Dynamic Resource Allocation with arcControlTower

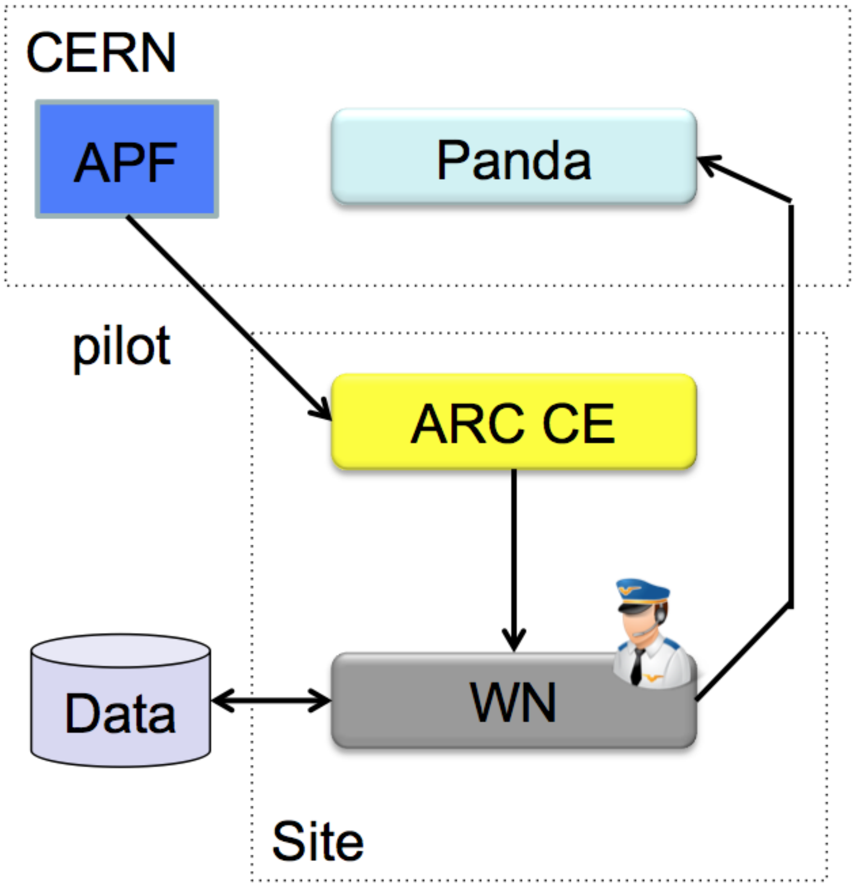
Andrej Filipčič, David Cameron, Jon Kerr Nilsen,
On behalf of the ATLAS Collaboration

Payload submission practice

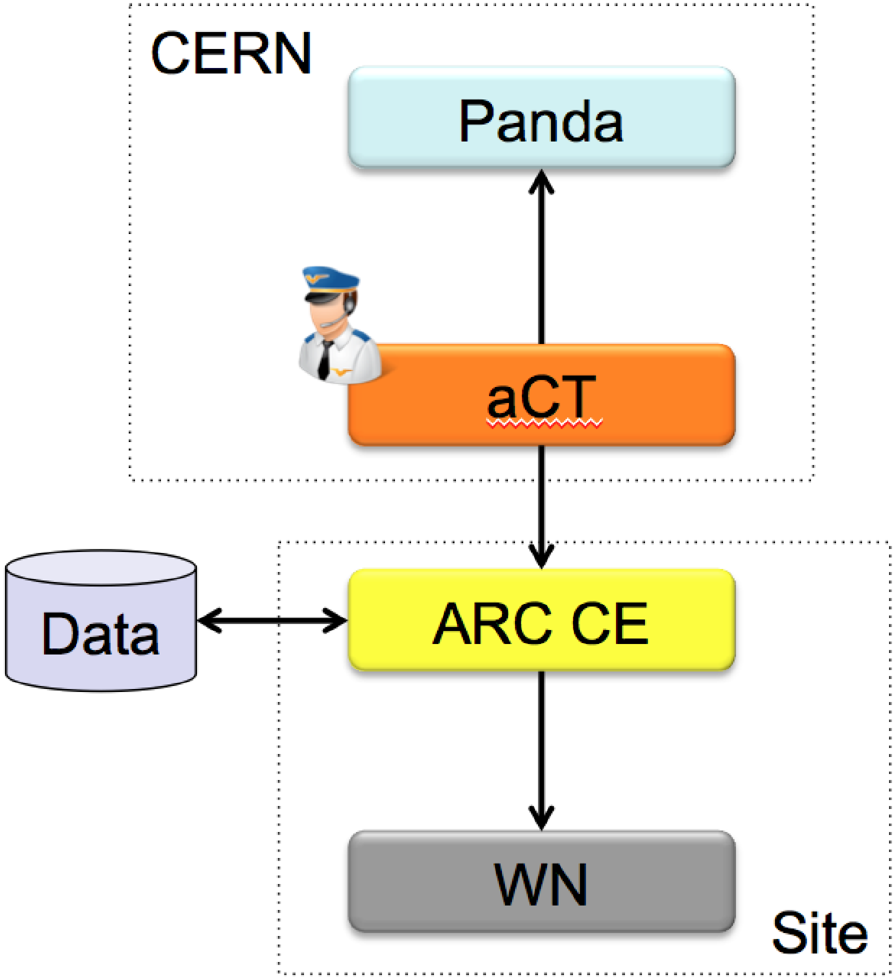
- Push model – direct (full) job submission to grid sites was terribly inefficient and unreliable 10 years ago:
  - failure rates were exceeding 50%
  - Workload management systems could not cope with the submission rate and complexity
- Pull model gained on popularity
  - Dummy batch jobs – pilots – pull the payload from central services
  - Local site instabilities have less impact on central submission service
  - But all the pilot jobs are the same – uniform memory, walltime and cpu requirements

Pilot mode works well only if everybody is happy with equal job resources

Push vs Pull Model



Worker Node pulls PanDA for payload



Payload is pushed to the Worker Node by intermediate service

Ideal distributed model

- An extended/distributed “batch” system
  - Worker nodes – full nodes allocated to external “batch” scheduler (PanDA)
  - Permanent pilots - “batch daemon slaves” - ask for payload
  - Central scheduling system (PanDA) distributes job to the pilots according to priorities and job requirements for resources
- Central scheduling system would manage all users (VOs)
  - Fair-sharing between VOs
  - Common job priority treatment

Was not even planned at the start-up of the grid computing

Distributed Reality

- Sites are still using the conventional batch systems to submit the jobs to clusters
  - We need to deal with multi-level scheduling
  - Central scheduling system and sites need to adapt to each other
- Pilots with uniform resource requirements not good enough any more:
  - ATLAS uses different workloads by memory, cputime, corecount requirements
  - Even worse if other VOs use completely different requirements – simple batch system configuration is not sufficient any more
- Workaround for ATLAS PanDA:
  - Each site has many custom queues, corresponding to different workload requirements:
    - RAL-LCG2_SL6 – default queue
    - RAL-LCG2_MCORE – 8-core
    - RAL-LCG2_HIMEM_SL6 – more memory
    - RAL-LCG2_VHIMEM – even more memory
    - ANALY_RAL_SL6 – analysis
  - When the tasks with new requirements are to be launched (“insane memory”) a new PanDA queue needs to be defined for each site
  - Difficult to maintain long term – after one year of multicore life, there are still sites without mutlicore support

Issues with uniform payloads

- Some sites are shared with other VOs, or are general purpose clusters (eg. supercomputers)
  - Fixed partition allocation does not make sense
  - Shorter jobs would get more cpu resources - backfilling
  - Long (2 day) jobs cannot start on empty extra worker nodes – draining is too expensive for sites
- ATLAS job resource requirements – wide spectrum:
  - 0.5GB to 6GB of memory
  - Minutes to 4 days of walltime
  - 1 to 32 cores
  - Massively parallel jobs coming into ATLAS production – AthenaMP spanning several nodes (Yoda)
- Static PanDA queues are becoming difficult to maintain and use

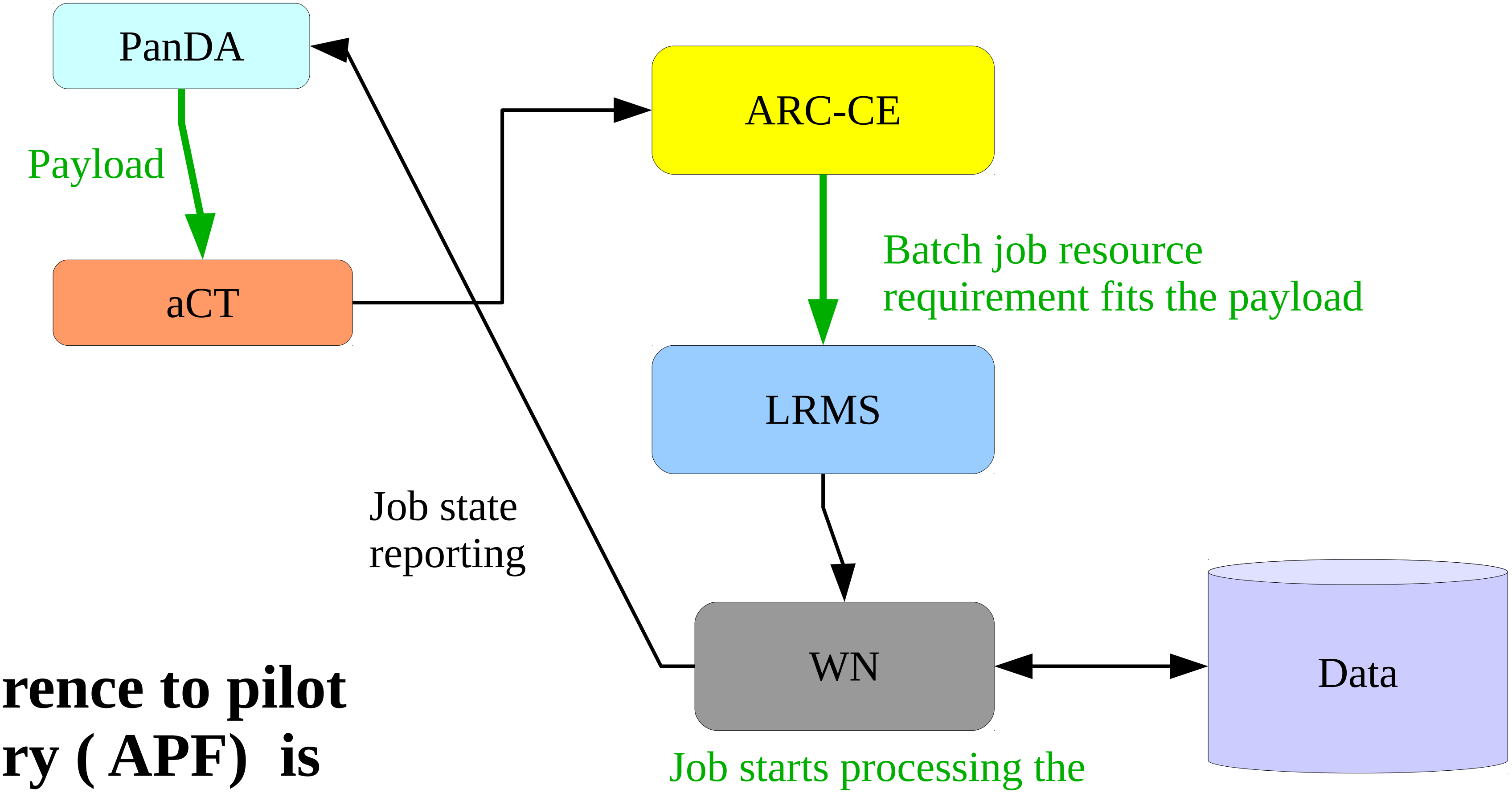
arcControlTower

- See presentation by Jon.K.Nilsen
  - http://indico.cern.ch/event/304944/session/4/contribution/263
- Used for submission to ATLAS Nordugrid sites since 2007
  - Relies on ARC Compute Element – ARC-CE
  - Most of the clusters are shared and have performant shared filesystems which enable input caching
  - Distributed NDGF-T1 is only partially local to the clusters – remote file transfers are expensive
- Version 2 rewritten from scratch to separate:
  - Generic ARC-CE submission interface
  - ATLAS PanDA interaction and payload management/submission

Modes of aCT job submission

- ARC native mode:
  - aCT communicates with PanDA and submits predefined payload to ARC-CE
  - ARC-CE transfers input and output files and submits to the batch
  - Pilot wrapper on worker nodes only executes the payload without accessing the external network
    - Outbound connectivity still used by CVMFS and Frontier
  - Worker nodes do not use grid middleware
  - Good for sites with capable shared filesystem with caching of input files, as well as HPC sites
- Truepilot mode:
  - aCT fetches the payload and submits it to the ARC-CE
  - ARC-CE submits the batch job with predefined payload
  - Pilot on the worker node does the same as on the conventional pilot sites, but skips the fetching of payload from PanDA
  - Good for worker node centric sites with capable local disk space and fast transfers to close storage site

aCT Truepilot



PanDA

Payload

aCT

ARC-CE

Batch job resource requirement fits the payload

LRMS

Job state reporting

WN

Data

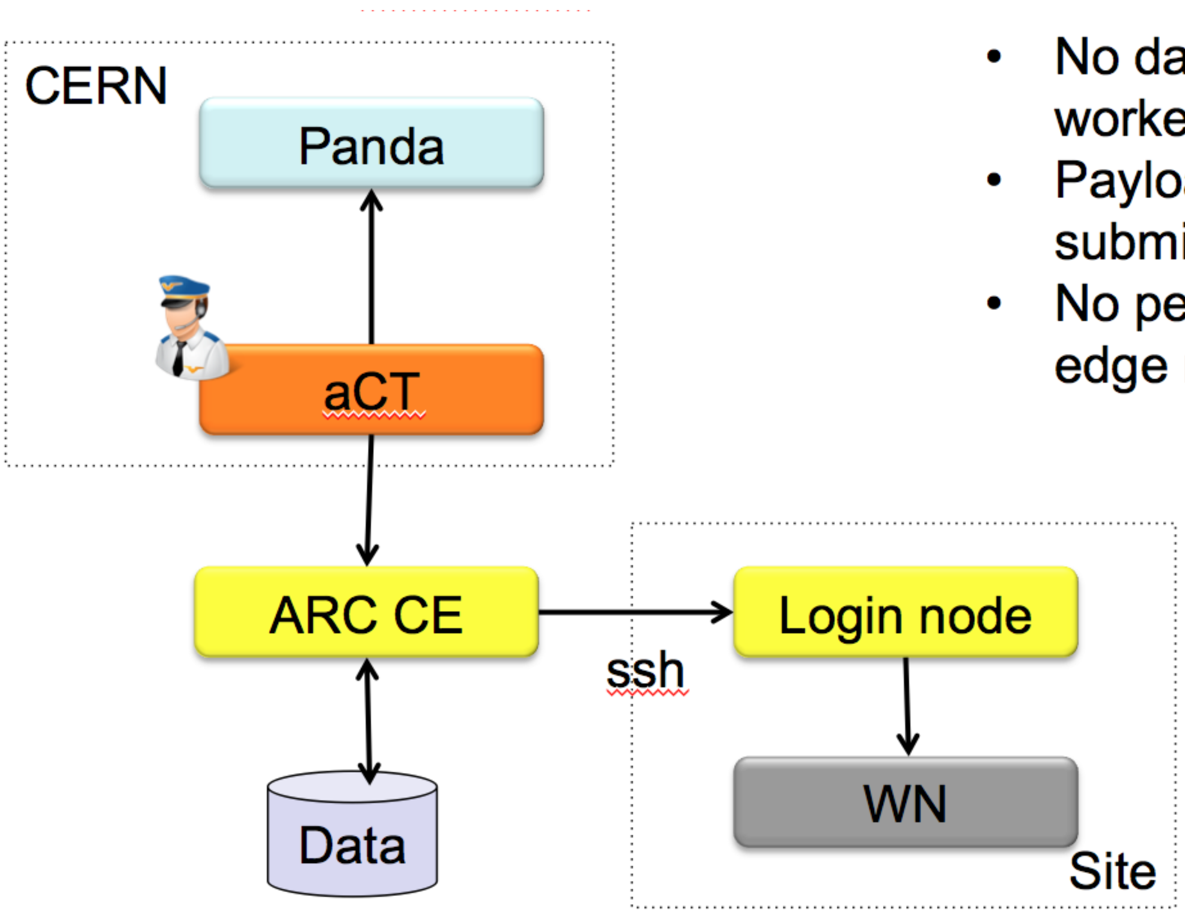
Job starts processing the given payload without asking PanDA

Difference to pilot factory (APF) is marked in green

Pilot factory vs aCT Truepilot

- Pilot factory:
  - Highest priority jobs start running first
  - But the batch jobs have all the same resources
- aCT truepilot:
  - Payload known in advance – the batch job has the resource requirements fit to the job
  - Payload can request any memory, cputime, corecount, of course in agreement with site capabilities
  - But the late-binding is partially lost – highest priority jobs need to wait some time in the batch
  - Bad worker nodes can cause black holes – fast resubmission cycle

aCT and Supercomputers - HPCs



- No data access from worker node ✔
- Payload known at job submission ✔
- No persistent service on edge node or open ports ✔

Using aCT native node

Experience

- aCT ARC native mode used for several years – payload resource description already tuned in PanDA
- Also used for 6 supercomputers (EU, China) where the pilot pull mode does not work due to site policies
- Fully in operation in LRZ-LMU Munich Tier-2 sites for two months
- Being tested on RAL Tier-1 with smaller amount of jobs

- Best suited for sites, where advanced resource limits (cgroups) are deployed
  - ATLAS can better fit the high-memory jobs to installed resources

Issues

- Predefined payload must be first queued in the batch – loosing the strict highest-priority execution order
  - Keeping the number of queued jobs low – 20% of running ensures the waiting time is maximum a couple of hours
- When resource specifications are too tight, the batch system would kill the job
  - Safety factor of 2 for the job walltime
  - Requested memory can be exceeded by some jobs – APF sets the maximum memory limit as specified for the queue, while aCT tunes it to the payload request, which can be lower
- aCT supports only ARC-CE sites

Future

- Try it on more sites and get higher statistic to analyze the benefits for ATLAS
  - Get more resources with short jobs
  - Provide fast turnaround for short analysis jobs
- Possible implementation for cream-CE and Condor-CE needs further discussion and development
  - ARC python clients support for other CEs

Conclusions

- arcControlTower has been successful for ATLAS on WLCG grid sites as well as in enabling opportunistic resources such as
  - HPC sites
    - https://indico.cern.ch/event/304944/session/10/contribution/92
    - https://indico.cern.ch/event/304944/session/9/contribution/161
    - https://indico.cern.ch/event/304944/session/9/contribution/153
  - Volunteer computing with BOINC, see talk by D.Cameron
    - https://indico.cern.ch/event/304944/session/7/contribution/170
- aCT provides a way to submit any kind of workload to any ARC-CE enabled ATLAS site:
  - Native ARC-CE mode tuned for shared sites
  - Truepilot mode for sites designed for the pilot approach