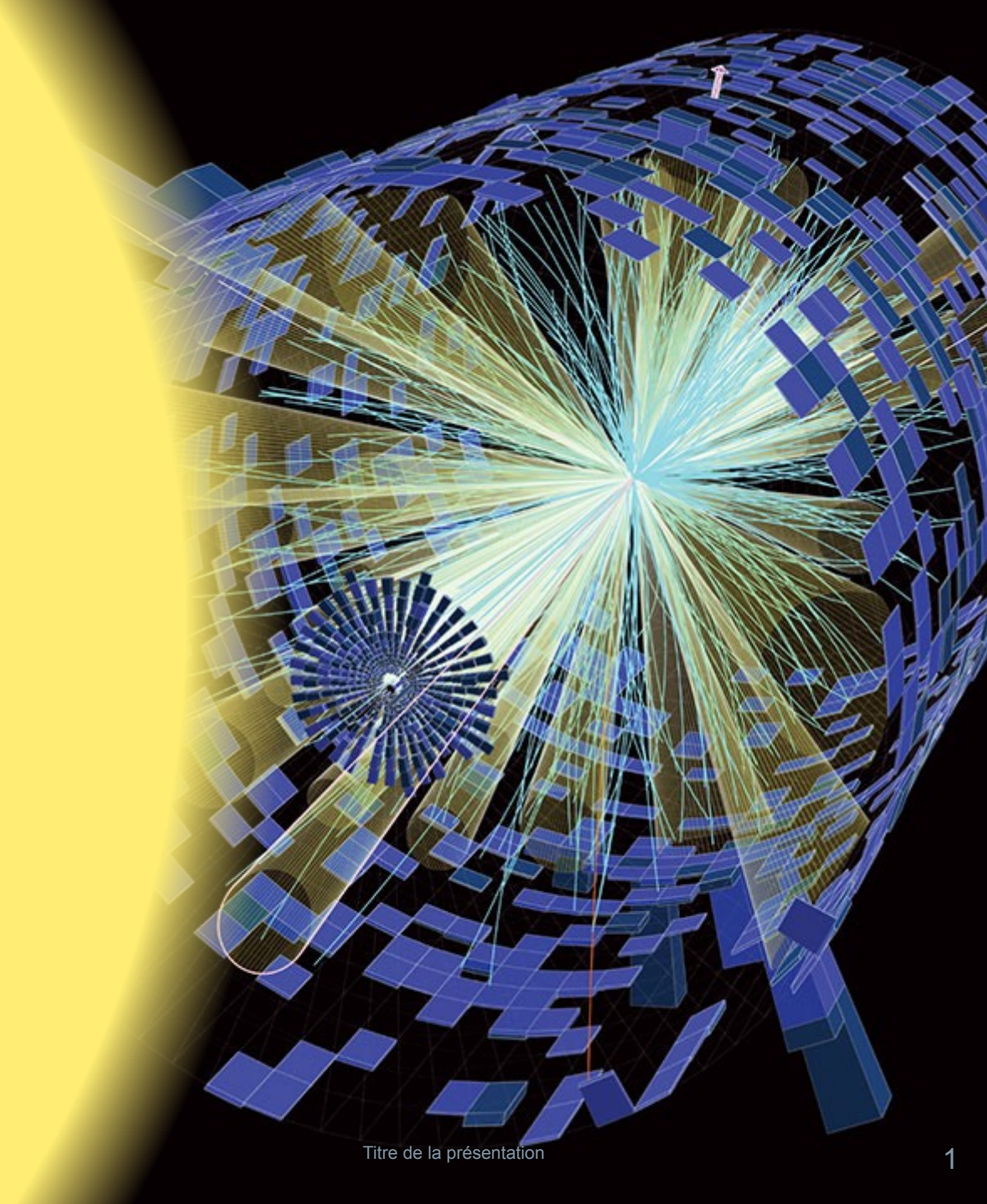
— IN2P3 —

LPSC GRENOBLE | MODANE

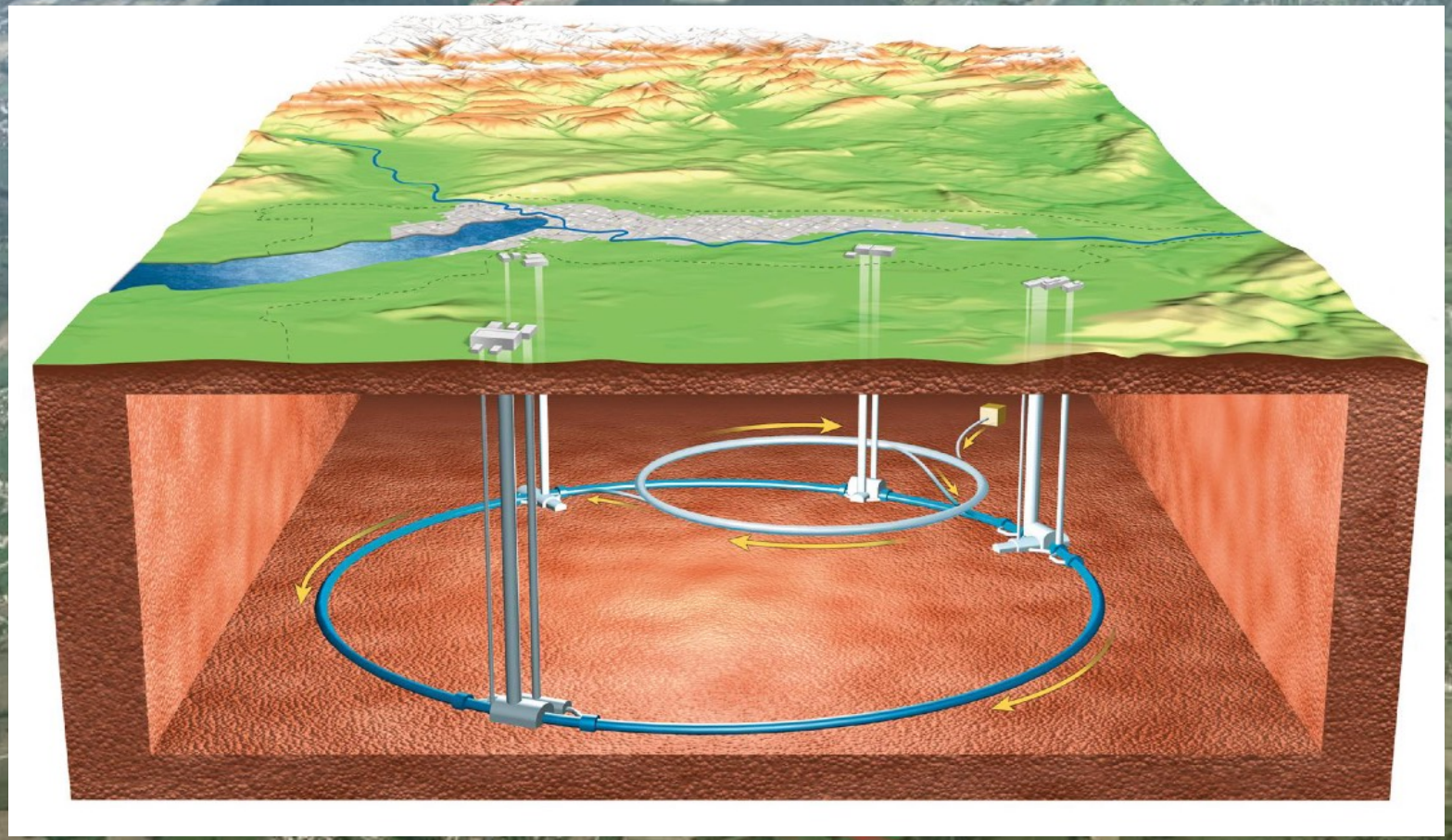


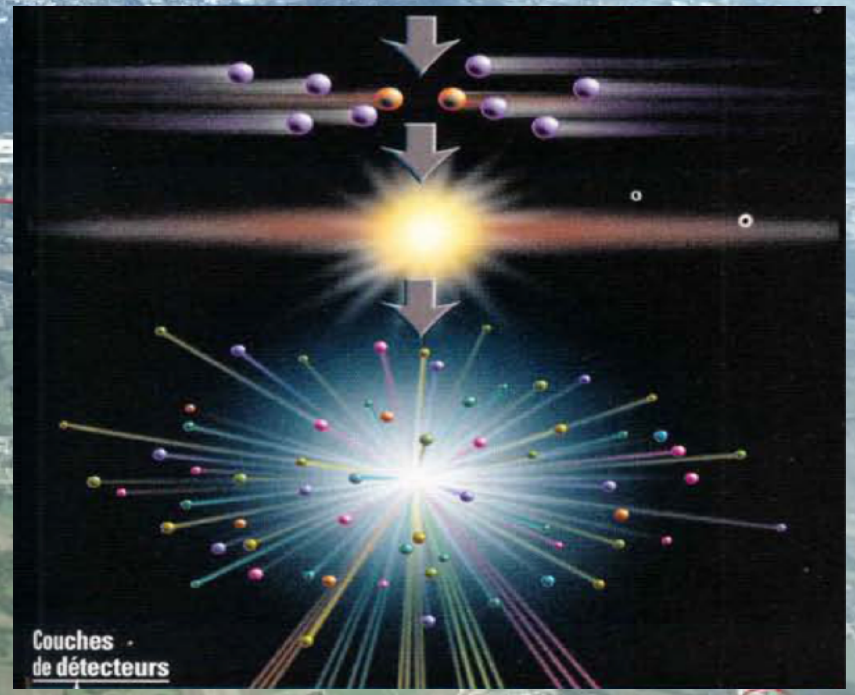
Organisation du traitement des données en physique des particules

Réunion Numpex IN2P3
28 janvier 2026

Sabine Crépé-Renaudin

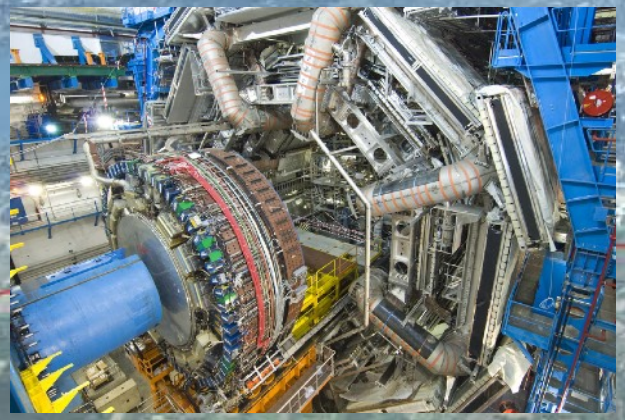
Le LHC



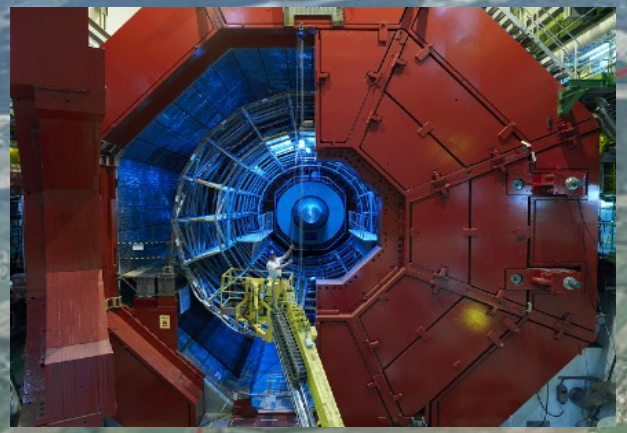


- 2 faisceaux composés de milliers de paquets de 100 milliards de protons accélérés à la vitesse de la lumière
- 40 millions de collisions par seconde

Les détecteurs

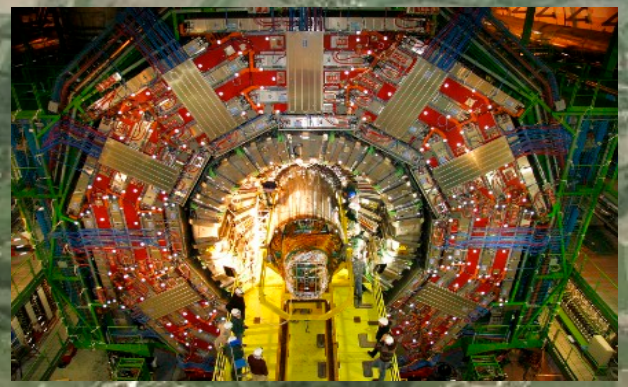


ATLAS
25x45 m, 7000 t



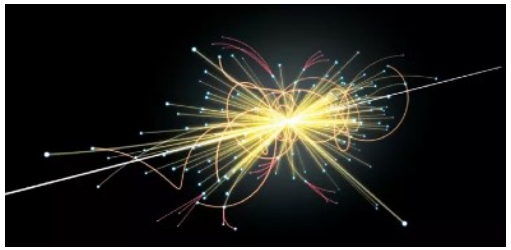
ALICE
16x26 m, 10000 t

LHCb
10x21 m, 5600 t

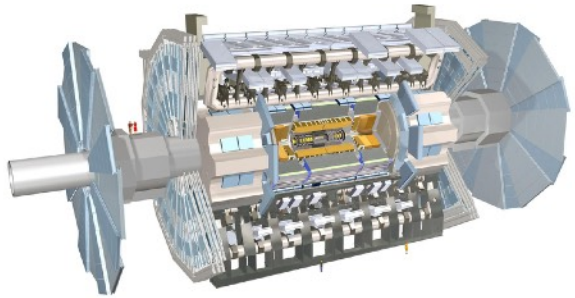


CMS
15x21 m, 10000 t

Une avalanche de données à traiter



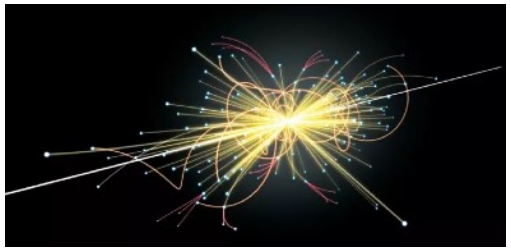
40 millions de
collisions par seconde



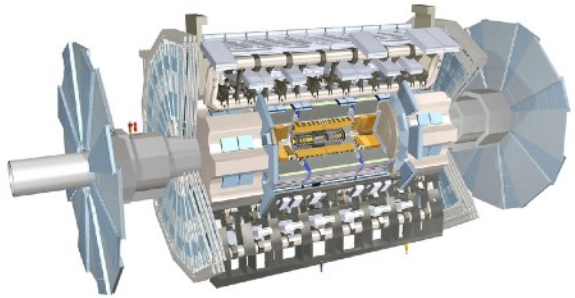
150 millions de canaux,
3Mo/collision

40 MHz
~100 To/s

Une avalanche de données à traiter



40 millions de
collisions par seconde

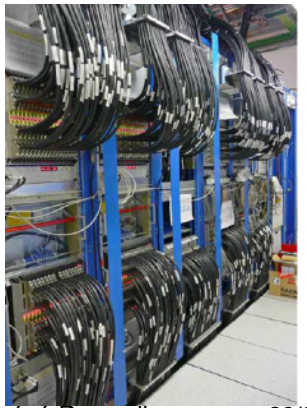


150 millions de canaux,
3Mo/collision

40 MHz
~100 To/s

Niveau 1, hardware
~ 2.5 µs

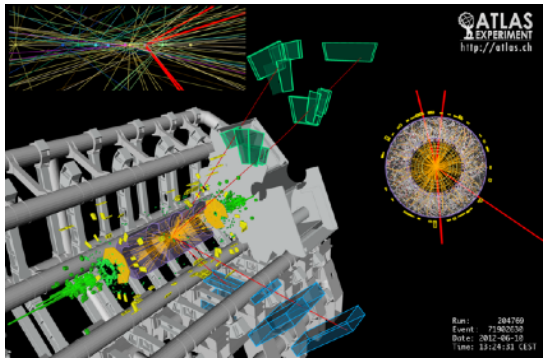
Système de
déclenchement



100 kHz
300 Go/s

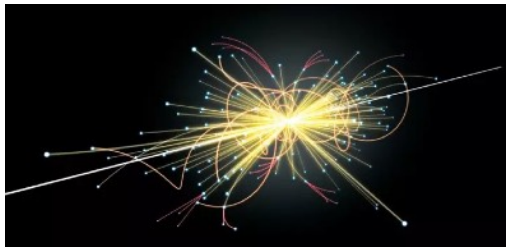
Niveau 2, software (60k coeurs CPU)
~ 500ms

3 kHz
6 Go/s

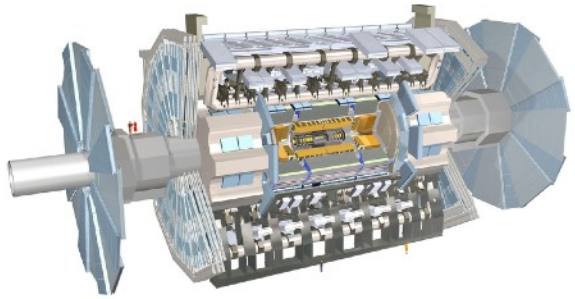


Stockage
Reconstruction
Analyse

Simuler les données

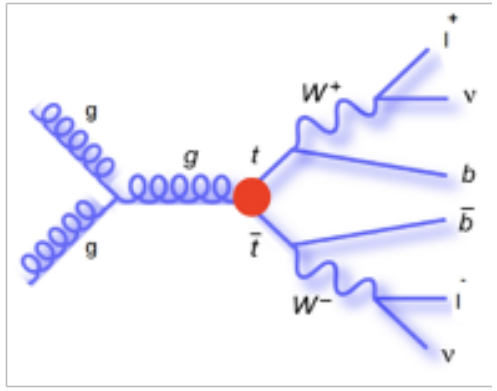


40 millions de
collisions par seconde

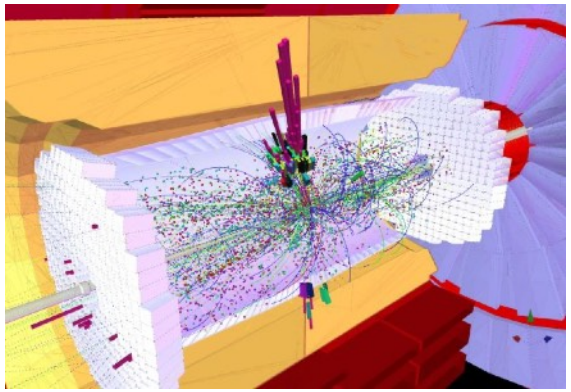


150 millions de canaux,
3Mo/collision

40 MHz
~100 To/s



génération



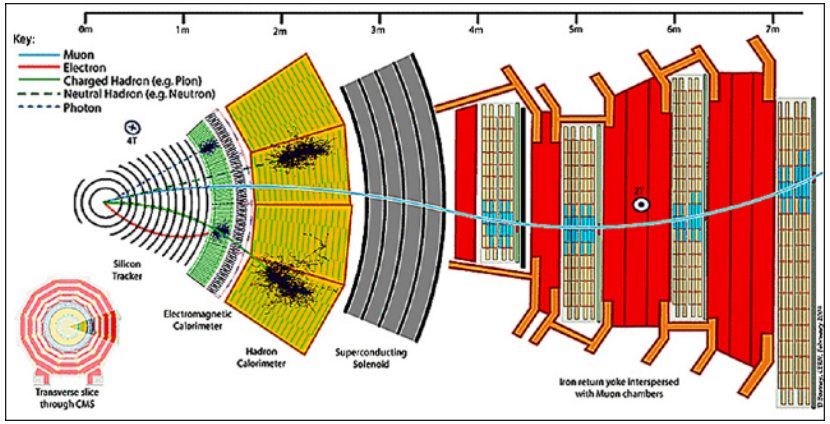
simulation de l'interaction
avec le détecteur

digitisation

Traiter les données du LHC

La reconstruction

- passer des signaux dans les détecteurs à des particules avec leurs caractéristiques

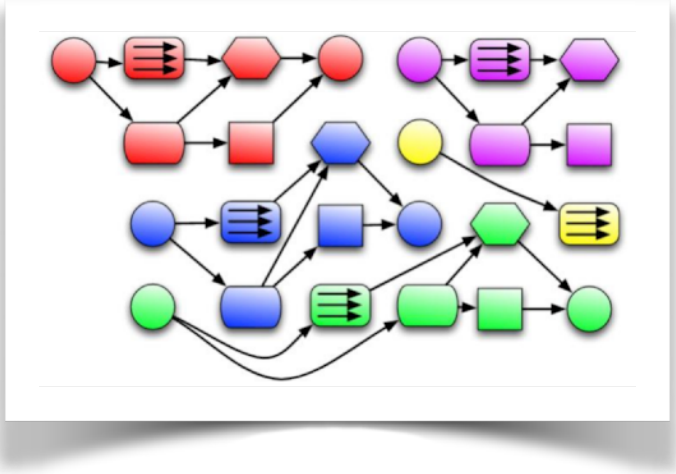


L'analyse

- sélectionner les collisions, les particules d'intérêts souvent avec un haut niveau de bruit de fond
  - ML employé depuis des décennies
- mesurer, comparer avec la théorie (simulation)

Des logiciels à la hauteur de la complexité des détecteurs

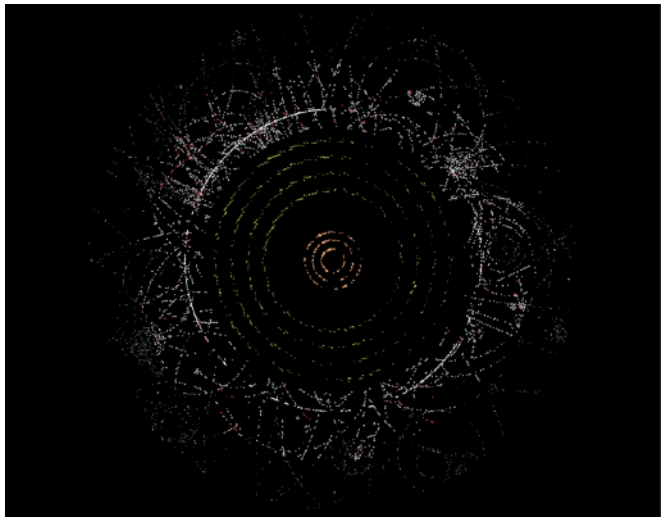
- développé par une centaine de personnes
  - ➜ 4 millions de lignes de code C++
  - ➜ 1 million de lignes de code en Python
- en évolution constante : multiprocess => multithreaded



Traiter les données du LHC

La reconstruction

- passer des signaux dans les détecteurs à des particules avec leurs caractéristiques

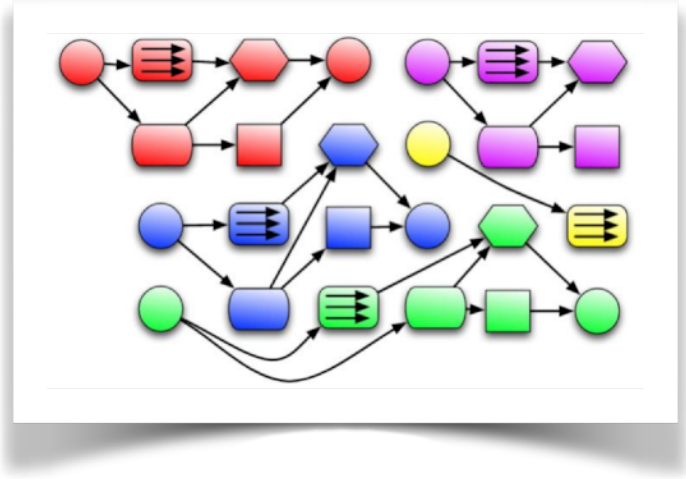


L'analyse

- sélectionner les collisions, les particules d'intérêts souvent avec un haut niveau de bruit de fond
- mesurer, comparer avec la théorie (simulation)

Des logiciels à la hauteur de la complexité des détecteurs

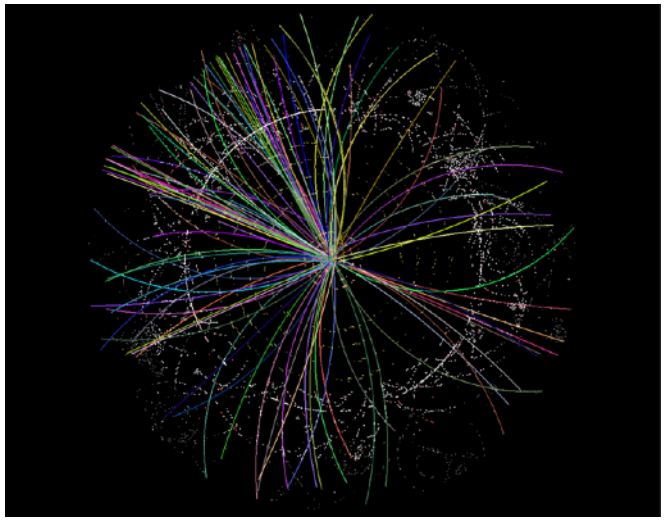
- développé par une centaine de personnes
  - ➜ 4 millions de lignes de code C++
  - ➜ 1 million de lignes de code en Python
- en évolution constante : multiprocess => multithreaded



Traiter les données du LHC

La reconstruction

- passer des signaux dans les détecteurs à des particules avec leurs caractéristiques

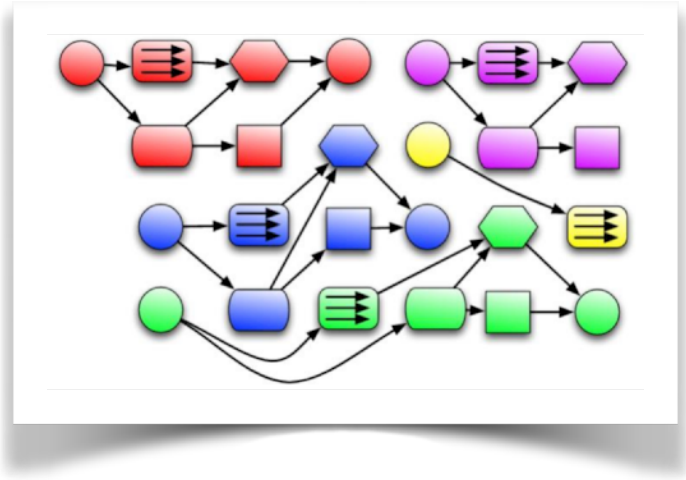


L'analyse

- sélectionner les collisions, les particules d'intérêts souvent avec un haut niveau de bruit de fond
- mesurer, comparer avec la théorie (simulation)

Des logiciels à la hauteur de la complexité des détecteurs

- développé par une centaine de personnes
  - ➜ 4 millions de lignes de code C++
  - ➜ 1 million de lignes de code en Python
- en évolution constante : multiprocess => multithreaded

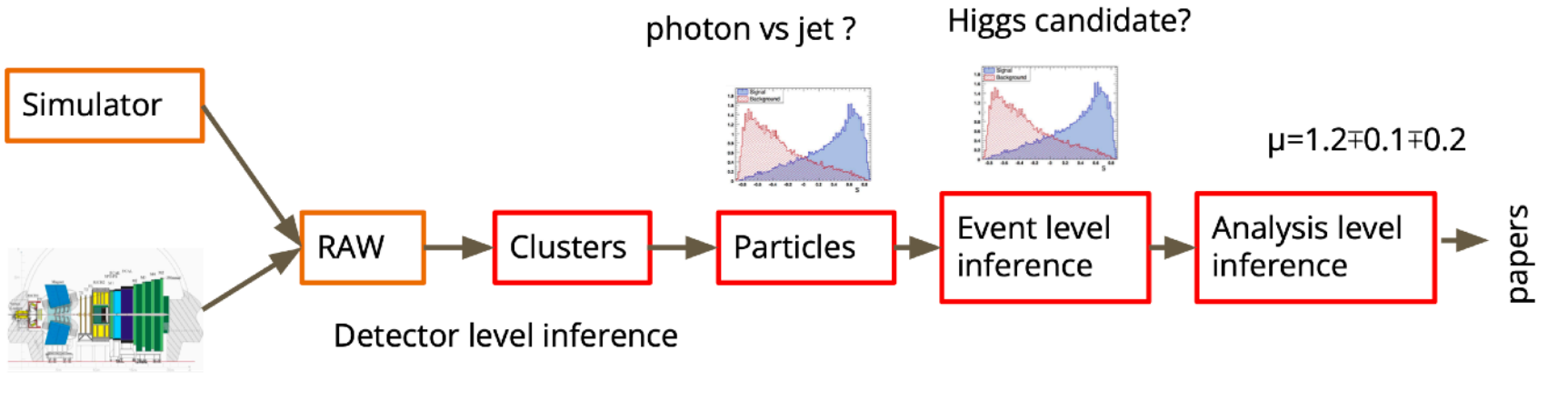


Des techniques d'analyse en évolution constante

Utilisation de l'IA depuis des décennies

- séparation signal/bruit de fond, identification des particules : BDT utilisés largement depuis les années 90

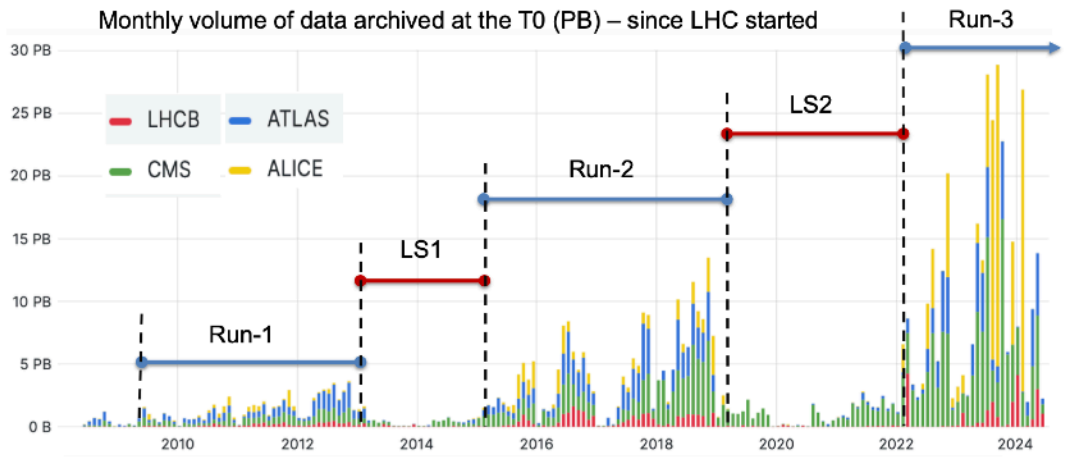
Utilisation de l'IA à tous les niveaux



- développement de l'IA pour accélérer la simulation (jumeaux numériques, model de substitution), pour la sélection, la reconstruction, le contrôle des accélérateurs, pour le déclenchement et l'analyse en ligne
- utilisation de techniques d'IA diverses : (BDT), conventionnal NN, variational auto-encodeur, Graph NN, DNN, generative adverserial NN
- des développements pour les adapter à nos spécificités : grande masse de données, simulations très précises
  - collaborations avec des chercheurs en IA
- implementation sur CPU, GPU ou FPGA

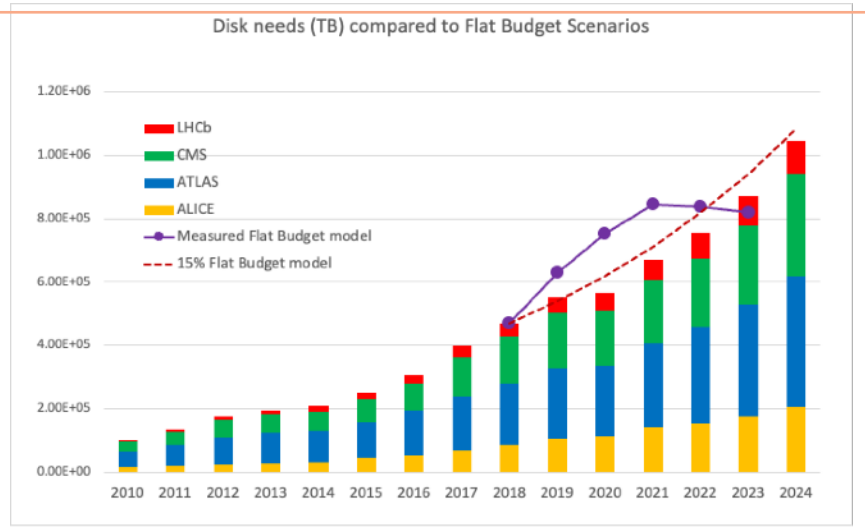
Utilisation actuelle : données

Au total : 3 Exaoctets de données stockées : disques + bandes

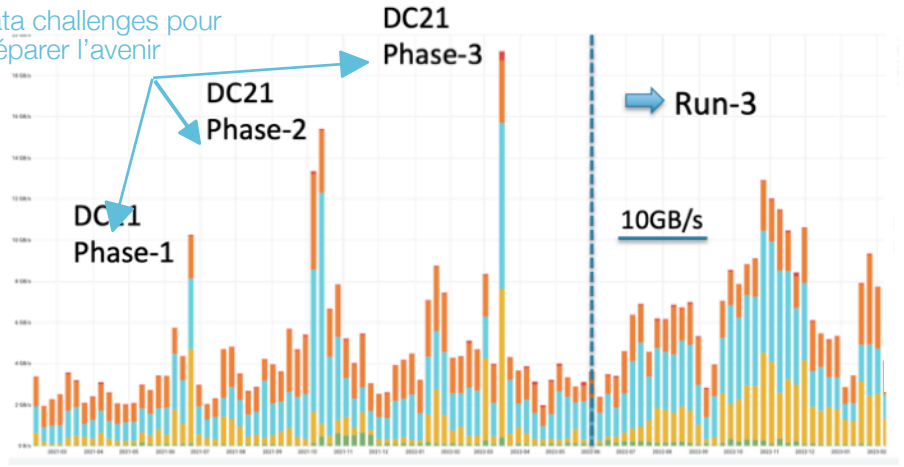


Pic à 14 Po/ jour

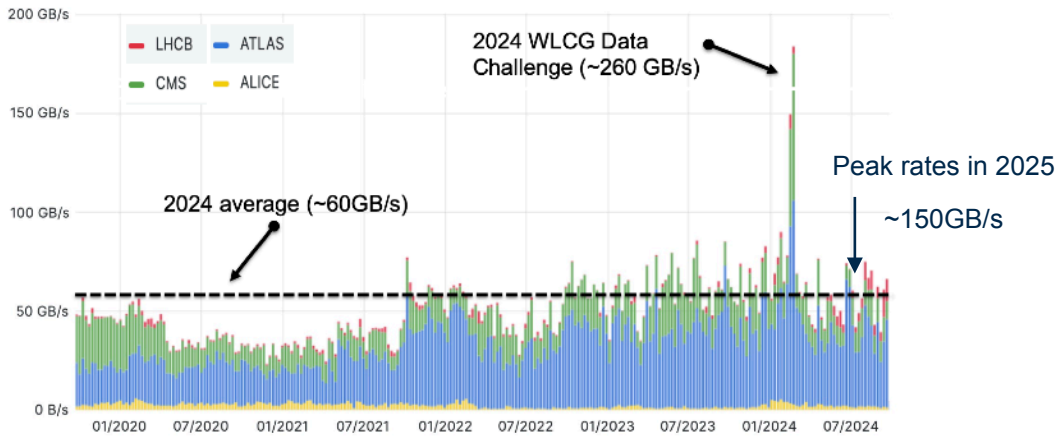
50Po/mois atteint



Data challenges pour préparer l'avenir



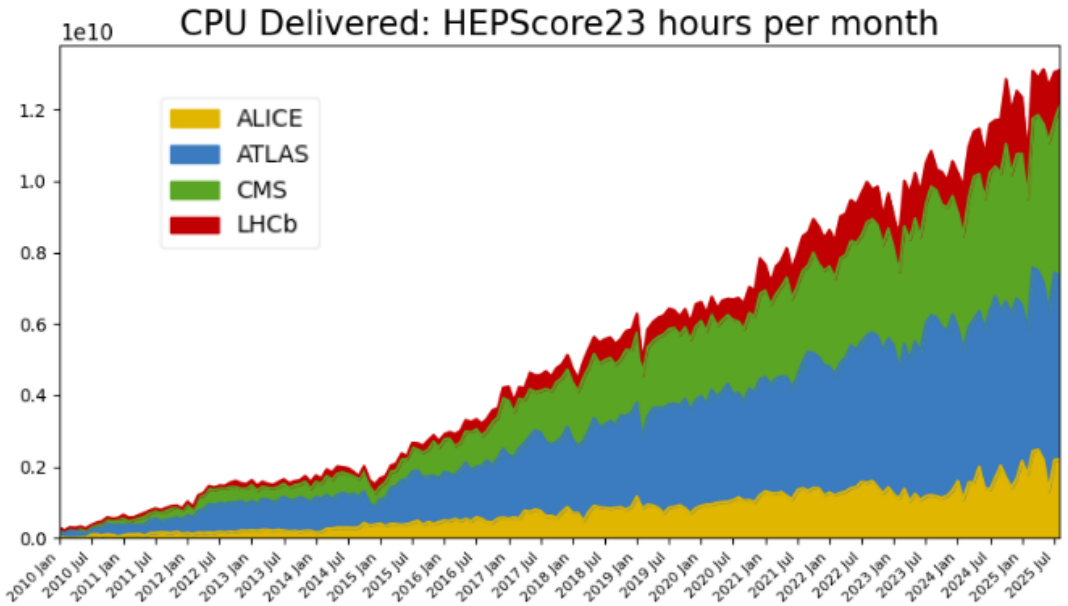
Transferts

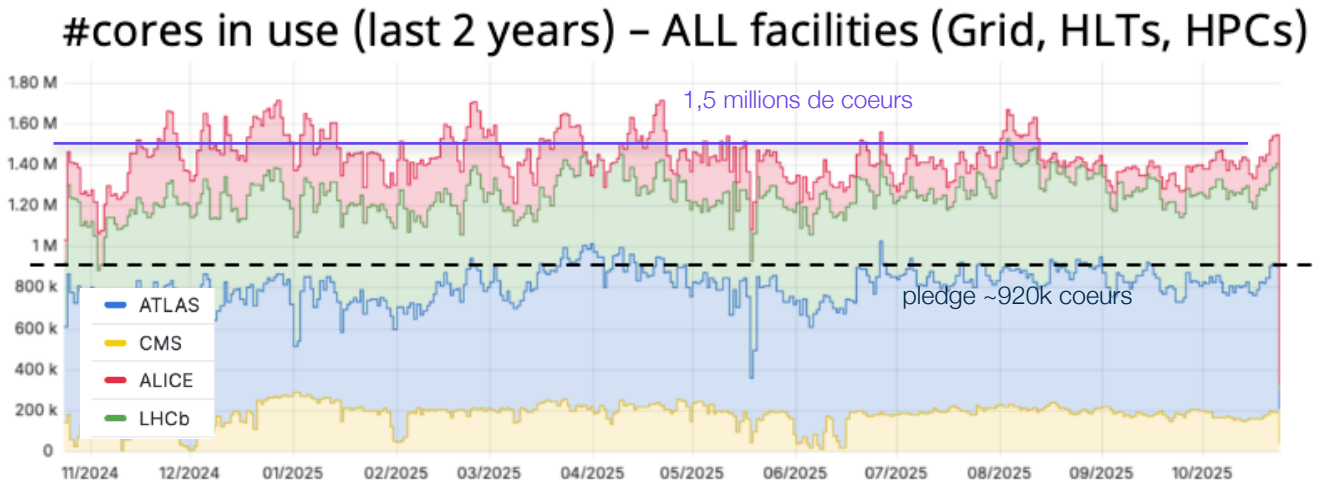


Utilisation actuelle : calcul

Évolution des besoins de calcul

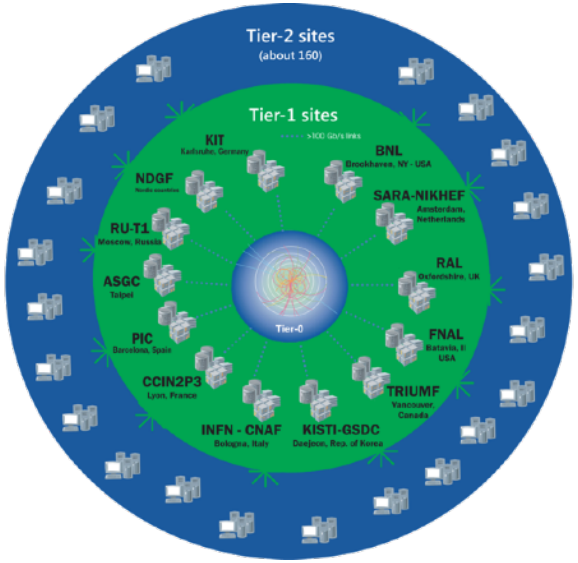
- unité HS23 = HEPScore23
  - 1 coeur récent ~10HS23 => 1,5 millions de coeurs
- nouveau bench déployé pour mieux prendre en compte les ressources hétérogènes
- CPU





L’infrastructure

Une infrastructure distribuée

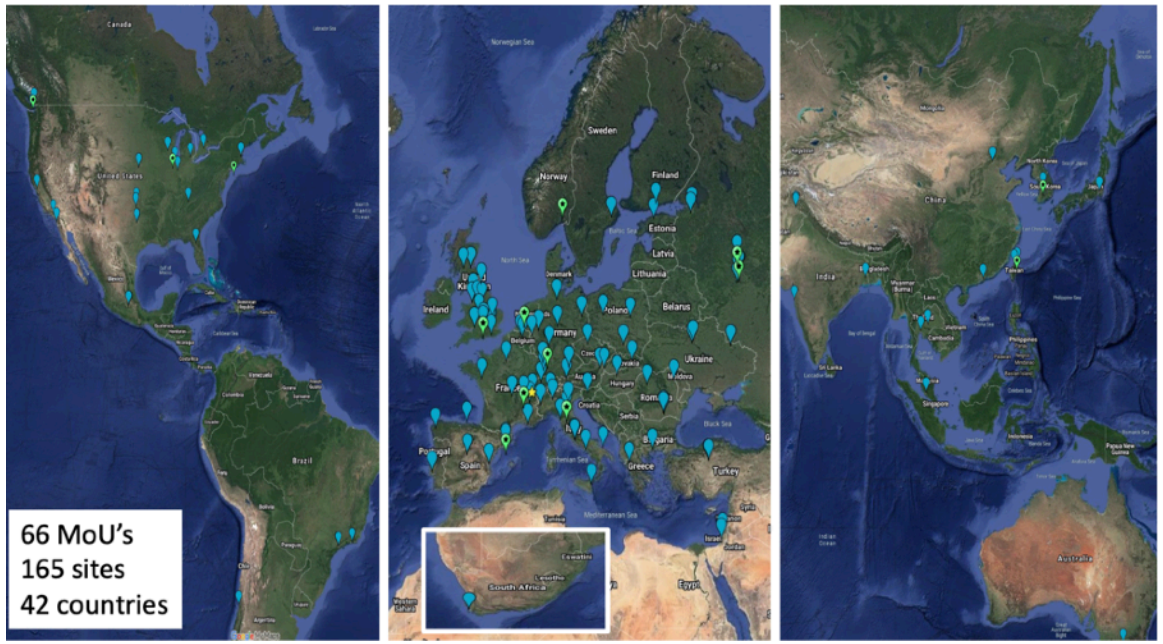


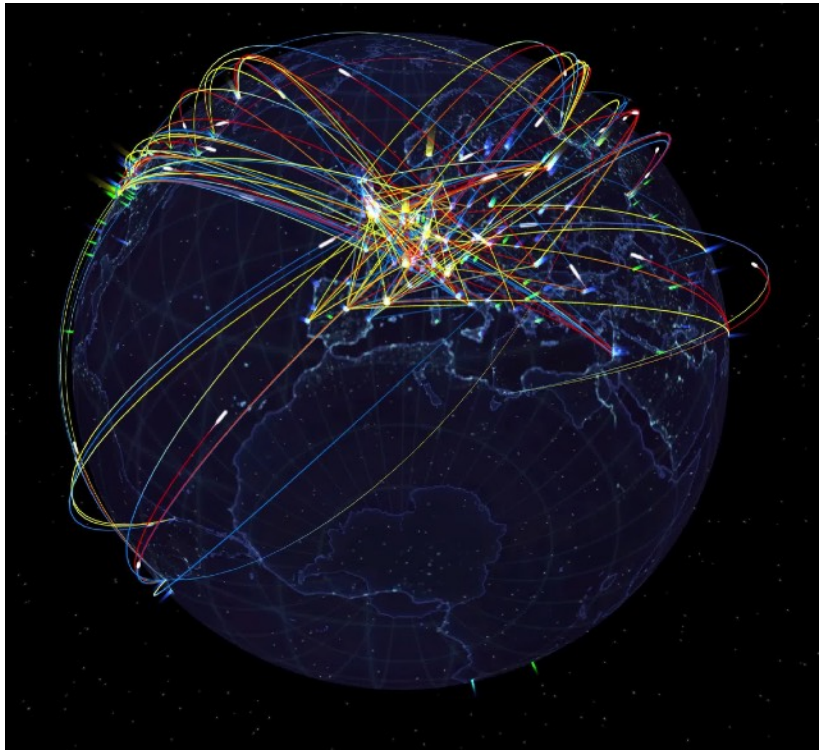
La grille de calcul WLCG

- 165 sites de calcul et de stockage dans 42 pays
  - niveaux : Tier 0 CERN, Tier 1 duplication des données brutes, Tier 2, Tier 3 uniquement analyse

La grille de calcul WLCG

- 1,5 millions de coeurs CPU utilisés 24/24 7/7 (x86 majoritairement)
- 3 Eo de données stockées
- un réseau performant (10-400 Gb/s)
- un accès transparent (mais sécurisé) pour les milliers de physiciens dans le monde qui analysent ces données
- des services permettant de gérer les données, leur traitement, leur référencement, le suivi de l'ensemble (monitoring)



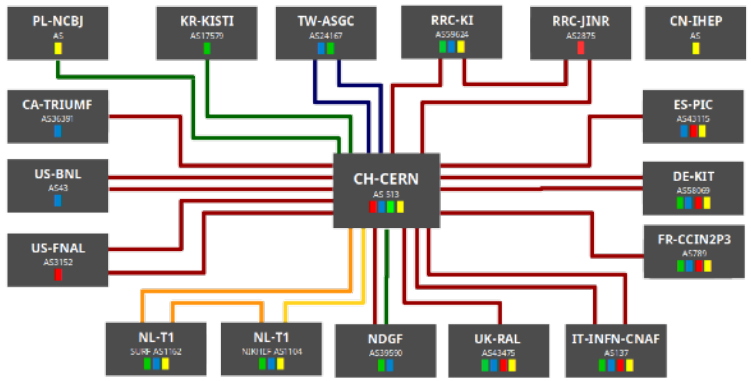


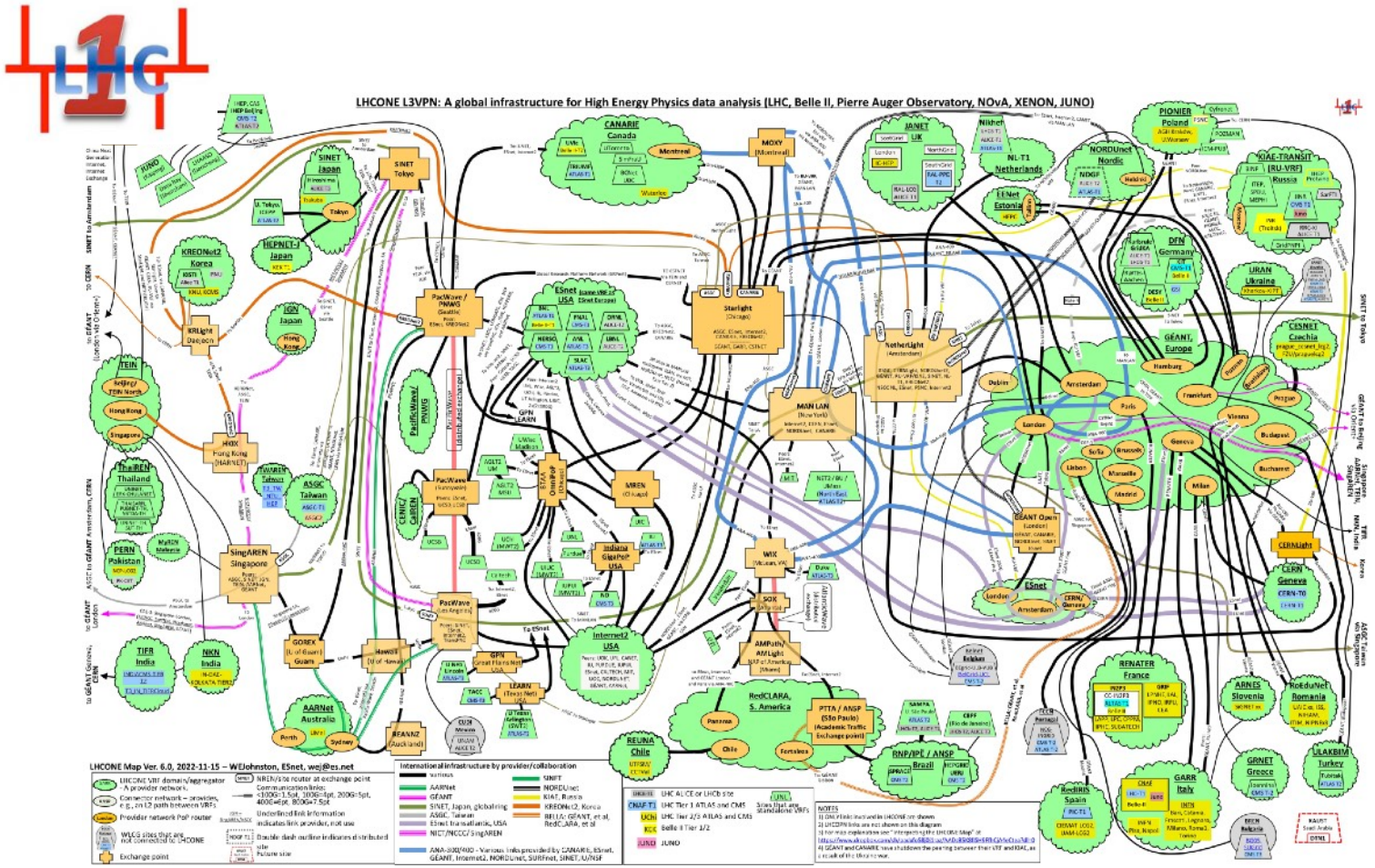
@

2 réseaux privés : LHCOPN pour les T1s et LHCONE pour une partie de la grille

- de 10 à 400Gb/s
- LHCONE maintenant ouvert à d'autres expériences : Belle2, Auger, Juno, Xenon…

LHCOPN

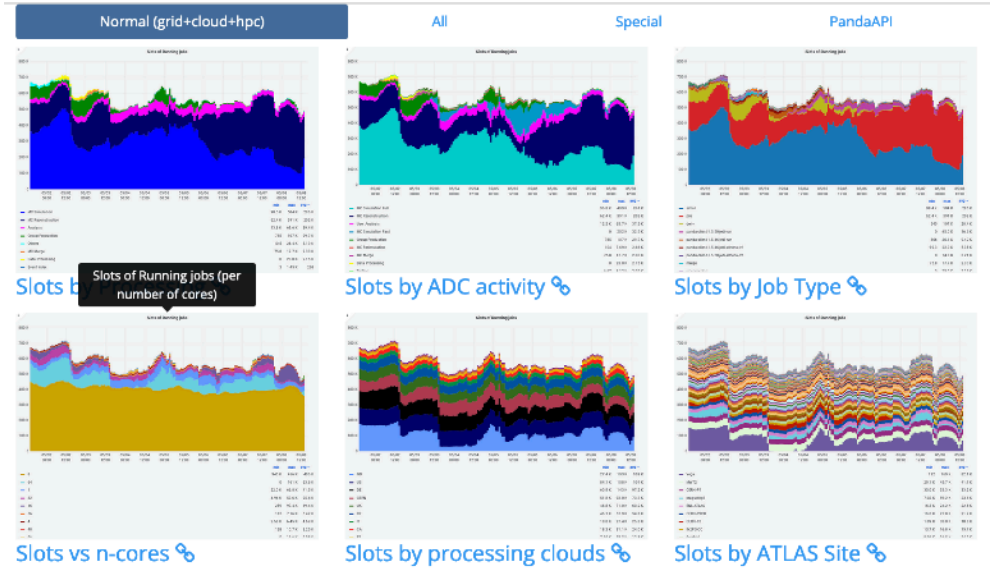




Les intergiciels, bases de données et outils de gestion

Des logiciels et bases de données pour orchestrer l'ensemble

- Distribuer les données de façon dynamique : FTS, xrootd, webdav, DDM, Rucio, DIRAC
- Distribuer les tâches de calcul : Panda, DIRAC, …
- Distribuer les logiciels : CVMFS = CernVM File System provides a scalable, reliable software distribution service.
- Bases de données : conditions de prises de données, détecteurs, logiciels, datasets, sites etc
- Monitoring : des sites, des tâches de calcul, des stockages, des transferts, du réseau…
- Systèmes de tickets : GGUS, JIRA



Fonctionnement

Authentification

VOMS X.509-based

- Validation des certificats utilisateurs au niveau des laboratoires/instituts d'appartenance
- Association du certificat à une VO (= Virtual Organisation) déléguée à la collaboration correspondant à la VO
- un site de WLCG décide quelle VO il supporte : il acceptera ttes les demandes d'utilisateur qui a la labellisation des VO qu'il supporte

Transition vers Indigo IAM (token)

- Seulement deux sources d'authentification : certificats X.509 ou SSO du CERN Single Sign On
  - identité des utilisateurs vérifiés et expiration dès que la personne n'est plus recensée dans la base de données du CERN

Fonctionnement

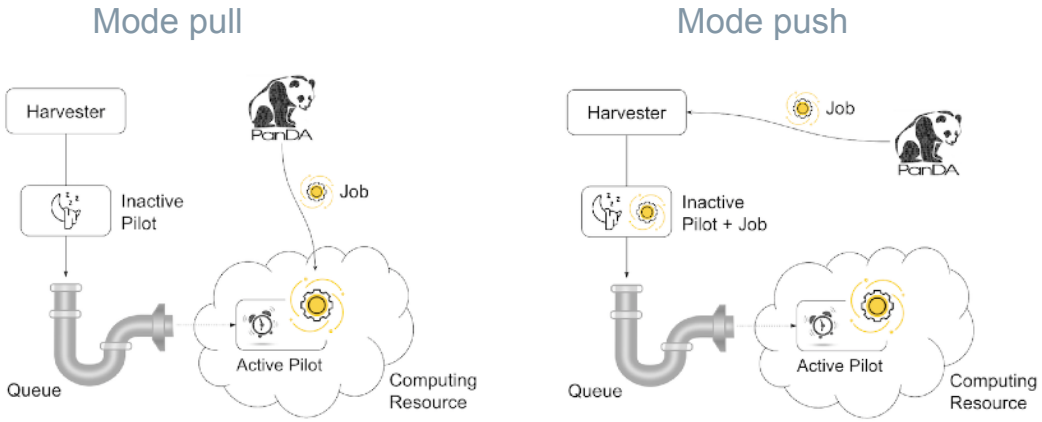
Distribution des tâches de calcul

Mode pull

- Les expériences envoient des jobs pilot légers aux centres de calcul qui supportent leur VO
- Ces jobs pilot
  - préparent l'environment,
  - se connectent à l'ordonnanceur central
  - récupère un job correspondant à la tâche à exécuter lorsqu'il y a des CPUs libres

Mode push

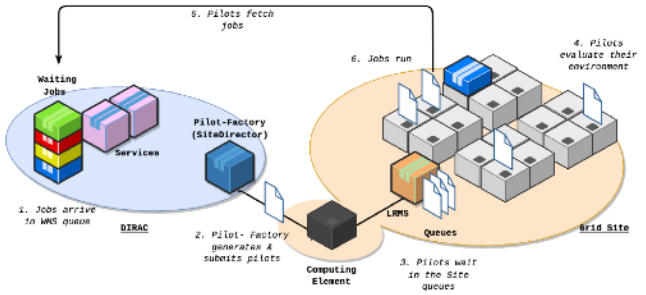
- Les pilots sont envoyés au centre de calcul avec des jobs de la tâche pré-assignés
- l'ordonnancement est fait à partir de l'ordonnancer du centre de calcul
- le pilot spécifie les besoins de chaque job et permettre la configuration du WN avec CPU, tps limite d'exécution …
- mode plus adapté au centre HPC et clusters GPU



Workload Management System (WMS): Transferring jobs to computing interfaces

Basics of DIRAC WMS

- Push model: Error-prone, but reintroduced to exploit HPCs with no external connectivity.
- Pull model: Pilot-Job paradigm is the most used way of submitting jobs.
- Vacuum model: (HLT Farm, VAC).



Ce dont nous avons besoin pour la production

- des collaborations internationales structurées avec une production continue centralisée et des logiciels en évolution constante
  - ➜ besoin d'accès à un dépôt de logiciels au CERN (CVMFS)
  - ➜ environnement logiciel (container)
    - Apptainer avec support des user namespace
  - ➜ accès centralisé via un robot (et pas une personne donnée)
  - ➜ accès continu
  - ➜ AAI inter-opérable

- un workflow de production centré sur les données et des données en quantité
  - flux de données en sortie pour la simulation, flux entrant et sortant pour la reconstruction
    - ➜ besoin de réseau performant
    - ➜ besoin d'espace disque
    - ➜ accès aux base de données des conditions des détecteurs (squid)

Ce dont nous avons besoin pour l’analyse

Besoins

- AAI
- Espace disque multi tera o10To
- Accès continu en deçà d’un certain seuil
- Container (idéalement singularity)

Cas d’usage pour le HPC

- gros training IA

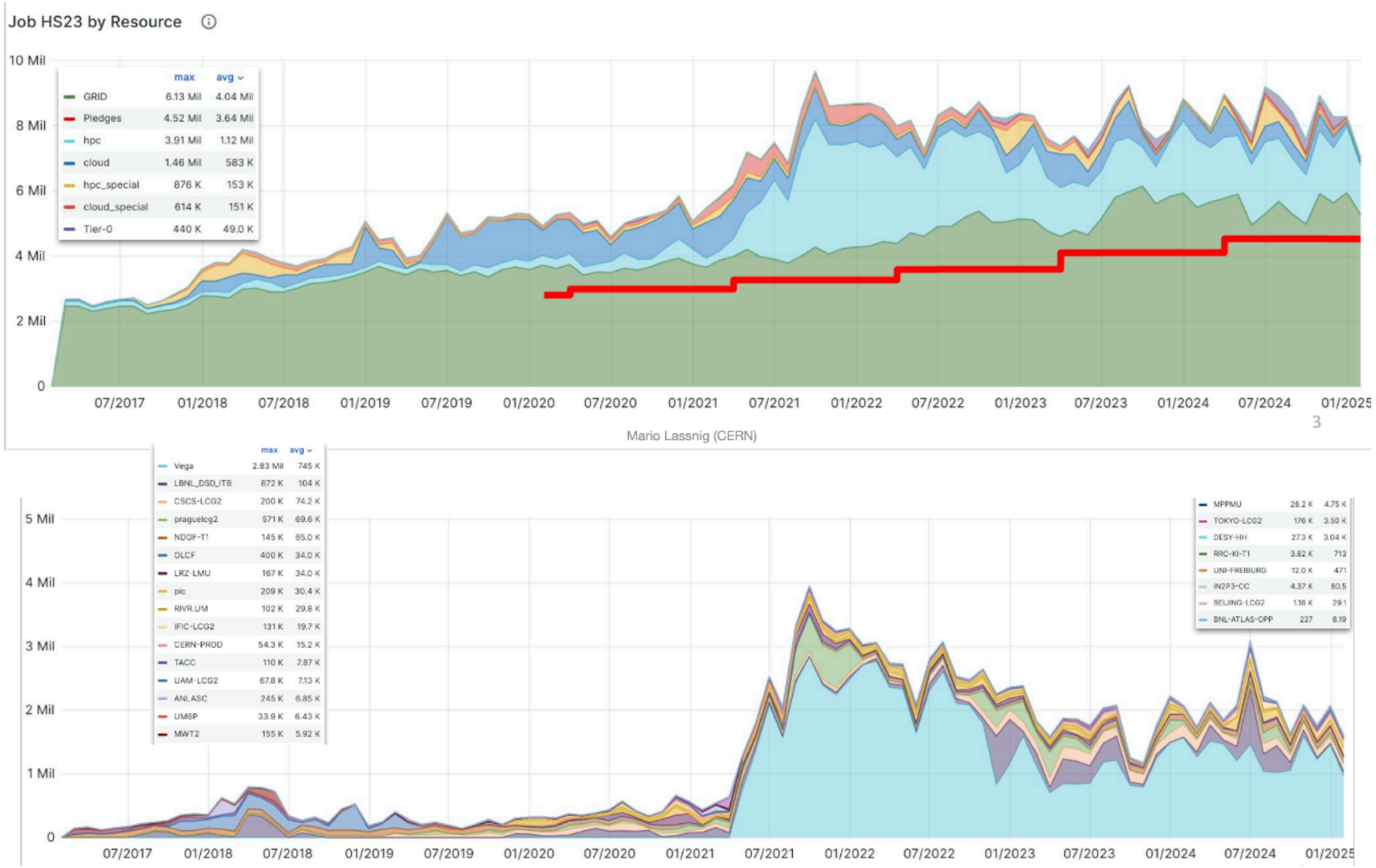
Bonus

Utilisation des machines HPC

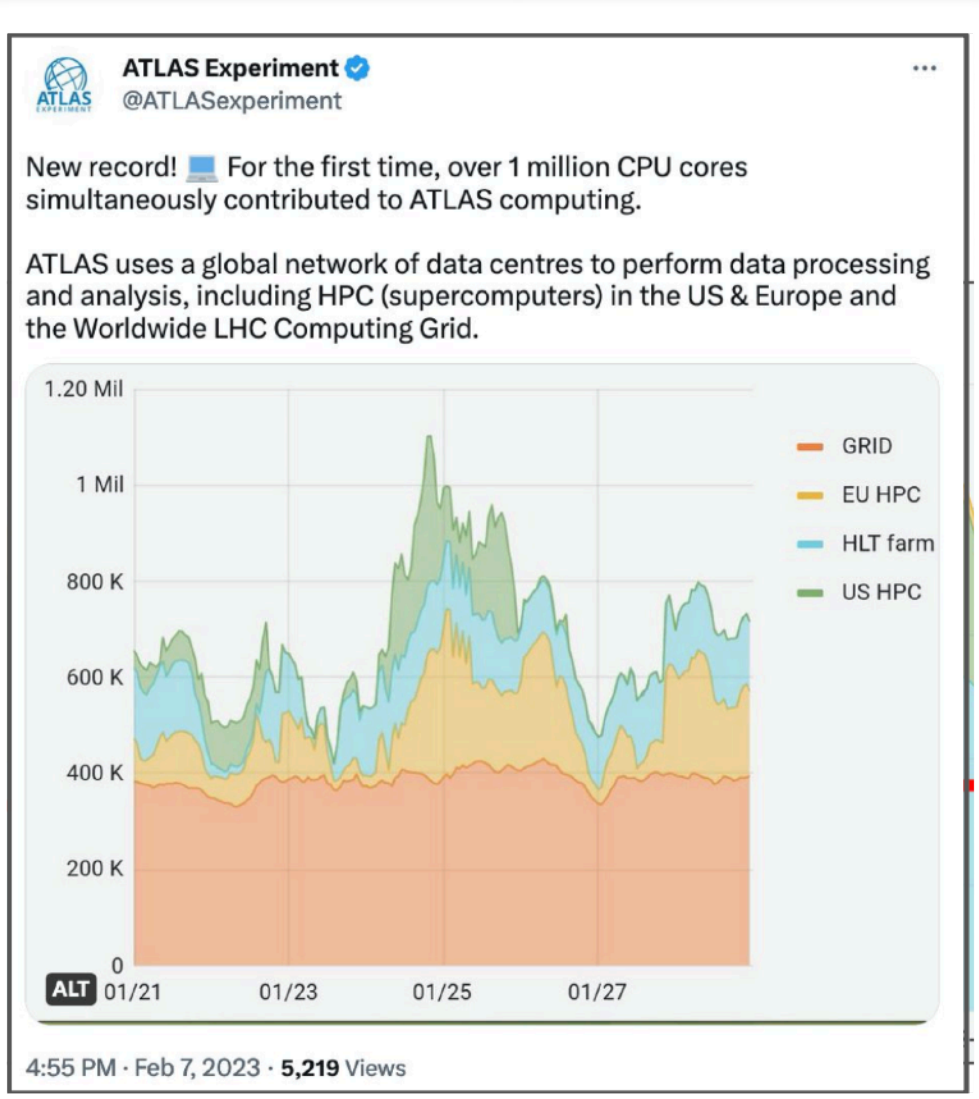
Utilisation actuelle du HPC

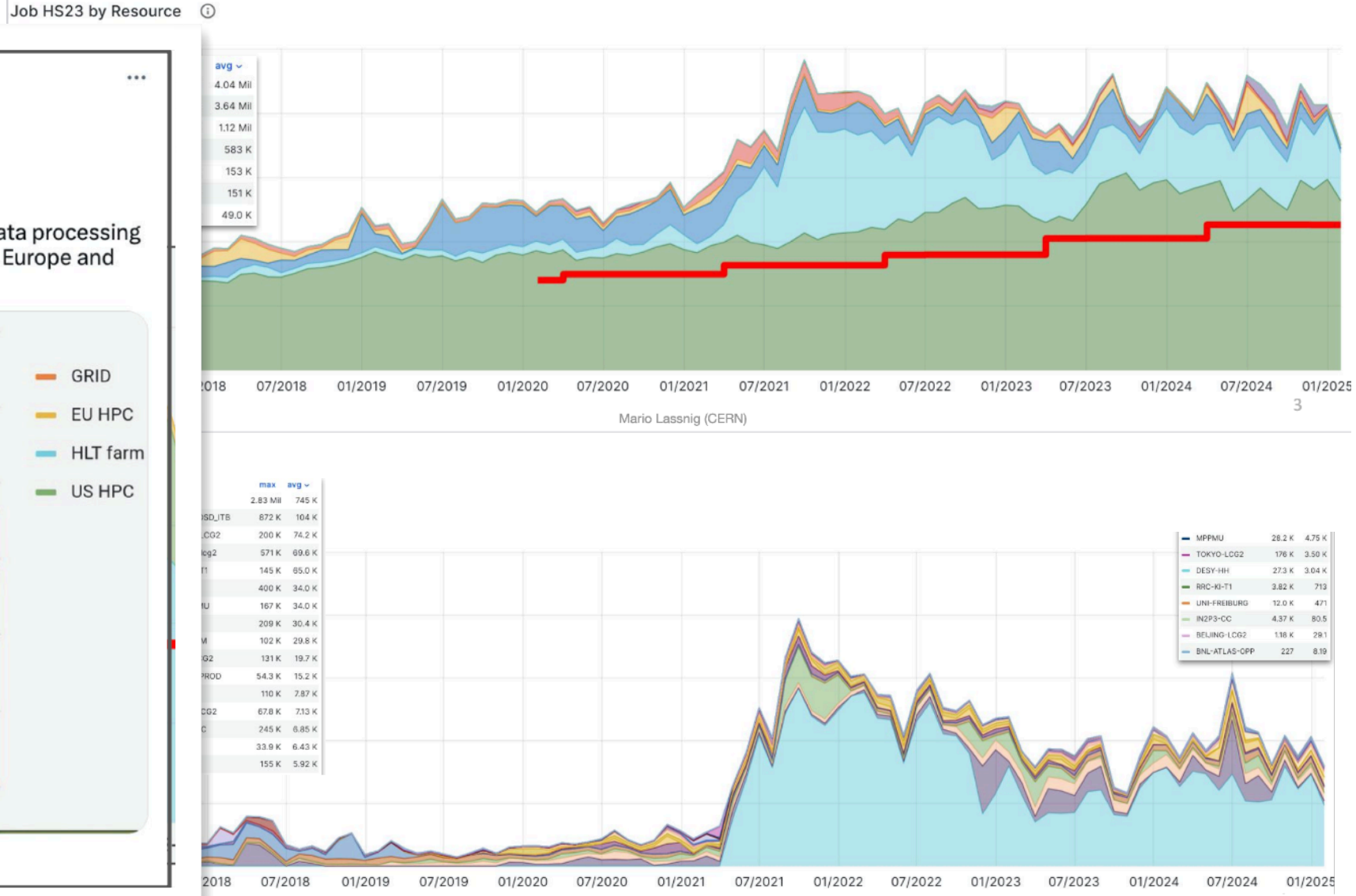
ATLAS :

- Utilisation transparente: Mare Nostrum, Norway, Vega, Karolina, CSCS, Leonardo, ..
- Utilisation complexe: Cori/Perlmutter, Toubkal, ...



Utilisation actuelle du HPC





Utilisation actuelle du HPC

CMS :

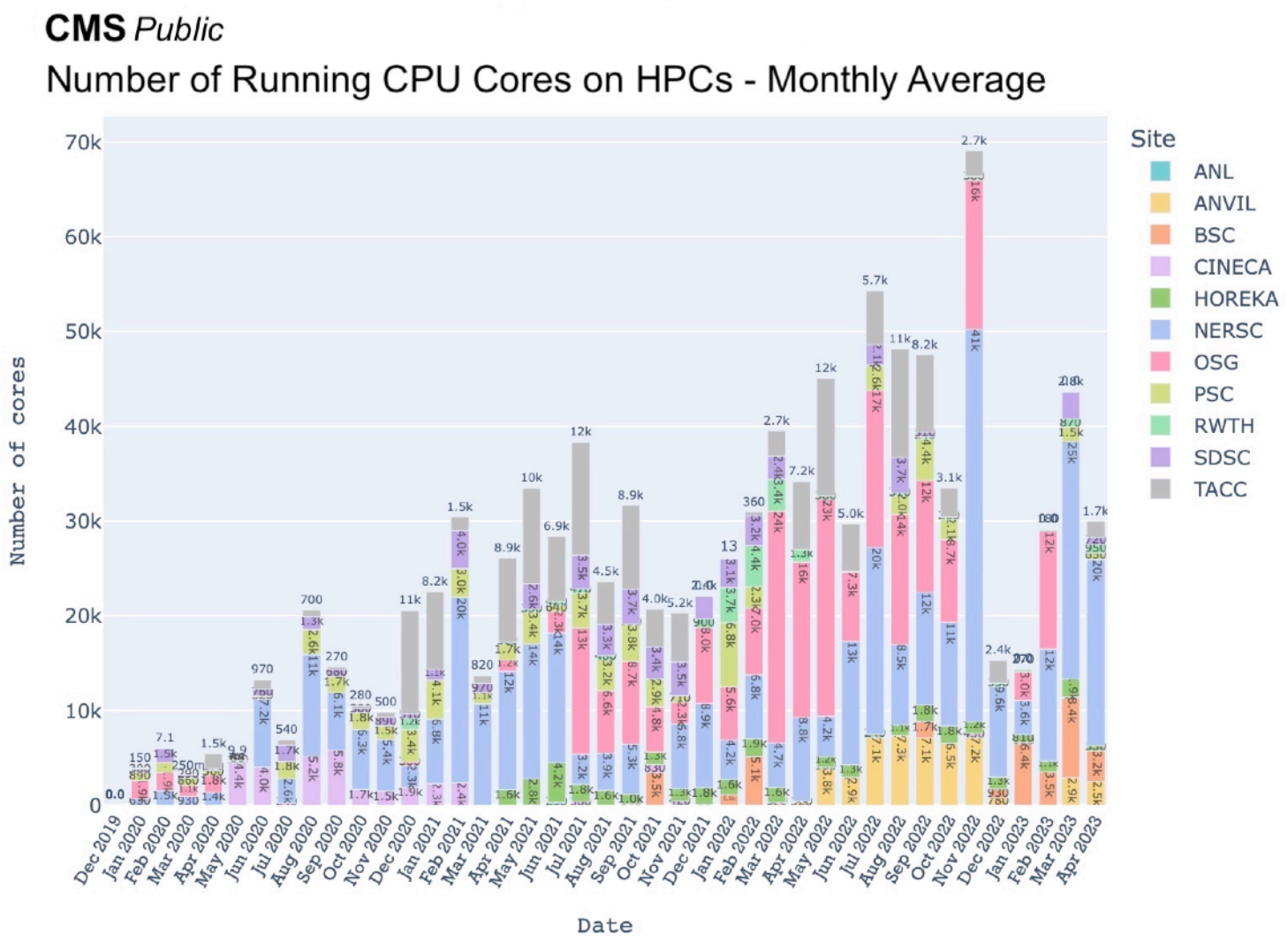
- Croissance de l'usage du HPC
- Sites avec utilisation transparente : RWTH, HOREKA, Marconi.
- Resources accédées via un service, ie HEPCloud, OSG : utilisé principalement pour la production Monte Carlo.

ATLAS et CMS utilisent de plus en plus de ressources HPC pour le traitement des données

- la grille de calcul est principalement constitué de CPU x86
- HPC utilisé de façon opportuniste (CPU) ou non, pour tous les processus ou uniquement pour la simulation

Pas de ressources françaises

- tests concluant à l'IDRIS il y a plusieurs années avec ATLAS
- mais pas implémenté
- aujourd'hui Equipex FITS (CC-IN2P3 - IDRIS -GENCI)
- possibilité via NUMPEX ?



Accès aux ressources HPC

Différentes adaptations en fonction des sites et des possibilités selon

- le mode d'authentification
  - ➜ convergence devrait être possible
- si les WN ont une connectivité vers l'extérieur ? ou seulement via le head node ?
- si le system batch est accessible de l'extérieur ?
  - ➜ possibilité de développer des **edge services** qui font le pont entre le stockage interne du site HPC et les protocoles externe (ex XRootD proxy, ARC Cache …) mais utiliser des protocoles standards serait plus efficace
- est-ce que CVMFS peut être accessible sur les WN ?
  - ➜ fat containers peuvent être utilisés mais restreignent les types de tâches
- le type d'allocation possible : Single core, multi-core, multi-node ?

Mode minimal à mode quasi transparent

- Mode minimal : accès via ssh et gros container
  - ➜ restreint les types de tâches
  - ➜ peut demander des ressources importantes au niveau des collaborations
- Mode transparent = quasi similaire site grille

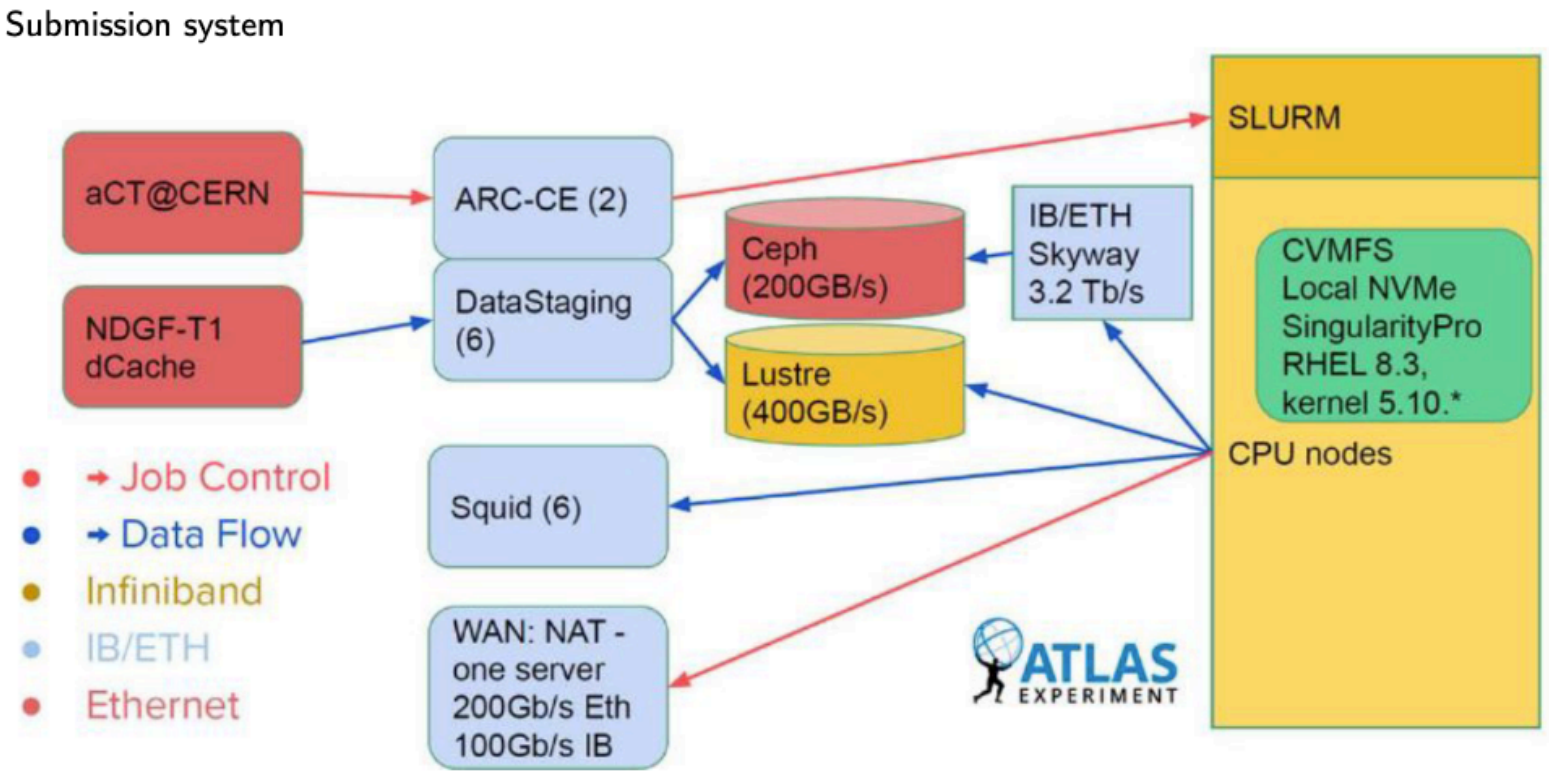
Exemple : CMS

Requirements for Production (not only testing)

- Submission. We need a Computing Element (a mediator between GWMS) and the local batch system (slurm, HTCondor,…):
  - Ok: a CE is available on site.
  - Ok-ish: connectivity to a subset of IP ranges (e.g. CERN or a « friend site »).
- Outgoing Network:
  - Ok: full outbound connection.
  - Ok-ish: connectivity to a subset of IP ranges (e.g. CERN or a « friend site »).
  - Showstopper: no connection to the ouside world.
    - The major obstacle derives from HTCondor, because worker nodes need to communicate externally to register the allocated resources to the central manager of the HTCondor pool, in order to be matched to requests, i.e. the payload jobs.
- Software distribution (CVMFS):
  - Ok: available on every compute node, served via a local squid.
  - Ok-ish: having just the client (rpm installed) or install it in user-space.
- Detector conditions distribution. We need a squid proxy towards CERN to cache them.
  - Ok: we have a local squid.
  - Ok-ish: we can use a frien site acting as the squid towards CERN.
- Architectures: any is good for CPUs (x86, aarch64) – for GPUs (AMD + NVIDIA, working on INTEL).
- Virtualisation: Ok: singularity. Ok-ish: docker/shifter/udocker, or Alma compatible OS if not virtualization is available.
- Storage: we don't really need storage on site if we have a friend site with ~40Gbit outgoing network.
- We can do non-opportunistic: actually, we prefer it.
- The site must be transparent to us: we can't redirect jobs there based on their characteristics.

Ex Vega

1. Vega



Easy environment

- ARC-CE + squid on site
- push jobs (ARC-CE orchestrates data transfers from/to dCache at NDGF)
- singularity Pro available
- OS: RHEL8
- CVMFS available
- running ATLAS all workloads

Ex Cineca

Overlay batch && site extension

Integration of CINECA into CMS Computing

CMS and CINECA were able to agree (2019) on a minimal set of changes to allow for CMS job processing.

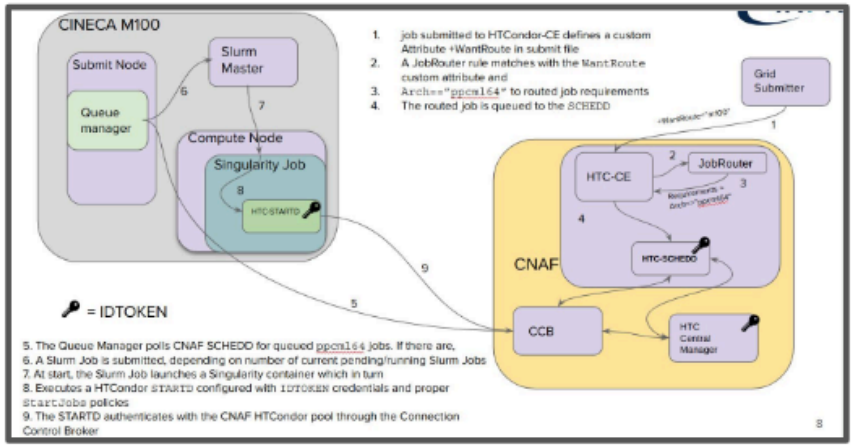
- CVMFS was installed on the systems; Network routing to CERN and CNAF IP ranges activated; Singularity audited and deployed;

CINECA nodes were configured as **an elastic extension of CNAF Tier-1** (SubSite concept) receiving all the jobs targeted for the standard WLCG site.

- Input data access to the AAA Data Federation via xrootd proxy (@CNAF)
- Worker Nodes provisioning: via site launched glidein Relying also on custom matching rules (defined at site)
- A cherry-picking method has been adopted allowing site-level specification of additional requests with respect to CNAF nodes in order to select most suitable workflows.

Same setup used also to integrate VEGA, a transnational site extension

Prototyping also a slightly evolved model transparent T1 batch extension



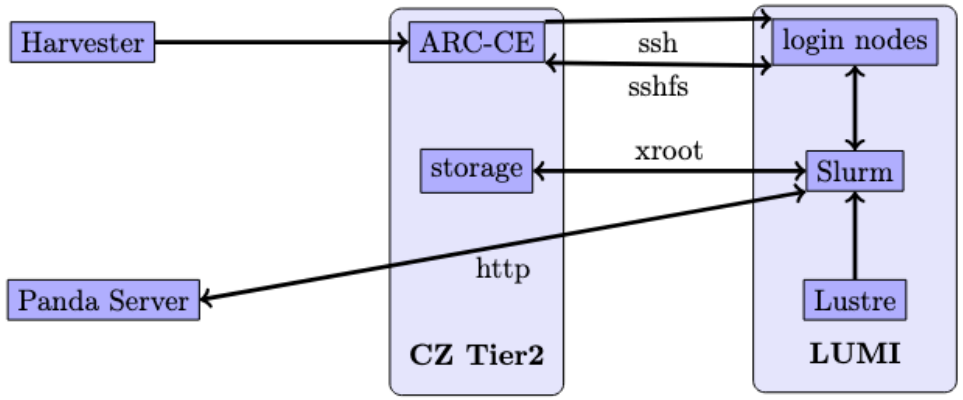
11

- CVMFS
- Singularity
- Lien CNAF T1
- xrootd proxy
- Tout type de job

Ex Lumi

3. LUMI

Submission system



- the ARC-CE (located at CZ Tier2) receives a pilot job, translates the job description into script that can be run in the Slurm batch system, puts necessary files into a folder shared with the HPC via sshfs, and submits the job via ssh connection to a login node
- when the batch job starts, pilot contacts panda server to receive payload job
- if it receives payload job, it gets input file from CZ Tier2 storage, starts the calculation (in software container), and sends outputs to CZ Tier2 storage when the payload finishes
- if the pilot can expect that another payload would finish, it requests it

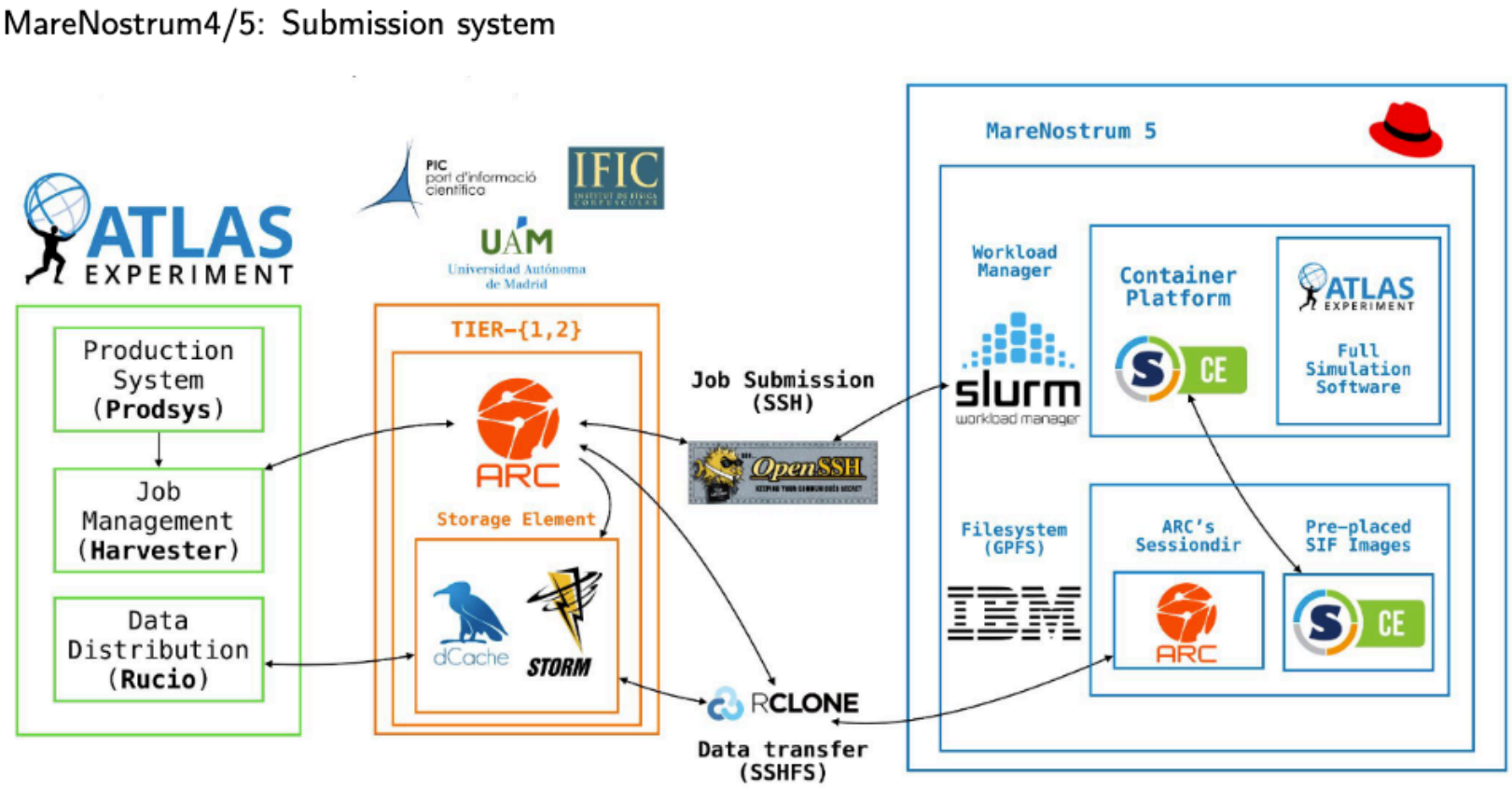
12/18

difficult environment

- OS is SUSE
  - grid middleware supported for RHEL only
- user namespaces will NEVER be enabled
- outgoing connectivity allowed
- environment compatible with centos8
- CVMFS not installed
  - => we were running in fat containers
- local apptainer installation not available but singularity is there
- =>Use
  - pilot needs to run on bare metal
  - stage-in/out runs in EL9 container
  - payload runs in fat containers → runs simulation only

Ex MareNostrum

4. MN4/5, MUC



Environment

- no outbound connectivity
- push jobs (ARC-CE orchestrates data transfers)
- running only fat container simulation