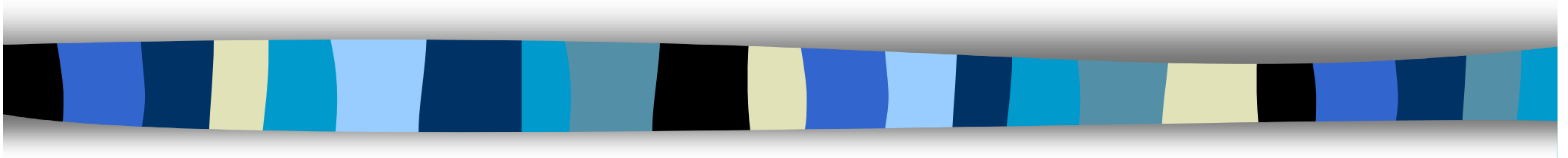


# Installing BaBarGrid over EDG at SLAC: a challenge ?



Gilbert Grosdidier  
*LAL-Orsay / IN2P3 / CNRS*



## BaBarGrid ? At SLAC ? Why ?

- BaBarGrid is meant to offer a common interface to the user
  - **FROM wherever** he submits (UK, Padova, Princeton, ...)
  - **TO whichever** site he submits (SLAC, Lyon, INFN, ...)
- Indeed, the datasets are not/will not be all & always available only at SLAC
  - they can be spread/shared among the different BaBar production centres
  - and this whichever type of data one considers
    - Objectivity, Root, Ntuples, ...
- This is for the time being mainly targeting data analysis use cases
  - but could be extended later on
  - when getting more experience



## BaBarGrid (2)

- On the Grid, one of the elements called Resource Broker selects the processing site by considering the **resources** specified by the user
  - Availability and load of the machines
  - Datasets he/she wants to process
  - Operating System he/she required (or not)
- This could decrease the amount of unnecessary transfers of data between remote sites and SLAC
  - One moves the analysis requests towards the data
    - instead of the contrary
- This could also allow for CPU load balancing between the different BaBar sites



## BaBarGrid (3)

- This means the BaBar user does not want/need
  - to care where his analysis job will be sent/executed
  - to learn/use a new type of batch interface
- But he only wants to care about one thing:
  - retrieving his log/output files **AT HOME**
    - **directly** on the machine from where he submitted the request
- In principle, the pair Globus+EDG should provide the tools to get all these
  - by now (1.1.4), the **automated** retrieval is not satisfactory
- It does not seem yet that another Grid Toolkit is announcing a facility similar to the RB



## Do we really need the EDG layer ?

- It is adding a more compact, integrated wrapper interface to user's jobs (simpler and better unified)
  - Even if it is not yet complete, IMHO
- The RB is adding the tool to select the execution site depending on the data resources available there
- It is adding the load balancing ability between execution sites in the RB as well (Resource Broker)
- It is adding the concept of Virtual Organisation (VO) to federate the sites able to offer resources to a given experiment (horizontal merging, user management)

# Very Tight Schedule Indeed

- Try to get a "proof of feasibility" by end of June '02
  - meaning in fact: identify the locks and show-stoppers and look for quick solutions and fixes ✂ **DONE**
  - and to have a few selected (and experts in this case) users able to run an analysis job if we are lucky ✂ **DONE**
- **Final target:** have some **production** environment ready for all users by the **end of this year**
  - with attractive interface tools
- Want to have this reached thru tailored install, customized to SLAC site
  - with only very limited software modifications: NO development must be foreseen (using **ONLY** standard tools)
    - Unfortunately, missed this one (see later) ☹



## Early show stoppers, as seen @ SLAC

- There were 3 types of issues raised thru EDG/Globus
  - use of LSF Batch Scheduler
  - AFS File System used for User Home Directories
  - Batch Workers located inside of the IFZ
- They are not specific to SLAC, indeed
  - can belong to untested areas of these 2 S/W layers
  - solved thru ad hoc workarounds, like in other sites, with minimal fixes/improvements
  - they are interleaved



## No Access to Home Dir on WNs

- LSF default: the Work Dir is the Home Dir
- But: NO AFS token is conveyed thru EDG/Globus
  - rather normal when remote submission
  - so: no write access to home dir
- In addition, NO EDG/Glob. command to chg Work Dir
  - lack of flexibility
  - better to implement one !
- SLAC/SCS currently studying implementation of Globus gssklog
  - security issues investigated
  - obviously, token creation will clear this access problem
  - not mandatory, IMHO.





## AFS & the area shared between CE & WNs

- This is the Globus gass-cache culprit:
  - Globus assumes a shared area (gass-cache) to transfer data (cert. proxy, among others) between CE & WN
  - For each user: `$HOME/.globus/.gass_cache/`
  - this area must be writeable (with no token, cf prev. slides)
  - and this happens to be **impossible** when lacking AFS token
- In addition, EDG (?) was writing the job-state-file directly inside of the `.globus` area !



# Remedy to the Gass Cache issue

- To allow for the EDG layer to write directly into the gass-cache area, I built a hack into the `globus_gram_job_manager` module
- To allow for Globus to work w/o a token, the idea was
  - split `.globus` & `.gass_cache` areas between AFS & NFS
    - for each and every user
  - move everything dedicated to temporary files into the latter
    - suitable for both S/W layers in fact
- In addition, each request is now assigned a specific subdir in the gass-cache area
  - and there is now a job-state-file for each request
- All fixes were implemented into the same module



## The IFZ case for the WNs

- At SLAC, the Batch Workers are located INSIDE the IFZ, for both inbound and outbound IPs
  - while the CE (and the SE) are located outside
- But, this prevents the EDG job wrapper to fetch the user script thru a globus-url-copy command (using gsiftp protocol) directly from the RB (located in UK)
  - this is rather inaccurate, could alleviate this if the script was split in several steps
- Since this setup (WNs inside of IFZ) is rather common (and sensible)
  - It is strongly suggested to EDG to adopt quickly the solution explained below (EDG 1.2.1 ?)



## Remedy to the IFZ issue

- I was able to build a safe hack around this issue
  - the job wrapper is now split in 3 parts
    - pre-fetch, run and post-download scripts
    - first and third ones are run on the CE
  - these 3 scripts are held in the new per-request subdir
  - this hack is implemented in the globus-lsf-job-submit script
  - there were no other hidden traps down the way
  - it is still required to fix EDG-1.2 for this
- As a by product, I checked that the WNs (LSF batch workers) can run either RH 6.2 or RH 7.2 with this fix
  - assuming the user's job contains NO call to any Globus/EDG tool (e.g. globus-url-copy)

# Conclusions for EDG-1.1.4 install @ SLAC

- Three parts of the Globus/EDG software were installed at SLAC: **CE, WN and UI**
- This exercise clearly showed that **they are running fine altogether, and also with the RB** 😊
  - meaning that the output stuff is actually returned to the RB
- Been able to build required hacks:
  - for some script links of this chain
  - for one module of the compiled stuff
    - even if this was not expected
- A minor point remains, for installing/running the UI
  - requires links in the `/opt` area to be installed on all front-end nodes
  - clearly wish to avoid this on next versions



# Here comes the RB !

- The lack of stability of the EDG-1.1.4 Resource Broker during the tests was really a pain in the neck
  - despite all efforts of our UK colleagues in I.C.
  - it was very tough to send more than 30 requests in a row without having one of the daemons dying
    - meaning **MTBF: 2-4 hours**
  - even when the network was stable
  - the jssparser was particularly fragile, but not only it
  - so the job retrieval was indeed very erratic as well
  - the nice Web monitor did not always show the break, and where the break was
  - in addition, **any hiccup on one of the links in the CE-Network-RB chain was sufficient to break the communications**
    - often requiring RB manual restart



## Remedies to RB instability (?)

- Don't even think to let any user experiment these kinds of trouble !
- This means, IMHO, that these daemons REQUIRE to be closely and actively monitored
  - meaning they need to be automatically restarted when dead or sick !
- EDG-1.2.0 seems to be very touchy as well
- Is there such monitoring within [EDG-1.2.1](#) ?
- If not, suggestion: could we cooperate with some experts to achieve this quickly and cleanly ?
  - this requires a very specific cross-check of the response of each daemon (and not only: is it alive ?)



## Near Future for EDG @ SLAC

- EDG 1.2 was due any day since end of April ...
  - install is now on achieved for EDG-1.2.0
- Install on RH 7.2 badly wanted
  - schedule was: Sept '02, within EDG 1.4. **Still true ?**
- The UI should be available on more platforms
  - True ? Which one ?
- What about Globus 3.0 integration ?
- Probable integration of PPDG/iVDGL tools/features





# Many Thanks

- To a lot of people in CCIN2P3, LAL/Orsay, SLAC/SCS, GridPP/UK, and more ...
  - Nadia, Fabio, Philippe, Dominique, Sophie ...
  - Cal, Serge, Christian, Michel, René ...
  - Adil, Ed, Karl, John, Richard ...
  - David S., David C., Rod ...
- For both their technical help, and encouraging support

# Request List (Wishlist ?)

- **[EDG]** Awareness of WNs located inside the IFZ
  - require transfers between RB and WN to be split in 2 steps:
    - RB  $\rightleftarrows$  CE, then CE  $\rightleftarrows$  WN (and vice-versa)
- **[Globus]** Gatekeepers running with NO AFS token
  - requires the possibility to relocate gass-cache into an NFS area
  - and also to relocate all temporary files into this gass-cache
- **[Globus]** Possibility to relocate the gass-cache area with a variable set at gatekeeper config level
  - seems to be forbidden right now
- **[EDG]** Possibility to set a `(EDG-) GLOBUS-DEPLOY-PATH` variable at config level to relocate the UI stuff
  - existed previously in Globus, missing in 2.0 ☹
  - missing in EDG 1.1.4 at least



## Requests (2)

- **[EDG]** Possibility of relocating default working area
  - thru a site-wide config variable, at sysadmin level
- **[EDG]** Possibility of relocating the user working area
  - thru a JDL directive at user level
- **[EDG]** Possibility of avoiding the LSF mail
  - thru a JDL directive at user level
  - at sysadmin level
- **[EDG]** Problem with use of Python in the UI
  - possibility to set a config variable pointing towards the local stuff



## Requests (3)

- [EDG] Availability over RH 7.2
- [EDG] Stability issue for RB daemons
  - lack of monitoring ?
    - is this still true in EDG-1.2 ?
  - they should be auto-restarted when failed/dead
- [EDG] Automated job output retrieval
  - Implement/improve direct delivery on user's node
- [EDG] Availability of the UI over several platforms
- [EDG] Avoid the `/opt` pointers for the UI install