Enabling Grids for E-sciencE

# FTS Developer Workshop

***Summary of issues presented by experiments and output of discussion on these issues***

***CERN, 16 November 2005***

**www.eu-egee.org**
**www.glite.org**

Information Society

# Outline

- Summary of issues from morning session
- Tasks arising prioritised by experiments
- Technical detail on specific tasks
- Feedback on SRM experiences from experiments
- Summary of individual experiment issues from morning sessions

# Summary of issues / features

- **Work out retry logic and Hold states**
- **Pre-staging**
  - Configurable to be on or off
- **“Central service” or equivalent: (data planner and scheduler)**
  - LHCb suggested road-map: client tool first
- **VO customisation (plug-ins)**
  - LFC interaction
  - Other catalog interactions
  - Pre/post steps
  - Agree on convention for this
- **How to handle T1 to T1**
  - Also T2 to T2 and T2 to T0?
- **Bandwidth monitoring**
  - Better channel monitoring
- **What are the experiments’ data movement model**
  - Store and forward or direct transfer

# Summary of issues / features

- **Fix advisoryDelete problem**
  - Deletion

- **Full transfer history**
- **Timing information + number of retries summary**
- **Check-summing and validation**
  - Simple file size check.. Check-summing harder via SRM
- **Interface stability**
  - Frequency of update?
- **Alert system**
  - This is the Site and Service Functional Tests
  - Generate alerts when FTS sees a problem

- **Process:**
  - Just who do we email when it goes wrong?

# Prioritised tasks

1. **Planning and routing**
2. **Monitoring**
3. **File integrity checking**
4. **Retry logic and Hold states**
5. **FTA Cataloguing**

---

6. **Staging support**
   - Should do the planning early

- **Items 1 to 5 seen by experiments as required for SC4**

# Planning and routing

- **Small steps?**
  - Experiments rather have a service that does it rather than having to do these steps in the expt. framework
- **Redirection**
  - Another service to handle the redirection and forward requests
  - Multi-hop is difficult to do
  - Yet another component…. This should be part of the “overall transfer service”
- **Otherwise.. Experiments do it themselves**
  - This is undesirable.. A waste of effort.
- **Initial plan.**
  - single service, lightweight, simple … no multi-hop
  - Expand to multi-hop later

# Planning and routing: data model

- **What are the experiments' data distribution models**
  - Store and forward or intermediate copy
    - Do we delete the copy on the multi-hop?
  - Do something easy first…
    - Dump into volatile storage
    - LHCb, Alice, Atlas: Don't make an entry in our Catalog – delete the intermediate copy!
  - Need to add broadcast jobs
    - e.g. AOD / ESD to all / many sites
  - Experiments don't want to know the topology!
    - If the experiment sees channels, we're doing something wrong!

# Monitoring

- **Fabric monitoring**
  - What is needed to effectively manage the service
    - Alarms on channel status, service functional tests
    - Debugging the fabric
  - Handled by operational side
- **Application level monitoring to VO users**
  - Like GridVIEW but for FTS jobs
    - Using R-GMA as message bus
  - Broadcasts only terminal states
    - With detailed job info: schedule time, transfer time, etc
- **FTA plug-in for experiment framework developers**
  - Provide messages on states changes that experiment frameworks need to react to
- **FTS clients**
  - Improve the API to provide full history status of transfer
  - Better statistics

# File integrity checking

- We currently do check the file size after transfer
- We cannot check the checksum inside FTS without a full re-read of the file across the WAN!
- We can only do it if storage systems could provide this to us
  - Issue to follow up with SRM developers

# Retry logic and Hold states

- **Currently retry policy is very simple**
  - Num retries (default == 3), interval between retries (default==10mins)
  - 3 retries is ~usually OK when things are working
- **Retry back-off**
  - Currently we don’t back-off
  - Back-off differently depending on different conditions e.g. file not staged yet
- **Parse error messages to FAIL terminal cases**
  - E.g. src not available, no proxy, no mapping for user
- **Hold state: the aim was to pause jobs to allow admin interventions to fix problems without draining queues**
  - If we have infinite retries with sensible back-off this state becomes defunct
  - Experiments need hooks to cancel jobs that are persistently failing
    - This is how the experiment frameworks expect the FTS to work
    - VO plug-in allows experiments to implement the complex retry policies that they want (we provide a default that can be overridden)

# FTA Cataloguing

- **Experiments use LFC / globus RLS / pool catalog**
- **Need plug-ins for FTA**
  - Atlas will do Pool and globusRLS
  - IT/GD will do LFC
- **FTA framework on FTS pilot will be used to integrate this**
  - Requires documentation, tutorials, dev RPMS

# Pre-staging support

- **Experiments stated that this is not a strong requirement for start of SC4**
  - It is a production exercise and data will not be coming from tape at T0
  - Intelligent retries and knowledge of file cache status may help us survive before full pre-staging is implemented

- **Understood pre-staging is still required for FTS to work efficiently in all cases**
  - As an operational issue having it would reduce the number of errors we see on analysing logs
  - This makes differentiation of real problems easier

# Feedback on SRM experiences from experiments

- **Problems on SRM.advisoryDelete, etc**
- **Issues of latency on calls**
  - SRM internal latencies are a large overhead
- **Integrity of the transfer**
  - Checksumming
- **Should be tolerant of clients that interpret the spec differently**
  - Underlying problem is that SRM v1 is not sufficiently well specified
  - Need *consistent* error message agreed across implementations
- **SRM v2 should solve these?**

# Summary of individual experiment issues

*Taken from experiment slides presented during the morning session*

www.eu-egee.org
www.glite.org

# CMS

- **Single most noteworthy lacking features:**
  - checksum (or at list file size) check after transfer
    - (file size already checked, but no check-summing yet)
- **Other desiderata:**
  - FTS stays in “waiting” even if file already exists at destination and SRM.put fails
    - DISCUSS retry policy
  - Would like to have full timing info for each transfer: request, start, end

# Alice

- **Transparent use**
  - ALICE requires the automatic discovery of the FTS Endpoints and the names of the FTS proxies servers through the information system
  - An upper layer able to hide the transfers among the different SRM
- **Tool for traffic monitoring**
  - Possibly R-GMA?
- **Closer coordination of FTS and LCG2 releases**
  - Updating the clients on VO-Box by hand is a pain

- **Things not needed soon (or at all?):**
  - Catalog update (handled externally)
  - LFN resolution (handled externally)
  - Tx-Ty predefined channels not needed [ Transparent use? ]
    - Deal in terms of site name or SRM names

# LHCb

- **Pre-staging needed: timeout period of FTS Transfer Agent**
  - ‘Failed on SRM get: SRM getRequestStatus timed out on get’
  - Currently LHCb pre-stage externally.
- **Different behaviour of dCache/Castor**
  - On file overwriting? Can we do better? Overwrite flag? Wait for SRMv2?
  - Related problem in SRM v1: No single method to perform physical removal
- **advisoryDelete problem**
- **Better retry**
  - Don’t retry ‘lost-cause’ files e.g. source isn’t there

# LHCb

- **Plugins**
  - Interested in LFC plug-in (can share some experience on this!)
- **T1 to T1 channel use-case**
  - Ideally full connected mesh over ~6 T1 sites
  - T2 to T1-CERN mentioned as well
    - DISCUSS model
- **Transparency:**
  - Central service to submit
  - Road-map suggested: what can we do *now*

# ATLAS

- **ReplicaVerifier**
  - Done already by FTS(?)
- **Staging**
  - Find out stage status of file
- **Plug-ins:**
  - Catalog interactions – not just grid catalog – multiple catalog updates
  - Zipping file plug-ins
  - Call-backs to avoid polling
- **Retry policy and Hold states**
  - Can FTS retry more! (except ‘permanent-error’ jobs)
- **Priorities**
  - Already done: maybe allow high-pri submit for VO manager as well
- **TierX to TierY transfers handled by the network fabric, so channels between all sites should exist**
  - Routing… data planner.
- **Bandwidth monitoring**
  - Statistics –success, failure
- **Error reporting issues (who do we call / mail?)**

# Other things that came up

- **All experiments poll**
  - Can we use the statistics from FTS to help with this – reduce the polling time?
  - Call backs?
- **Retry logic and Hold state**
  - We should discuss the model here
- **VO manager role**
  - Who gets to modify what?
- **Interface stability**
  - How often is it OK to make changes that may break things