Exploiting Virtualization & Cloud Computing in ATLAS

Fernando Harald Barreiro Megino[α], Doug Benjamin[β], Kaushik De[γ], Ian Gable[δ], Val Hendrix[ε], Sergey Panitkin[ζ], Michael Paterson[δ], Asoka De Silva[η], Daniel van der Ster[α], Ryan Taylor[δ], Roberto A. Vitillo[ε], Rod Walker[θ] on behalf of the ATLAS Collaboration

α: CERN: European Organization for Nuclear Research, Geneva, Switzerland
β: Duke University Durham, NC 27708
γ: The University of Texas at Arlington, TX, USA
δ: University of Victoria, BC, Canada
ϵ: Lawrence Berkeley National Lab, 1 Cyclotron Road, Berkeley, CA, USA 94720
ζ: Brookhaven National Laboratory, Upton, NY, USA
η: TRIUMF, Vancouver, BC, Canada
θ: Ludwig-Maximilians-Universitaet Munich

CAPACITIES

e-infrastructure

ATLAS Cloud Computing R&D

EFFICIENCY, ELASTICITY

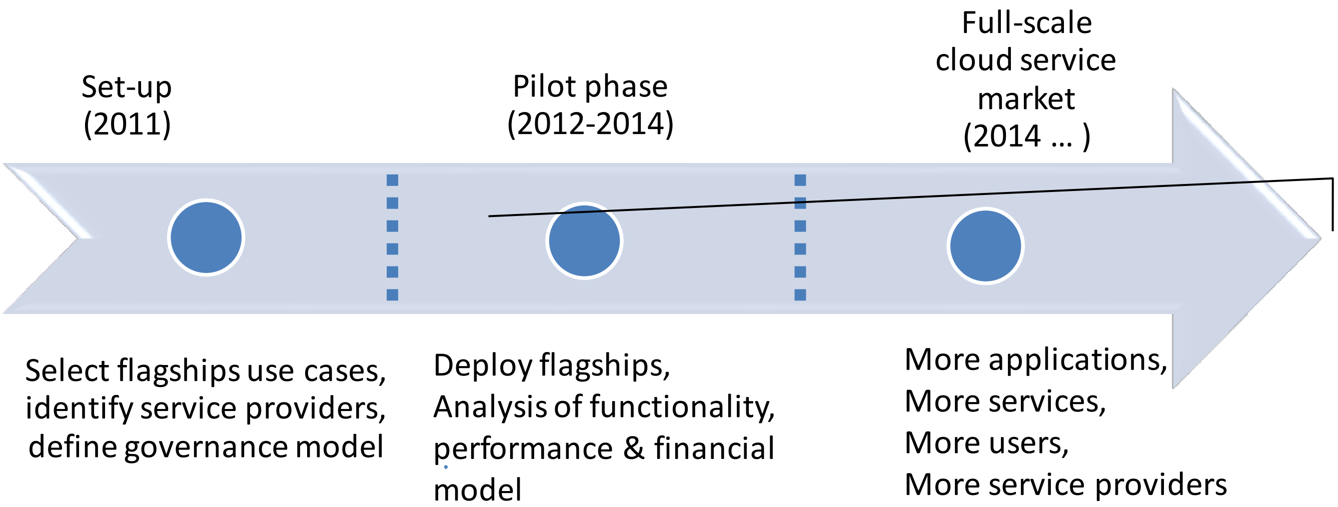
- ATLAS Cloud Computing R&D is a young initiative
  - Active participation, almost 10 persons working part time on various topics
  - Goal: How we can integrate cloud resources with our current grid resources?
- **Data processing and workload management**
  - PanDA queues in the cloud
    - Centrally managed, non-trivial deployment but scalable
    - Benefits ATLAS & sites, transparent to users
  - Tier3 analysis clusters: instant cloud sites
    - Institute managed, low/medium complexity
  - Personal analysis queue: one click, run my jobs
    - User managed, low complexity (almost transparent)
- **Data storage**
  - Short term data caching to accelerate above data processing use cases
    - Transient data
  - Object storage and archival in the cloud
    - Integrate with DDM

Data processing and workload management

- PanDA queues in the cloud
- Analysis Clusters in the Cloud
- Personal PanDA Analysis Queues in the Cloud

Helix Nebula: The Science Cloud

- European Cloud Computing Initiative: CERN, EMBL, ESA + European IT industry
  - Evaluate cloud computing for science and build a sustainable European cloud computing infrastructure
  - Identify and adopt policies for trust, security and privacy
- CERN/ATLAS is one of three flagship users to test a few commercial cloud providers (CloudSigma, T-Systems, ATOS...)
- Agreed to run MC Production jobs ~3 weeks per provider



First step completed: Ran ATLAS simulation jobs (i.e. small I/O requirements) on CloudSigma

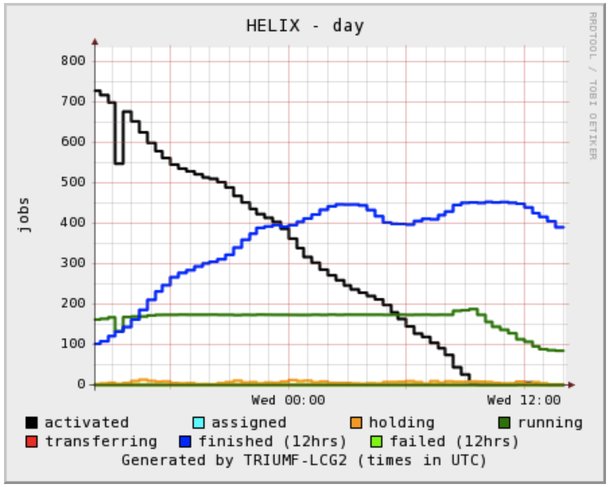
Basic idea: KISS principle

- The most simple model possible
- Use CernVM with preinstalled SW
- Configure VMs at Cloud Sigma which join a Condor pool with master at CERN (one of the pilot factories)
- Create a new PanDA queue *HELIX*
  - Real MC production tasks are assigned manually
- I/O copied over the WAN from CERN (lcg-cp/lcg-cr for input/output)

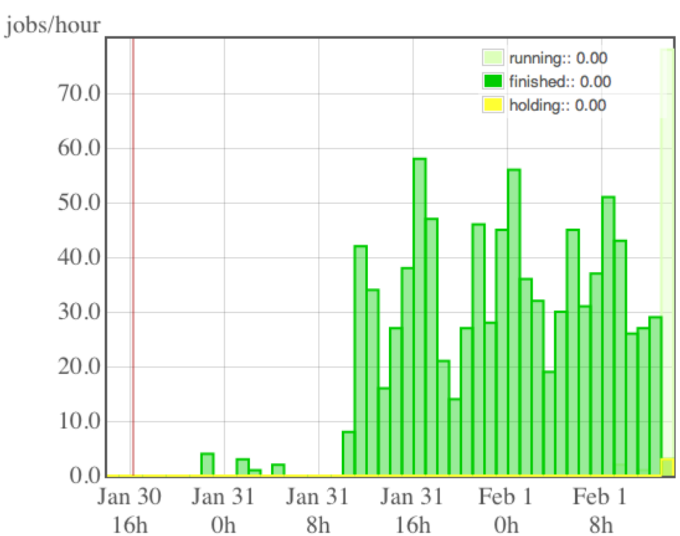
Results

- 100 nodes/200 CPUs at Cloud Sigma used for production tasks
- Smooth running with very few failures
- Finished 6 x 1000-job MC tasks over ~2 weeks
- We ran 1 identical task at CERN to get reference numbers

	HELIX	CERN
Success Rates	265 failed, 6000 succeeded	36 failed, 1000 succeeded
Mean Running Times	$16267s \pm 7038s$	$8136.6s \pm 765.5s$



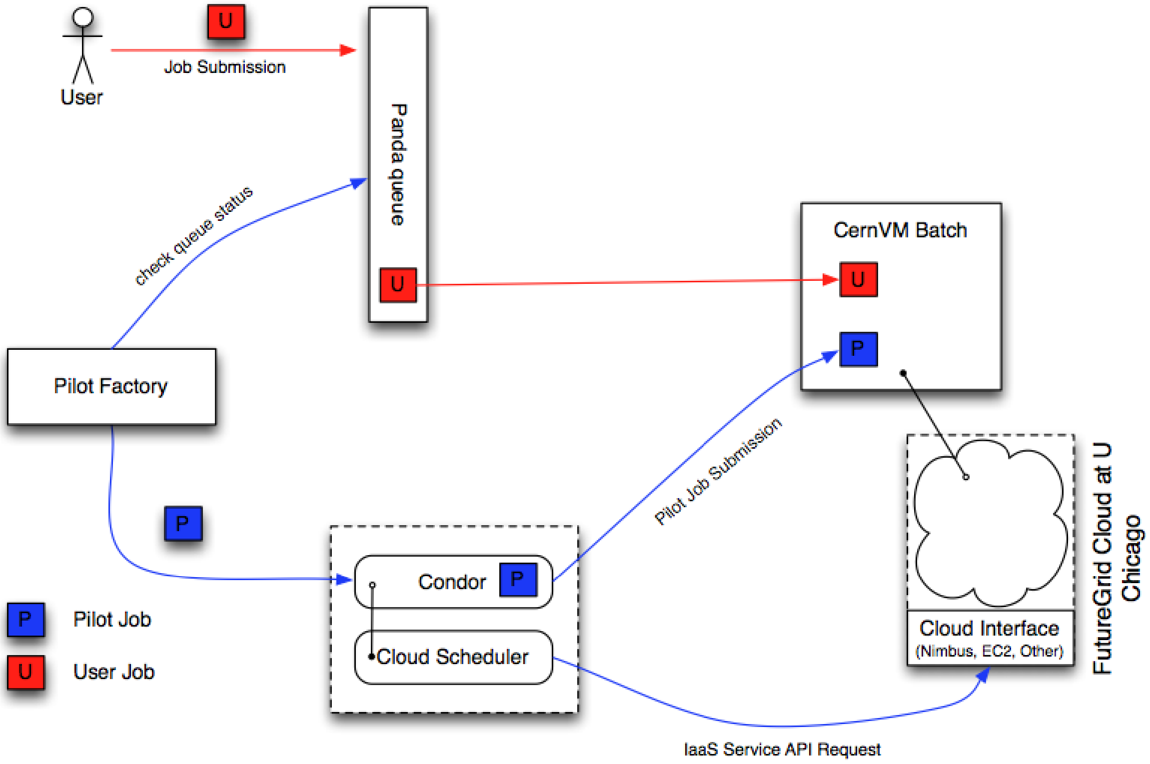
The jobs/hour for all sites progress



- Wall clock performance cannot be compared directly, since we don't have the same hardware on both sites
  - CloudSigma has ~1.5Ghz of AMD Opteron 6174 per jobslot, CERN has a ~2.3GHz Xeon L5640
- Best comparison would be CHF/event, which is presently unknown

Cloud Scheduler

- Allow users to run jobs on IaaS resources by submitting to standard Condor queue
- Simple python software package
  - Monitor state of a Condor queue
  - Boot VMs in response to waiting jobs
  - Custom Condor job description attributes to identify VM requirements
- Control multiple IaaS cloud sites through cloud APIs
- Same approach used successfully for BaBar jobs



Cloud Scheduler: Results

- Cloud resources are aggregated and served to one analysis and one production queue
- **Analysis queue:** HammerCloud benchmarks
  - WNs in FutureGrid (Chicago), SE in Univ. Victoria
  - Evaluate performance and feasibility of analysis jobs in the cloud
  - I/O intensive: Remote data access from Victoria - depends on WAN bandwidth capacity between sites
  - Data access tests through WebDAV and GridFTP
  - Initial results: **692/702** successful jobs
- **Production queue:** in continuous operation for simulation jobs
  - Low I/O requirements
  - Several thousand jobs have been executed on over 100 VMs running at FutureGrid (Chicago) and Synnefo cloud (Uvic)
  - Plan to scale up number of running jobs as cloud resources become available

Data processing and workload management

- PanDA queues in the cloud
- Analysis Clusters in the Cloud
- Personal PanDA Analysis Queues in the Cloud

Analysis Clusters in the Cloud

- A site should be able to easily deploy new analysis clusters in a commercial or private cloud resource
  - easy and user-transparent way to scale out jobs in the cloud is needed
  - scientists may spend time analyzing data and not doing system administration
- Goal: migrate functionality of a physical Data Analysis Cluster (DAC) into the cloud
- Support in the cloud:
  - Services that support job execution and submission: Condor
  - User management: LDAP
  - ATLAS software distribution: CVMFS
  - Web caching: squid
- Evaluate cloud management tools that will enable us to define these DACs in all their complexity: **CloudCRV, Starcluster, Scalr**
  - Scalr found to have most robust feature set and active community

Data processing and workload management

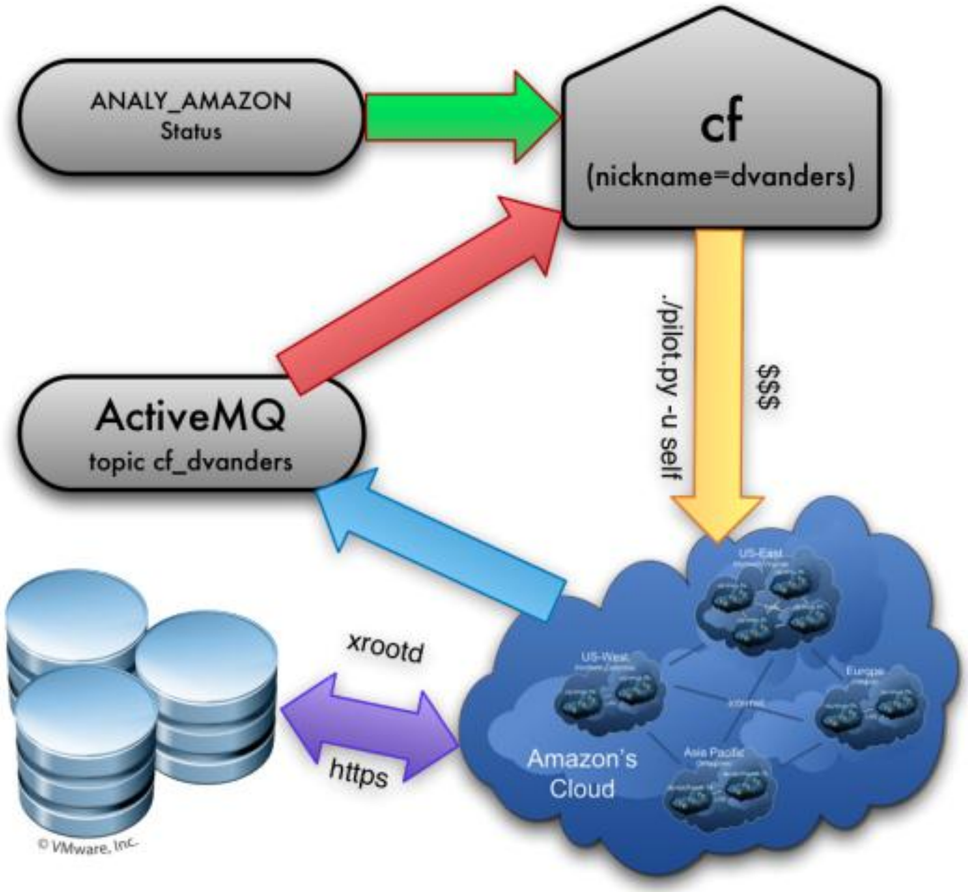
- PanDA queues in the cloud
- Analysis Clusters in the Cloud
- Personal PanDA Analysis Queues in the Cloud

Personal PanDA Analysis Queues in the Cloud

- Enable **users** to access extra computing resources on-demand
  - Private and commercial cloud providers
- Access new cloud resources with minimal changes to their analysis workflow on grid sites
  - Submit PanDA jobs to a virtual cloud site
  - Virtual cloud site has no pilot factory associated: user has to run PanDA pilots to retrieve their jobs
  - **Personal PanDA pilot** retrieves only user's jobs and ignores others

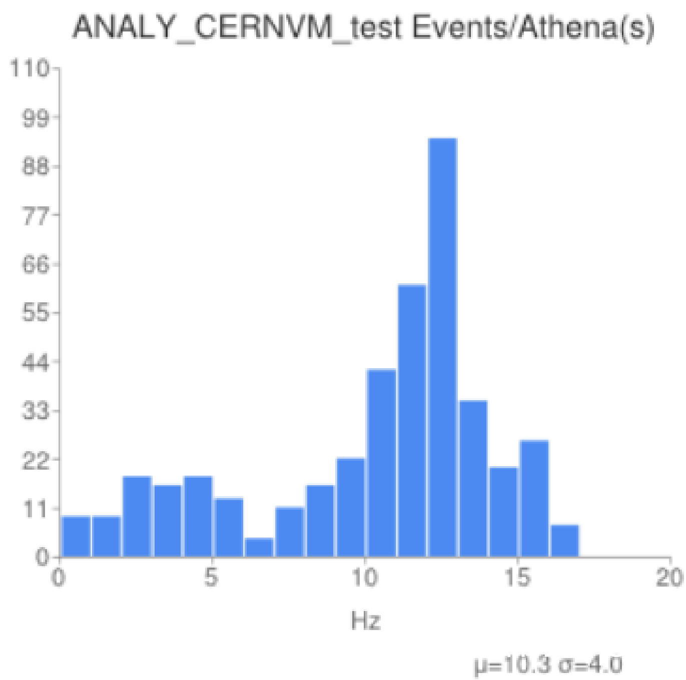
Cloud Factory

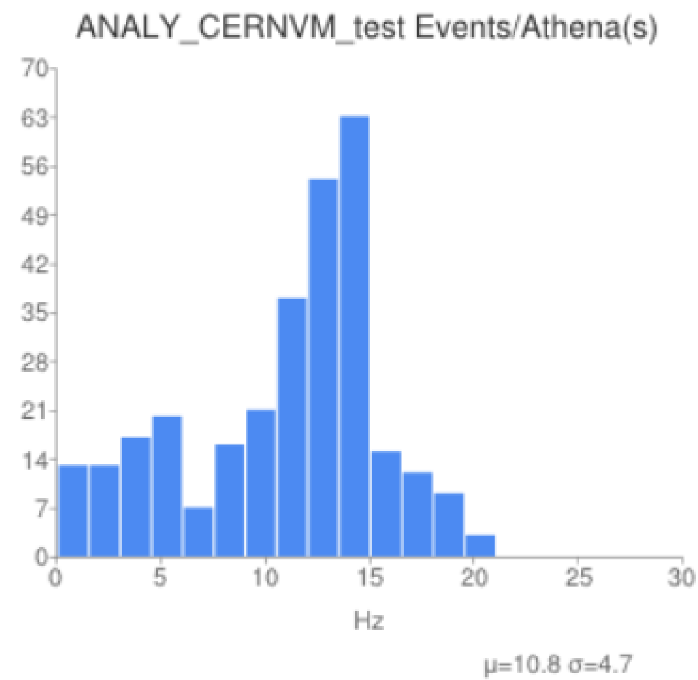
- Need a simple tool to automatize the above workflow
  - Sense queued jobs in the personal queue
  - Instantiate sufficient cloud VMs via cloud API
  - Run the personal pilots
  - Monitor VMs (*alive/zombie, busy/idle, X509 proxy status*)
  - Destroy VMs after they are no longer needed (zombies, expired proxy, unneeded)



Cloud Factory Testing at LxCloud

- LxCloud: OpenNebula testing instance with EC2 API at CERN
- CVMFS for ATLAS releases
- CERN storage element as data source
- HammerCloud stress-tests ran to compare job performance in LxCloud and CERN batch facility LSF
  - No difference in reliability
  - Small decrease in average performance: **10.8Hz in LSF** to **10.3Hz in LxCloud**

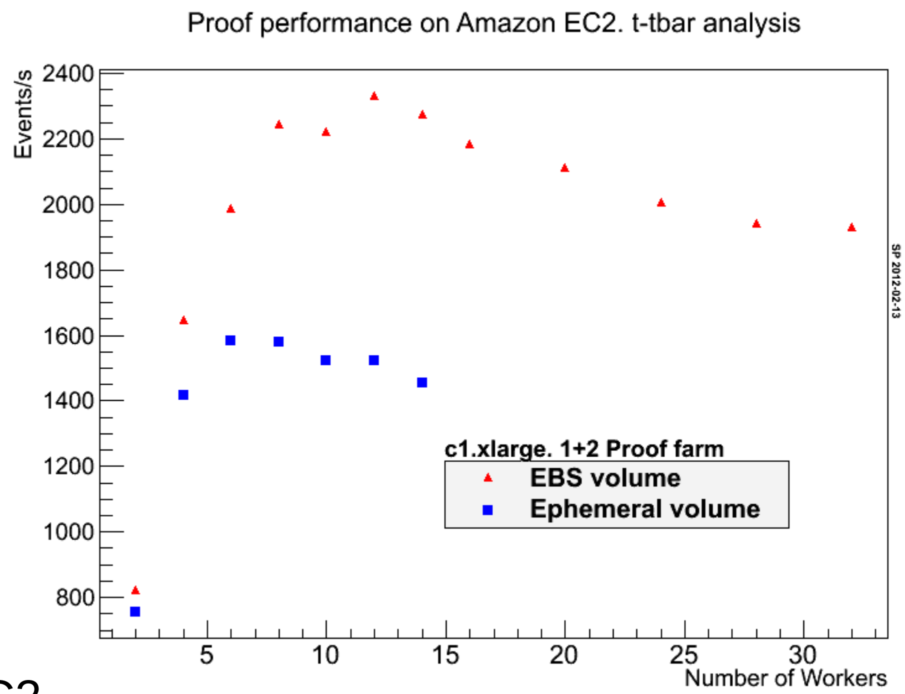




Storage and Data Management

Data Access Tests

- Evaluate the different storage abstraction implementations that cloud platforms provide
- Amazon EC2 provides at least three storage options
  - Simple Storage Service (S3)
  - Elastic Block Store (EBS)
  - Ephemeral store associated with a VM
  - Different cost-performance benefits for each layout that need to be analyzed
- Cloud storage performance on 3-node PROOF farm
  - EBS volume performs better than ephemeral disk
  - But ephemeral disk comes free with EC2 instances
    - Scaling of storage space and performance with the size of the analysis farm



Xrootd Storage Cluster in the Cloud

- Evaluate write performance of cluster where data comes from outside the cloud
- 2 data servers and 1 redirector node using Amazon EC2 large instances and either
  - Amazon ephemeral storage
  - Amazon EBS
- Transfers using Xrootd native copy program set to fetch files from multiple sources
- Most useable configuration: 2 ephemeral storage partitions joined together by Linux LVM
  - Average transfer rate of 16 MB/s to one data server
  - Around 45 MB/s for the three
  - Startup time of the VM increased due to assembling, formatting and configuring the storage
- Using 1 or 2 EBS partitions
  - Average transfer rate under 12 MB/s
  - Large number of very slow transfers (< 1 MB/s)
  - Very high storage costs given current Amazon rates

Future Evaluation of Object Stores

- Integration of Amazon S3 with ATLAS DDM
- DDM team will demonstrate compatibility of their future system with S3-API implementations of various storage providers
  - Huawei, OpenStack Swift and Amazon
- Main questions to be answered:
  - how to store, retrieve and delete data from an S3 storage
  - how to combine data organization models
    - S3 bucket/object model
    - ATLAS dataset/file
  - how to integrate a cloud storage with existing grid middleware
  - how to integrate authentication and authorization mechanisms

Conclusions

Conclusions and cloud futures

- Data processing
  - Many activities are reaching a point where we can start getting feedback from users. In the next months we should
    - Determine what we can deliver in production
    - Start focusing and eliminate options
  - Improve automation and monitoring
  - Still suffering from lack of standardization amongst providers
- Cloud storage
  - This is the hard part
  - Looking forward to good progress in caching (XRootd in cloud)
  - Some “free” S3 endpoints are just coming online, so effective R&D is only starting now
    - ATLAS DDM S3 evaluation and integration proposal written recently
- Support grid sites who want to offer private cloud resources
  - Develop guidelines, best practices
  - Good examples already, e.g. LxCloud, PIC, BNL, and others

Thank you for your attention