aCT and ARC CE

- Rod Walker, LMU Munich, 4th Dec 2014

- ARC Control Tower
  - Past, present and future
- ARC Compute Element
  - unique capability
  - matter of taste

aCT – ARC Control Tower

- Developed so that Nordugrid sites could integrate to Panda
  - pilot jobs required WN network – not allowed
    - Ah but...cvmfs,Frontier work! Selected IPs is allowed.
    - want known job with custom limits
  - WN pulls data from SE – no local SE on NG
    - must know inputs and pre-fetch
- aCT spoofs WN pilots
  - pulls job from Panda, sends heartbeats, final update

The payload

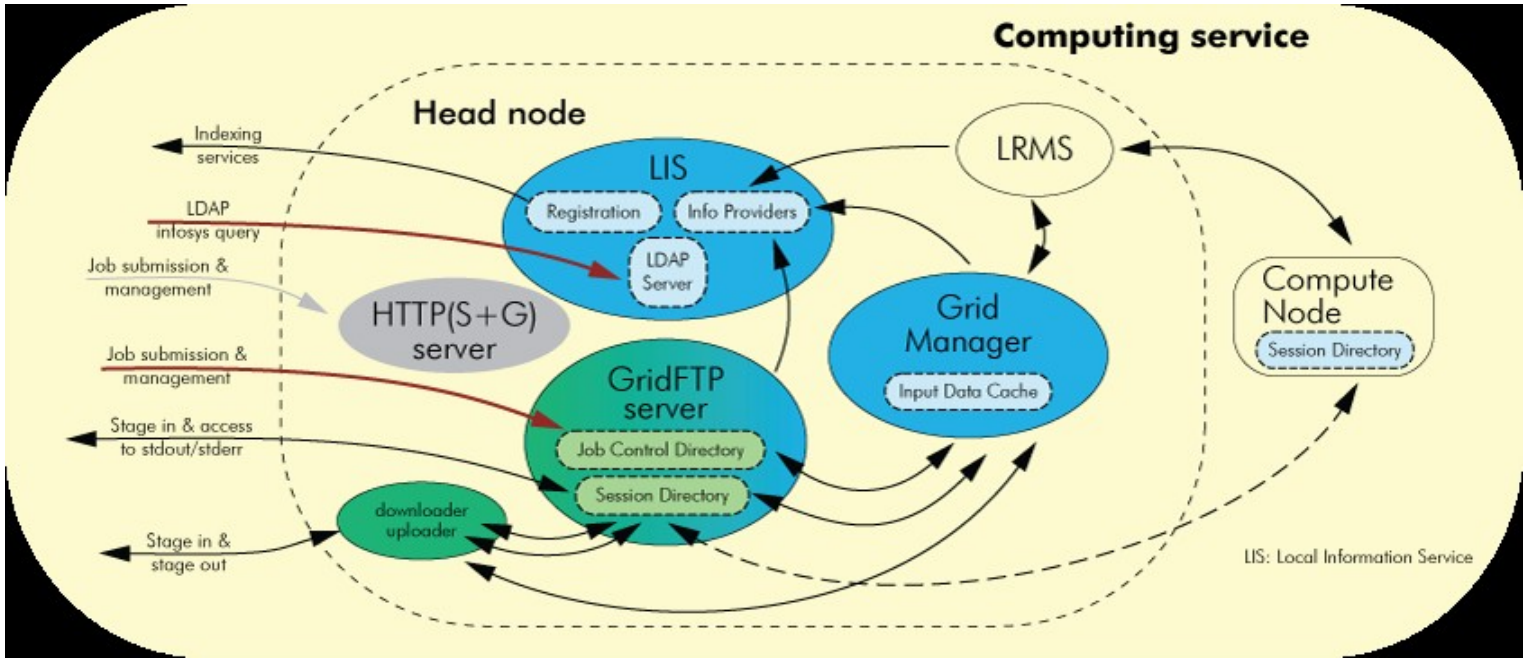
- In between preloaded pilot is submitted to ARC CE
  - input data known
  - resource limits, RSS+wall, known and set
- Lose late binding
  - do we care? Gain a lot.
  - instead must set BS priority for eg. merge job
  - modifying priorities possible(with dev) but maybe not necessary

Dynamic queues vs aCT

- Range of limits(mem,wall,cores) in Panda queues(static or chosen by JEDI)
  - send pilots to each queue with activated jobs
- Late binding messed up here too
  - cannot choose which pilot starts next
  - priorities between jobs in different queues NOT respected
- aCT sets priority per job
  - Panda chooses next job to run – better.

ARC Compute Element

- Fully functioned Grid Compute Element part of UMD
  - BDII, APEL, GOCDb, SAM, …
  - ATLAS, CMS, LHCb, ALICE can use it
    - single users/small VOs have/will adapt...and benefit!
- Unique functionality
  - data input and output done outside batch job



Matter of taste

- Installation straightforward
  - SL, Debian, SUSE(builds ok)
  - single config file
  - major batch systems supported
    - PBS, SGE, Slurm, HTCondor, Loadleveler, LSF, Boinc
      - bash scripts for submit,stat,cancel
      - perl for infosys
- Operation
  - no scale limit observed yet: challenge to a T1?
    - RAL runs 3 ARC CEs but not due to scaling
      - 1 per Condor sched for RAM reasons & redundancy
  - debug is easier, e.g. log file per job

GridKa has 8 CreamCEs for 12k jobs
Try 1 ARC CE?

Use cases for ARC CE

- Traditional Computer centers/HPC
  - networkless WNs
  - job limits with no surprise workloads
- No local SE
  - data handling by CE, outside batch job
  - maybe suitable for Clouds – asynchronous data movement
- ssh backend to ARC CE
  - no service on site, just passwordless ssh
    - sshfs, batch commands via ssh – works well.
- Replace CREAM CE for pilot submission
  - ignore data handling – needs large shared FS, not common on WLCG sites
    - just use to submit pilots reading/writing to local SE
      - without pre-loaded jobs(works now), or with pre-loaded job(dev)

ARC CE to replace Cream CE

- Use to submit normal ATLAS pilot jobs
  - no data staging required(goes via SRM)
- Why do some need an alternative?
  - RW: “Cream CE is unreliable, difficult to configure/debug/upgrade, not scaling well, high RAM usage”
    - not alone: read lcg-rollout list
  - interfaced to aCT to receive preloaded pilots
    - with custom per job resource limits
    - increased entropy without multiple pilot streams
      - kill misbehaving jobs. Backfill MCORE.
  - passing resource limits to batch works
- Flexibility of choosing custom job limit
  - recall surprise RAM usage of pile MCORE – no manual action required
    - aCT served sites with higher BS queue limits, just received jobs with bigger limits

Conclusions

- aCT and ARC CE crucial for certain use-cases
  - HPC, strict CC, Boinc, no local SE, Cloud(?)
- Useful for ATLAS production
  - fine-grained & flexible job limits via aCT
    - aCT currently only supports ARC CE
      - no strong case/manpower to support preloaded pilots to Cream
    - no more empty pilots!
  - resource limit setting for dynamic queues
- personal recommendation rather than ATLAS