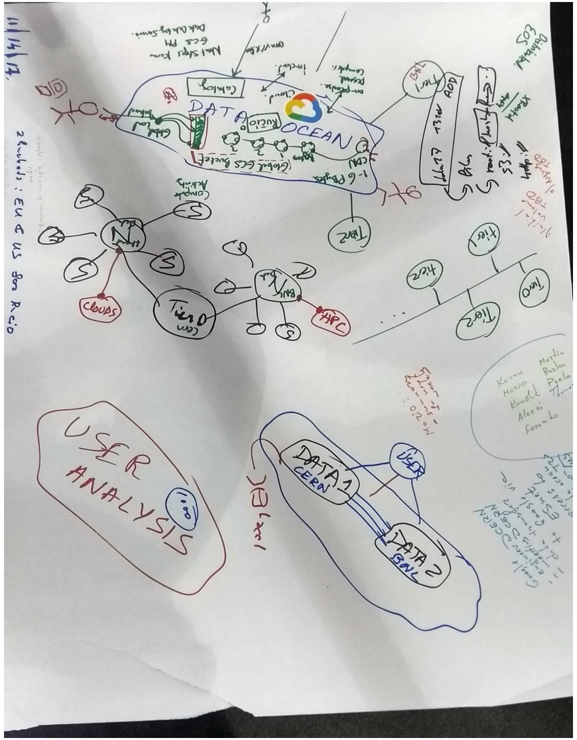
The *Data Ocean* Project

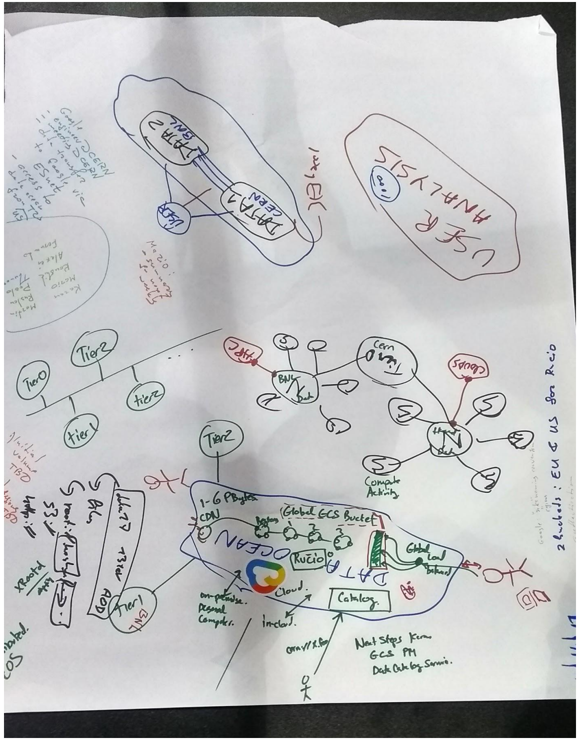
An ATLAS and Google R&D Collaboration

Mario.Lassnig@cern.ch
on behalf of the ATLAS Collaboration

Contributors: Karan Bhatia, Andy Murphy, Wyatt Gorman, Alexei Klimentov, Kaushik De, Fernando Barreiro, Johannes Elmsheuser, Sergey Panitkin, Tobias Wegner, Martin Barisits, Thomas Beermann, Ruslan Mashinistov, Peter Love, Arnaud Dubreuil, Tadashi Maeno, Paul Nilsson

In the beginning …





WHERE DATA AND DISCOVERY CONVERGE.
Google Cloud
THE UNIVERSITY OF NEW MEXICO
HGST

SC17
Denver, CO
hpc connects.

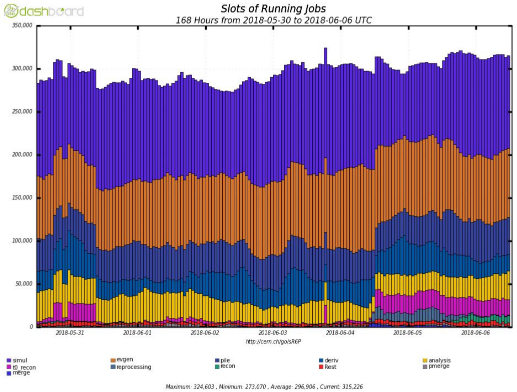
Motivation and objective

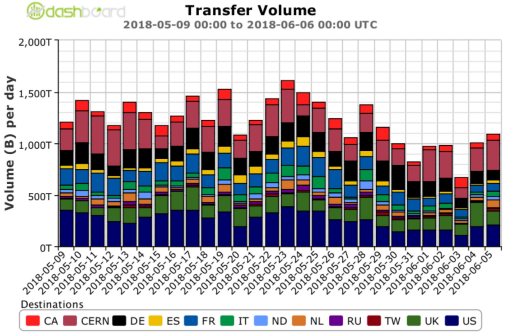
- ATLAS is facing several challenges for LHC Run-3 (2020-2023) and HL-LHC runs (2025-2034)
  - These challenges are not specific for ATLAS but common for the HENP computing community
  - Storage continues to be the driving cost factor
  - At the current growth rate we cannot absorb the increased physics output of the experiment
  - Novel computing models with more dynamic use of storage and computing need to be considered

- The *Data Ocean* project is an R&D project for evaluating and adopting novel IT technologies
  - Allow ATLAS to explore the use of different computing models to prepare for High-Luminosity LHC
  - Allow ATLAS user analysis to benefit from the Google infrastructure
  - Give Google real science use cases to improve their cloud platform

The first use cases

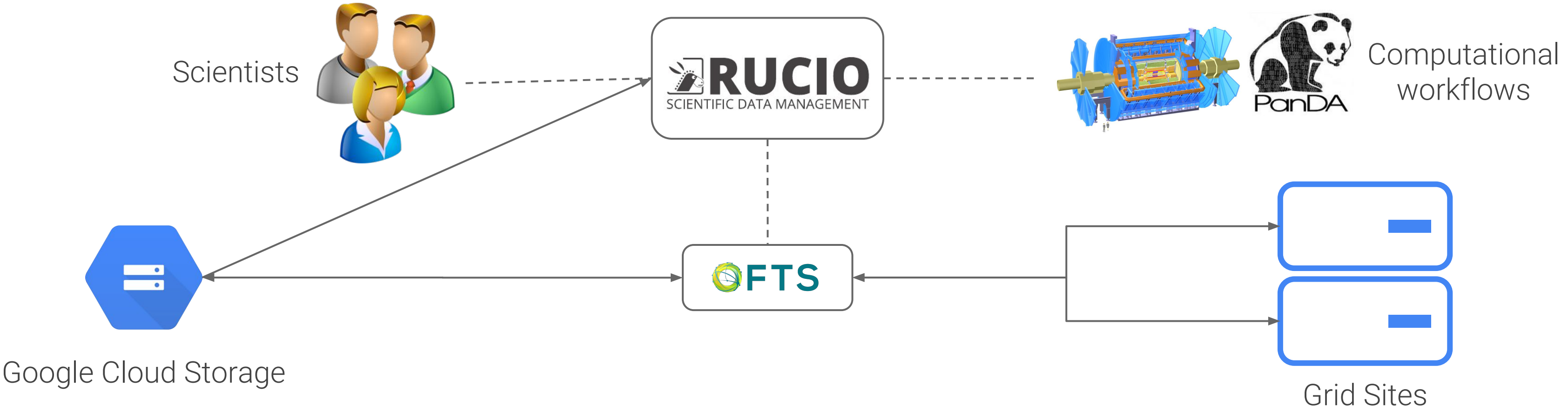
- User analysis
  - Ensure 100% output availability through additional cloud replicas
  - Overflow CPU to cloud compute
- Data placement, replication, and popularity
  - Dynamically expand experiment storage capacity with cloud storage
  - Use cloud networks for additional replication throughput
  - Use cloud internal replication mechanisms for popular data
- Data formats and streaming
  - Unravelling ROOT files into their constituents
  - Cloud-based marshalling of events from files





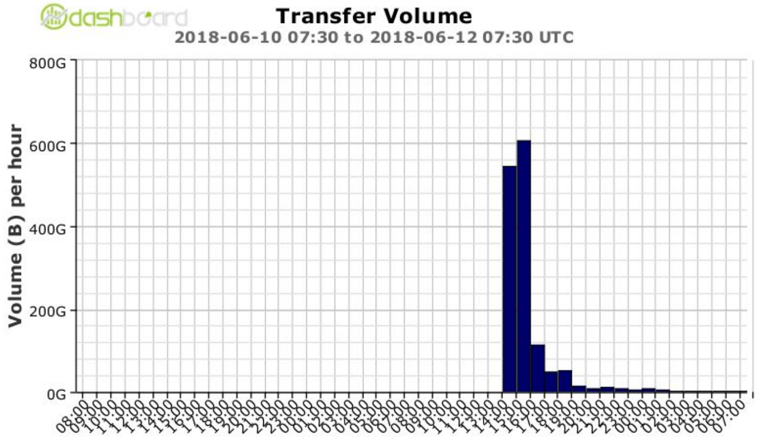
Getting data into GCS

- The ATLAS Data Management system *Rucio* orchestrates all experiment transfers
  - S3 used in the first iteration, since support is already available from both sides
  - Tests successful, however not usable for client-based access (key distribution, server-side signing)
  - Parallel third-party copy is rate-limited to 100MB/sec because we were not using the native GCS API
- Decision to move to GCP-native client-side signed URLs

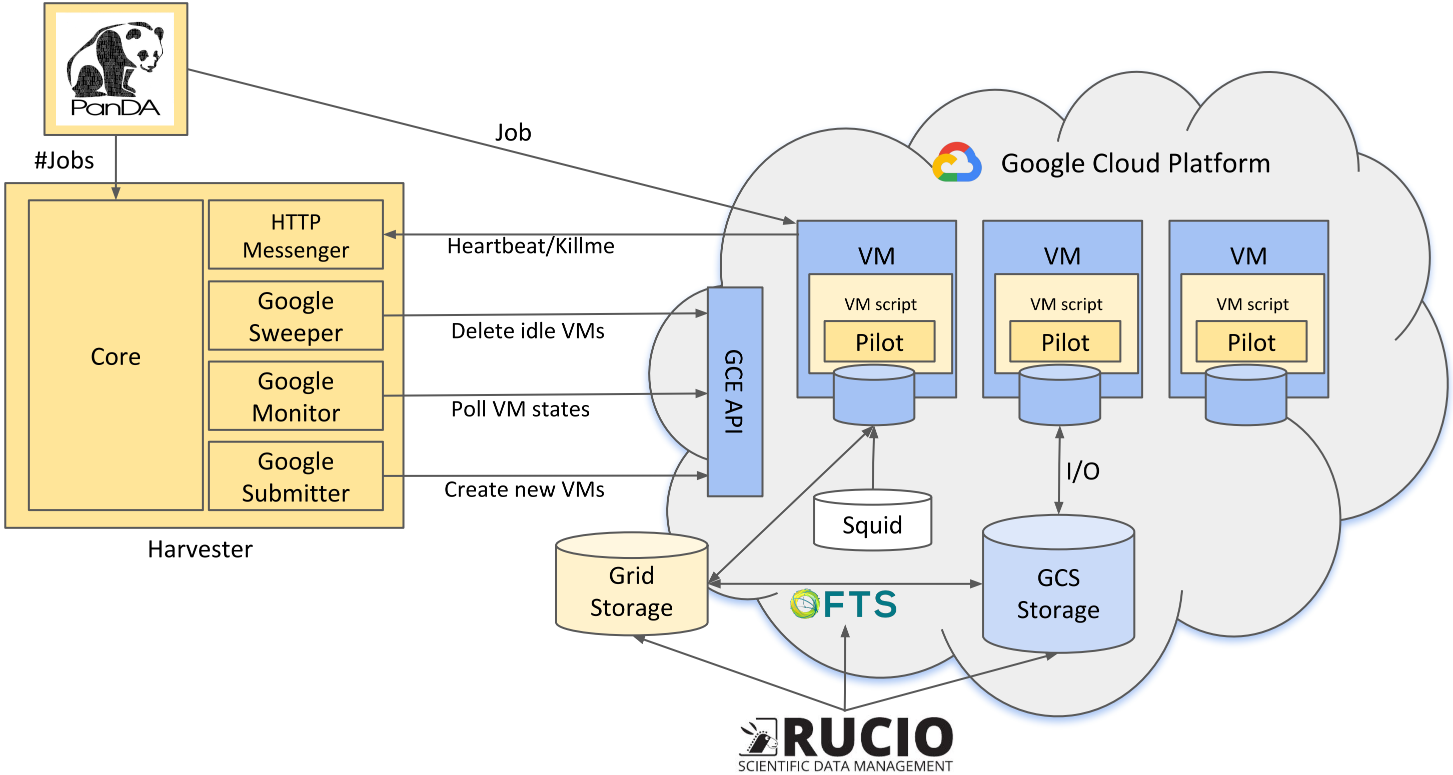


Data evaluation

- The first datasets were moved manually
  - To allow the compute evaluation to go in parallel with the data management evaluation
  - Slow and tedious due to S3 and manual registration
- Using the signed URLs we can use Google Storage like any other WebDAV Storage
  - Implemented full support in Rucio — clients now can transparently access cloud storage
  - FTS development underway to create signed URLs
- Terabyte transfer test
  - Created rules to transfer 1 TB of user analysis data
  - 1TB each to both US and EU Google Data Centres
  - Worked off at 0.6TB/h aggregate
  - Maxing out FTS intermediate stream



Job submission through Harvester edge service

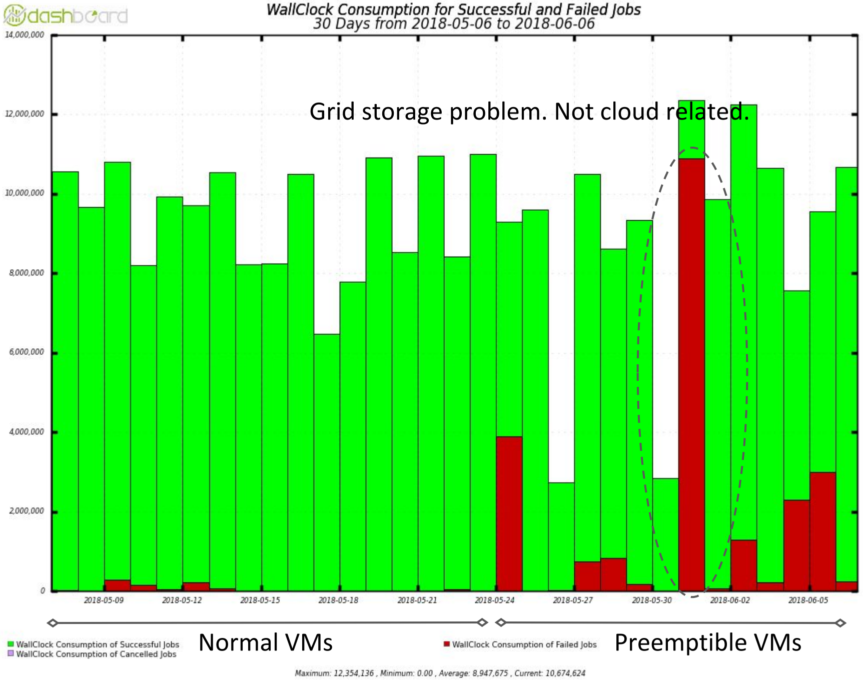


Harvester integration details

- Purest PanDA-GCE integration: no translation layers
    - Plugins talk to GCE via Python API
    - HTTP messenger interface
- Uses unaltered CernVM4 image and cloud-config contextualization
    - CVMFS, Squid, Proxy, Queue, Harvester URL, Log endpoints and startup script
    - VM startup script: ~200 lines of Python run the pilot while sending VM heartbeat/killme messages
    - VMs are recycled once per day based on timefloor option in PanDA pilot
    - Squid deployed in GCE for caching
- Reducing the cost
    - Custom VMs adjusted to ATLAS simulation (8 vCPUs, 16GB RAM, 50GB disk)
    - Stable setup should be profiting from “sustained use & inferred instances” discounts
    - Recently also running on preemptible VMs (20% of the cost)
    - Preemptible VM can be evicted any time and the maximum lifetime is 24 hours

Compute evaluation

- Operated a 120 core cluster running standard **simulation** jobs for 1.5 months
  - I/O to CERN storage
  - Excellent success rate (<<5% errors) using normal VMs
- Preemptible VMs
  - Significantly higher error rate (20-30%)
  - Still gain on a $/event basis
- **Analysis** queue ramping up
  - I/O intensive workloads reading from GCS



Grid storage problem. Not cloud related.

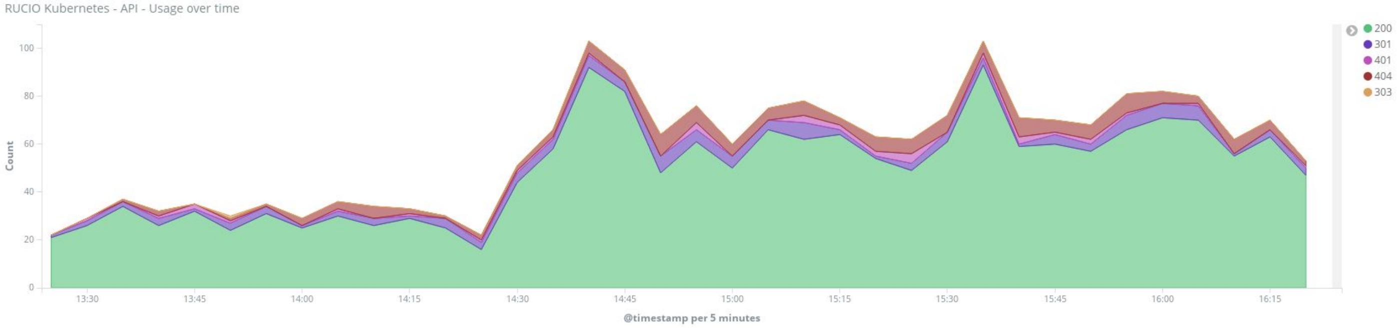
Normal VMs

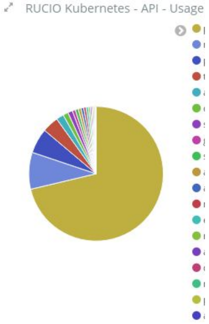
Preemptible VMs

Efficiency of preemptible VMs can be optimized through usage of Event Service.

Kubernetes

- Container orchestration system
  - Originally developed by Google
  - Available in CERN IT and Google Cloud Platform
- PanDA and Rucio are moving towards Kubernetes-based deployment
  - Single-click startup and shutdown of instances on GCP
  - Kubernetes-based PostgreSQL/MySQL backend running on GCP
- Gradual deployment of components— load distributed using HAproxy


RUCIO Kubernetes - API - Usage over time


RUCIO Kubernetes - API - Usage

Summary and outlook

- ATLAS+Google R&D project to evaluate computing models with real use cases
- Interface Compute (PanDA+Harvester) and Data (Rucio+FTS) with GCP native APIs
- First use case evaluations very promising
  - Compute
    - Fully integrated with ATLAS Workflow Management
    - Excellent success rate (10M hours/day) of simulation, cost-saving through preemption
    - Analysis jobs coming online
  - Data
    - Fully integrated with ATLAS Data Management
    - User analysis transfer (0.5+ TB/h) promising, looking forward to native support in FTS
    - Users get automatic and transparent access to cloud storage
- Turnkey deployment of Rucio service using GCP & Kubernetes
- Both Google and ATLAS are committed to a long-term collaboration