VO Box Requirements and Future Plans

ATLAS

GDB Bologna - October 12, 2005

Miguel Branco - miguel.branco@cern.ch

Outline

- Usage and Motivation for VO Box
- ‘Baseline’ Requirements considering current model
  - LCG VO Box
- An improvement
- A proposal for a future model

Goal

- In this presentation we will try to distinguish what the current, short-term needs for SC3 are, and what a future model should look like (>= SC4)
  1. Current model
  2. A small improvement
  3. A possible long-term model

(sorry for the “text-intensive” style of presentation but it’s easier to present remotely!)

Usage of VO Box

- ATLAS expects the VO box to be deployed on all sites participating on ATLAS production
  - Currently deployed in few sites only
- It is used by the ATLAS data management to:
  - ‘validate’ files on LRC (eg. Many jobs running at a site - perhaps more than one doing the same task (zombie) - but only one output should be registered
  - Launch and ‘babysit’ FTS/g-u-c/SRM requests (eg. many jobs request a file but only one copy should arrive and be registered onto LRC)
  - Real-time monitoring/management of ATLAS transfers to the site
  - Validation/registration of ATLAS datasets onto ATLAS-specific metadata catalogs
  - … (a few other data management use cases)

Usage of VO Box

- Site subscribes to dataset
  - And automatically gets all latest versions
- Jobs/Users claim datasets
  - ATLAS can enforce its VO policy and accounting (eg. PWGs quotas)
  - Also used to make sure data is ready on disk
- Datasets, Files and Metadata
  - VO Box provides the consistency layer for these interactions

Usage of VO Box

- In theory, a ‘huge’ box at CERN accessible by WNs all around the world *would* work:
  - But it doesn’t scale; Harder to maintain
  - Would require outbound access to all WNs
- ATLAS deploys onto the VO Box
  - A service container (Apache + mod_python)
  - Uses the security infrastructure (mod_gridsite + MyProxy client)
  - A persistent database (MySQL, SQLite, …)
  - A set of ‘agents’ acting on the requests

Motivation for VO Box

- To provide more efficient usage of the resources
- To introduce the concept of *site* into the ATLAS distributed computing software stack
  - It is a Computing Model concept!
- To allow site information to stay local
- Part of a broader strategy to improve the experiment's middleware, and allow sites to administer their systems without 'central' intervention
  - (we also requested LRC per site)

A look into LCG Data Management

- FTS: A huge leap but difficult to use in isolation
- Use Case: ATLAS wants to move data from site A to site B
- “Insert FTS request” ?
  - What about intermediary sites (hop) ?
  - And what prevents multiple similar (or the same) transfer requests from being inserted multiple times ?
  - What prevents similar (or the same) set of files from being requested to ‘stage to disk’ many times over?
- Big lesson from DC-2/Rome: putting grid ‘business logic’ onto job scripts at the worker node is naïve and highly inefficient - uncontrollable, difficult to stop, …

A look into LCG Data Management

- Fact for SC3: LCG DM tools are still insufficient although they are a major improvement!
- We are (and will continue) to put forward our requirements for the “generic” parts of our VO Box s/w
- Nevertheless we believe the VO Box is a *1st class component on the Grid* and should be kept in the future
  - But **not** as it stands today! (see next slides)

Requirements (based on current model)

https://uimon.cern.ch/twiki/bin/view/Atlas/DDMVOBoxRequirements

- Operating System: SLC3
- Performance: Any WN class PC sufficient
- Disk space: 20 MB for DDM software
  - 1-2 GB for LCG s/w (POOL)
  - around 20 GB for DBs & logging information
  - ~10 MB writable directory space for file based catalogs
  - In total: 30GB should be enough
- Connectivity:
  - Login via ssh/gsi (no root access required)
  - Insecure and secure port available for apache server: (port 8000 and 8443 only)
  - Outbound access to DDM global catalogs, FTSes, SRMs, LRCs, MyProxy

Requirements (based on current model)

- An external connection to the MySQL DB at the site is desirable (especially during start-up phase for debugging) ***but not mandatory***
  - All messages are designed to be “TCP traced”: XML or "plain-text" messages. Minimal traffic.
- No requirement to access s/w installation area at the site from VO-box
  - VO-box can be “independent” from the cluster setup as long as it is at the site with in/outbound connectivity)
- WNs at site can access these services
  - e.g. there is a published environment variable at the WN with the hostname of the VO-box; or it’s published in the information system
- Backups: system can reconstruct information if backend DBs are lost

LCG VO box

- Current status:
  - An ‘advanced’ form of a SSH server
  - vobox-proxy-init is *extremely* useful!
    - This should have been done a **long** time ago!
    - Extremely valuable for those doing secure applications!
  - While a *temporary* workaround it provides all necessary functionality
    - Simple, but *excellent* handling of security infrastructure (certificate cache)
    - but ssh is a bit too much functionality…

A small improvement

- The model needs to be improved as sites feel their security is compromised:
  - **Agreed**, but LCG VO Box is a very good start! Excellent work by LCG on this quick development
- Looking at the ‘Usage of the VO Box’ it is evident that what ATLAS requires is a:
  - Secure container for services
  - With dynamic deployment/management of services
  - *And this is what should be provided as an LCG service*
- Looking at EGEE, Globus (cf. OGSA), even ARDA initial specification, service architectures are a useful paradigm
  - Interesting to see that ATLAS came to the same conclusion not by academia studies but by running Data Challenges!

A small improvement

- A first improvement would be choosing a ‘bare minimum’ set of applications *(not really services at this stage)* - common across experiments? - and deploy them as part of the standard installation
  - Apache & MySQL are generic apps but critical security-wise
    - They could be included by default and configured to open a few ports to the outside
  - Sites (LCG?) get responsibility to update Apache / MySQL
  - More work for the sites, but full control of security
    - Interactions with mod_gridsite could be possible as well
- Later we would deal with deployment of services (instead of applications) and on using a common security model (eg. mod_gridsite + MyProxy)

A possible future approach

- A generic, dynamic, secure services container
  - This was my first request to ARDA back in the days (not very “high profile” request at the time)
    - Still to be discussed and agreed in ATLAS and elsewhere
  - Now we *really* need it if we are seriously developing secure applications
- Even when LCG provides a set of DM components that match current ATLAS-VO box usage, ATLAS will always want to profit from the extensibility of a services container
- Requires:
  - Generic security infrastructure for the VO: sites see the ‘VO’
  - Grid Computing is all about VO policies anyway

Final Message

- Naïve to assume the ‘grid’ middleware will handle all scenarios, all usage patterns on all conditions
  - Experiments need the flexibility to handle the middleware in a way that’s efficient - according to the experiment’s needs - and secure!
  - Difficult - impossible? - to find generic usage pattern across all experiments, just because experiments **are** different
    - We can keep pushing for a single File Placement Service that handles all usage scenarios or…
    - focus on the base components (FTS), get them right and allow the experiment to efficiently extend them as necessary
- Alternative:
  - (try to) agree on all sets of distributed software stack (I’m pessimistic) and implement it for SC3/4
  - Create mechanisms to validate/authorize/sign experiment software at the sites and sites handle it (no flexibility)
- Seems preferable to create security services framework (remember that this is a core Grid concept - perhaps the most basic concept and we do not yet have the corresponding s/w)

Final Message

- Are the services running on the VO Box generic?
  - Some of them, yes (eg. the ‘FTS’ babysitters)
    - They should go to the m/w
  - But not all of them are:
    - some depend on eg. ATLAS dataset definition, datablock definition, ATLAS metadata model
- An important point is that *even if they were generic* there *is no uniform (and secure) way* to deploy and use them
  - A generic container would provide just that
- Services would co-exist and for some time compete on functionality (coopetition :-)
  - (again, it was my first request to ARDA)
  - Less work for the sites (only support the container)

Final Message

- VO box is a work-around as it stands
- Many of its services (ATLAS) should go to the grid m/w
- The fact that we deploy services per site may turn out to be irrelevant:
  - Gain geographical and load scalability, lose on administrative scalability
  - But it seems to be a good approach because other services are per site (FTS, LRC) and WNs should only access a restricted set of *local* services (local information stays local)
- The security is a fundamental aspect - and it still needs to be sorted out
- Dynamic, Secure Service Container is the desirable approach
  - Aids interoperability, compatible with EGEE model
  - It would *finally* allow for seamlessly integration between the application distributed software stack and that of the grid