



Enabling Grids for E-scienceE

File Transfer Software Multi-VO improvements

Gavin McCance – JRA1 Data Management Cluster

*GDB Meeting
July 20 2005, CERN*

www.eu-egee.org



- **Overview of Components**
- **Multi-VO support**
 - Changes from current version
- **Other improvements being worked on**

- **File Transfer Service is a **fabric** service**
- **It provides point to point movement of SURLs**
 - Aims to provide reliable file transfer between sites, and that's it!
 - Allows sites to control their resource usage
 - Does **not** do 'routing' (e.g like PheDEX)
 - Core FTS does **not** deal with GUID, LFN, Dataset, Collections
- **It's a fairly simple service that provides sites with a reliable and manageable way of serving file movement requests from their VOs**
- **We are understanding together with the experiments the places in the software where extra functionality can be plugged in**
 - Improved FTS framework is much more flexible in this regard
 - Primary reason why the changes were made

- **LCG and gLite teams worked on original architecture and design document for the file transfer software and the service**
 - A prototype (radiant) was created to test out the architecture and was used in SC1 and SC2
 - Architecture and design have worked well for SC2
 - Current released version of FTS was based on this design
- **Original design had no concept of VO (i.e. was entirely VO-unaware)**
 - The hope was with minimal extension we could deploy this design in multi-VO mode: it didn't work out
 - Would have caused serious deployment and operational headaches
 - Chose to deploy FTS VO-unaware and live with it for throughput phase
- **We now have an updated FTS design and implementation with multi-VO support included**
 - This is currently being testing

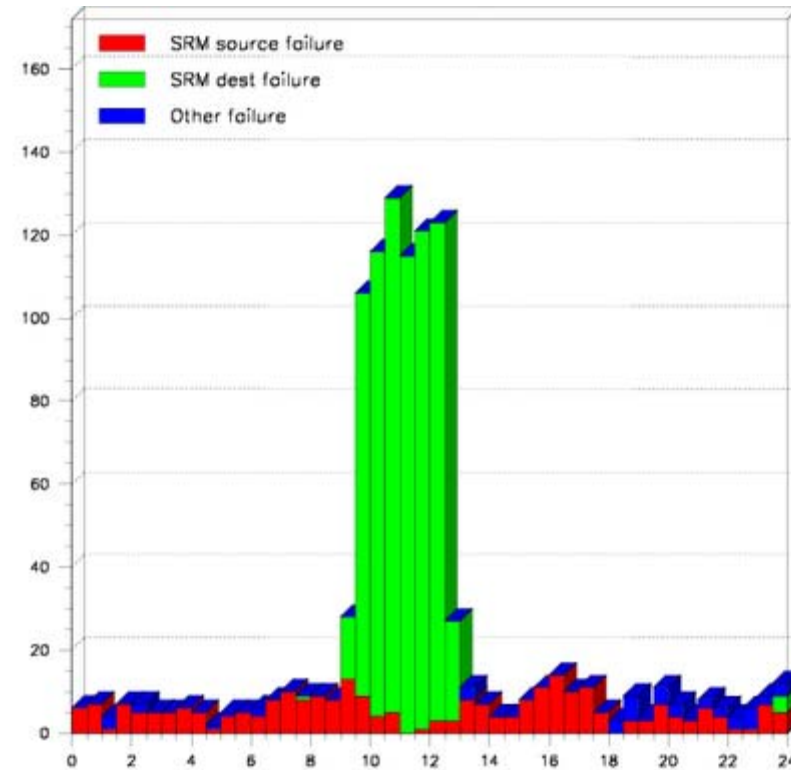
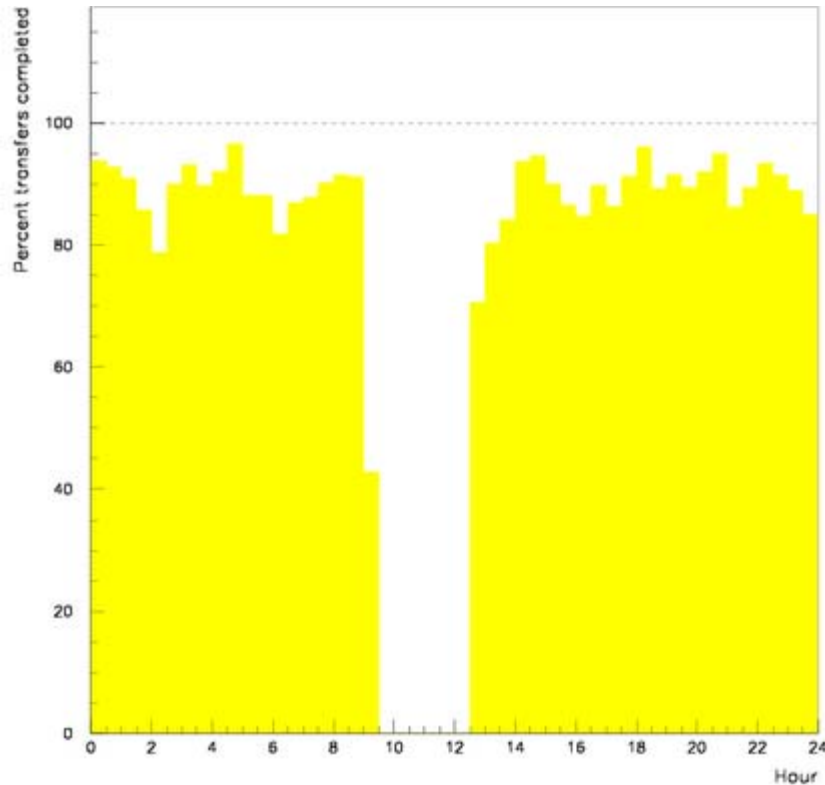
No VO support: why this is bad

- **No fair-share between VOs**
 - Current scheduling policy is FIFO
 - ...and service does not know about VOs
 - Experiment X submits 10000 jobs
 - Experiment Y then submits 1 job
 - → Experiment Y will wait a long time
- **No VO-plugin framework for standard operations in current production FTS version**
 - e.g. standard VO-specific catalog update, file resolution
 - VO “plugins” currently needs to be implemented as separate processes (or “agents) connecting to the production FTS DB
 - This can be appropriate for some things, but not all...

- **New FTS version is VO-aware**
 - Gets your VO from VOMS or mapfile on job submission
- **Scheduling**
 - Channel (site) managers can set VO fair-share per channel
 - Algorithm untested in production
 - We'll see how it goes... await feedback...
- **New concept: VO production manager**
 - Can control, view and cancel all VO jobs on all channels
- **Existing concept: channel (site) manager that can manage all jobs on a given network pipe regardless of VO and manage that pipe**
- **Site can maintain control of the fabric**
- **VO managers given full control over their VO jobs**

- **FTS File Transfer Agent (FTA) actually performs the file transfer and can be extended by VO specific plugins**
 - Plugins allow VO-specific operations (retry, cataloguing, allocation, ..staging soon).
 - We provide reasonable default plugins for these
 - Override if you want:
 - e.g. a staging plugin that knows about other things you do with your stage pools. Cataloguing plugin that updates your own VO-specific replica / metadata catalogs.
 - We provide a basic framework for these plugins to run in
- **You also still have the option of running a program externally against a view of the production DB**
 - Less favoured option: since it's easier to corrupt production DB, harder to debug, harder to deploy and operate. More work...
 - Preference to restrict access to the production DB
 - But the option is there if you need it

- **Better monitoring capabilities**
 - We store much more in the schema now
 - Already have some monitoring / reporting in current version:



- **Better monitoring capabilities**
 - We store much more in the schema now
 - Already have some monitoring / reporting in current version:
 - There is room for improvement
 - More timely monitoring / specific alarm conditions
 - More easily digestible stats / current status
 - *What is actually going wrong now?*
 - The new schema gives us better hooks for getting hold of this monitoring information, doing tracing diagnostic jobs, etc
- **Working on support for SRM copy as an option for a channel instead of SRM put/get**
 - Still seem to be problems with SRM copy interoperability
 - SRM teams are working them out

- **FTS plugin framework and multi-VO code developed**
 - Based on the existing code-base
- **Deployment module missing**
 - Being written now
 - Deployment model will change somewhat for the FTAs from the current version
- **Testing now on pilot FTS cluster (gLite test box)**
- **Proper stress-testing after throughput phase (August)**
- **Need to get experience of running it in production**
 - Majority of the existing experience we're building up in SC3 is still pertinent

- **Stick with the old version**
 - No VO support
 - No fair-share
 - Harder to debug, deploy and manage VO specific plugins
 - Increased support load
 - Restricted monitoring capabilities
 - Hack around monitoring deficiencies
 - But will allow us to finish SC3

- **File transfer service running in production for SC3**
 - Service behavior and service procedures being understood
- **Multi-VO support being added**
 - Allows fair-share among VOs
 - Allows VO production manager to control the activities of a VO
 - Channel manager continues to control the overall activities on a given network channel – involved sites retain control of their fabric
- **Other improvements being added**
 - Better monitoring
 - Support for SRM copy
- **Current throughput phase of SC3 is providing valuable experience in running the service**
 - Build on this with the improved version