impact NICs, TORs and Routers
  - Managing expectations:
    - Need to know what our stakeholders want: less data and more reliable or more date but less reliable?
      - 100PB of data at 10^{-5} anual-reliablity or 200PB at 10^{-4} anual-reliablity? … **or a mix of both?**

Large scale disk storage

The tunning challenge

- Application knowledge
  - Understading the access **patterns** is fundamental to tailor a service, ie. HPC centers invest a lot to align code to maximize resources exploitation
  - Many different **workflows** are needed in HEP before getting the final data products for scientists
    - And access patterns are very different, from nearly zero I/O and pure CPU for montecarlo (*HPC-like*) to intense I/O for reconstruction (*HTC-like*)
  - Can a single storage **system** provide High Throughput (HT) and High IOPS?
  - Can a single **hardware** provide HT and High IOPS (keeping costs under control)?
  - Should shared **filesystems** be treated different?
    - Home directories requiring high posix compliance, checkpointing capabilities and “infinite” uptime
    - Is there a current *optimal* “solution”?
      - CEPH-FS, Lustre, GPFS, NFS appliance servers,…?

Document Classification: Restricted

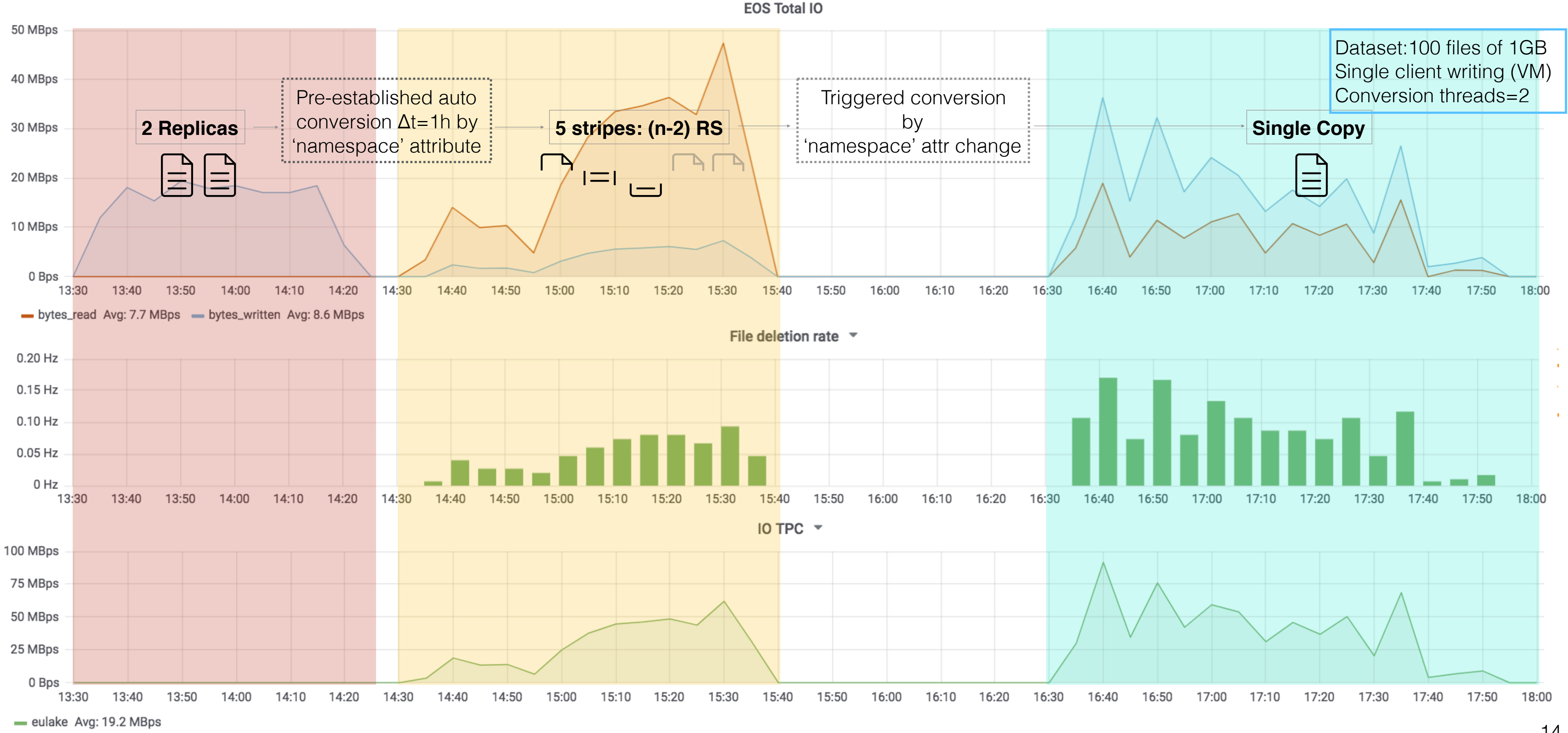
Storage paradigms shifting

Storage paradigms shifting

- Re-evaluate (or give-up) on local disk redundancy:
  - RAID, duplication, EC extra costs
  - Data can be reproduced, except RAW data (primary data coming from the detectors). Candidate for tape archive.
  - Reproducing data costs money (CPU cycles) but how much in comparison with the potential gain in storing more data?
    - ~1% of anual disks failure rate (for 100k disks installation -> 3 disks failures per day)
- Leverage *byte-costs* by QoS (Quality of Service)
  - Data gets cold with time. Likelihood to be accessed decreases rapidly.
  - File workflows orchestration? ie. from 2 replicas to EC (8+3) to tape (or cost equivalent) backup
- Data federations: concentrate big storage services on few sites and push for high performance I/O centres driven by data caching and latency hidding mechanisms
  - Maintain caches require less effort (stateless service) and resources could be re-oriented to computing infratsructure

Storage paradigms shifting

Distributed redundancy
and QoS example



Dataset:100 files of 1GB
Single client writing (VM)
Conversion threads=2

2 Replicas

Pre-established auto conversion Δt=1h by 'namespace' attribute

5 stripes: (n-2) RS

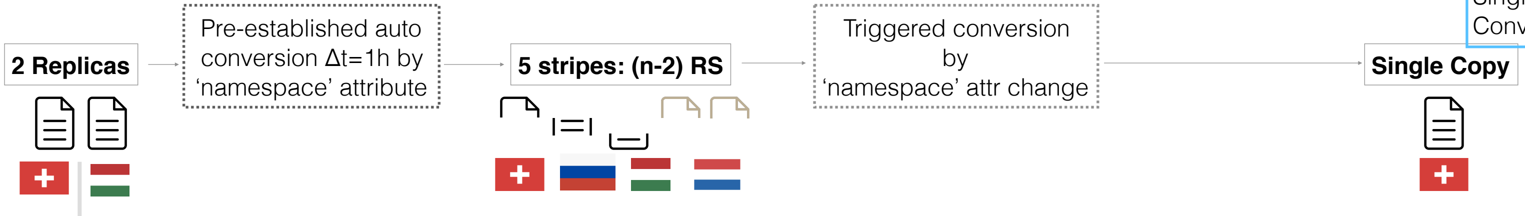
Triggered conversion by 'namespace' attr change

Single Copy

Storage paradigms shifting

Distributed redundancy
and QoS example

Dataset:100 files of 1GB
Single client writing (VM)
Conversion threads=2


2 Replicas
Pre-established auto conversion Δt=1h by ‘namespace’ attribute
5 stripes: (n-2) RS
Triggered conversion by ‘namespace’ attr change
Single Copy

```
180315 14:04:36 func=open path=/eulake/lcg/test/conversion/2replicas-to-rain32/file-workflow-2r-rain32.175.file
op=write target[0]=(p05799459m56401.cern.ch,33) target[1]=(p05798818t49625.cern.ch,80)

                    180315 15:04:58 time=1521123718.328306 func=open path=/eulake/lcg/test/conversion/2replicas-to-rain32/file-workflow-2r-rain32.175.file
                    op=read  target[0]=(p05799459m56401.cern.ch,33) target[1]=(p05798818t49625.cern.ch,80)

180315 15:04:58 func=open path=/eos/eulake/proc/conversion/0000000000001819:default#20640442
op=write  eos.layout.nstripes=5&eos.layout.type=raid6
target[0]=(fst2.grid.surfsara.nl,130) target[1]=(p05496644k62259.cern.ch,1) target[2]=(dvl-mb01.jinr.ru,122) target[3]=(p05798818t49625.cern.ch,97)
target[4]=(fst1.grid.surfsara.nl,124)

                    180315 17:22:17 func=open path=/eulake/lcg/test/conversion/2replicas-to-rain32/file-workflow-2r-rain32.175.file
                    op=read  target[0]=(fst2.grid.surfsara.nl,130) target[1]=(p05496644k62259.cern.ch,1) target[2]=(dvl-mb01.jinr.ru,122)
                    target[3]=(p05798818t49625.cern.ch,97)

180315 17:22:17 func=open path=/eos/eulake/proc/conversion/00000000000018e2:default#00100001
op=write eos.layout.nstripes=1&eos.layout.type=plain tpc.stage=copy  redirection=p05799459m56401.cern.ch?
```

Archive technologies (tapes & co.)

- Tape is *roughly* 1/4 cheaper than disks. Easy argument to gain x4 in storage(?)
  - True. Tape is *fast* but single *stream*. Doing OK in orchestrated workflows, suffer with random access. Probably need to stay as cold/nearline storage
- Tape is known as reliable and users stick to this idea. Reluctant to change
  - But double replicas in different disks also provides extremely good reliablity
- Tape market was shaken by *O* dropping out and *I* taking the lead (tape density battle)
  - LTO-only future. Recent findings hinting much better positioning+seek performance (LTO-8)
- Tape evolution under the spotlight? is there interest in increasing TB/in^2 density? will still be market for tape in 10+ year time?
  - My personal take: YES, but the market is changing as main customers already changed
- Periodical Media Change (repack) is still a heavy process
  - Bring 100PB from tape to disk and to tape again took 7 months at CERN with a new infrastructure with SSDs in front of tapes to speed up the campaign
- Rumors for big SSD nodes (~1PB/U). Good for WORN approach and Wake-on-LAN? but will they endure enough for long term archival? A real alternative for us?
- DNA storage not for 2026

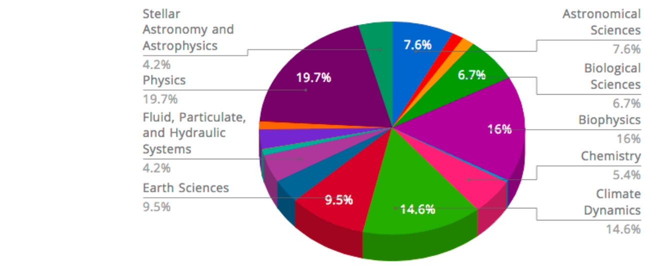
IaaS: could this be the solution?

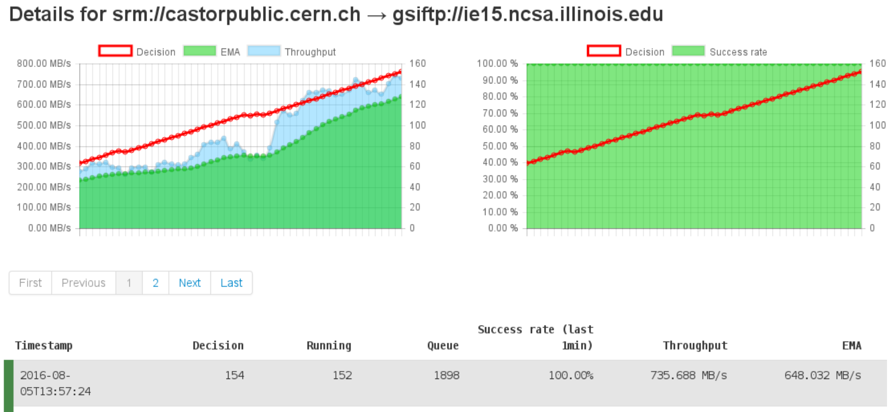
- Evaluated and continue being evaluated in HEP community
- Successful projects with main LHC experiments
  - Interoperablity is ready (HTCondor integration)
- Perceived as a good mechanism for handling unforeseen workloads
  - Maximal exploitation of local resources remains the priority
  - IaaS reserved instances could be an option for expected (if any) computing capacity gaps
  - On-demand IaaS (*stock market*) could be an option for emergency computing
- Benefits from IaaS not clear as largely depends on: providers, type of workflows, performance and market evolution

HPC and HTC: Bringing T closer to P

- Common interest and implication from experiments and HPC centers.
  - CSCS in the front line alredy successfully running HEP workflows
- Proven for simulation/montecarlo. What about data intensive workloads?
  - Active caching for latency hiding
  - Smart application access by optimizing data structures
  - Efficient workload orchestration (maximising cache efficiencies)

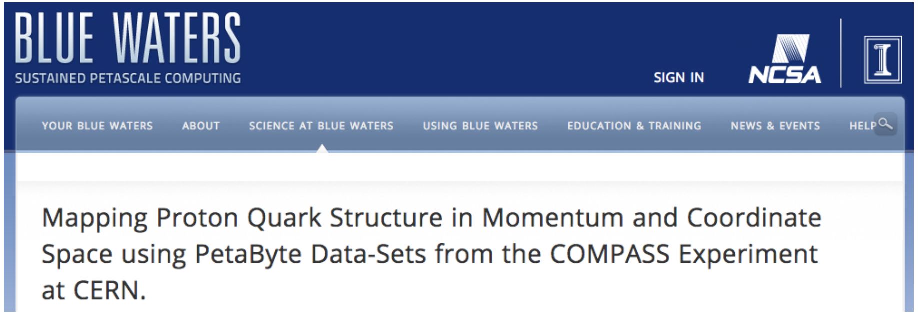
CURRENT RUNNING JOBS BY SCIENCE AREA





Details for srm://castorpublic.cern.ch → gsiftp://ie15.ncsa.illinois.edu

Timestamp	Decision	Running	Queue	Success rate (last 1min)	Throughput	EMA
2016-08-05T13:57:24	154	152	1898	100.00%	735.688 MB/s	648.032 MB/s



BLUE WATERS

SUSTAINED PETASCALE COMPUTING

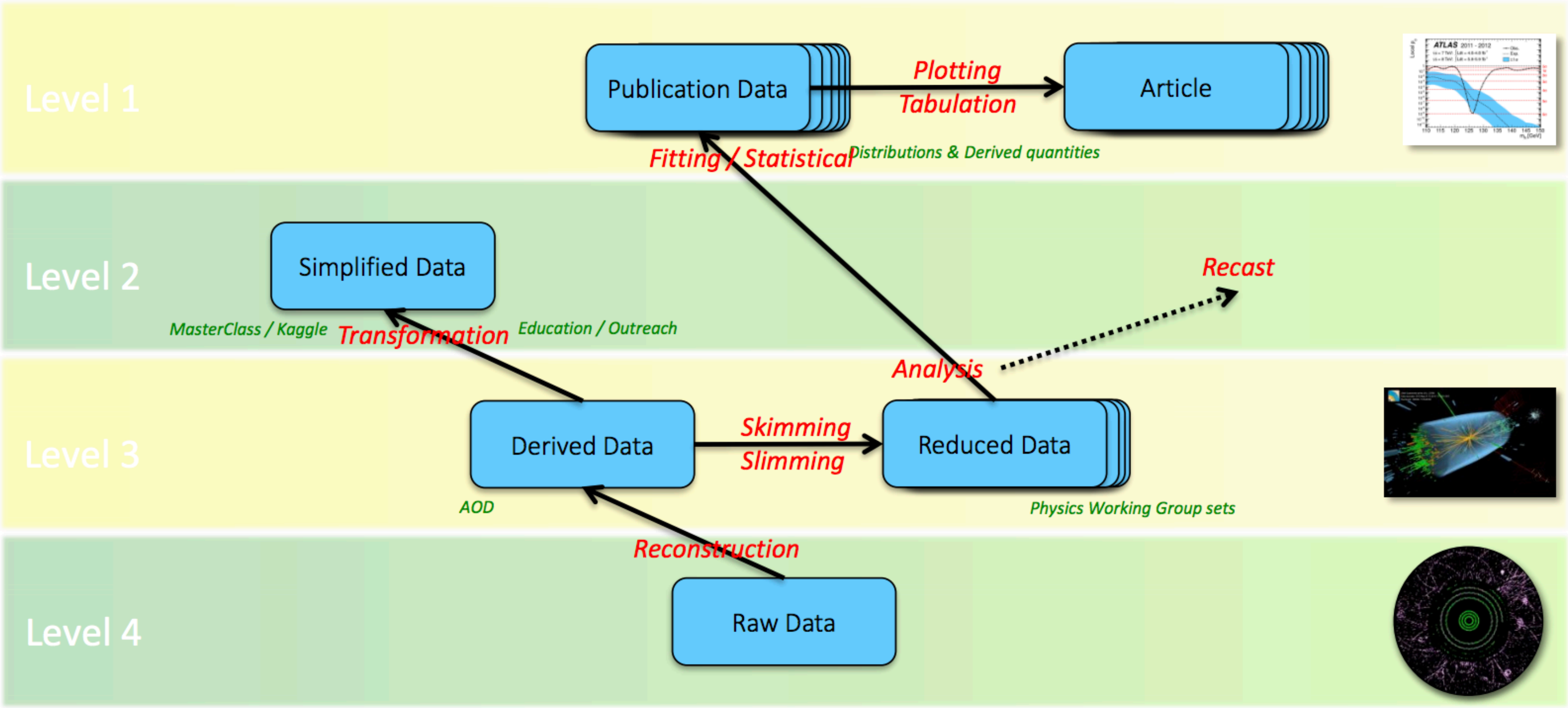
Mapping Proton Quark Structure in Momentum and Coordinate Space using PetaByte Data-Sets from the COMPASS Experiment at CERN.

(re)analysis and knowledge preservation

- Preservation of data
- Reusability of data
- Reroducibility of results

Document Classification: **Restricted**

(re)analysis and knowledge preservation



Level 1

Publication Data

Plotting

Tabulation

Article

Fitting / Statistical

Distributions & Derived quantities

Level 2

Simplified Data

Recast

MasterClass / Kaggle

Transformation

Education / Outreach

Level 3

Analysis

Derived Data

Skimming

Slimming

Reduced Data

AOD

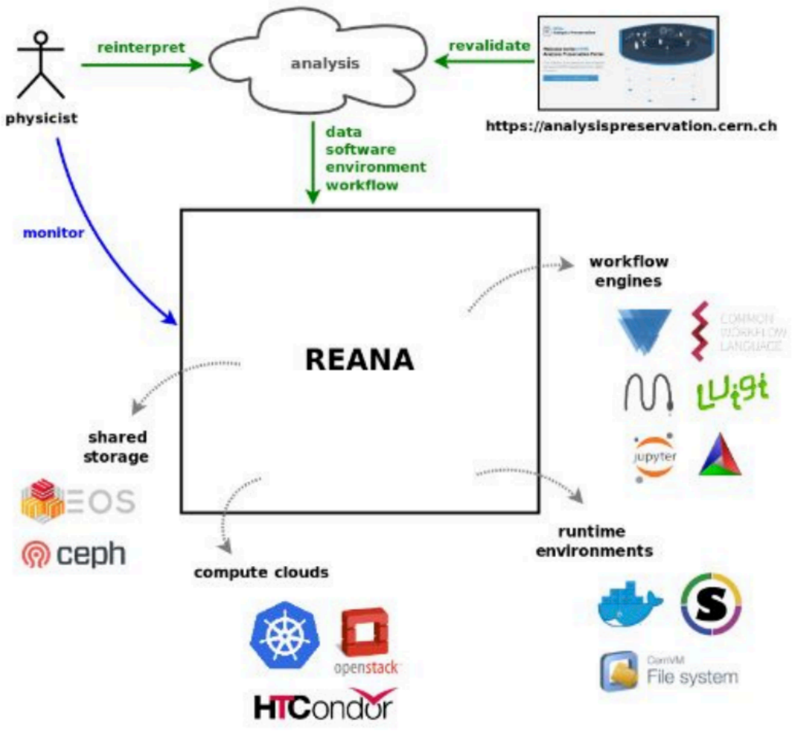
Physics Working Group sets

Level 4

Reconstruction

Raw Data

(re)analysis and knowledge preservation



http://www.reanahub.io/

reana

Reproducible research data analysis platform

Flexible

Run many computational workflow engines.

Scalable

Support for remote compute clouds.

Reusable

Containerise once, reuse elsewhere. Cloud-native.

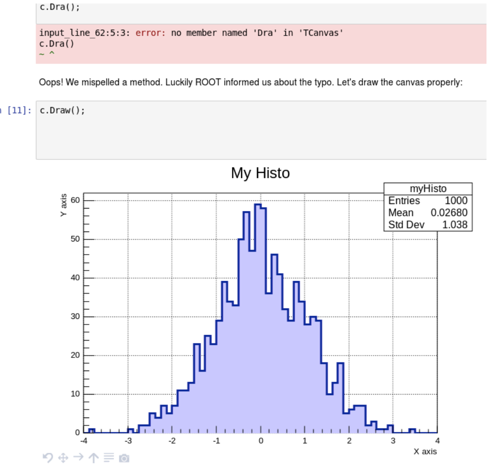
Free

Free Software. GPL licence. Made with ❤ at CERN.

CERN Analysis Preservation & REANA Workshop (30/06/2018)
https://indico.cern.ch/event/720455/

New ways of accessing data: notebooks

https://swan.web.cern.ch/



```
c.Dra();
```

```
input_line_62:5:3: error: no member named 'Dra' in 'TCanvas'
c.Dra()
~ ^
```

Oops! We mispelled a method. Luckily ROOT informed us about the typo. Let's draw the canvas properly:

```
In [11]: c.Draw();
```

ROOT
Data Analysis Framework

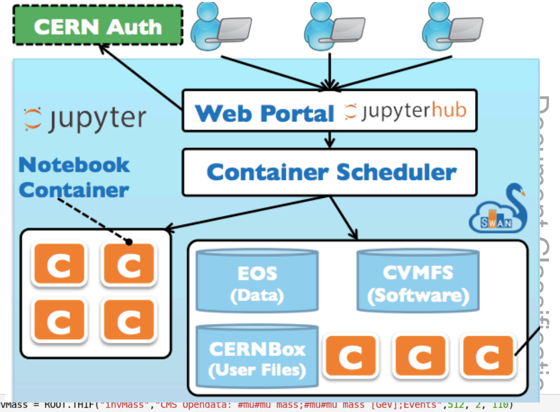
CERNBox

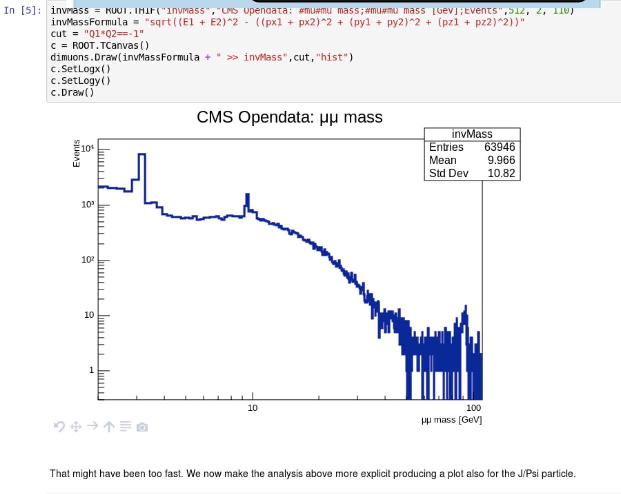
jupyterhub

SWAN

EOS

CERN





```
In [5]: invMass = ROOT.TH1F("invMass","CMS Opendata: #mu#mu mass;#mu#mu mass [GeV];Events",512, 2, 110)
        invMassFormula = "sqrt((E1 + E2)^2 - ((px1 + px2)^2 + (py1 + py2)^2 + (pz1 + pz2)^2))"
        cut = "Q1*Q2==-1"
        c = ROOT.TCanvas()
        dimuons.Draw(invMassFormula + " >> invMass",cut,"hist")
        c.SetLogx()
        c.SetLogy()
        c.Draw()
```

That might have been too fast. We now make the analysis above more explicit producing a plot also for the J/Psi particle.

Web based **computing interface** combining: **data**, **code**, **equations**, text and **visualisation**

Summary

- Future scientific computing scenario force us to **re-evaluate** the current model
  - How we understand data storage
  - How we understand data access
  - How we understand data preservation
- Storage technology trends and funding not helping
- Revisiting **redundancy**, **caching, interoperability** and **reproducibility** should give us some of the hints to address the future of data storage in scientific computing
- Dedicated working groups starting **now** to set direction and start R&D projects:
  - Content delivery and caching (latency hidding, bandwidth and space optimization)
  - Protocols (http/xrootd/tpc) and networks (tcp/udp, DTNs)
  - Interoperability and quality of services in storage systems