The Future of Distributed Computing Systems in ATLAS: Boldly Venturing Beyond Grids

Fernando Barreiro[1], Doug Benjamin[2], Taylor Childers[2], Kaushik De[1],
Johannes Elmsheuser[4], Andrej Filipcic[3], Alexei Klimentov[4], Mario Lassnig[5], Tadashi Maeno[4], Danila Oleynik[1], Sergey Panitkin[4], Torre Wenaus[4]
on behalf of the ATLAS collaboration

[1]University of Texas at Arlington, USA
[2]Argonne National Laboratory, USA
[3]Jozef Stefan Institute, Slovenia
[4]Brookhaven National Laboratory, USA
[5]European Center for Nuclear Research, Switzerland

CHEP 2018, 9-13 July 2018, Sofia, Bulgaria

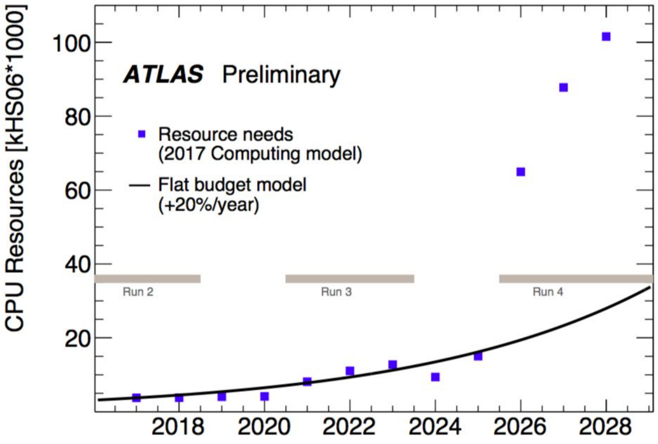
Acknowledgements

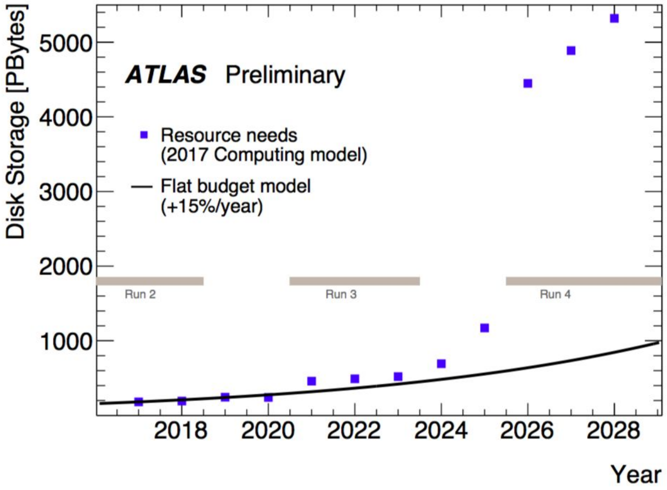
This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357.

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility operated under Contract No. DE-AC02-05CH11231.

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

Motivation

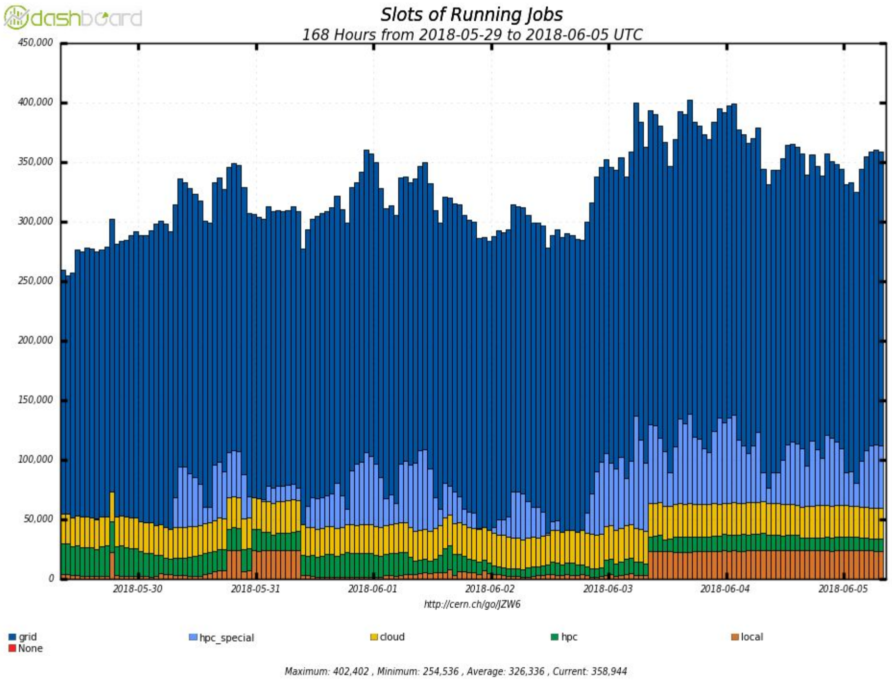




- LHC computing needs keep increasing, while budget is flat at best
- IT landscapes, computing infrastructures and funding models change
- Heterogeneous workloads, architectures, resource types, storages
- We need to be able to use every resource available and use it efficiently
- ...and there is a general manpower limitation

ATLAS usage on opportunistic resources

- Cloud, HPC & volunteer resources used successfully for >5 years
- Resources not always tailored for ATLAS: adaptation needed and inherent limitations in suitable workflows
- This presentation will focus on the effort to harmonize the adaptations and overcome some of the most challenging limitations using Harvester[1]



[1] *See T Maeno's Harvester talk in this conference*

Revised architecture: Server - Harvester - Pilot

Harvester as edge service, capable of integrating heterogeneous resources through plugin interface

HPC

- Run on edge node of each HPC, or potentially centrally if HPC provides a CE
- Data pre-placement and output transfer through download/upload or 3rd party transfer
- Job management
    - Combine jobs into multi node submission
    - Jumbo jobs management with Yoda
- Exploited in US DOE HPC facilities and available for other HPCs

Cloud

- Can run anywhere, usually centrally in shared instance
- VM lifecycle management: create, monitor and delete VMs
- Plugins existing for Google Compute Engine and Openstack

Grid

- Can run anywhere, usually centrally in shared instance
- Standard Pilot submission in different modes
    - Push/pull
    - Closer integration with PanDA server and can receive commands for e.g. Unified PanDa queues

HPC: architectures and software

- Each HPC has own set of architectures and restrictions
  - Different operating systems
  - SW installation: local installations, CVMFS, trend on containers
  - Possibility to provide a Computing Element in the future
  - Different CPU architectures and increasing presence of co-processors
    - Effort on ATLAS SW compilation methodology
    - Currently unable to use GPU co-processors
    - Nodes without disk, using shared filesystem
      - Concurrent file access can create a bottleneck and needs to be optimized

Specifications and Features

Processor: IBM POWER9™

GPUs: NVIDIA Volta™

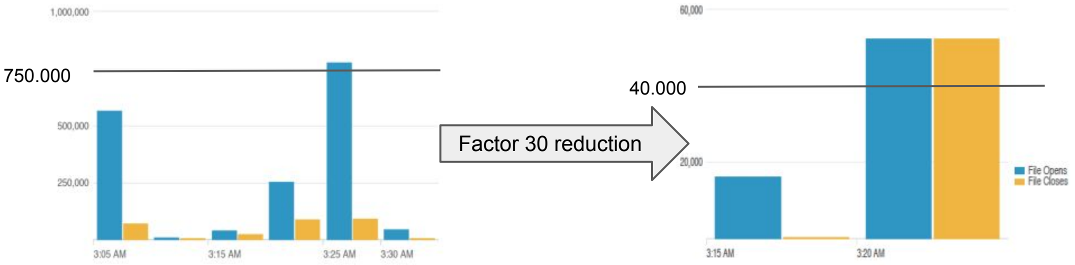
Nodes: ~4600

Node Performance: >40TF

Memory/node: 512GB DDR4 + HBM

NV Memory/node: 1600GB

ORNL Summit specs



16 Aug 2017

4 Sept 2017

I/O operations reduction on Lustre filesystem at ORNL Titan

HPC: data management

- Not always storage element present at HPC
- HPCs with external I/O can use a remote grid storage element
- Restrictive HPCs require data pre-placement to local storage or shared filesystem
    - Download
    - 3rd party transfers managed by Rucio
        - FTS
        - Globus Online
    - Difficult to converge on one solution

HPC: internal scheduling

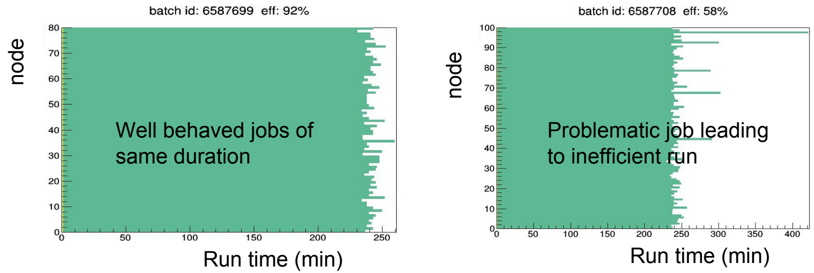
- HPC allocations usually awarded by n million node-hours over a period
- HPC internal scheduling policies optimize the usage of their infrastructures while honouring users' fair shares
  - Usually only multi-node slots
  - Large requests often prioritized
  - Max walltime can depend on the size of the request
  - Backfill opportunities outside your allocation
    - Fill out leftovers with limitation on running time
- However ATLAS workloads are loosely coupled (pleasantly parallel)
  - Typically each job needs 1-16 cores, 2-4 GB RAM/core
  - Runs over a file with few hundred events over several hours

Bin	Min Nodes	Max Nodes	Max Walltime (Hours)	Aging Boost (Days)
1	11,250	–	24.0	15
2	3,750	11,249	24.0	5
3	313	3,749	12.0	0
4	126	312	6.0	0
5	1	125	2.0	0

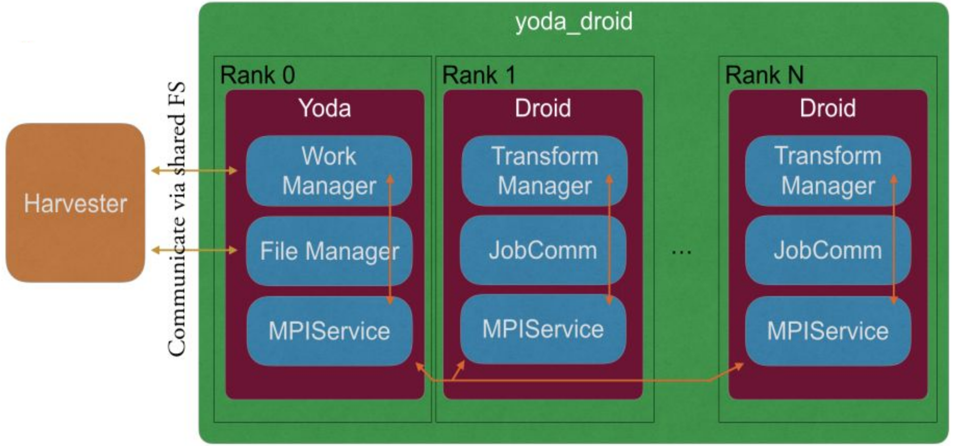
ORNL Titan scheduling policies

HPC: improving the efficiency

- Combining ATLAS jobs into HPC multi-node jobs
  - Manual task assignment to ensure same duration of jobs
  - Failure of one node leads to failure of all concurrent ATLAS jobs
  - Turnaround time not guaranteed, limiting to non-urgent jobs
- Jumbo jobs and Yoda: manage finer granularity jobs through MPI
  - Jumbo jobs package together multiple related jobs and manage these at event level
  - Yoda runs on the HPC and feeds event ranges to subsequent ranks through MPI
  - Further down the line envisage event level streaming

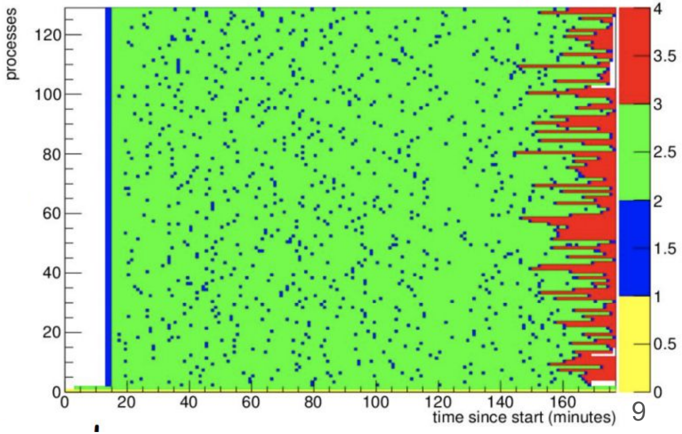


NERSC utilization per node



A Yoda job at Theta/ALCF with 16k cores

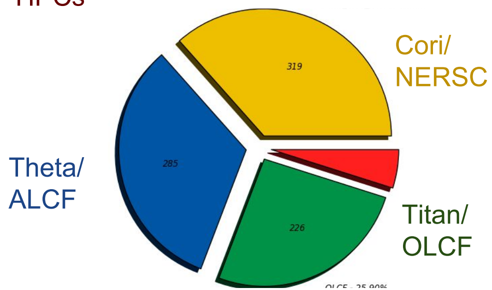
Idle, Processing events which completed, Processing events which didn't complete

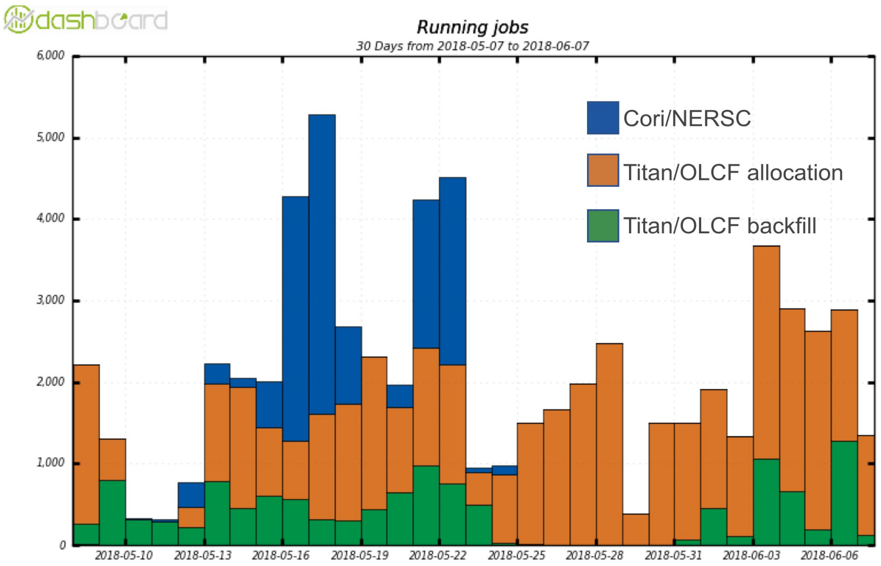


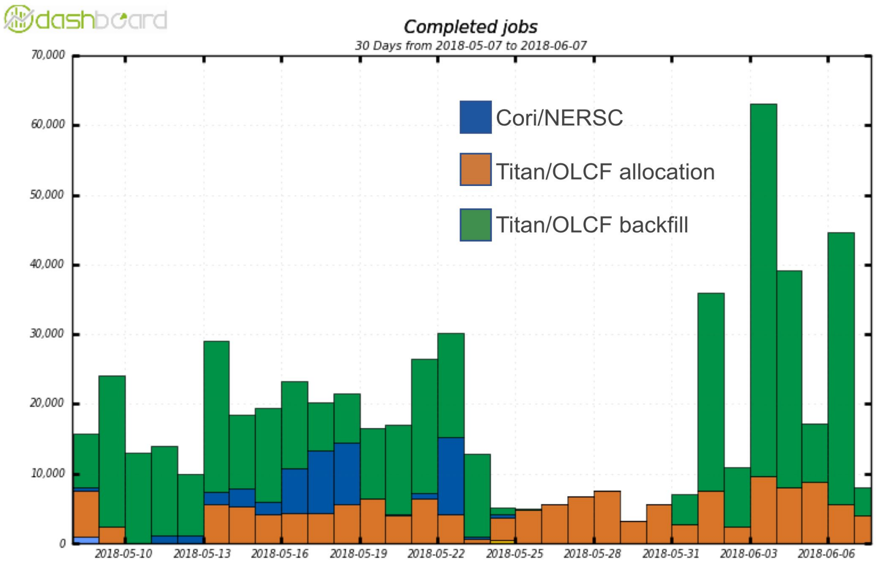
HPC: status

- Flexible plugin architecture in Harvester to integrate very different HPCs
- In use at US DOE HPC facilities
  - Inclusion of other HPCs in EU or US NSF under discussion

Total number of events (in M) processed in 6 months at US DOE HPCs







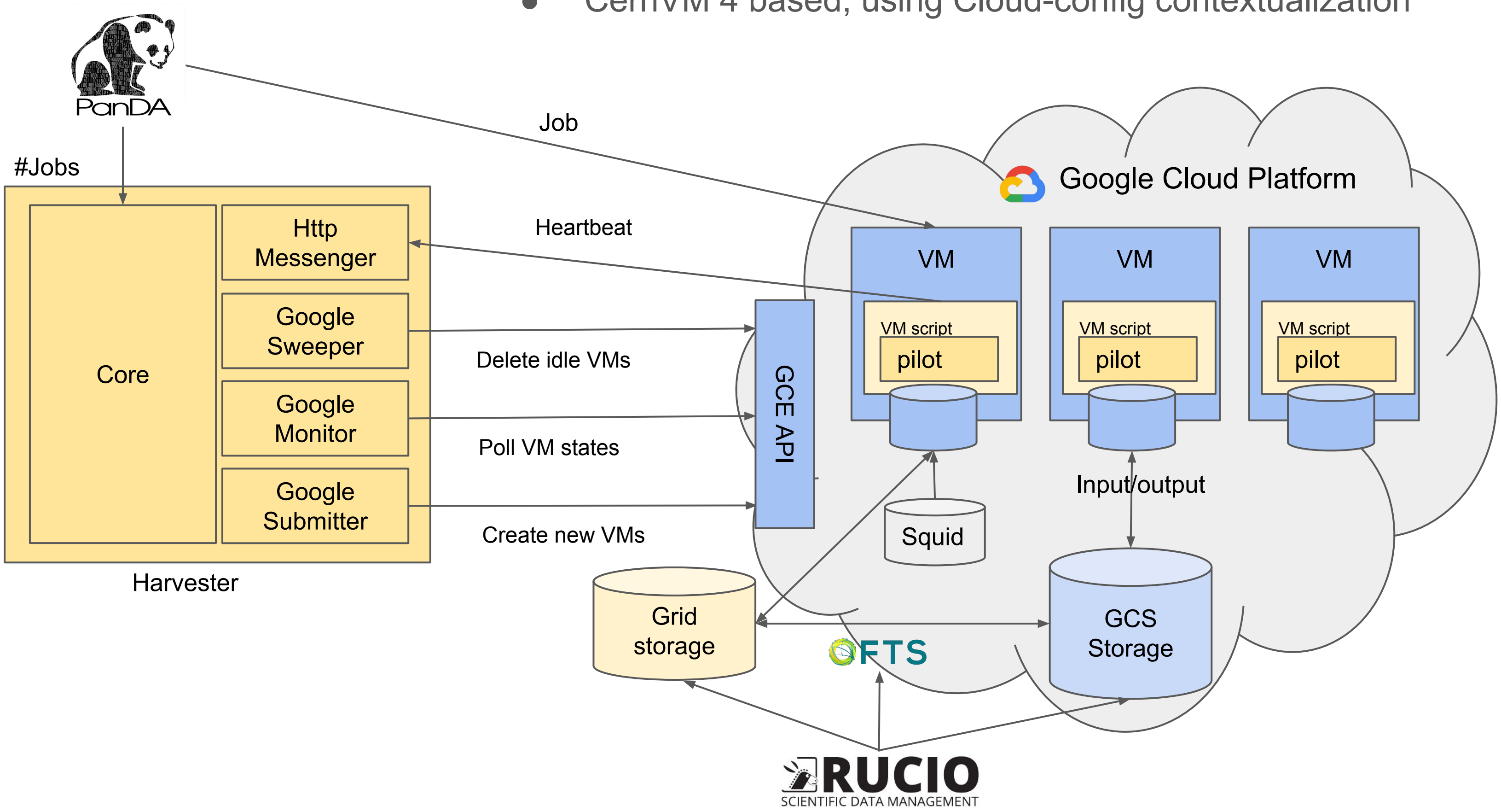
ATLAS Google Data Ocean Project[1]

- Storage becoming a driving cost factor for High Luminosity LHC
  - ATLAS-Google common project to evaluate more dynamic use of storage
  - Store ATLAS data on Google Cloud Storage and access anywhere in the world
- First ATLAS attempt to run both storage and compute on a commercial cloud
- **Data** management: Google Cloud **Storage** like any other storage element for data transfer and accounting
  - Based on signed URLs
  - Third party transfer through FTS
    - Possible from all recent DPM and dCache WebDav endpoints
  - Download and upload of files through Rucio clients
- **Workload** management: manage Google **Compute** Engine resources through Harvester
  - Running a queue for simulation and a queue for analysis

[1] *See also M Lassnig's talk in this conference*

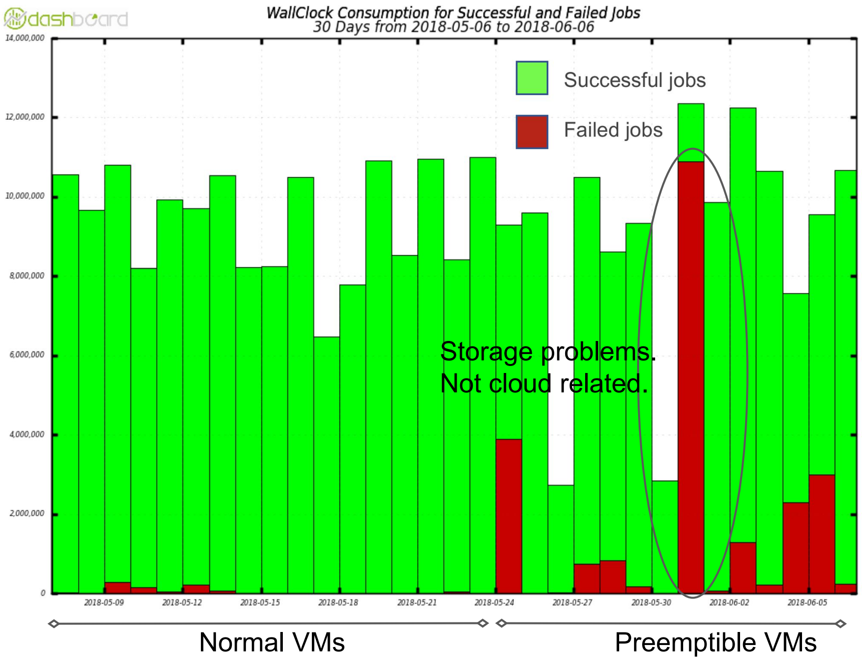
Block diagram

- Top-down, pure PanDA-GCE implementation
- CernVM 4 based, using Cloud-config contextualization



Results

- Google Cloud Platform completely integrated in Rucio for data and PanDA for workload management
- Analysis use case in progress using cloud storage
- Expand on performance, scalability and cost studies



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

Conclusions

- Increasing HL-LHC computing needs
- Grid funding stagnates, but other public and private resources appear
- Harvester edge service with its plugin infrastructure allows interfacing them all
- Examples with key players of today’s IT landscape have been shown
- Current focus on improving efficiency, demonstrating scale and thriving towards standardization to reduce operational costs