Extending the ATLAS PanDA Workload Management System for New Big Data Applications

XLDB 2013 Workshop

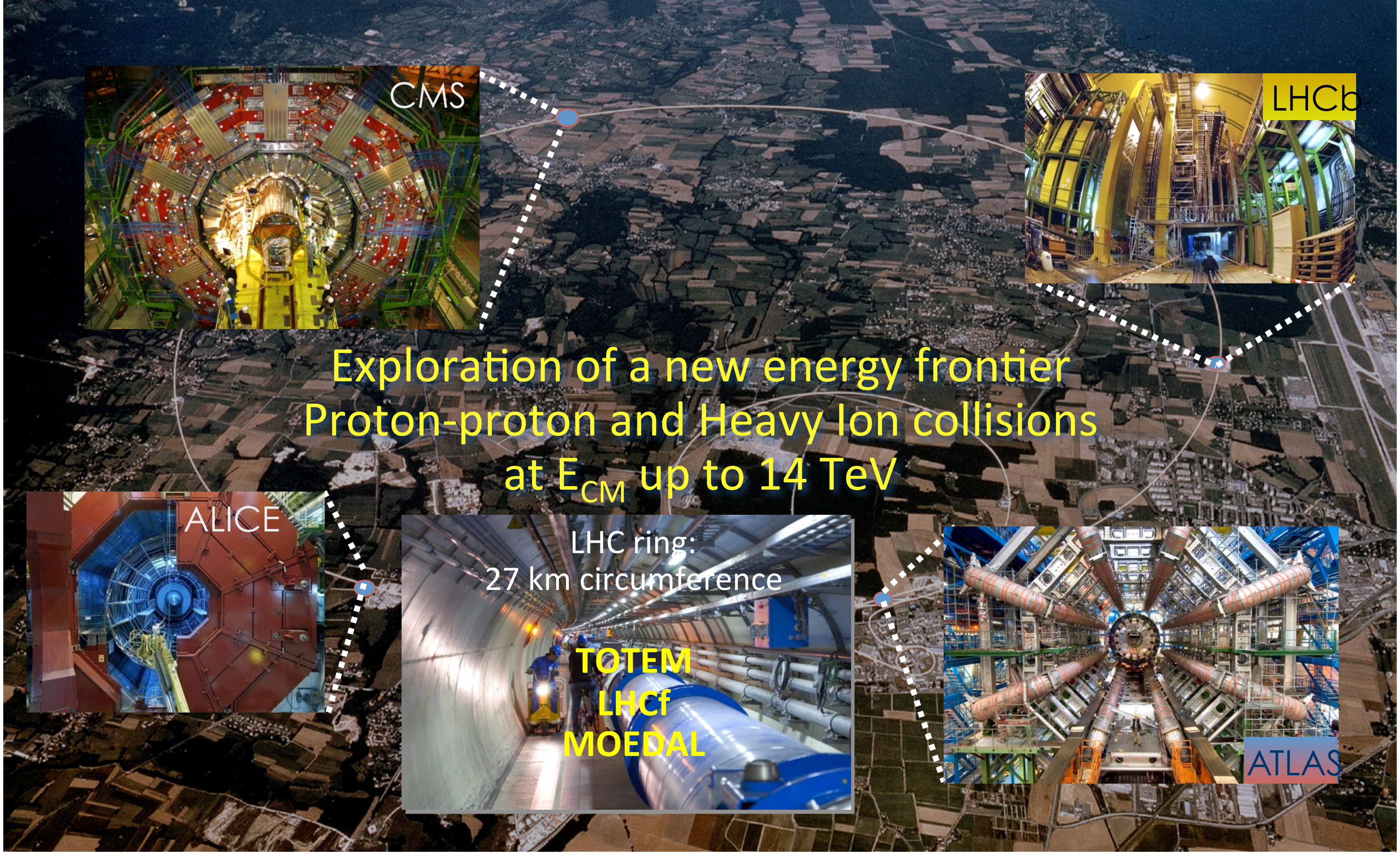
CERN, 29 May 2013

Alexei Klimentov

Brookhaven National Laboratory

Enter a New Era in Fundamental Science

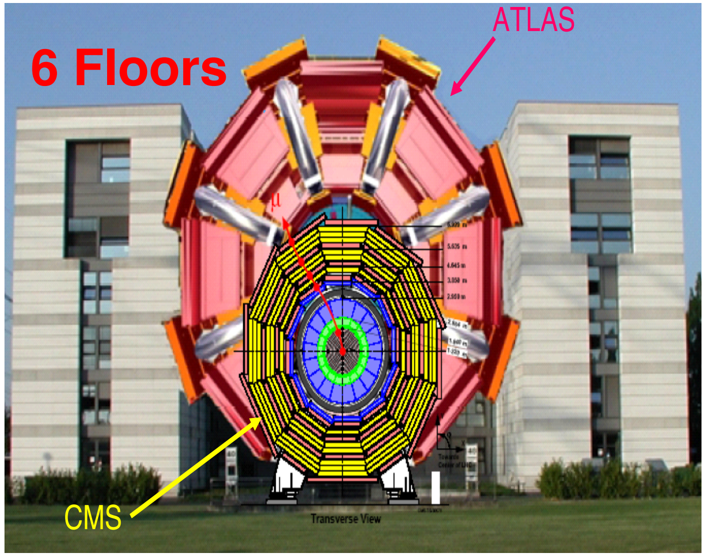
The Large Hadron Collider (LHC), one of the largest and truly global scientific projects ever, is the most exciting turning point in particle physics.



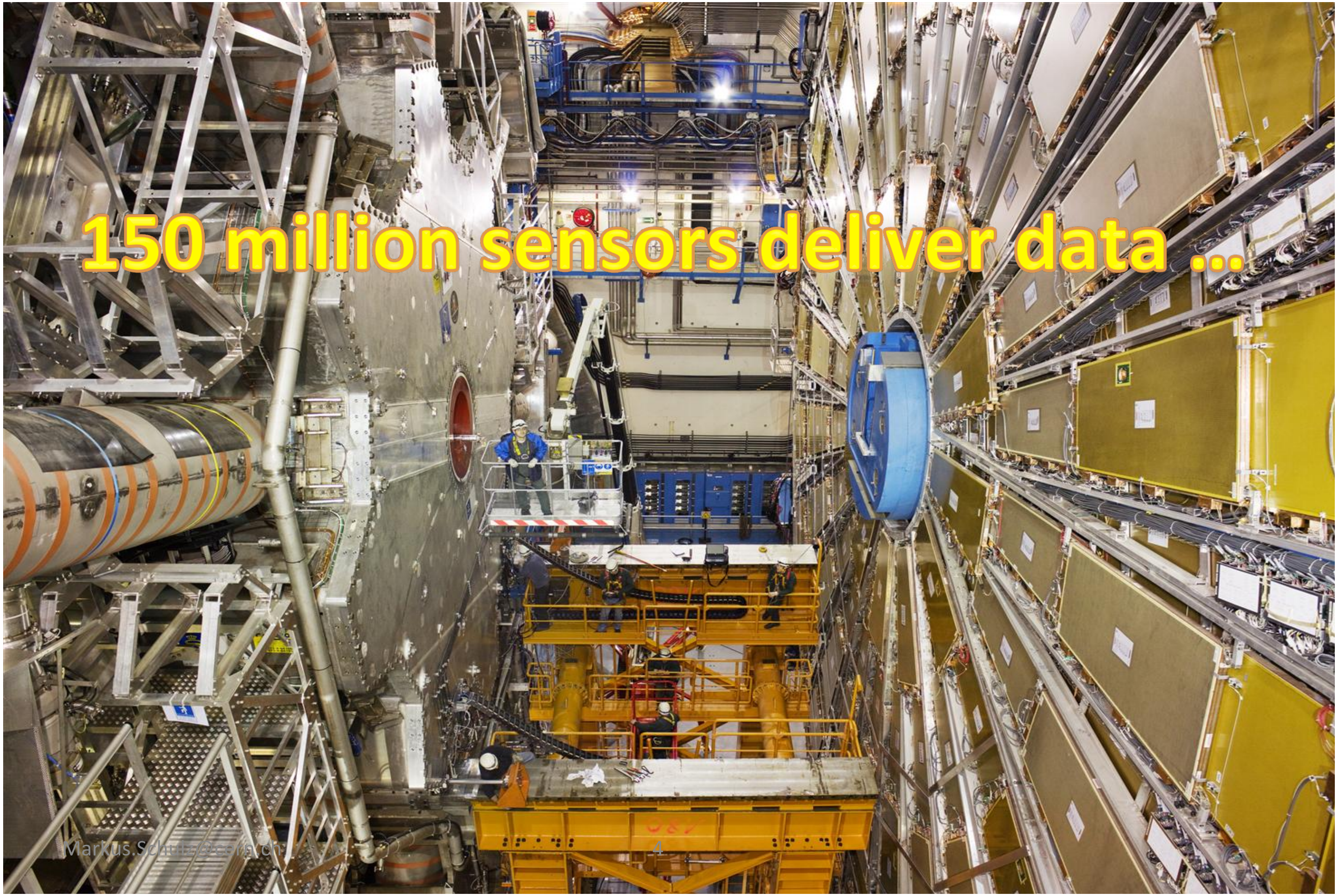
ATLAS

- A Thoroidal LHC ApparatuS is one of the six particle detectors experiments at Large Hadron Collider (LHC) at CERN
- The project involves more than 3000 scientists and engineers in ~40 countries
- ATLAS has 44 meters long and 25 meters in diameter, weighs about 7,000 tons. It is about half as big as the Notre Dame Cathedral in Paris and weighs the same as the Eiffel Tower or a hundred 747 jets


Notre Dame de Paris

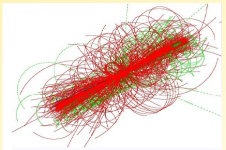


ATLAS. BigData Experiment



Our Task

Reality



We use experiments to inquire about what "reality" (nature) does

ATLAS Physics Goals

- Explore high energy frontier of particle physics
- Search for new physics
  - Higgs boson and its properties
  - Physics beyond Standard Model – SUSY, Dark Matter, extra dimensions, Dark Energy, etc
- Precision measurements of Standard Model parameters

We intend to fill this gap

Theory

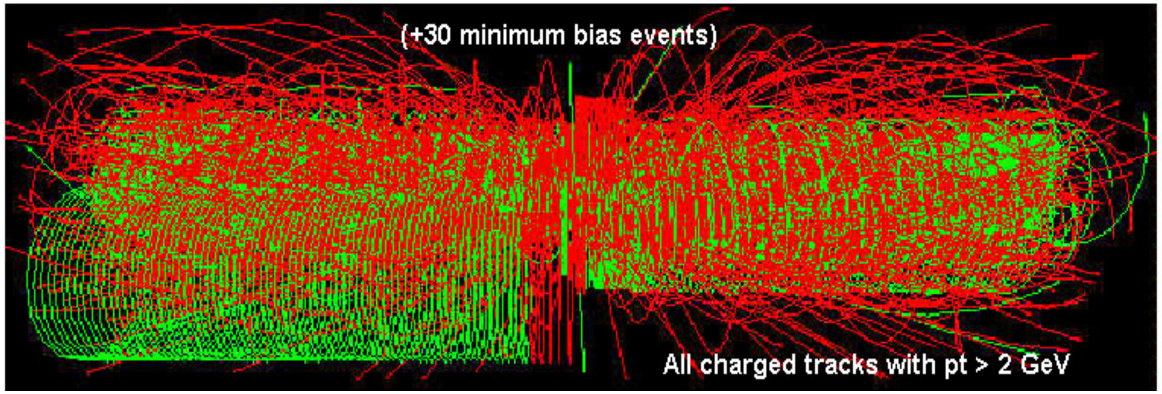
$$S = \mathrm{i} \int \mathrm{d}^4 x \, \mathcal{L}(x)$$

The goal is to understand in the most general; that's usually also the simplest.

- A. Eddington

ATLAS Data Challenge

- 800,000,000 proton-proton interactions per second
- 0.0002 Higgs per second
- ~150,000,000 electronic channels
- >10 PBytes of data per year

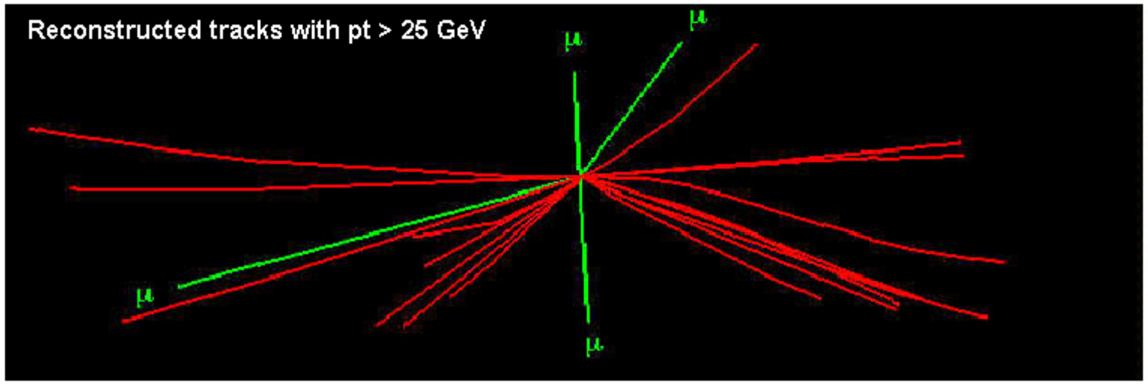


Selectivity: 1 in 10^{13}

Like looking for 1 person in a thousand world populations

Or for a needle in 20 million haystacks!

We are looking for this "signature"



1 in 10,000,000,000,000

Like looking for a single drop of water from the Jet d'Eau over 30 minutes

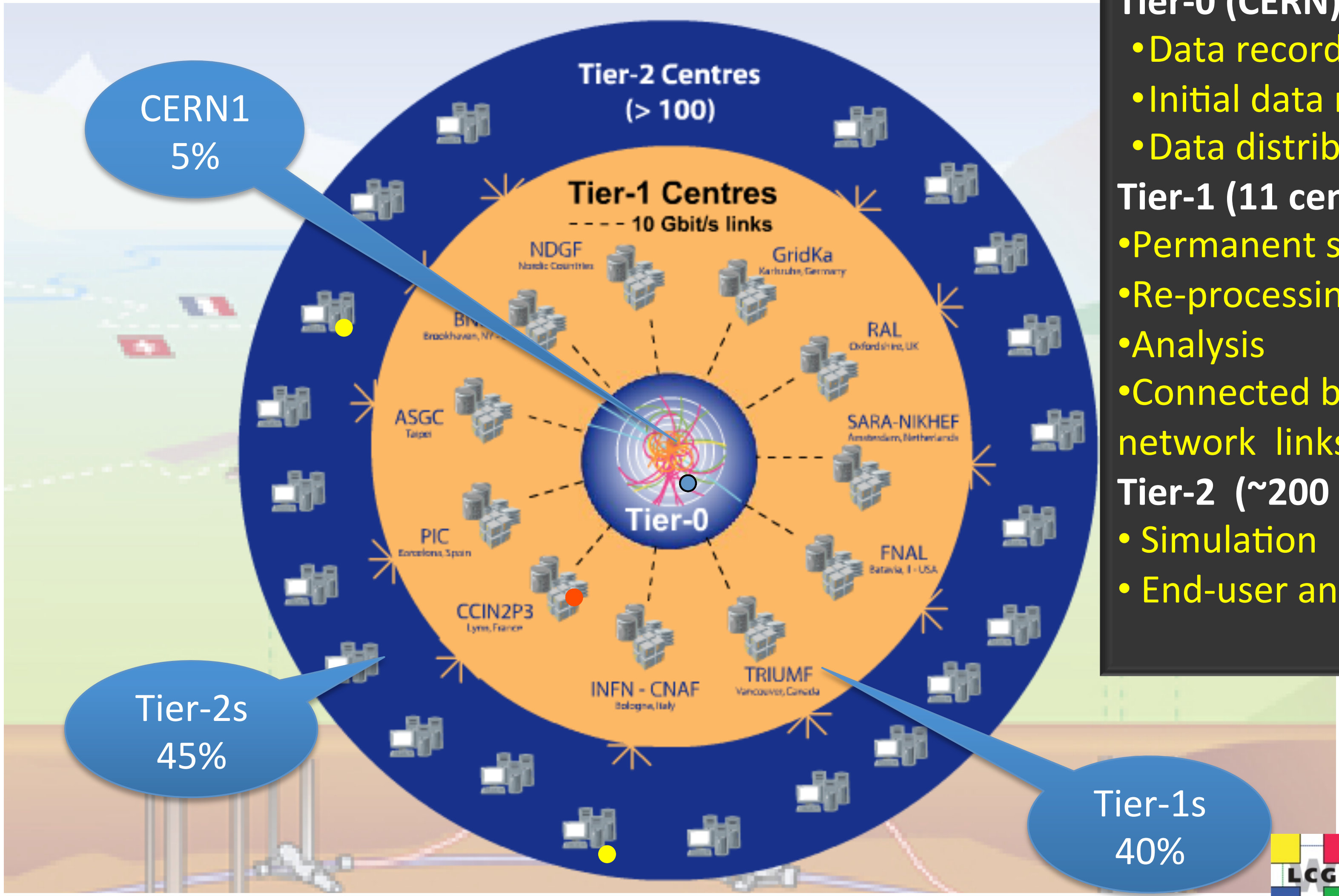
ATLAS Computing Challenges

- A lot of data in a highly distributed environment.
  - Petabytes of data to be treated and analyzed
  - ATLAS Detector generates about 1PB of raw data per second – most filtered out in real time by the trigger system
  - Interesting events are recorded for further reconstruction and analysis
  - As of 2013 ATLAS manages **~140 PB** of data, distributed world-wide to O(100) computing centers and analyzed by O(1000) physicists
  - Expected rate of data influx into ATLAS Grid ~40 PB of data per year in 2014
- Very large international collaboration
  - O(140) Institutes and Universities in ~40 countries
  - Thousands of physicists analyze the data

ATLAS uses grid computing paradigm to organize distributed resources

A few years ago ATLAS started Cloud Computing RnD project to explore virtualization and clouds

- Experience with different cloud platforms : Commercial (EC2, GCE), Academic, National

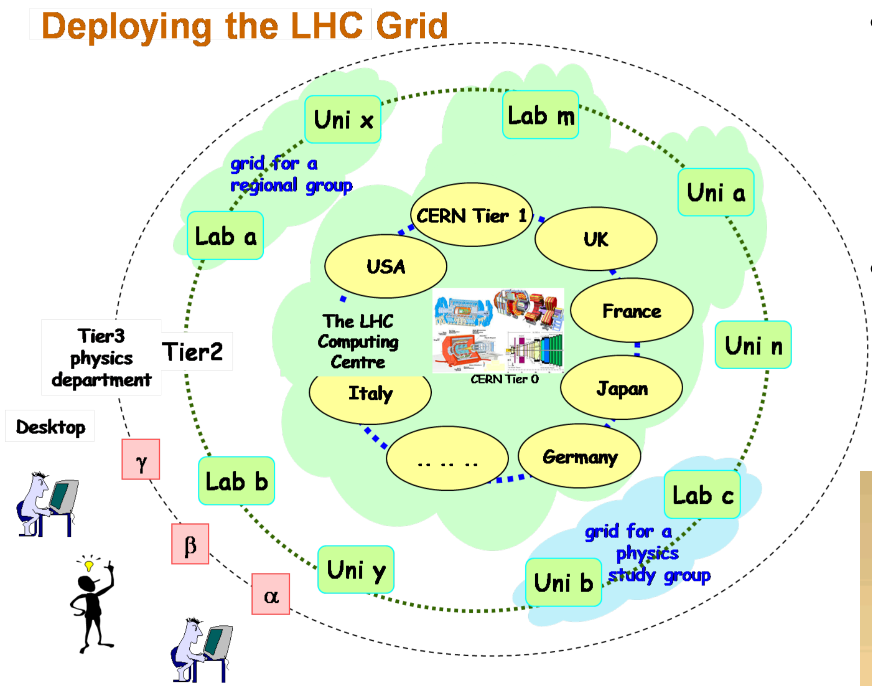


Tier-0 (CERN): (15%)
- Data recording
- Initial data reconstruction
- Data distribution

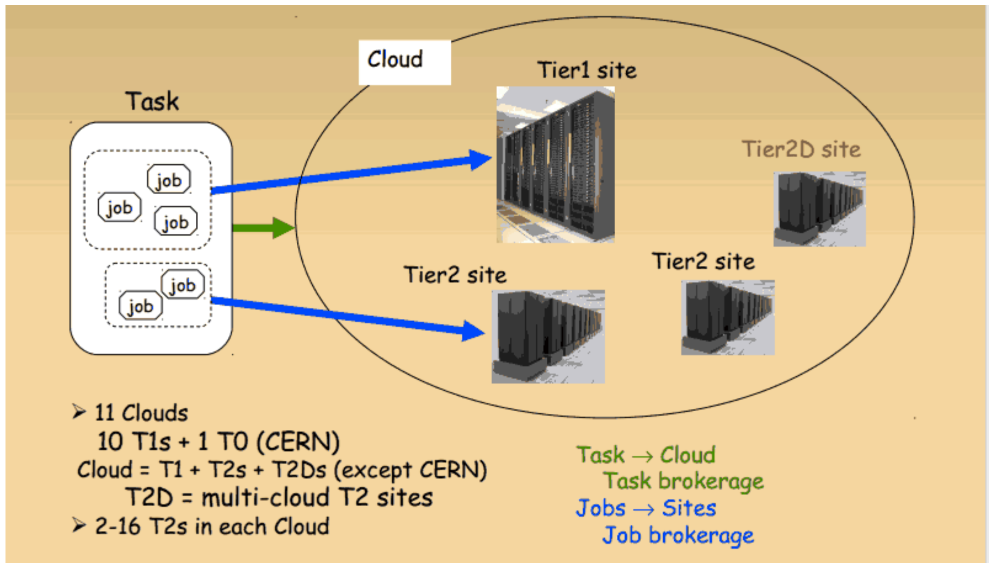
Tier-1 (11 centres): (40%)
- Permanent storage
- Re-processing
- Analysis
- Connected by direct 10 Gb/s network links

Tier-2 (~200 centres): (45%)
- Simulation
- End-user analysis

ATLAS Computing Model



- WLCG Computing Facilities in ATLAS are organized into Tier's
  - CERN (Tier0) – source of primary data
  - 10 Tier-1s hierarchically supports 5-18 Tier-2s
  - Tier-1 and associated Tier-2s are formed a clod
- PanDA is deployed at all ATLAS Grid centers



PanDA in ATLAS

- ATLAS computational resources are managed by PanDA workload management system
- PanDA project was started in Fall of 2005 by BNL and UTA groups
  - Production and Data Analysis system
  - An automated yet flexible workload management system (WMS) which can optimally make distributed resources accessible to all users
    - Originally developed in US for US physicists
    - Adopted as the ATLAS wide WMS in 2008 (first LHC data in 2009) for all computing applications
- Now successfully manages $O(10^2)$ sites, $O(10^5)$ cores, $O(10^8)$ jobs per year, $O(10^3)$ users
- Through PanDA, physicists see a single computing facility to run all data processing

PanDA Philosophy

- PanDA WMS design goals
  - Deliver transparency of data processing in a distributed computing environment
  - Achieve high level of automation to reduce operational effort
  - Flexibility in adapting to evolving hardware, computing technologies and network configurations
  - Scalable to the experiment requirements
  - Support diverse and changing middleware
  - Insulate user from hardware, middleware, and all other complexities of the underlying system
  - Unified system for central Monte-Carlo production and user data analysis
    - Support custom workflow of individual physicists
  - Incremental and adaptive software development

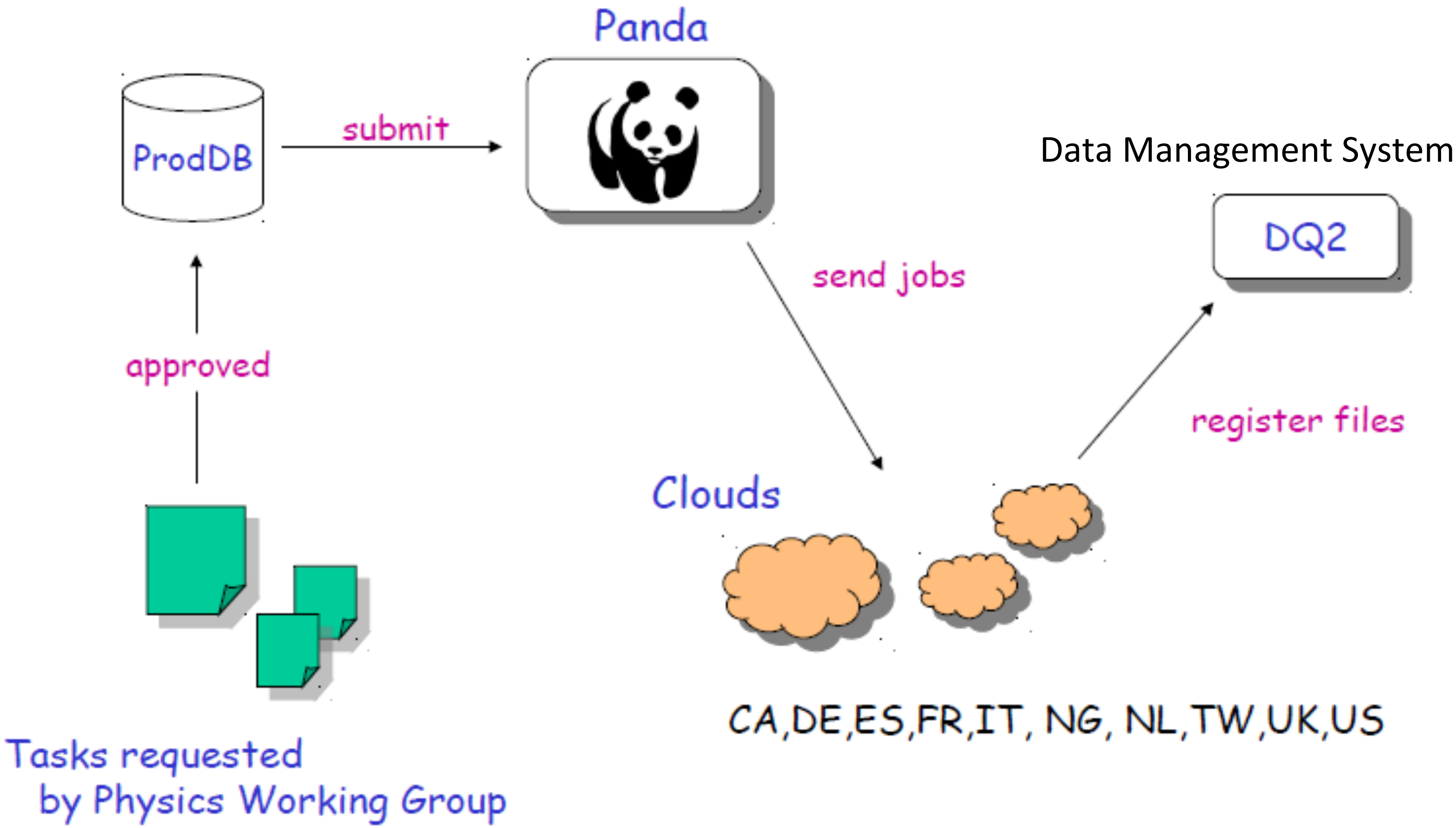
Key Features of PanDA

- Pilot based job execution system
  - HTCondor based pilot factory
  - Payload is sent only after execution begins on Computing Element
  - Minimize latency, reduce error rates
- Central job queue
  - Unified treatment of distributed resources
  - SQL DB keeps state - critical component
- Automatic error handling and recovery
- Extensive monitoring
- Modular design
- HTTP/S RESTful communications
- GSI authentication
- Workflow is maximally asynchronous
- Use of Open Source components

PanDA Components

- Server
  - Implemented in python and run under Apache as a web service
- Database back-end
  - System-wide job database (RDBMS) to keep static and dynamic information on all jobs in the system
- Pilot System
  - Job wrapper
  - Pilot factory
    - An independent subsystem manages to deliver of pilot jobs to worker nodes. A pilot once launched on a worker node contacts the dispatcher and receives an available job appropriate to the site
- Brokerage
  - Module to prioritize and assign work on the basis of job type, priority, SW and input data availability and its locality
- Dispatcher
  - A component of the PanDA server which receives requests from pilots and dispatches job payloads
- Information system
  - a system-wide site/queue information database recording static and dynamic information used throughout PanDA to configure and control system behavior
- Monitoring systems
- PD2P – dynamic data caching

PanDA Workflow



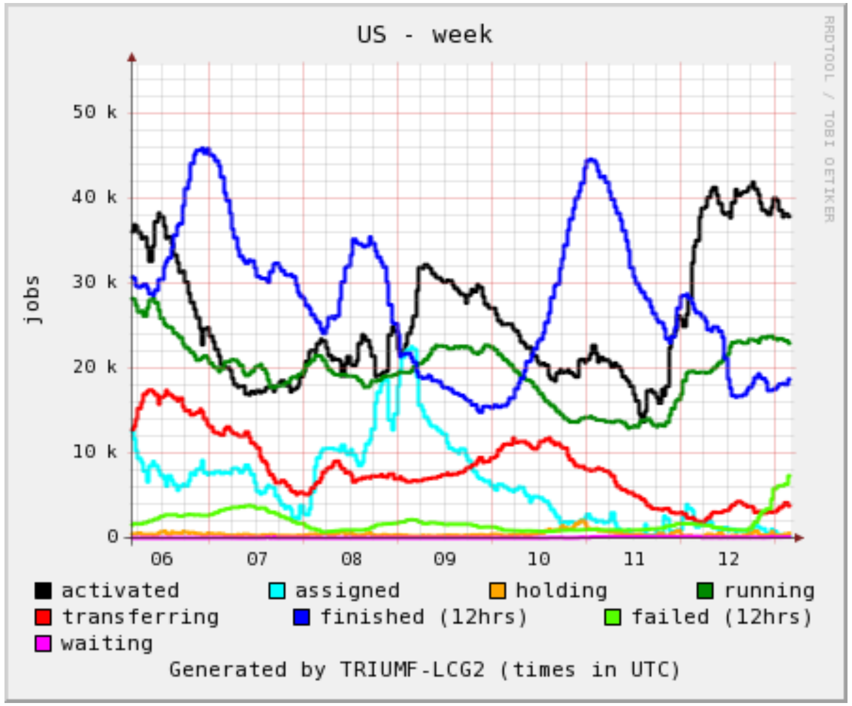
What is Job ?

- Basic unit of work is a job :
  - Executed on a CPU resource/slot
  - May have inputs
  - Produces output(s)
- Two major types of jobs
  - Production
    - Data processing, Monte-Carlo simulation, Physics Groups production
      - Organized and predicted activities
  - User analysis

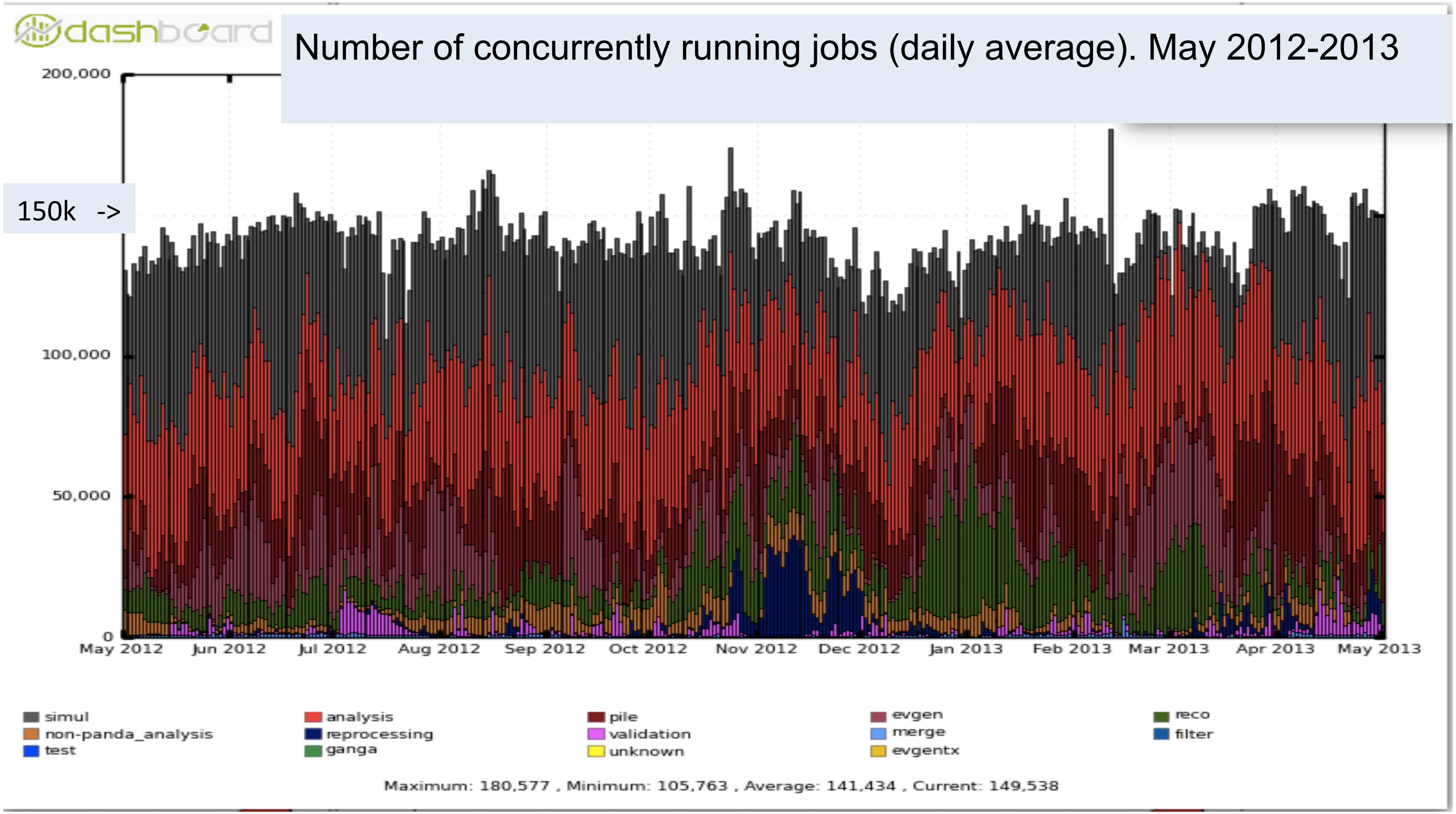
Current scale – million jobs per day

Job States

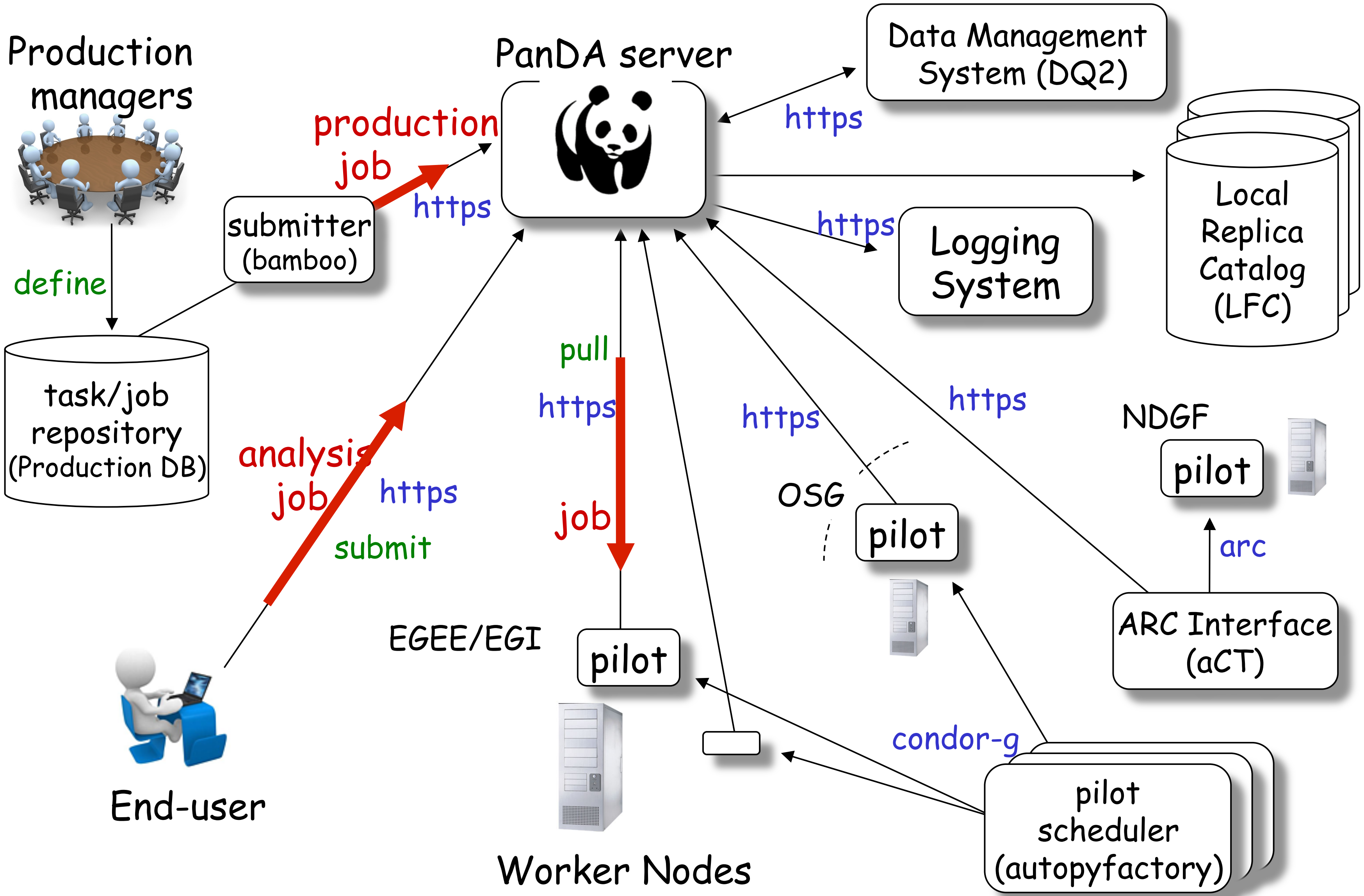
- Panda jobs go through a succession of steps tracked in DB
  - Defined
  - Assigned
  - Activated
  - Running
  - Holding
  - Transferring
  - Finished/failed



PanDA ATLAS Running Jobs



Workload Management



Data Management

- PanDA supports multiple DDM solutions
  - ATLAS Distributed Data Management (DDM) System
  - Pandamover file transfer (using chained Panda jobs)
  - CMS PHEDEX file transfer
  - Federated Xrootd
  - Direct access if requested (by task or site)
  - Customizable lsm (local site mover)
  - Multiple default site movers are available

PanDA's Success

- The system was developed by US ATLAS for US ATLAS
- Adopted by ATLAS Worldwide as Production and Analysis system
- PanDA was able to cope with increasing LHC luminosity and ATLAS data taking rate
- Adopted to evolution in ATLAS computing model
- Two leading HEP and astro-particle experiments (CMS and AMS) has chosen PanDA as workload management system for data processing and analysis.
- PanDA was chosen as a core component of Common Analysis Framework by WLCG

Evolving PanDA for advanced scientific computing

- DoE ASCR and HEP funded project “Next Generation Workload Management and Analysis System for Big Data” - BigPanDA. Started in September 2012.
  - Generalization of PanDA as meta application, providing location transparency of processing and data management, for HEP and other data-intensive sciences, and a wider exascale community.

- There are three dimensions to evolution of PanDA
  - Making PanDA available beyond ATLAS and HEP
  - Extending beyond Grid (Leadership Computing Facilities, Clouds, University clusters)
  - Integration of network as a resource in workload management

PanDA for Leadership Computing Facilities

- Adding new class of resources supported by PanDA
  - HEP and LCF
    - CRAY XMP in the past (CERN)
    - HEP code on x86 CPUs
    - Trivial parallelism uses too much memory
    - Data-intensive as always
    - Need much more computing
  - Present trends in computing technologies to which HEP must respond
    - Many-core processing

PanDA for Leadership Computing Facilities

- Expanding PanDA from Grid to Leadership Class Facilities (LCF) will require significant changes in pilot system
- Each LCF is unique
  - Unique architecture and hardware
  - Specialized OS, “weak” worker nodes, limited memory per WN
  - Code cross-compilation is typically required
  - Unique job submission systems
  - Unique security environment
- Pilot submission to a worker node is typically not feasible
- Pilot/agent per supercomputer or queue model
- Tests on BlueGene at BNL and ANL. Geant4 port to BG/P
- PanDA/Geant4 project at Oak-Ridge National Laboratory LCF (ORNL LCF)

PanDA project on ORNL LCF

- Get experience with all relevant aspects of the platform and workload
  - job submission mechanism
  - job output handling
  - local storage system details
  - outside transfers details
  - security environment
  - adjust monitoring model
- Develop appropriate pilot/agent model for Titan
- Geant4 and Project X at OLCF proposal will be initial use case on Titan
  - Collaboration between ANL, BNL, ORNL, SLAC, UTA, UTK
  - Cross-disciplinary project - HEP, NP, HPC

Cloud Computing and PanDA

- ATLAS Distributed Computing set up a few years ago cloud computing project to exploit virtualization and clouds in PanDA
    - Utilize private and public clouds as extra computing resource
    - Mechanism to cope with peak loads on the Grid
- Experience with variety of cloud platforms
    - Amazon EC2
    - Helix Nebula for MC production (CloudSigma, T-Systems and ATOS – all used)
    - Futuregrid (U Chicago), Synnefo cloud (U Vic)
    - RackSpace
    - Private clouds OpenStack, CloudStack, etc…
    - Recent project on Google Compute Engine (GCE)

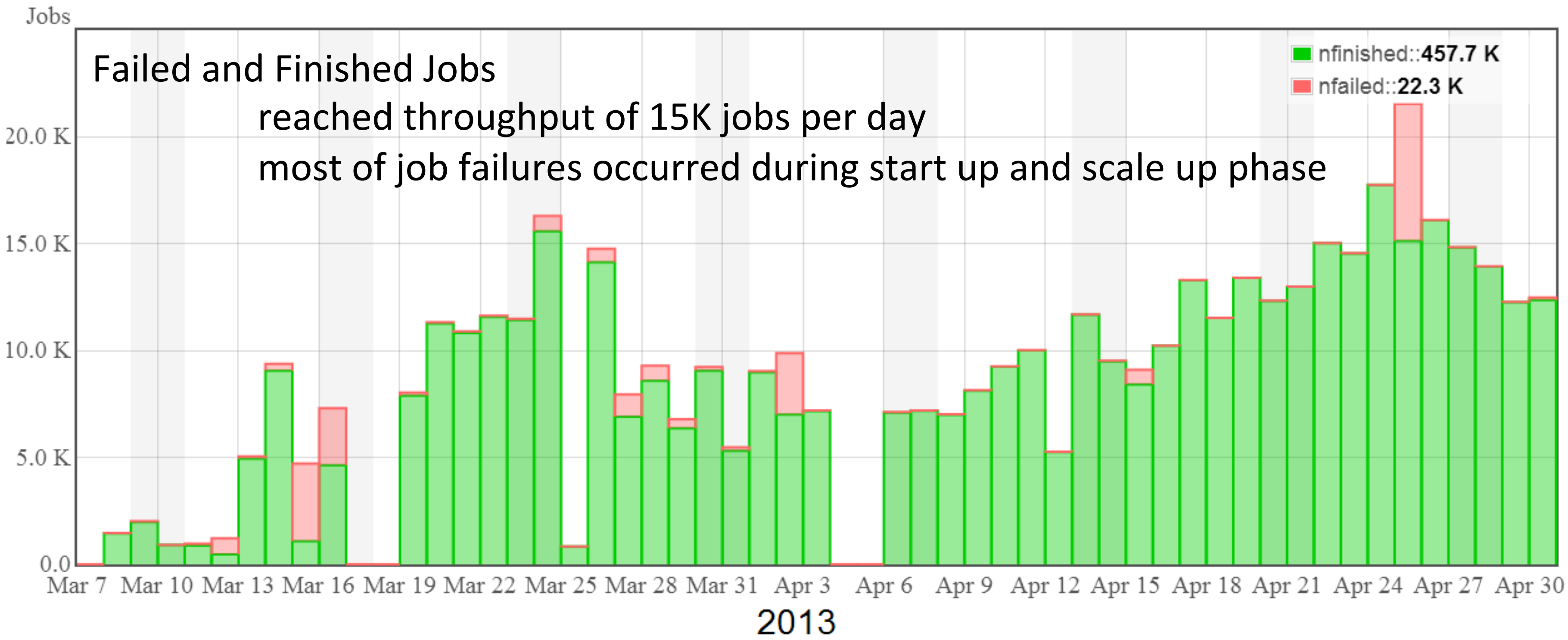
Running PanDA on Google Compute Engine

- Google Compute Engine (GCE) preview project

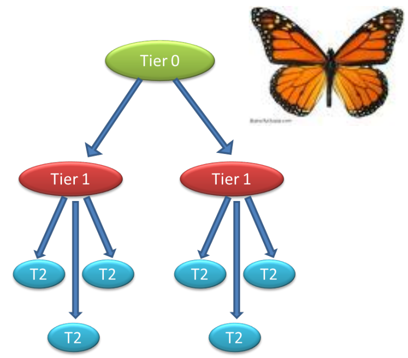
- Google agreed to allocate additional resources for ATLAS for free
  - ~5M cpu hours, 4000 cores for about 2 month, (original preview allocation 1k cores)
- These are powerful machines with modern CPUs
- Resources are organized as HTCondor based PanDA queue
  - Centos 6 based custom built images, with SL5 compatibility libraries to run ATLAS software
  - Condor head node, proxies are at BNL
  - Output exported to BNL SE
- Work on capturing the GCE setup in Puppet
- Transparent inclusion of cloud resources into ATLAS Grid
- The idea was to test long term stability while running a cloud cluster similar in size to Tier 2 site in ATLAS
- Intended for CPU intensive Monte-Carlo simulation workloads
- Planned as a production type of run. Delivered to ATLAS as a resource and not as an R&D platform.

Running PanDA on Google Compute Engine

- We ran for about 8 weeks (2 weeks were planned for scaling up)
- Very stable running on the Cloud side. GCE was rock solid.
- Most problems that we had were on the ATLAS side.
- We ran computationally intensive jobs
  - Physics event generators, Fast detector simulation,, Full detector simulation
- Completed 458,000 jobs, generated and processed about 214 M events

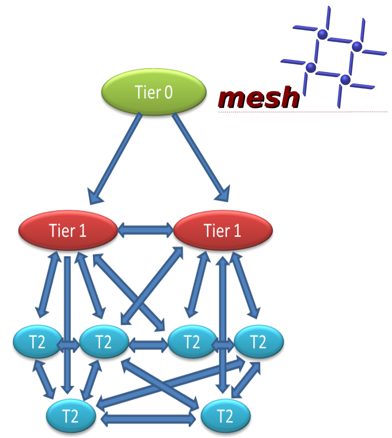


Adding Network Awareness to PanDA



- LHC Computing model for a decade was based on MONARC model
  - Assumes poor networking
    - Connections are seen as not sufficient or reliable
      - Data needs to be preplaced. Data comes from specific places
  - Hierarchy of functionality and capability
    - Grid sites organization in "clouds" in ATLAS
    - Sites have specific functions
    - Nothing can happen utilizing remote resources on the time of running job
  - Canonical HEP strategy : "Jobs go to data"
    - Data are partitioned between sites
      - Some sites are more important (get more important data) than others
      - Planned replicas
        - » A dataset (collection of files produced under the same conditions and the same SW) is a unit of replication
    - Data and replica catalogs are needed to broker jobs
    - Analysis job requires data from several sites triggers data replication and consolidation at one site or job splitting on several jobs running on all sites
      - A data analysis job must wait for all its data to be present at the site
        - » The situation can easily degrade into a complex n-to-m matching problem

There was no need to consider network as a resource in WMS in static data distribution scenario



- New networking capabilities and initiatives in the last 2 years (like LHCONE)
  - Extensive standardized monitoring from network performance monitoring (perfSONAR)
  - Traffic engineering capabilities
    - Rerouting of high impact flows onto separate infrastructure
  - Intelligent networking
    - Virtual Network On Demand
    - Dynamic circuits

...and dramatic changes in computing models
- From strict hierarchy of connections becomes more of a mesh
- Data access over wide area
- "no division" in functionality between sites

We would like to benefit from new networking capabilities and to integrate networking services with PanDA. We start to consider network as a resource on similar way as for CPUs and data storage

Intelligent Network Services and PanDA

- Quick re-run a prior workflow (bug found in reconstruction algo)
  - Site A has enough job slots but no input data
  - Input data are distributed between sites B,C and D, but sites have a backlog of jobs
  - Jobs may be sent to site A and at the same time virtual circuits to connect sites B,C,D to site will be built. VNOD will make sure that such virtual circuits have sufficient bandwidth reservation.
    - Or data can be accessed remotely (if connectivity between sites is reliable and this information is available from perfSONAR)
      - In canonical approach data should be replicated to site A
- HEP computing is often described as an example of parallel workflow. It is correct on the scale of worker node (WN) . WN doesn't communicate with other WN during job execution. But the large scale global workflow is highly interconnected, because each job typically doesn't produce an end result in itself. Often data produced by a job serve as input to a next job in the workflow. PanDA manages workflow extremely well (1M jobs/day in ATLAS). The new intelligent services will allow to dynamically create the needed data transport channels on demand.

Intelligent Network Awareness and PanDA

- In BigPanDA we will use information on how much bandwidth is available and can be reserved before data movement will be initiated
- In Task Definition user will specify data volume to be transferred and deadline by which task should be completed. The calculations of (i) how much bandwidth to reserve, (ii) when to reserve, and (iii) along what path to reserve will be carried out by VNOD.

BigPanDA work plan

- Factorizing the code
  - Factorizing the core components of PanDA to enable adoption by a wide range of exascale scientific communities
- Extending the scope
  - Evolving PanDA to support extreme scale computing clouds and Leadership Computing Facilities
- Leveraging intelligent networks
  - Integrating network services and real-time data access to the PanDA workflow
- 3 years plan
  - Year 1. Setting the collaboration, define algorithms and metrics
  - Year 2. Prototyping and implementation
  - Year 3. Production and operations

Conclusions

- ASCR gave us a great opportunity to evolve PanDA beyond ATLAS and HEP and to start BigPanDA project
- Project team was set up
- The work on extending PanDA to LCF has started
- Large scale PanDA deployments on commercial clouds are already producing valuable results
- Strong interest in the project from several experiments and scientific centers to have a joined project.

Acknowledgements

- Thanks to I.Bird, J.Boyd, K.De, I.Fisk, R.Heuer, T.Maeno, R.Mount, S.Panitkin, L.Robertson, A.Vaniachine, T.Wenaus and D.Yu whose slides were used in this presentation.
- Thanks to many colleagues from the ATLAS experiment at LHC