DODAS @ CMS

Daniele Spiga

HPC4L - Training. Beirut, Lebanon
21th22th October 2020

WLCG A Success Story

4/15/2014 17:39:22

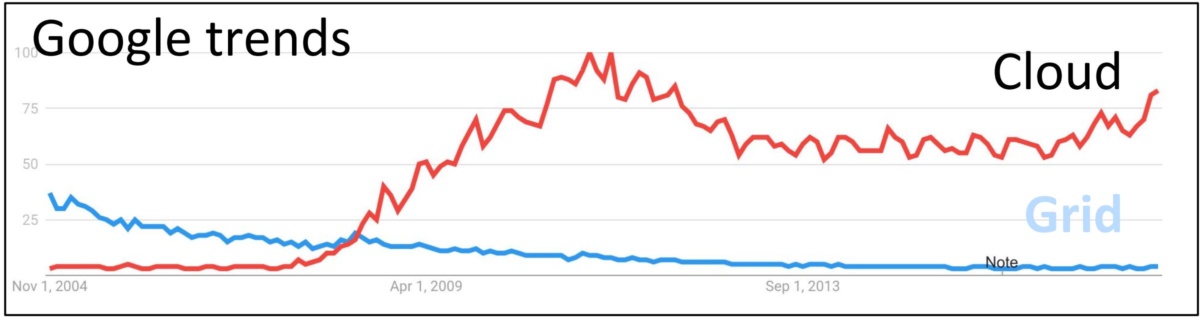
Running jobs: 262487
Transfer rate: 11.61 GiB/sec

CERN July 4th 2012

UKI-SCOTGRID-ECDF
UKI-SCOTGRID-GLASGOW
UKI-SCOTGRID-DURHAM
UKI-NORTHGRID-LANCS-HEP
UKI-NORTHGRID-MAN-HEP
UKI-NORTHGRID-SHEF-HEP
UKI-SOUTHGRID-BHAM-HEP
UKI-SOUTHGRID-CAM-HEP
EFDA-JET
UKI-LT2-BRUNEL
UKI-SOUTHGRID-SUSX
NDGF-T1
DESY-HH
RUG-CIT
DESY-ZN
PSNC
ICM
UNI-DORTMUND
GOEGRID
WUPPERTALPROD
UNI-SIEGEN-HEP
BELGRID-UCL
RWTH-AACHEN
WCSS64
TUDRESDEN-ZIH
GSI-LCG2
PRAGUELCG2
CYFRONET-LCG2
IFJ-PAN-BG
GRIF
ITWM
FZK-LCG2
IN2P3-IRES
IEPSAS-KOSICE
IN2P3-SUBATECH
UNI-FREIBURG
LRZ-LMU
MPPMU
FMPHI-UNIBA
T3_CH_PSI
BMEGRID
HEPHY-UIBK
RO-16-UAIC
RO-14-ITIM
AUVERGRID
CERN-PROD
IN2P3-LPC
IN2P3-LAPP
CSCS-LCG2
IN2P3-LPSC
INFN-MIB
INFN-PADOVA
INFN-TRIESTE
EGEE.SRCE.HR
INFN-PAVIA
INFN-LNL-2
INFN-FERRARA
RO-02-NIPNE
RO-11-NIPNE
USC-LCG2
IFCA-LCG2
INFN-BOLOGNA-T3
INFN-T1
IN2P3-CPPM
SNS-PISA
INFN-PISA
EGEE.FESB.HR
LIP-COIMBRA
BIFI
INFN-PERUGIA
BG01-IPP
TR-10-ULAKBIM
TR-03-METU
CIEMAT-LCG2
UAM-LCG2
IFAE
UB-LCG2
NCG-INGRID-PT
INFN-ROMA3
INFN-ROMA1
INFN-BARI
GR-01-AUTH
IFIC-LCG2
GRISU-UNINA
INFN-NAPOLI-ATLAS
INFN-LECCE
GR-07-UOI-HEPLAB
INFN-CAGLIARI
GRISU-CYBERSAR-CAGLIARI
INFN-COSENZA
HG-06-EKT
INFN-CATANIA
FI_HIP_T2
T2_ESTONIA
RU-PNPI
ALICE
LHCb
dashboard
Google earth
US Dept of State Geographer
© 2014 Google
Map Data © 2014 AND
Image Landsat

So what

On one side technology evolves

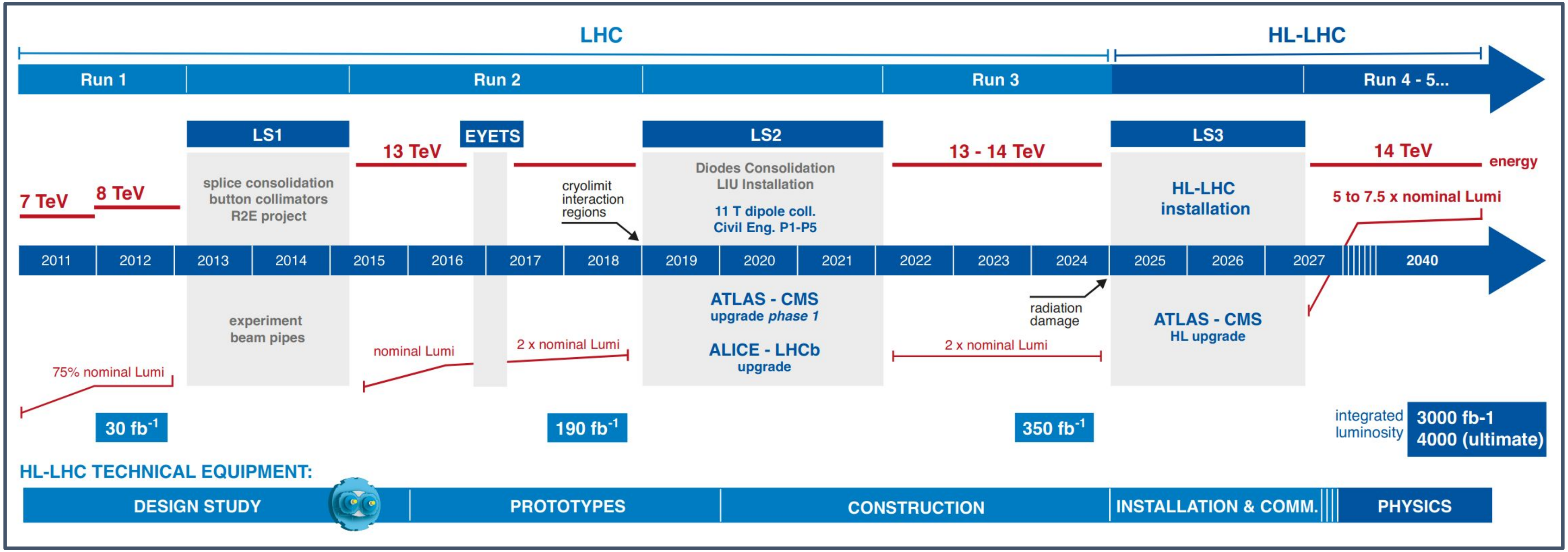


It is in this context that we developed a solution to “easily” bring Clouds to CMS system..

DODAS

LHC Schedule

On the other side the computing needs change (increase)



CPU and DISK needs

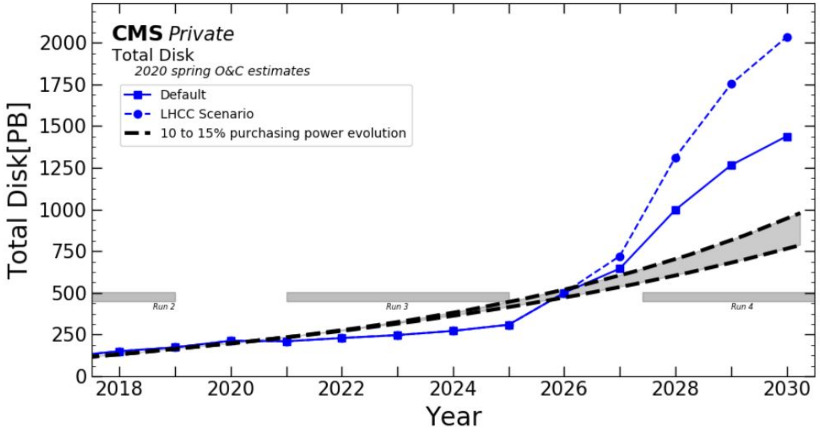
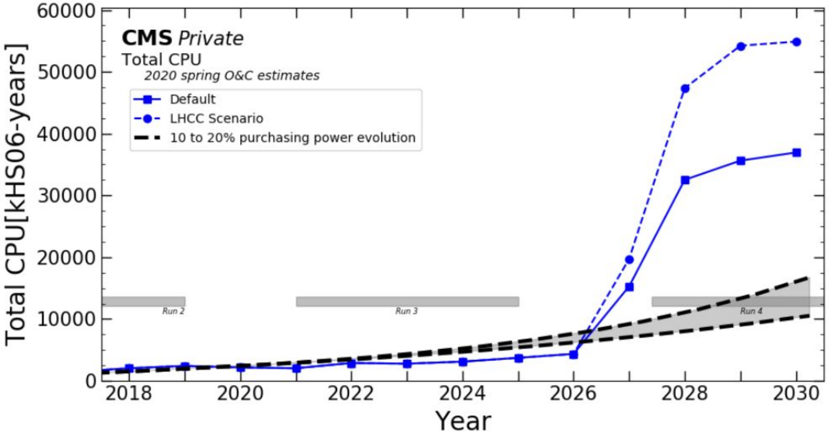


Figure 5: Estimated evolution of CMS disk resource needs (summed over all tiers) vs. time through Run-4.



CMS (2030) will need disk which is still 2x with respect to what we will be able to buy and 3x CPU with respect to what we will be able to buy

How to address these challenges

R&D and new ideas

So we need two main ingredients :

- R&D on infrastructure, software, analysis models
- Ideas on how to improves the current systems as well as on how to introduce new approaches and solutions, at any level

No we focus on Infrastructural R&D currently on-going in our domain

- CMS/HEP/WLCG

Computing Infrastructure R&D

Vision for evolution of CMS (WLCG) infrastructure

- Will evolve towards a data lake model
- The Data Lake model currently discussed in WLCG makes clean regional distinctions (US Region / EU Region)

Key aspects in the model are caching and content delivery solutions

- **Today we have opportunities learns about best uses of caching**
  - We will come to this later

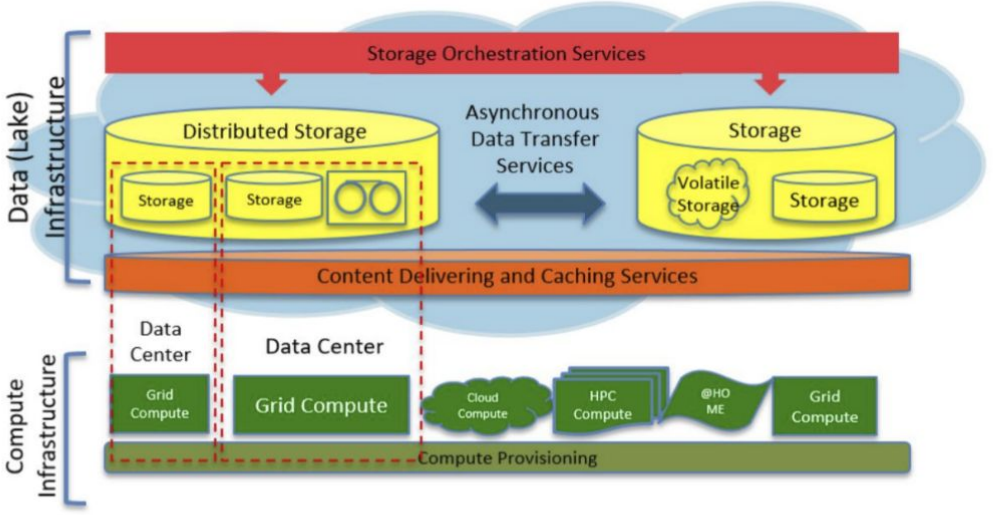
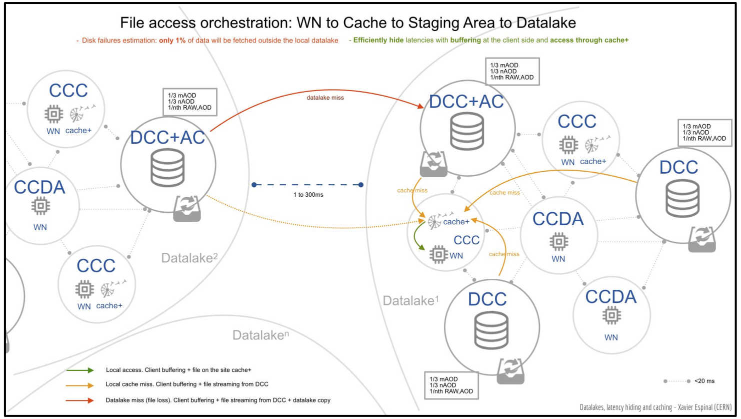


Figure 20: The data lake design from DOMA/WLCG.

Something more on Data Lake

From Data Lake Straw Proposal @ DOMA



File access orchestration: WN to Cache to Staging Area to Datalake

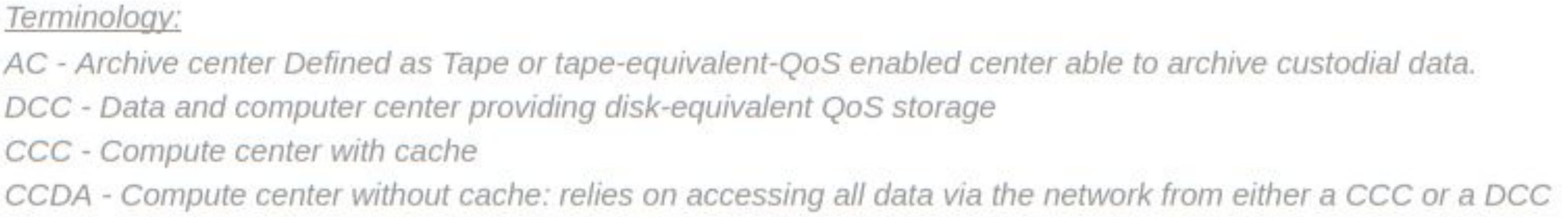
- Disk failures estimation: **only 1%** of data will be fetched outside the local datalake - **Efficiently hide** latencies with **buffering** at the client side and **access through cache+**

Local access. Client buffering + file on the site cache+

Local cache miss. Client buffering + file streaming from DCC

Datalake miss (file loss). Client buffering + file streaming from DCC + datalake copy

Datalakes, latency hiding and caching - Xavier Espinal (CERN)



Terminology:

AC - Archive center Defined as Tape or tape-equivalent-QoS enabled center able to archive custodial data.

DCC - Data and computer center providing disk-equivalent QoS storage

CCC - Compute center with cache

CCDA - Compute center without cache: relies on accessing all data via the network from either a CCC or a DCC

A small number of Data Lakes across the world

- Reduced number of storage endpoints wrt the current WLCG model

Envision a mix of distributed caches directly accessed from compute nodes

DODAS main concepts

- Support **user tailored** computing **environments**
- **Automate** configuration and deployment of custom services and/or dependencies
- Support declarative approach to define input parameters, customize workflows, **treating a collection of resources together as a single unit**

DODAS main concepts (cont)

Highly flexible and modular solution enabling multiple usage patterns:

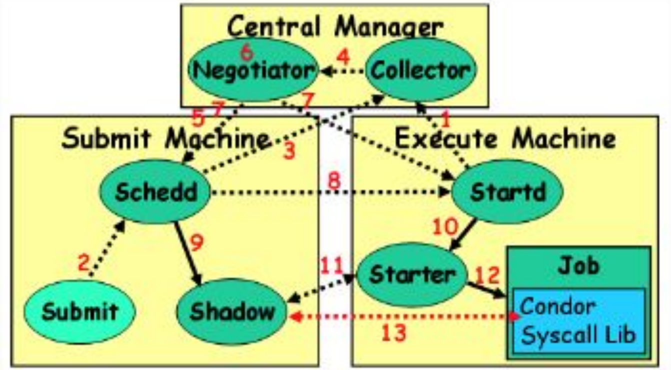
- Leverage clusters, possibly customising the stack, building highly reliable, highly scalable, applications
  - without worrying about creating and configuring the underlying infrastructure
    - **TOSCA + Ansible + Application setup (e.g Helm)**
- Generate Clusters on demand (**K8s on demand**), possibly customizing the underlying infrastructure
  - **TOSCA + Ansible**
  - And leave users **to focus just on applications**
    - **(e.g. Helm)**
- (Abstract the IaaS provisioning VMs/DB etc…)

NOT CMS specific

Recap From Tommaso

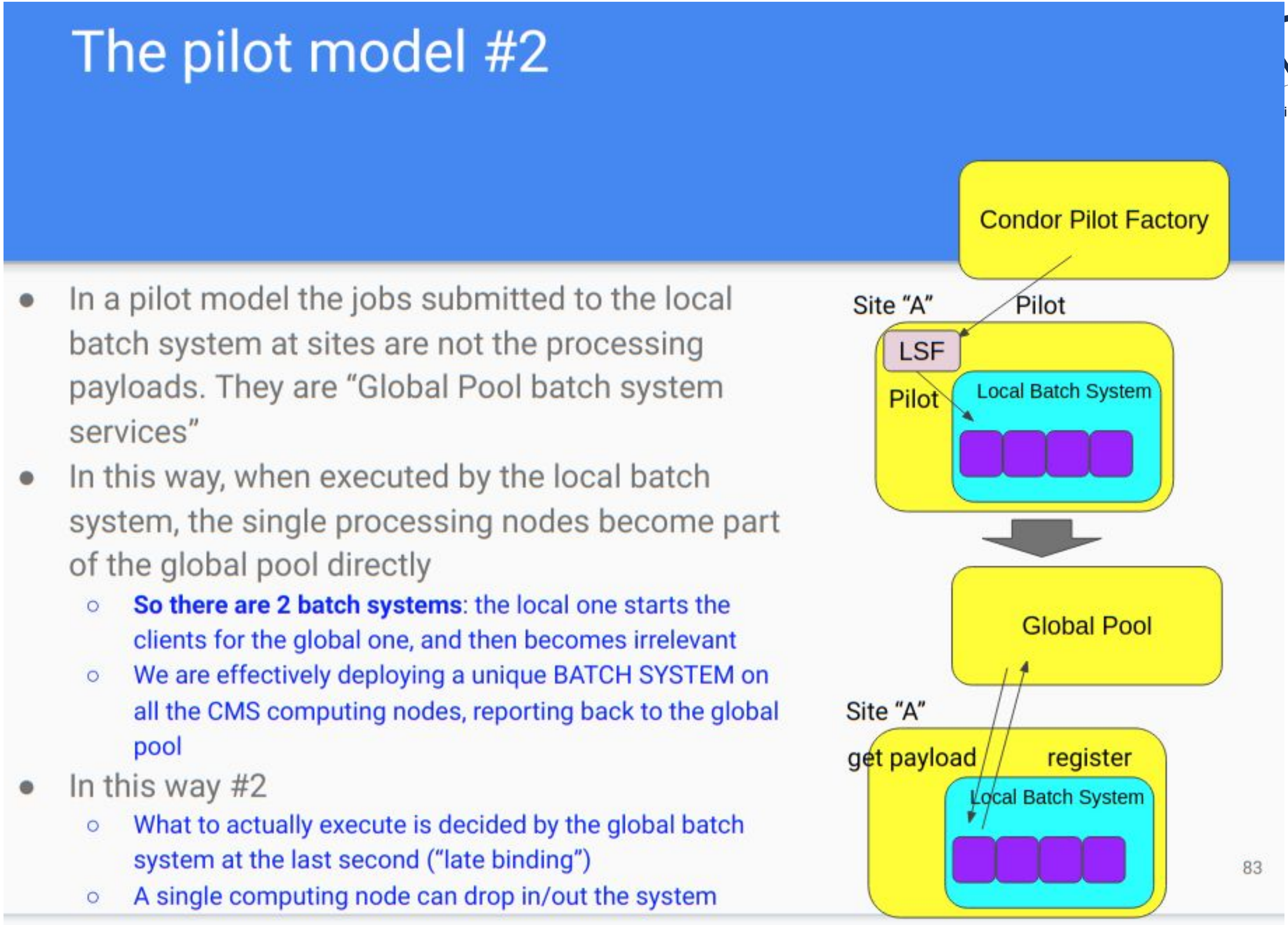
HTCondor

- (HT)Condor: “*Our goal is to develop, implement, deploy, and evaluate mechanisms and policies that support High Throughput Computing (HTC) on large collections of distributively owned computing resources*”
- It has central services (Negotiator, Collector) which
  - Receive jobs to be processed (from a submit machine)
  - Send them to available machines
  - The model is push: computing nodes are not listed in a DB, but are connecting to central services (“Hey, I am available!”)



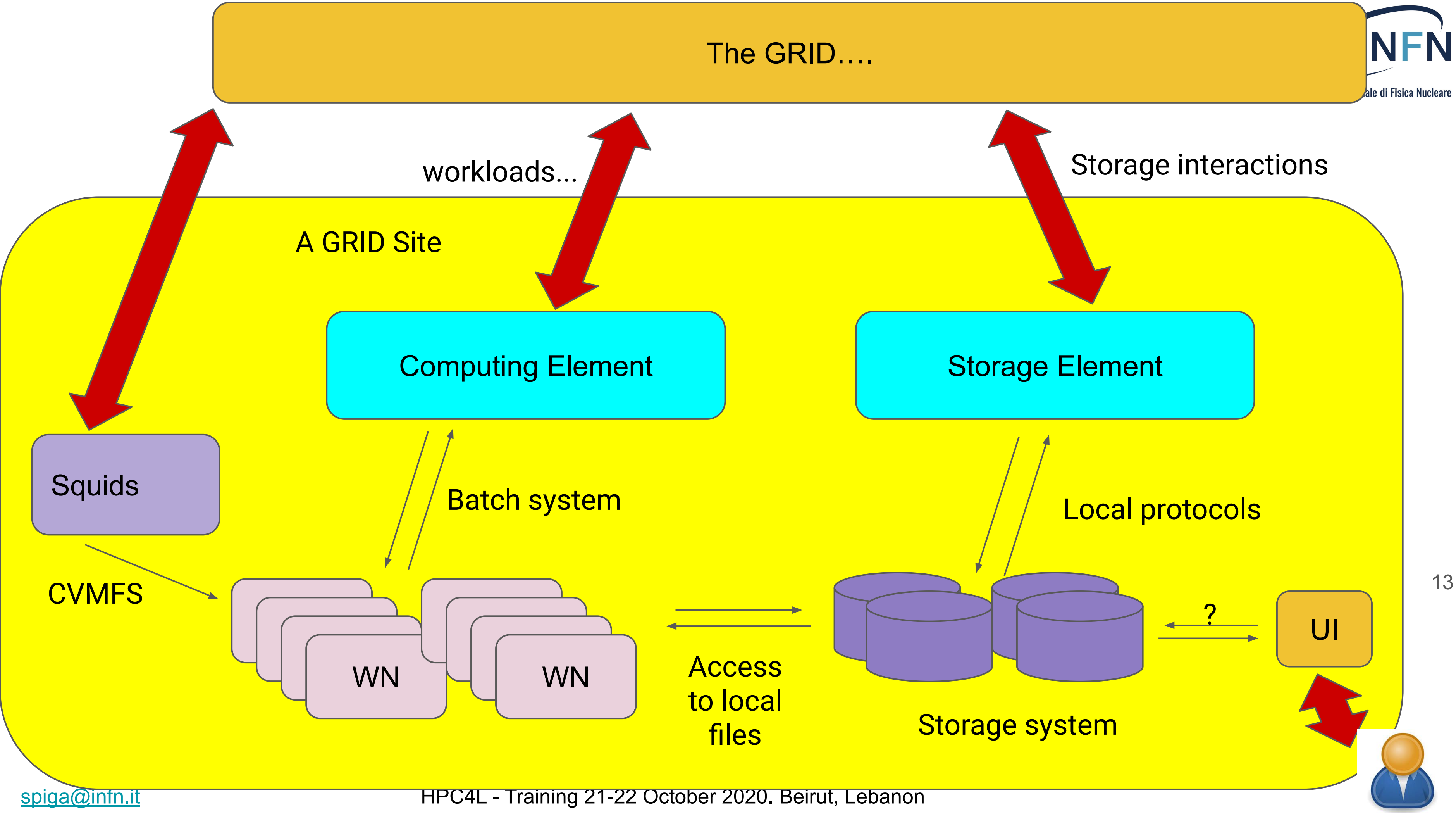
The central services have a global view of the system, and implement a sort of “global batch system in CMS” (the “global pool”)

Recap From Tommaso

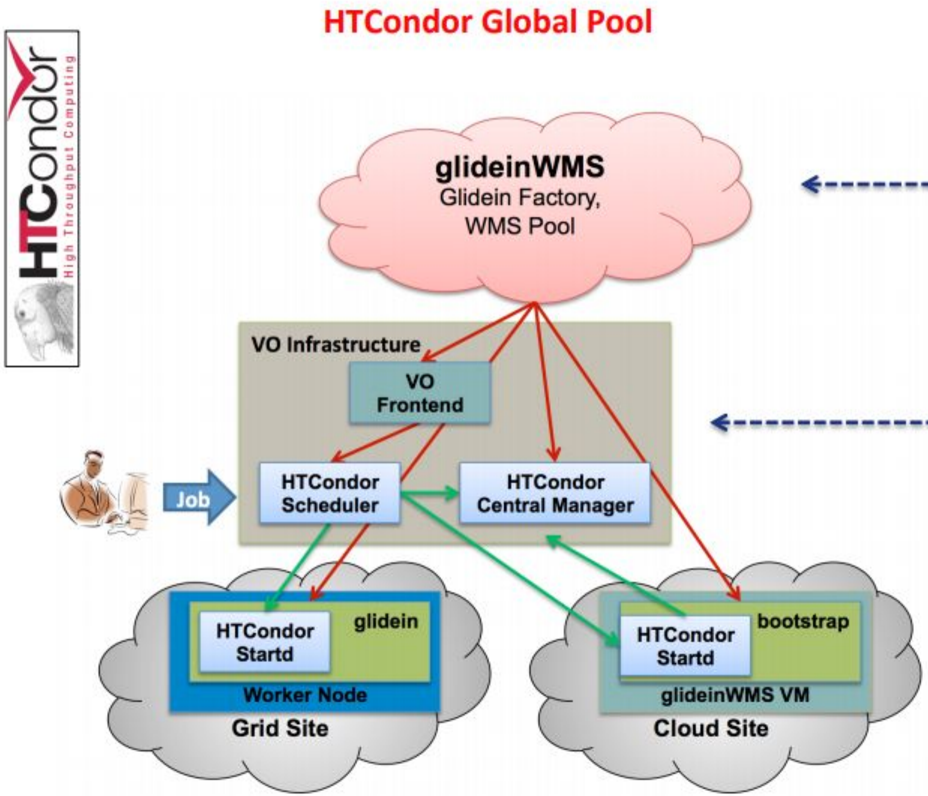


The pilot model #2

- In a pilot model the jobs submitted to the local batch system at sites are not the processing payloads. They are “Global Pool batch system services”
- In this way, when executed by the local batch system, the single processing nodes become part of the global pool directly
  - **So there are 2 batch systems**: the local one starts the clients for the global one, and then becomes irrelevant
  - We are effectively deploying a unique BATCH SYSTEM on all the CMS computing nodes, reporting back to the global pool
- In this way #2
  - What to actually execute is decided by the global batch system at the last second (“late binding”)
  - A single computing node can drop in/out the system


The GRID....
workloads...
Storage interactions
A GRID Site
Computing Element
Storage Element
Squids
Batch system
Local protocols
CVMFS
WN
WN
Access to local files
?
UI
Storage system

CMS Job Submission Infrastructure

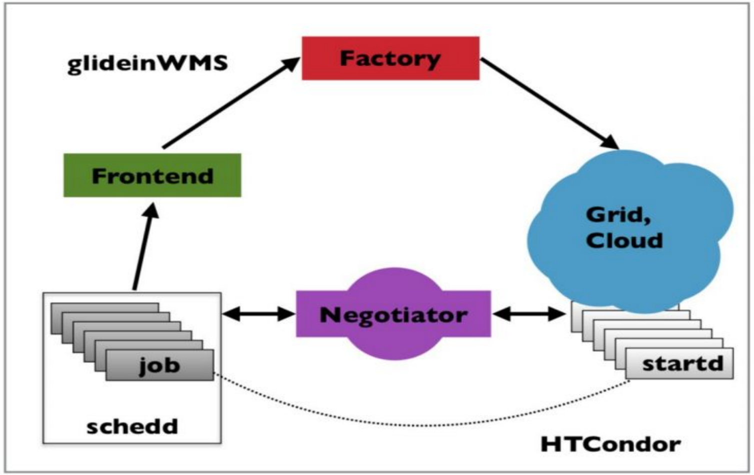


- In the first stage of matchmaking, glideinWMS frontend matches jobs to their desired sites and requests the glideinWMS factory to send glideins (properly configured condor tar ball)

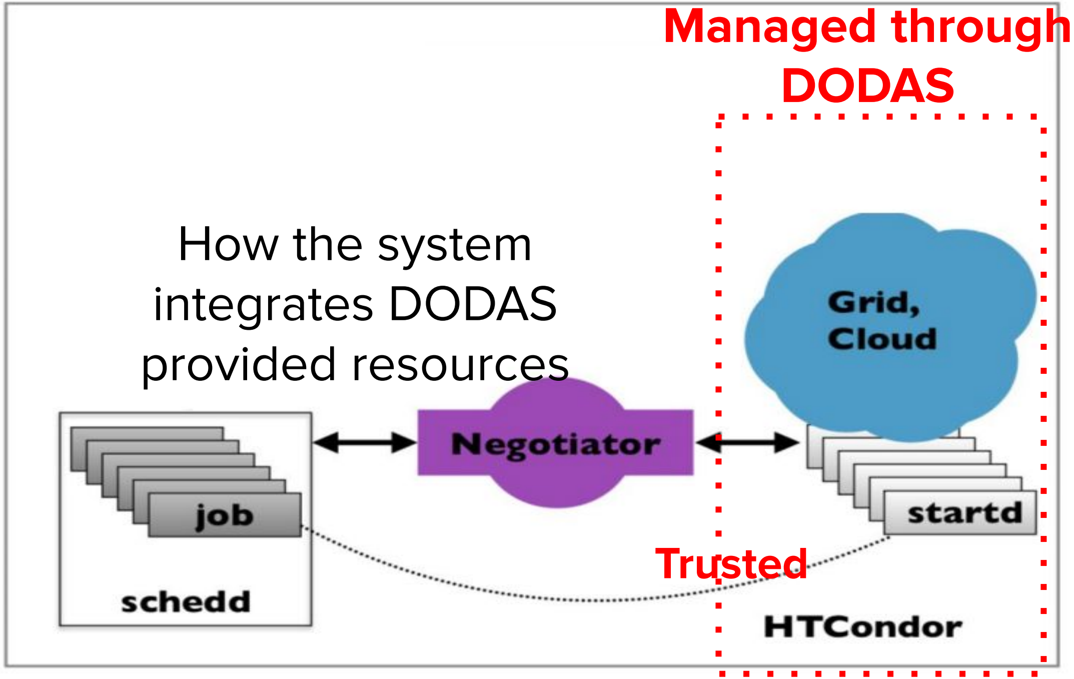
- The 2nd stage of matchmaking is when a job gets matched to a slot once the condor starts on the worker node and makes itself available in the pool

- Glidein pulls in the job and then GLExec is used to switch to central production or analysis user's credentials

The HTCondor view of DODAS



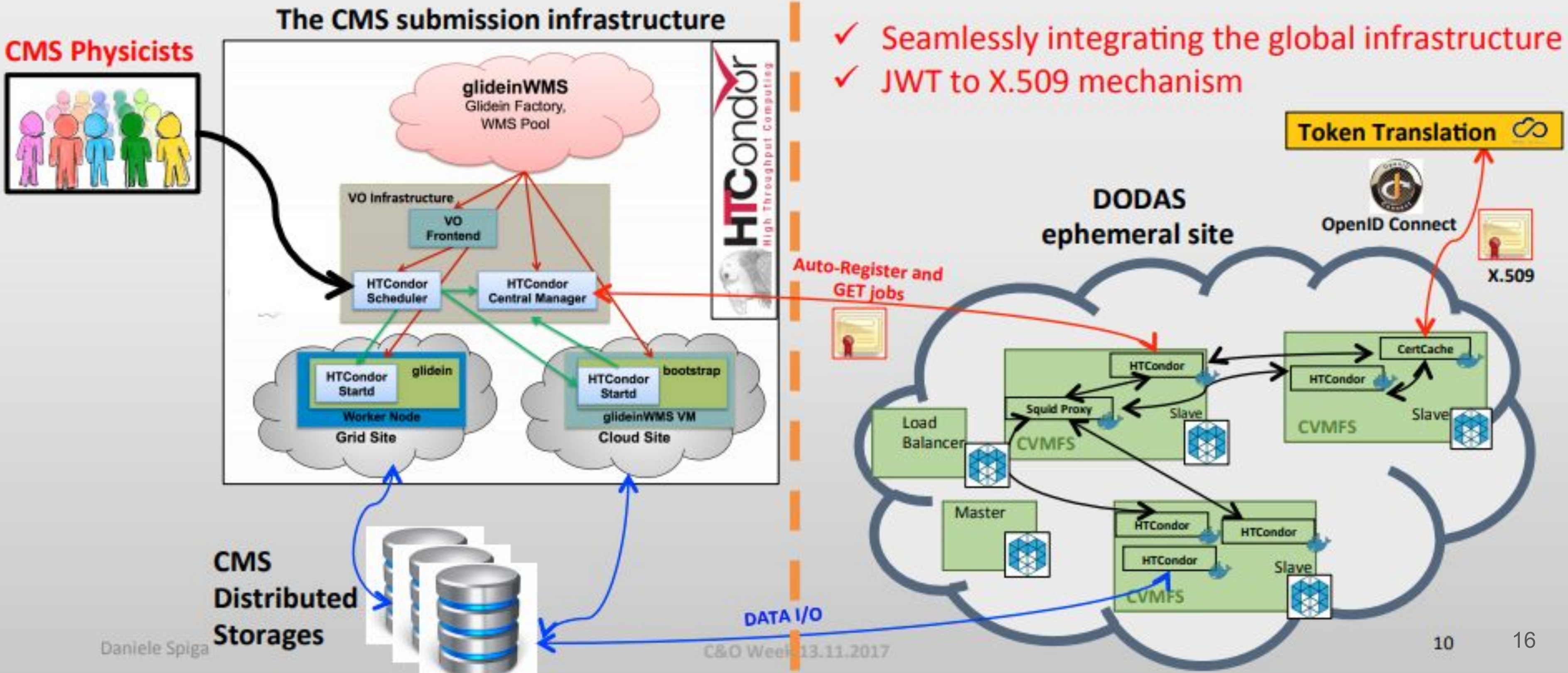
High level view of the CMS Global pool and its main elements



How the system integrates DODAS provided resources

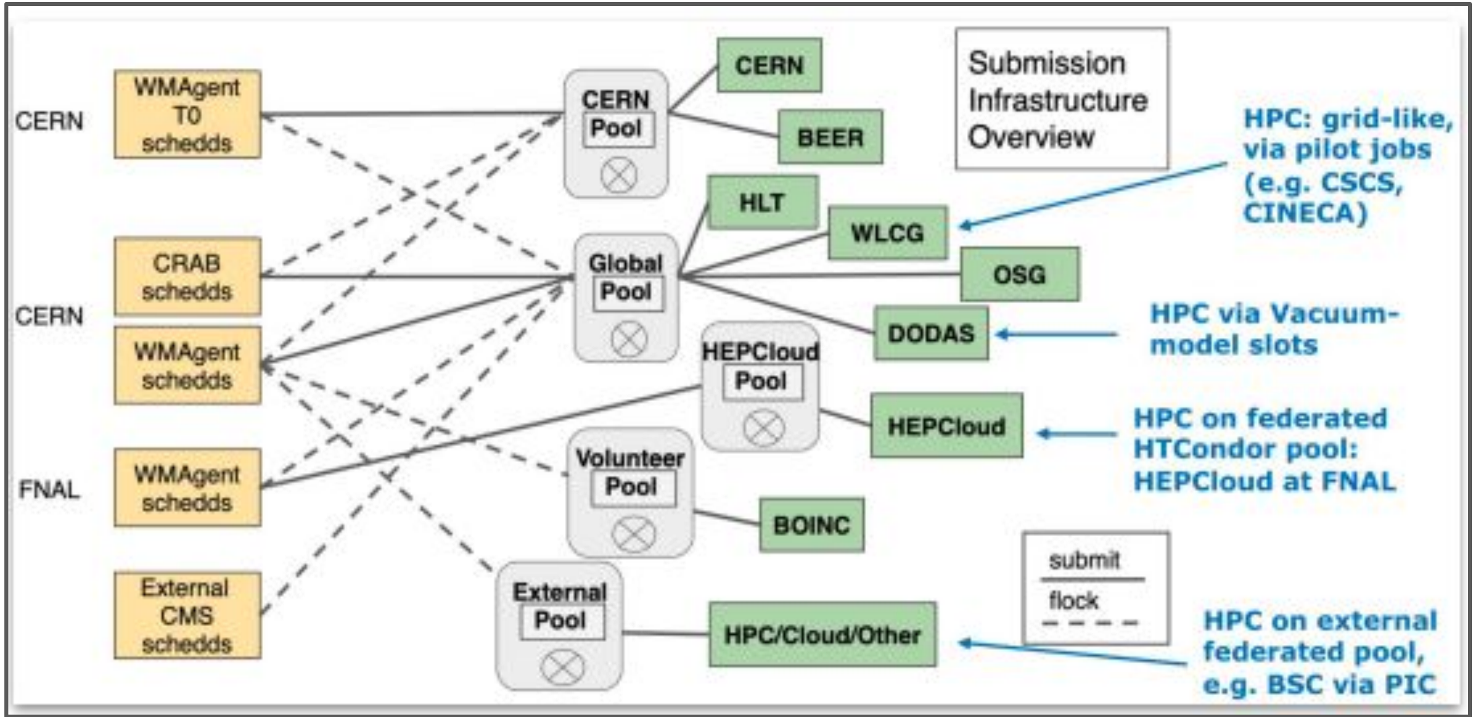
Integration with Submission Infrastructure : Schema

- ✓ Seamlessly integrating the global infrastructure
- ✓ JWT to X.509 mechanism



CMS Submission Infrastructure

A different perspective: CMS see DODAS as yet another solution to provision resources to the experiment



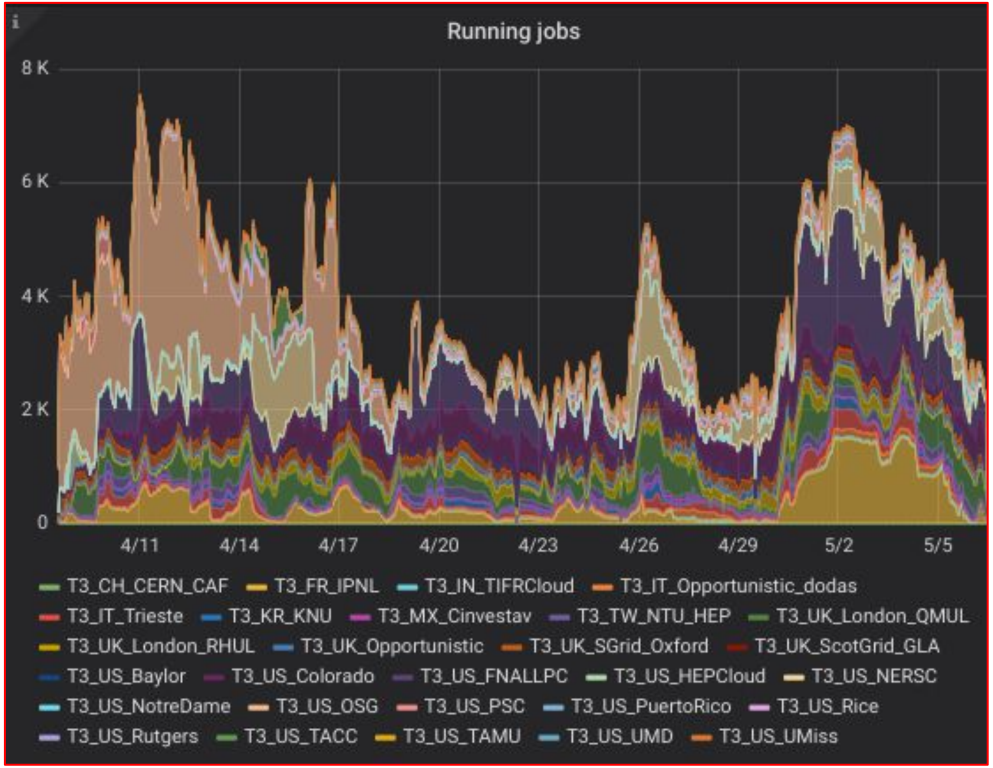
HPC: grid-like, via pilot jobs (e.g. CSCS, CINECA)

HPC via Vacuum-model slots

HPC on federated HTCondor pool: HEPCloud at FNAL

HPC on external federated pool, e.g. BSC via PIC

WLCG Monit integration



But it is also fully integrated in the CMS Job monitoring running on CERN MONIT infrastructure for monitoring WLCG resources

How DODAS fits into this ?

DODAS: automatizes the process of dynamically, on demand, creating a Grid site

- On top of Cloud
- In a cloud-native manner
- Seamlessly integrating the submission infrastructure of CMS
- Automatically fetching jobs
    - Implementing well defined **authN/Z** policies
- Eventually reducing the need of extra services such as
    - Computing Element
        - Implementing the vacuum model, the pilots (glidein) starts themselves “by magic”
    - Storage Element
        - Relying on friendly sites

What does it means concretely?

DODAS performs the following actions:

- Bare-hosts (e.g. VMs) instantiation based on user requirements, defined at TOSCA level
  - Virtual hardware can be scaled up/down (elasticity)
- Services and software configurations at host leves (docker engine etc) via ansible
- Container orchestrator deployment (K8s) via ansible
- Deployment and execution of services/microservices (e.g. Worker Nodes/HTCondor, squid proxy, CVMFS and JWT-->x509 certificate translator) as k8s Application
  - **this is how worker nodes are "spontaneously produced" and scaled up/down**

Ok but in practice ?

We have all the pieces now:

- TOSCA define the topology embedding :
  - Ansible: prepare the infrastructure and customisations and execute helm install
  - HELM: install the CMS Site (from auto-pilots up to squids, cvmfs etc)

Here we go:

https://github.com/DODAS-TS/dodas-apps/blob/master/templates/applications/k8s/experiments/cms/template-cms_cluster.yml

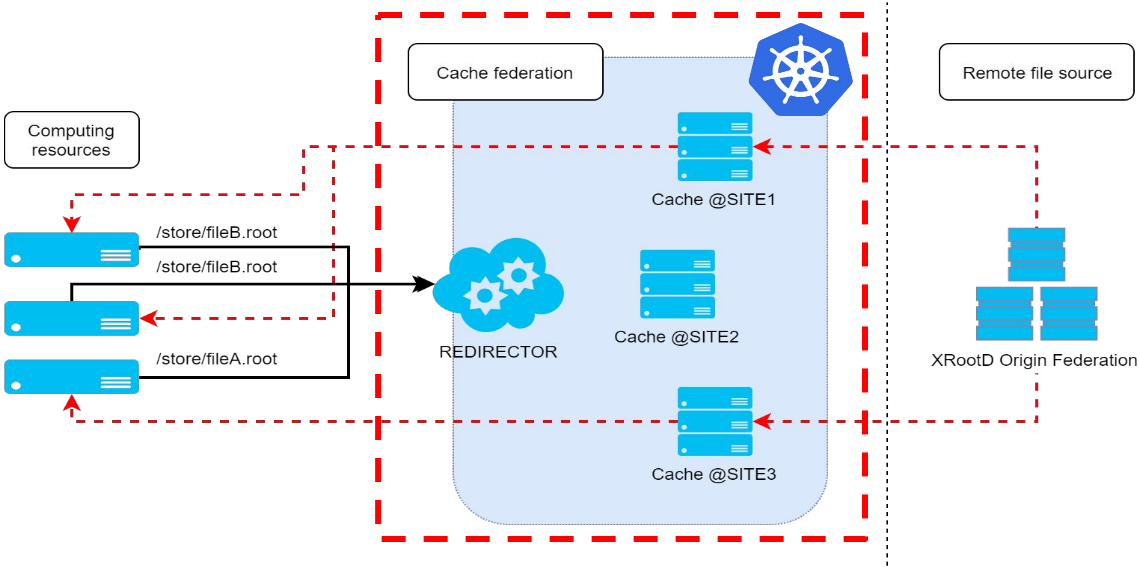
Optionals features

- Current incarnation is based on HTCondor as a mean to manage (aka overlay) distributed worker nodes (startd/glideins)
  - Does not deploy Computing Element (CE) but it could be added (example of modularity)
  - DODAS Vacuum system is integrated in the **CMS Computing infrastructure aka HTCondor Global pool** (see next slides)
- Current incarnation is based on Storage-less configuration, this means that the generate Site will use a friendly remote storage
  - As an optional it is possible to deploy storage to create a cache layer which allow to optimize I/O. The implementation is based on XrootD (XCache)
    - We will not go through the details here

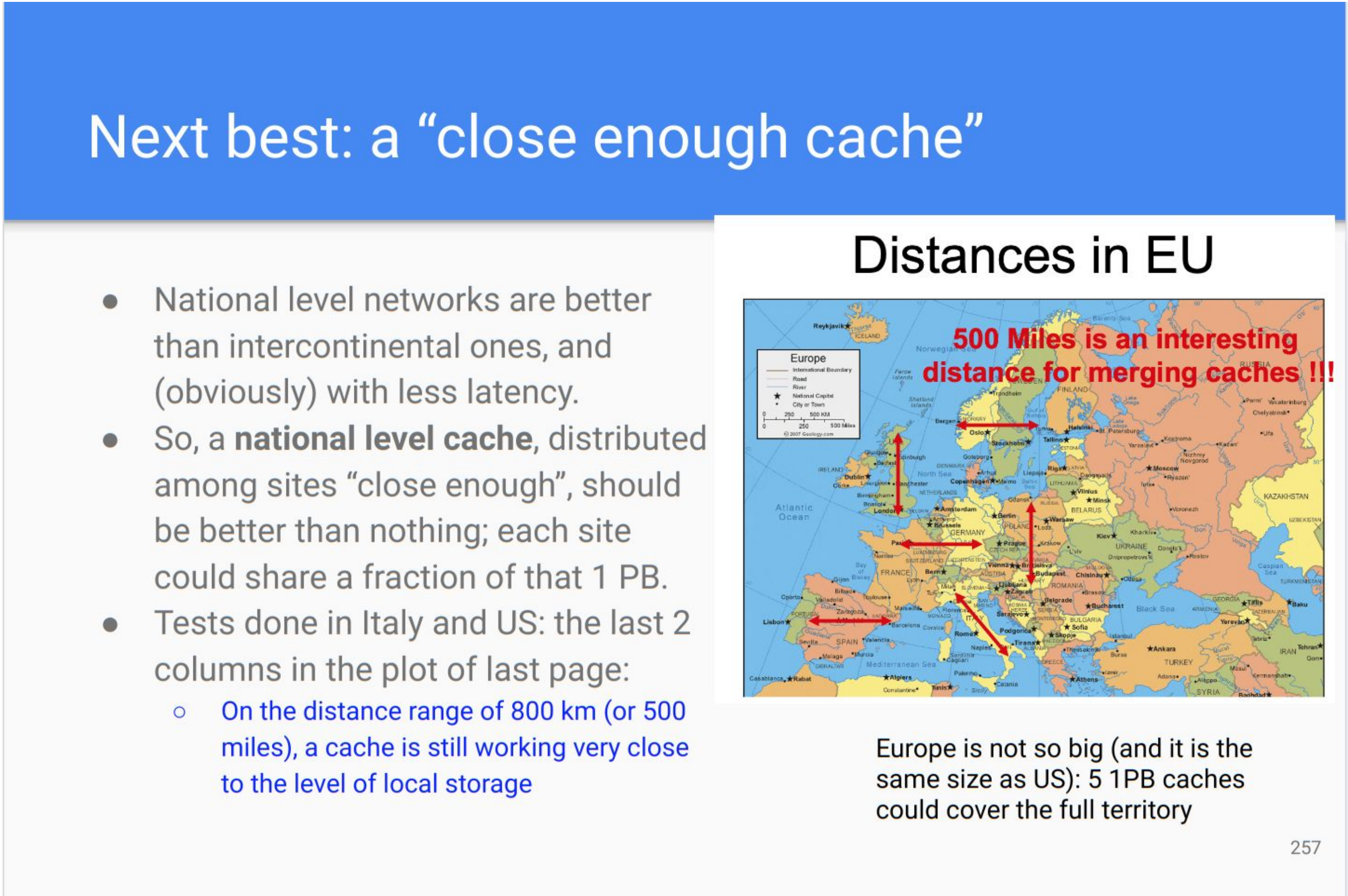
Storage

The data cache part leverages XRootD XCache technology to cache data near the WNs

- Configurable through Pod environment variables and/or configMap
- New cache servers can be scaled and automatically registered on the redirector
- Limit to one cache per VM and using net:host for performance reason



Recap From Tommaso



Next best: a “close enough cache”

- National level networks are better than intercontinental ones, and (obviously) with less latency.
- So, a **national level cache**, distributed among sites “close enough”, should be better than nothing; each site could share a fraction of that 1 PB.
- Tests done in Italy and US: the last 2 columns in the plot of last page:
    - On the distance range of 800 km (or 500 miles), a cache is still working very close to the level of local storage

Distances in EU

500 Miles is an interesting distance for merging caches !!!

Europe is not so big (and it is the same size as US): 5 1PB caches could cover the full territory

257

Ok but in practice ?

We can have a look to the “last mile”

HELM chart to deploy a complete xcache system

- https://github.com/DODAS-TS/helm_charts/tree/master/stable/cachingondemand

This can be embedded in TOSCA

- But now all this is already understood and straightforward

Analysis and Production jobs

The described solution is fully compatible for any type of CMS jobs, being them Production / Montecarlo / private Monte Carlo and Analysis job

However in this demo we will focus on Analysis jobs:

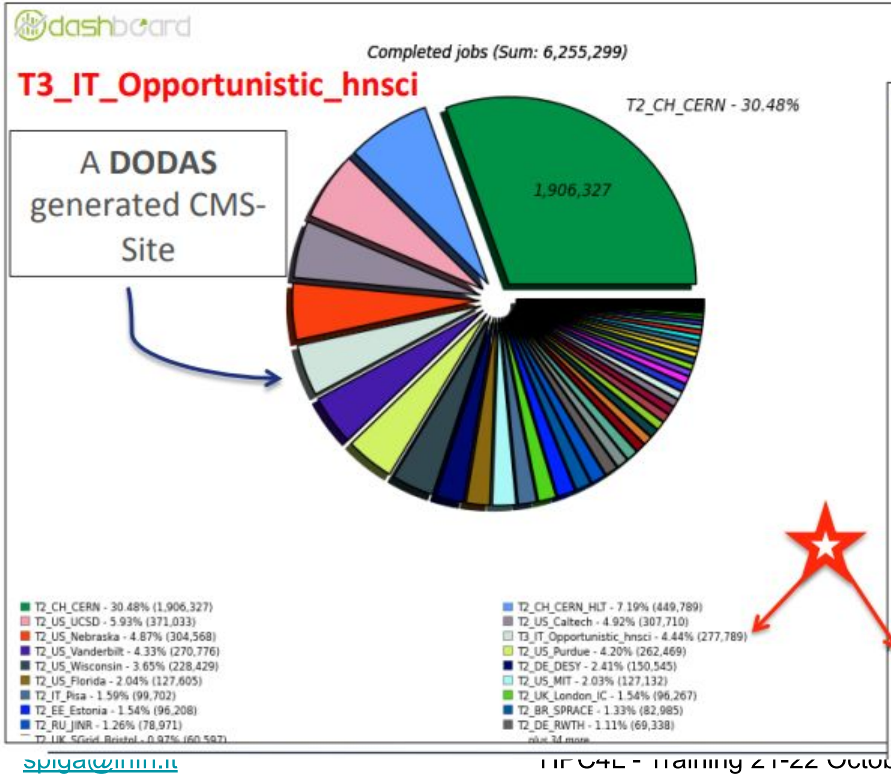
Link to the demo:

http s://drive.google.com/drive/folders/173FV1iRFQirKNsdxinKO3o7mmnJBIlC1?ths=true

DODAS and High Energy Physics Part2

EOSC-hub Handling Opportunistic Resources: Real Life

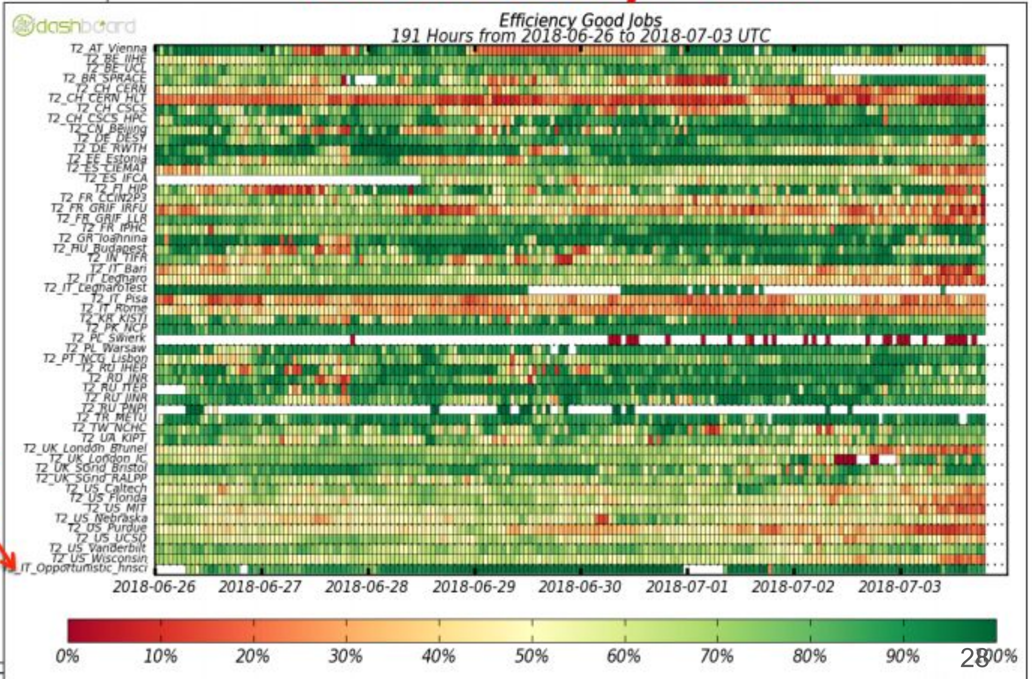
DODAS within top 6 CMS Tier2s for 10 days



T3_IT_Opportunistic_hnsci

A **DODAS** generated CMS-Site

Job Efficiency > 85%

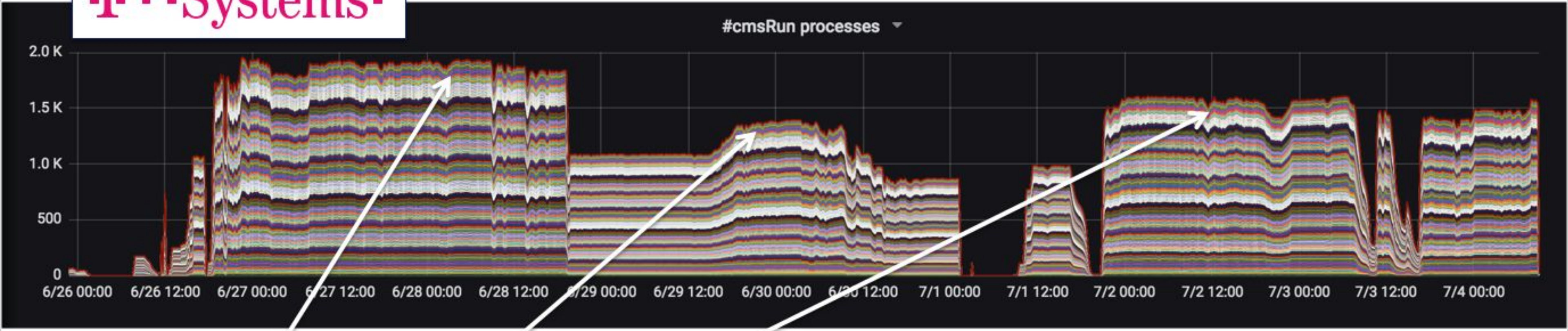


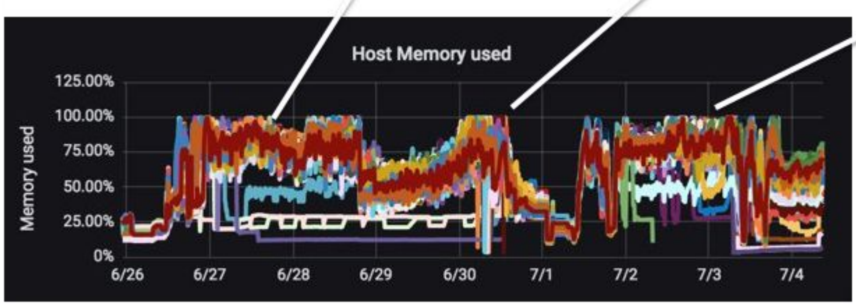
EOSC-hub

... From resources perspectives

INFN Istituto Nazionale di Fisica Nucleare

T · ·Systems·





- Elasticity and self-healing
- Stability over days/weeks (120k jobs)
- Handling “special requirements” high memory jobs

HELIX NEBULA THESCIENCECLOUD

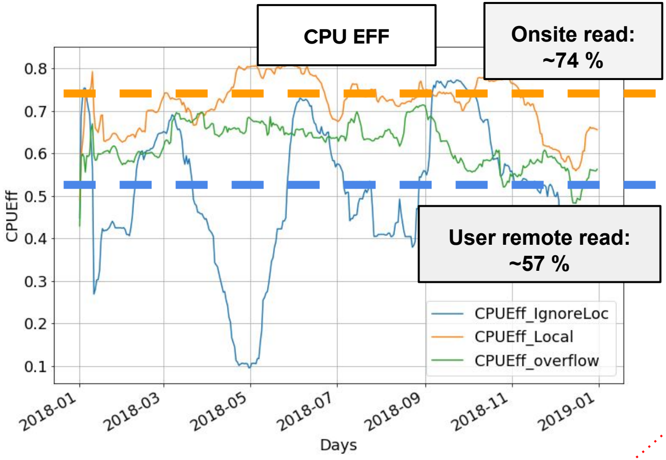
Cache effect on CPU efficiency @ CMS

- We studied monitoring data of the whole **2018 CMS analysis workflows**
  - Remote data read costs on average **about 15% of CPU time w.r.t. Onsite data reading**

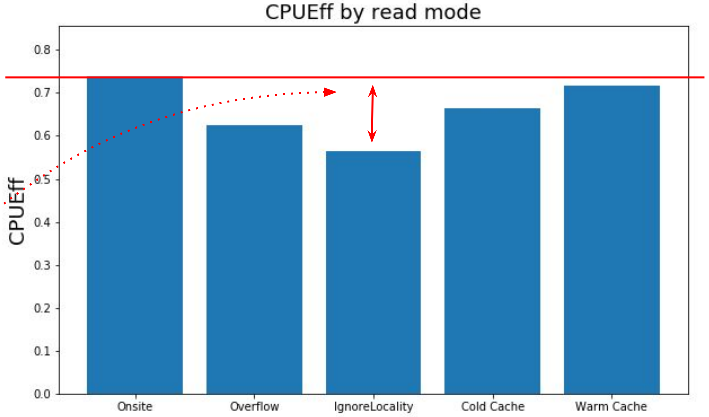
CPUEff = sum(cpu time) / sum (job time)

@Italian Tier2's

From data@CERN MONIT hdfs



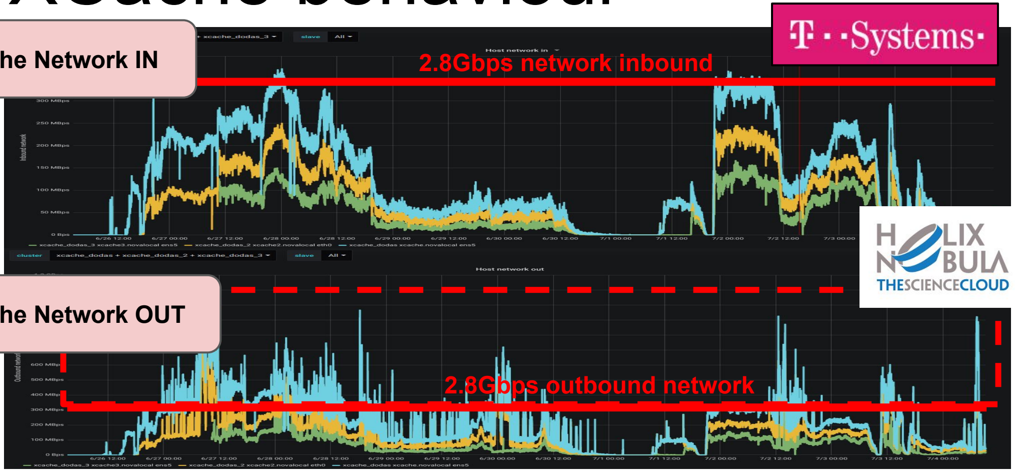
Measured on testbed



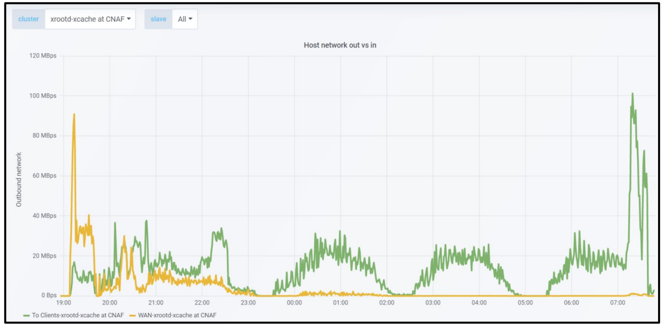
Caches allow to reduces the overall WAN traffic and, makes the processing job that requested the data **more efficient** by reducing I/O wait time for remote data.

XCache behaviour

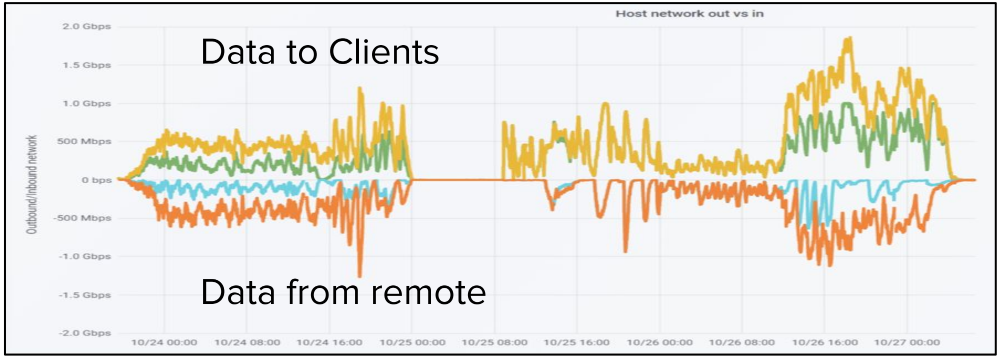
Cache Network IN



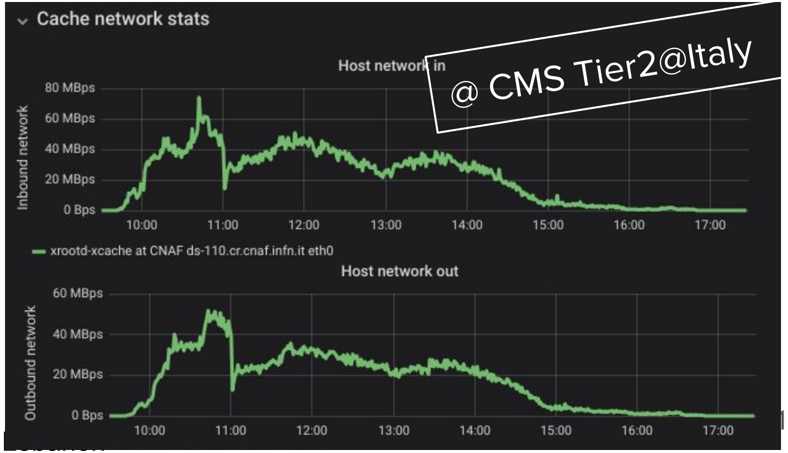
Cache Network OUT



See: [Extension of the INFN Tier-1 on a HPC system]



See: [The DODAS experience on EGI Federated Cloud]

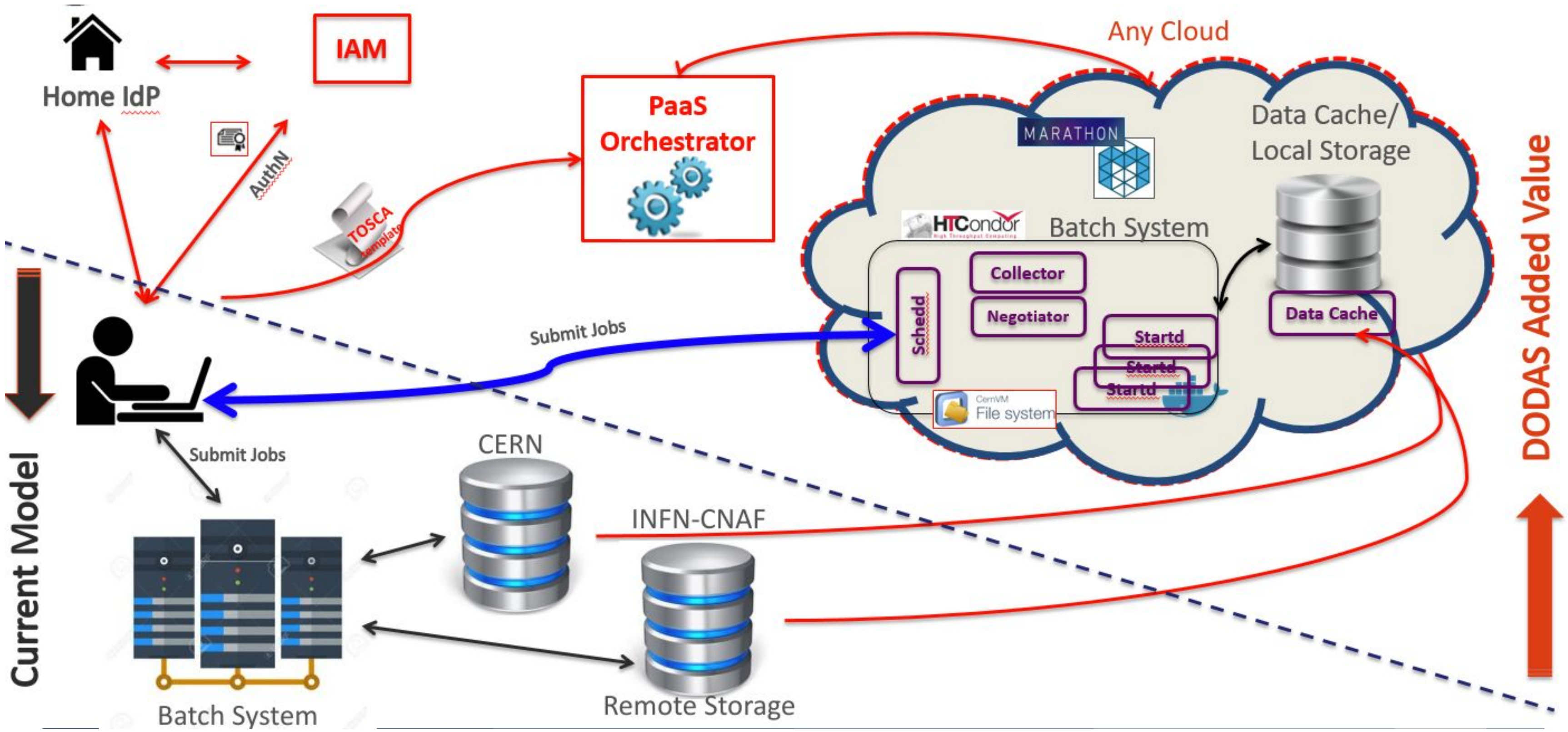


We said that DODAS is experiment free… is that confirmed?

Any concrete example ?

….

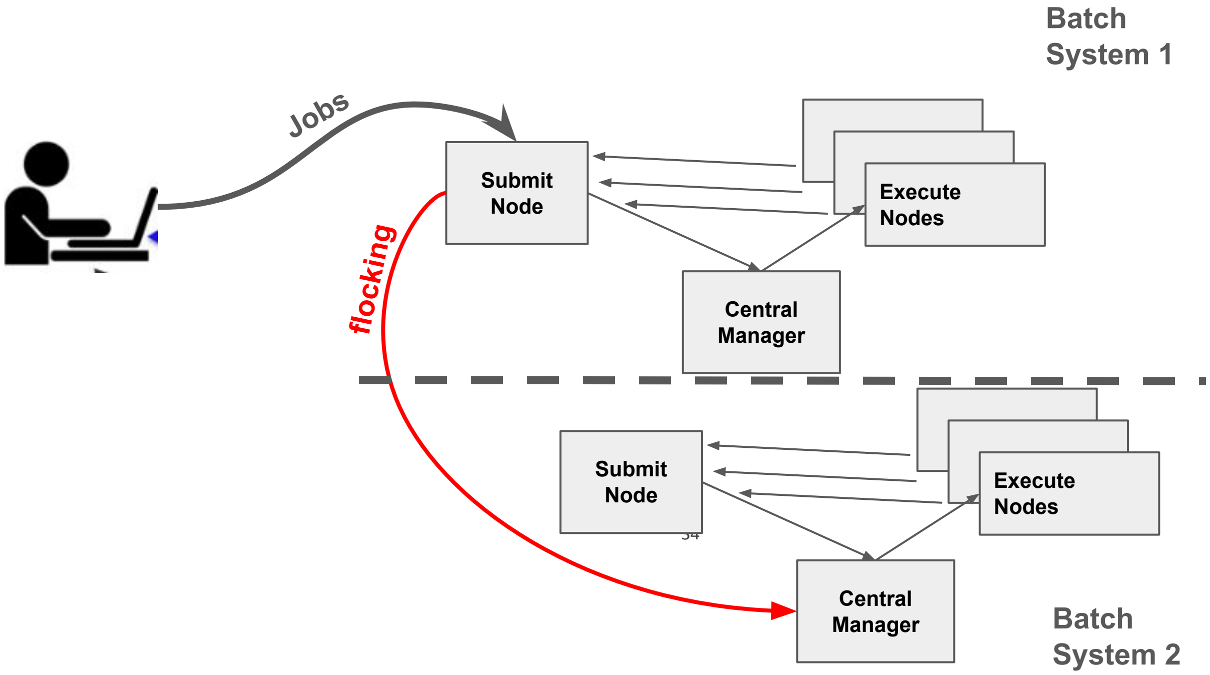
The first end-to-end integration



Federation through HTCondor Flocking

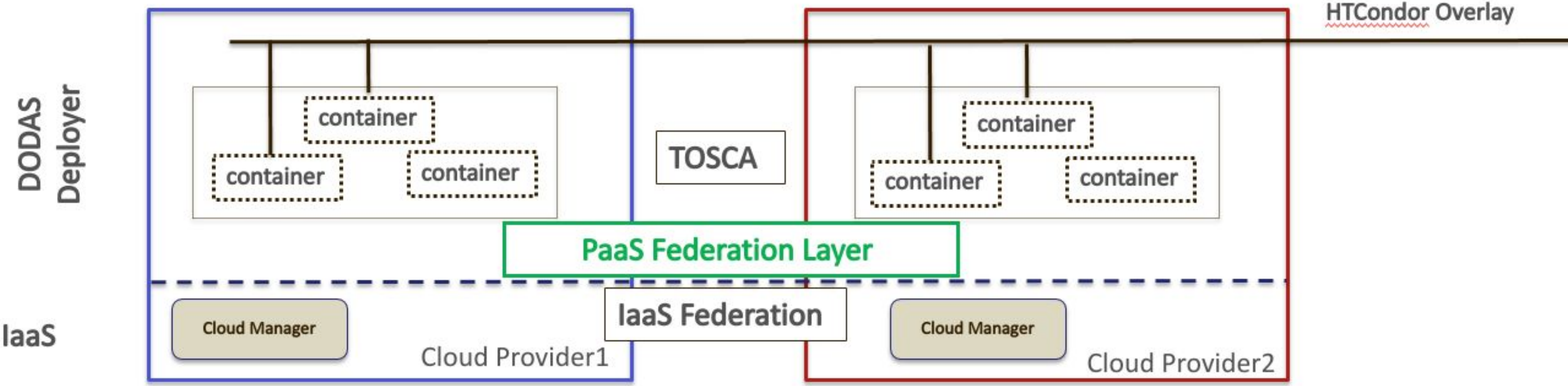
AKA a native HTCondor federation solution

- easy to implement from the HTCondor perspectives
  - few lines of config file (flock from, flock to)
  - plus AuthN/Z
- It is transparent from the user point of view
  - just keep submitting jobs as before



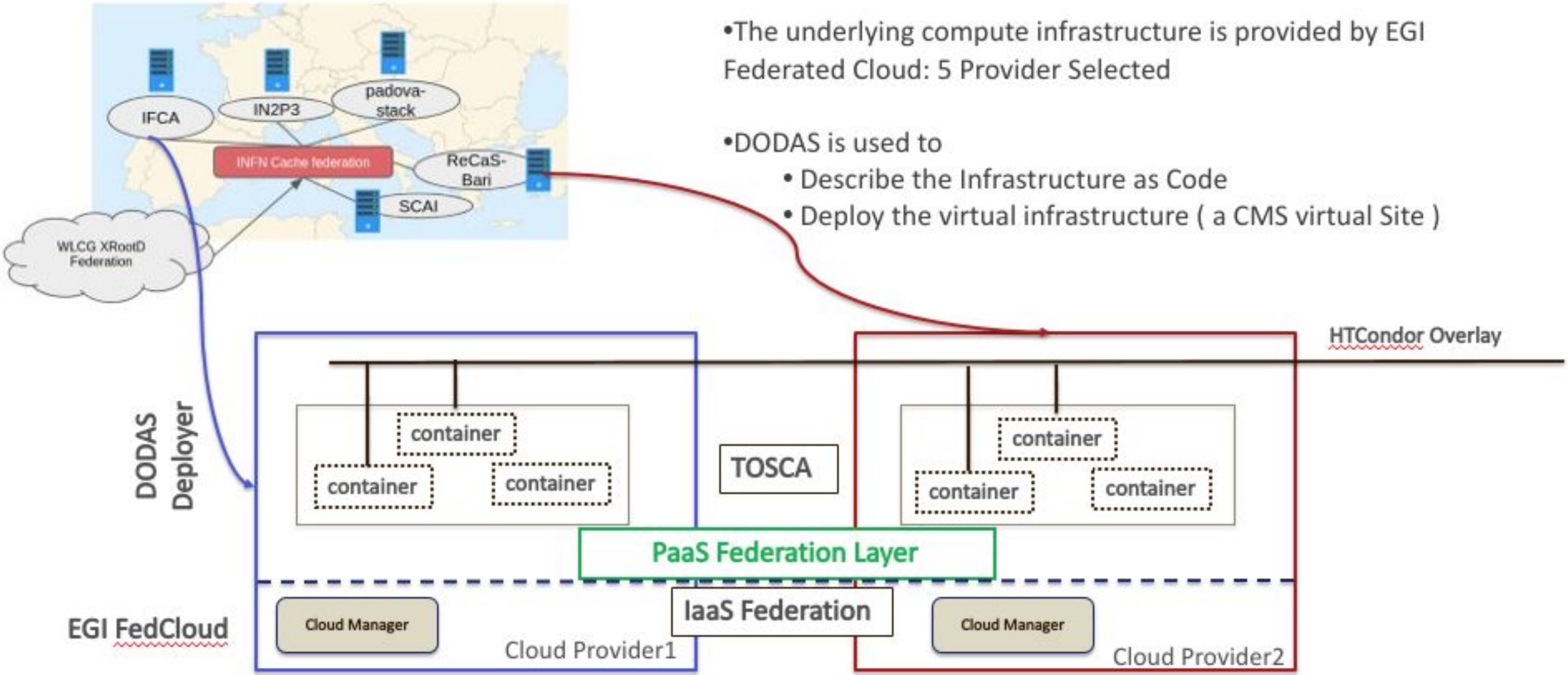
From the infrastructure point of view

PaaS federation layer allow to build container based HTCondor overylay

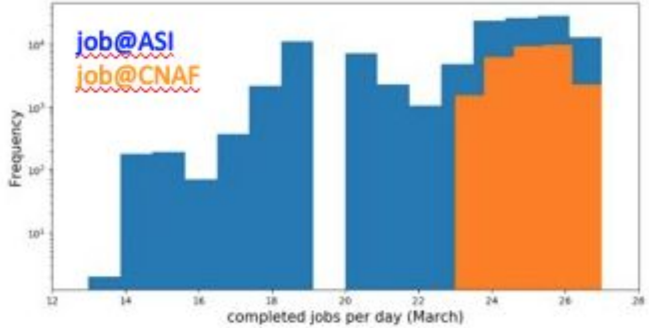


DODAS & Providers: Exploiting EGI Resources

- The underlying compute infrastructure is provided by EGI Federated Cloud: 5 Provider Selected
- DODAS is used to
  - Describe the Infrastructure as Code
  - Deploy the virtual infrastructure (a CMS virtual Site)



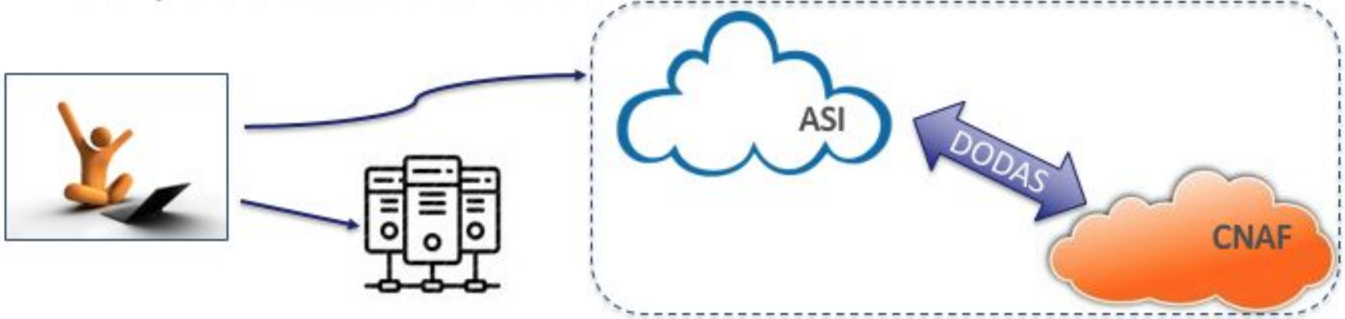
Managing stateless providers with DODAS



The Space Scientific Data Center (SSDC) of the Italian Space Agency (ASI) host an AMS farm.

- no experiment dedicated manpower
- no specific expertise on AMS software and computing environment

An example of Stateless Site Providers



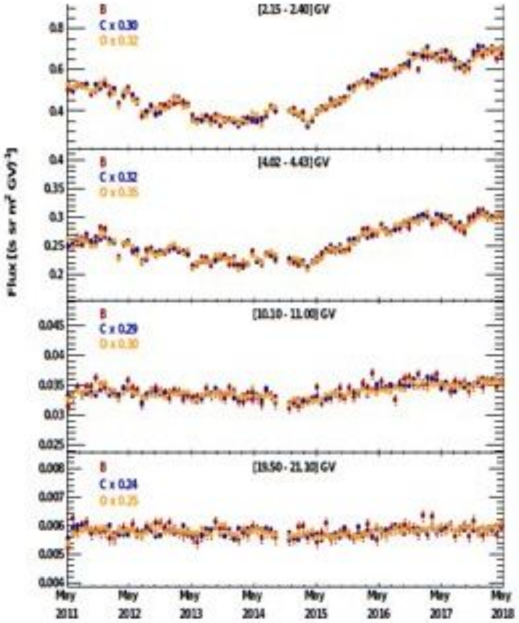
DODAS centrally manages federated and possibly **stateless sites, completely transparent for the end-user.**

AMS analysis in 2020 using DODAS

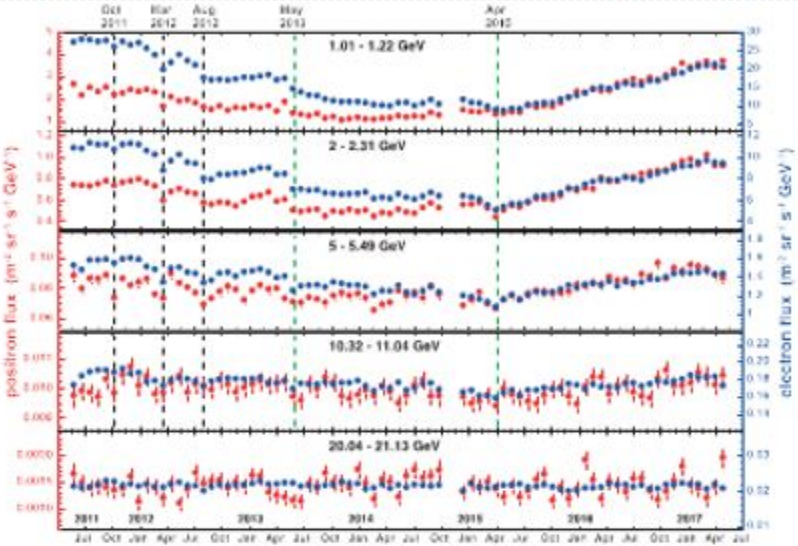
- The Alpha Magnetic Spectrometer measures Charged Cosmic Rays (0.1 – 2000 GV) in space since 2011, May 19th
- The ~ 160 billion events (~ 180 k science runs), once reconstructed in ROOT format, weight ~ 1 PB
- The analyses are performed on stripped samples (i.e. streams) with a lighter data format (i.e. ntuples)

- a job to produce a single run needs ~ 2 hr and produces O(102 MB) ntuples
- every analysis target (e.g. electrons/positrons vs. ions) requires its own ntuples set

Dr. M. Duranti
Dr. V. Formato



left - Boron, Carbon (scale = 0.24) and Oxygen (scale = 0.25) fluxes as function of time (Δt = 27 days)
bottom- Electron and positron fluxes as function of time (Δt = 27 days)



1. AMS collaboration previously published B, C and O fluxes only as a function of energy and time-integrated. **This new analysis, by INFN-RM2, has been performed using the ntuples produced running on DODAS;**
2. Electrons and positrons fluxes, as a function of time have been already published with 27 days time granularity. **A new analysis, by INFN-PG, and using the ntuples produced on DODAS, is extending the time range and producing the electron (positron) fluxes on a daily (weekly) basis;**

EOSC-hub

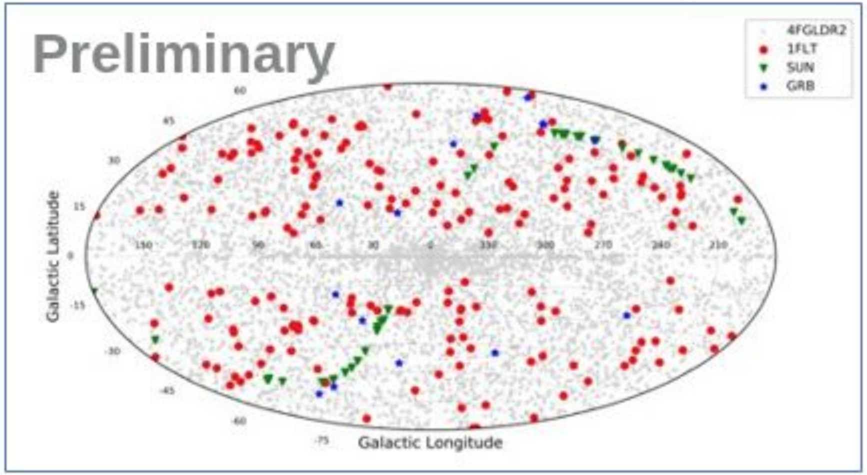
(FIRST) FERMI-LAT ANALISYS USING DODAS

Fermi Gamma-ray Space Telescope

FERMI GAMMA-RAY SCIENCE TELESCOPE

- The LAT instrument onboard of Fermi gamma-ray science telescope (Atwood et al 2009) observes the sky in the gamma rays range between 30MeV - 300 GeV since August 2008.
  - Extract catalog of transient sources (monthly basis) from Fermi-LAT data (1FLT; Fermi-LAT collaboration in preparation)
    - 10 years of data in monthly time scale—> 120 independent skies + 120 (15-day shifted month)
  - Detected ~1000 seeds/monthly skies
    - ~260 binned maximum likelihood analysis (ML) for each month.

- **Submitted roughly 60k ML analysis jobs —>Very time consuming! ~ 960 h of computing time without interruptions**



Aitoff projection of 1FLT in red, sun detections from transient catalog in green, and GRB detection in blue. Standard catalog (4FGLDR2) in gray

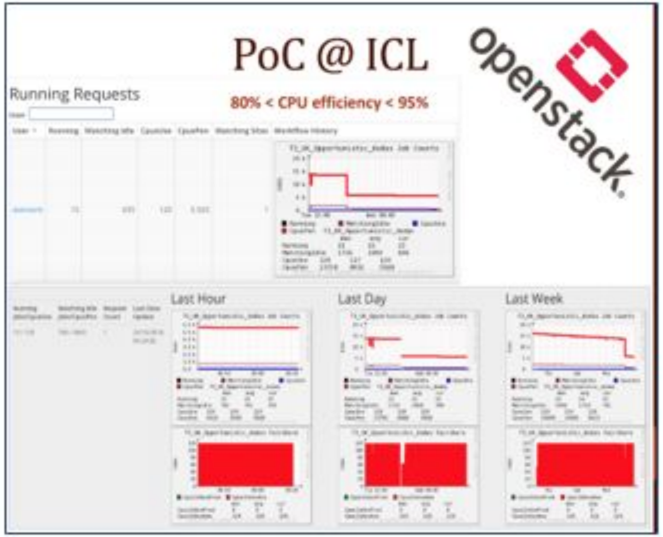
507 new detections —>extraction of standard products: monthly light curves **(120 ML jobs per source)** and Spectral energy distributions **(4 ML jobs per source)**

DODAS clusters allows to reduce the computation time, scaling with cluster size. This implies a faster turnaround in the data analysis steps **(VERY PROMISING)** Dr. Sara Cutini

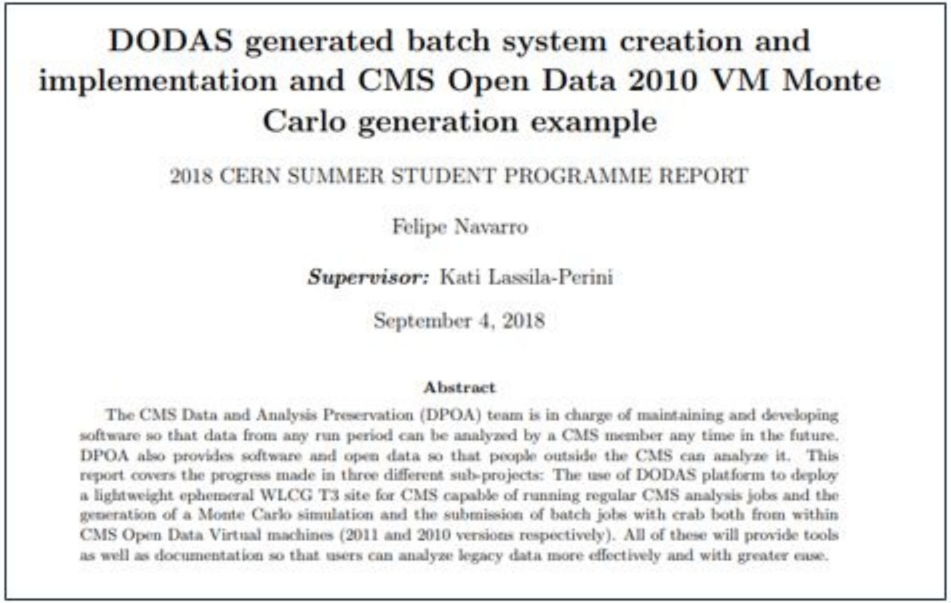
DODAS & support to requirement-specific workflow

- Producing CMS nanoAOD format for IC SMP Analyses from miniAOD (CMS tree slimming stage, EDM trees -> flat trees);
- Producing Gen-Sim from LHE (Pythia parton shower, hadronisation and particle decays, Geant4 detector simulation);
- LHE Creating (Madgraph5 aMC@NLO of V+j and V+jj processes, generate events from the Matrix Element only)

Dr. Riccardo di Maria,
Imperial College London



Data Preservation @CMS



DODAS generated batch system creation and implementation and CMS Open Data 2010 VM Monte Carlo generation example

2018 CERN SUMMER STUDENT PROGRAMME REPORT

Felipe Navarro

Supervisor: Kati Lassila-Perini

September 4, 2018

Abstract

The CMS Data and Analysis Preservation (DPOA) team is in charge of maintaining and developing software so that data from any run period can be analyzed by a CMS member any time in the future. DPOA also provides software and open data so that people outside the CMS can analyze it. This report covers the progress made in three different sub-projects: The use of DODAS platform to deploy a lightweight ephemeral WLCG T3 site for CMS capable of running regular CMS analysis jobs and the generation of a Monte Carlo simulation and the submission of batch jobs with crab both from within CMS Open Data Virtual machines (2011 and 2010 versions respectively). All of these will provide tools as well as documentation so that users can analyze legacy data more effectively and with greater ease.

Data preservation and open access (DPOA) team at CMS exploits DODAS to allow people from outside the collaboration to perform analysis on open legacy data 2010/2012, as part of the open access project.

Summary

DODAS is a high modular deployer manager built on the concept of Infrastructure as a code to create and provision infrastructure deployments, **automatically and repeatedly**.

- We discussed how we use DODAS to support **CMS stateless sites on K8s**
    - includes compute and data creation and orchestration and federation (XCache service)

DODAS also supports an **on-demand analysis facilities** offering (**on top of K8s**):

- HTCondor batch systems on demand, supporting HTCondor federations
    - Floking, routing and HTC/HPC mixing

DODAS is experiment independent and its community free. As such it can be adopted by "any experiment/use case"