Hiding the Complexity: Building a Distributed ATLAS Tier-2 with a Single Resource Interface using ARC

Stuart Purdie, Graeme Stewart, Mike Kenyon, Sam Skipsey, Wahid Bhimji and Andrej Filipcic
Hang on a moment!

Aren't all the Tier-2's distributed?
• Fortunately, I can point to Ian Bird's talk from yesterday about caching models of data storage!
• Fortunately, I can point to Ian Bird's talk from yesterday about caching models of data storage!

• It's about how it is distributed
• Fortunately, I can point to Ian Bird's talk from yesterday about caching models of data storage!

• It's about how it is distributed
• It's about how the data is distributed
ATLAS gLite model

User

Pilot Factory

gLite CE

gLite CE

gLite CE

Panda

Hiding the complexity
gLite jobs start running on the WN, then access data

- Several options here; commonly job stages against DPM
• gLite jobs start running on the WN, then access data
  - Several options here; commonly job stages against DPM

• ARC stages all the data to a local cache before job start
  - Job accesses data locally
Tiers before bedtime

One Tier 0
Hiding the complexity
Tiers before bedtime

One Tier 0

Few Tier 1's

Many Tier 2's
Tiers under our model

One Tier 0

Few Tier 1's

Several Tier 2's
• Three compute clusters
• Storage at each cluster
Proposed Model

- Three compute clusters
- One Storage Element
  - Plus modest cache at each compute cluster
  - Cache no harder to maintain than home directories
• VO storage management complexity reduced
What's the advantage?

• VO storage management complexity reduced
• Easier to add in further (smaller) compute clusters
  - CE installation is easy
Compute vs Data

• Compute is:
• Data is:
Compute vs Data

- Compute is: cheap
- Data is:
Compute vs Data

- Compute is: cheap, idempotent
- Data is:
Compute vs Data

- Compute is: cheap, idempotent and fungible
- Data is:
• Compute is: cheap, idempotent and fungible
• Data isn't
Compute vs Data

• Compute is: cheap, idempotent and fungible
• Data isn't

• Moving compute jobs around is solved
• Data requirements is the hard, and expensive, part
Our situation

- ECDF - University wide compute cluster, on an internal changing basis, and administered centrally
- Has data facilitates, but not quite at the scale for Atlas use to peak efficiency
Our situation

- ECDF - University wide compute cluster, on an internal changing basis, and administered centrally
- Has data facilitates, but not quite at the scale for Atlas use to peak efficiency
- Glasgow - Pure Grid cluster, plenty of data (1.4PB)
- It would be really, really, nice to be able to use Edinburgh as compute, backed by data at glasgow
• ECDF - University wide compute cluster, on an internal changing basis, and administered centrally
• Has data facilitates, but not quite at the scale for Atlas use to peak efficiency
• Glasgow - Pure Grid cluster, plenty of data (1.4PB)
• It would be really, really, nice to be able to use Edinburgh as compute, backed by data at glasgow
• Site: Simpler to run - and would mean we could pick up smaller compute clusters too
• VO: Simpler to administer, simpler to use
Early Attempts

• When the Grid was young, we tried this
• Had data at Glasgow, and marked it as a close SE to Edinburgh
• Crippled by the transfer times
• Got to the level of looking at the costs of putting in a dedicated lightpath ...
• Not cheaper!
What's changed?

• ARC handles data differently
• Pre-stages to a local cache before job execution
  - Cache is managed by the computer
• Therefore less sensitive to distance between data store and compute cluster
What's the plan again?

- Three sites collectively make up a Tier-2
- Tricky to have enough data at each for optimal usage
  - In particular for smaller additional sites
- ARC's cache allows for one data store to be shared
- Aim to consolidate the data stores
Does it work?

• Step one was to install ARC at Glasgow, and join it the the NorduGrid Cloud for Panda work

• More extreme than the long term plan; a good test environment
Does it work?

- Step one was to install ARC at Glasgow, and join it to the NorduGrid Cloud for Panda work.
- More extreme than the long term plan; a good test environment.
- 13k jobs completed.
- Prestaging from NDGF storage works well.
- CPU efficiency is good.
• Step one was to install ARC at Glasgow, and join it the NorduGrid Cloud for Panda work
• More extreme than the long term plan; a good test environment
• 13k jobs completed
• Prestaging from NDGF storage works well
• CPU efficency is good
• Cache maintenance required thus far: 0
What didn't go so well?

• A/A system integration
  - X509 certificates work fine, but LCMAPS integration had linking issues due to bitness concerns
What didn't go so well?

• A/A system integration
  - X509 certificates work fine, but LCMAPS integration had linking issues due to bitness concerns

• Different interface for job submission + management
  - Already used in ATLAS, so trivial in this case
Net Benefits / Challenges

• Simpler data distribution model for users
• Smaller sites relived of the storage upgrade treadmill
• Last mile of data storage is automated
• Usable now

• Different submission and control interfaces
Thank you