



SuperB DIRAC report



- SuperB DIRAC works well
- SimulationProductionSystem works well
 - All code under SimulationProductionSystem until further notice
- Dummy Service successfully tested
- Dummy Agent successfully tested
 - I added a page on DIRAC official guide
 - <http://diracgrid.org/files/docs/DeveloperGuide/AddingNewComponents/DevelopingAgents/index.html>
- Porting SBKService (mylosz's code) in SuperB DIRAC
 - Work in progress
 - **See next slides**



Porting SBKService in SuperBDIRAC



- I'm working on bbrbuild01
 - SuperBDIRAC v0-pre10
- Good news
 - I can edit my code online, just need to restart SimulationProduction SBKService via `dirac-admin-sysadmin-cli` to see changes in log
- Created SuperBDIRAC.SimulationProduction.Client section
- SBKService is a service (follows DIRAC conventions)
- Support code for SBKService placed under Client folder
- Replicated Mylosz's folders' structure
 - Client.sa.manager... etc. etc.
- Continue on next slide....



Porting SBKService in SuperB DIRAC



- **Troubles executing SBKService as DIRAC Service**
 - **Failures in import chain**
 - **Some modification in Mylosz's code to follow DIRAC conventions**
 - **Solving in progress**
- **To install on DIRAC server (bbrbuild01)**
 - SQLAlchemy – done
 - Psycopg2 – done
 - PostgreSQL – done
 - pg_config – done



Next meeting reminder



- SuperBDIRAC meeting
 - 6 December 2012 on Skype
- SuperB Collaboration Meeting @ Frascati
 - <https://agenda.infn.it/conferenceDisplay.py?ovw=True&confId=4442>
 - 10-14 December 2012
 - Participants:
 - **Armando - confirmed**
 - Bruno (quite sure)
 - **Cristian - confirmed**
 - **Giacinto - confirmed**
 - **Marcin - confirmed**
 - Mylosz (to confirm)
 - Rafal (?)
 - Please, update participation info in order to plan an hour face-to-face meeting to plan next steps in SuperBDIRAC (more precisely, the Simulation Production porting under DIRAC)
 - especially to understand what of DIRAC's features we could use in a profitable way (Trasformation System ? Workflow ?)



Discussion time



- Proposal ?
- Suggestion ?
- Something not clear ?



BACKUP SLIDES



SuperBDIRAC module installation status



- `bbrbuild01.cr.cnaf.infn.it`
 - Production
- `sb-serv04.cr.cnaf.infn.it`
 - Development
- `gridtest-05.ba.infn.it`
 - Installation procedure test



SuperB DIRAC shared development



- /opt/dirac/pro
 - SuperB DIRAC.main
 - SuperB DIRAC.bruno
 - SuperB DIRAC.mylosz
 - SuperB DIRAC.marcin
 - SuperB DIRAC.rafal
- Symbolic link
 - SuperB DIRAC → SuperB DIRAC.main (for example)
- Working directly on sb-serv04
- Update SVN directly from sb-serv04



SVN usage



- Now, only sb-serv04 should be used for development purposes
 - If needed, also bbrbuild01 can be used for same task
 - especially when new DIRAC head node will be available
- `svn co https://sbrepo.pd.infn.it:8910/WebUi/trunk/DIRAC`
- `svn commit -m "Update from git release"`
- `svn push`
- SVN now available on sb-serv04 (thanks to Armando)



SVN usage improvements



- <https://sbrepo.pd.infn.it:8910/WebUi/trunk/DIRAC/SuperBDIRAC>
- Introduce trunk, tags and branches (one branch foreach developer
- .../SuperBDIRAC/trunk
- .../SuperBDIRAC/tags/<tagged_release>
- .../SuperBDIRAC/branches/<bruno|mylosz|rafal|marcin>
- Sync (automatically or via script) personal branches with personal SuperBDIRAC version
 - See previous slide



SuperBDIRAC status



- SuperBDIRAC installed and usable
 - New system defined: SimulationProductionSystem
 - Dummy service tested
- Now we can place all our code in proper places under SuperBDIRAC
- VM DIRAC properly installed
 - Together to SuperBDIRAC (SuperBDIRAC doesn't depends by VM DIRAC)
 - If occi or amazon cloud resources available, we can test job submssion to cloud



SimulationProductionSystem



- Easy to configure, transparent usage
- We can start using Mylosz code to create a working service (named SBKAlchemy ?)
 - To return site's data
 - Info needed to create Site Management webpage



Task assignement proposal



- Bruno
 - Solve repo troubles
 - Complete Severus porting under DIRAC
- Mylosz
 - Prepare a working service
 - To return Site's list
- Marcin, Rafal
 - Webpage to diplay site's list (taking data from Mylosz's service)
 - Permission management:
 - The same page (site's list) but adding an “edit” button foreach site
 - Page should be availble only for sb_productionmanager members



Credits



- Armando Fella – Pisa
- Bruno Santeramo – Bari
- Christian De Santis - Rome
- Giacinto Donvito - Bari
- Marcin Chrzaszcz - Kracow
- Miłosz Zdybał - Kracow
- Rafał Grzymkowski - Kracow

Thanks to DIRAC developers, expecially to
Andrei Tsaregorodtsev, Federico Stagni, Matvey Sapunov,
Krzysztof Daniel Ciba, Ricardo Graciani, Adrian Casajus



Groups



Groups and Role in VO and DIRAC

Group/Role	Type	Description	VO share	DIRAC mapping
/superbvo.org	Standard User	Generic user group	0	sb_user
/superbvo.org/Role=ProductionManager	Production Manager	User role related to MC production management	20	sb_productionmanager
/superbvo.org/Role=SoftwareManager	Software Manager	VO Software administrator	0	sb_softwaremanager
/superbvo.org/Role=Analysis	Standard User	User role related to analysis task	0	sb_analysis
	DIRAC administrator	DIRAC administrator		dirac_admin
	DIRAC pilot job	Useful for DIRAC		sb_pilot



Collaborative tools update



- SVN: <https://sbrepo.pd.infn.it:8911/projects/WebUi/browser/trunk/DIRAC>
 - Uploaded SuperBDIRAC module
 - Uploaded Severus (modified to work with DIRAC)

- Severus/version24/modules/fileManager.py
 - Line 109
 - Original
 - *out = self.lcgUtil.lfcLs(path)*
 - Modified
 - *out = self.lcgUtils.lcgLs(path)*
 - Why?: Workaround because surl starting with lfn:/ supposed to be accessed from CNAF (so Severus drop initial lfn:/ from surl and try to access input file locally)



Modifying Severus for DIRAC



- DIRAC job wrapper install its own python environment
 - Python version 2.6.6
- Severus tested to work with Python 2.4
 - Severus perform an intial python version check
 - Version24 directory contains all components
- Proposal
 - Test modified Severus with Python 2.6
 - New code under version26 directory
 - Modify severus.py to enable Python 2.6 usage



MC production via SuperBDIRAC



- SBK5 queries to be prepared - via SQLAlchemy and/or REST interface
 - SELECT count(runnum) as tot FROM "<session_name>"_Job " WHERE (prod_series = '<prod_series>') AND (runnum BETWEEN "<minrun>" AND "<maxrun>") AND (status!='prepared');
 - SELECT * FROM "<session_name>"_Job " WHERE (prod_series = '<prod_series>') AND runnum = '<runnum>';
 - UPDATE "<session_name>"_Job SET grid_job_id = "<job_id>", status = 'submitted', ts_submitted = cast(now) as timestamp(0) WHERE prod_series = '<prod_series>' AND runnum = "<runnum>" AND status = 'prepared';
 - UPDATE "<session_name>"_Submission SET bulk_job_id || ('<site>' => '<bulk_job_id>') WHERE uid = '<uid>' AND ts_prepared = '<timestamp>';



MC production via SuperB DIRAC



- Issue: SuperB DIRAC/ProductionSystem need to interact with SBK5
 - N.B. Now Severus can be properly launched only from CNAF (?)
 - SBK5 psql-like access only from CNAF
 - SBK5 access outside CNAF only via REST interface
 - Solution proposal:
 - Create a Service (temporary named SBK5Service) into SuperB DIRAC to cope (all?) SBK5 queries
 - Each site equipped with a DIRAC server and SBK5Service must be able to directly query SBK5
 - Enhance SBK5 REST interface by adding some views to simply perform needed queries from every DIRAC Server Instance
 - No changes to SBK5 access policy



SBK operations



- DIRAC webportal must replicate all WebUI features
- DIRAC/WebUI operations
 - Production creation → write
 - Production data retrieve → read
 - Request creation → write
 - Request data retrieve → read
 - Bulk job creation → write
 - Bulk job submission → read data for Severus
- Severus job wrapper
 - Job status report → write
 - Job Log registration → write



DIRAC API



- DIRAC offers a powerful set of API
 - We need a good quality documentation to speed up development
- API documentation
 - <http://diracgrid.org/files/docs/CodeDocumentation/API/index.html>
 - Epydoc generated – updated – maintained by developers



Useful links



- **DIRAC official site**
 - <http://diracgrid.org/>
- **DIRAC official forum**
 - <https://groups.google.com/forum/?hl=en&fromgroups#!forum/diracgrid-forum>
- **DIRAC repository**
 - <https://github.com/DIRACGrid>
- **DIRAC API documentation**
 - <http://diracgrid.org/files/docs/CodeDocumentation/API/index.html>
- **DIRAC testbed for SuperB wiki page**
 - http://mailman.fe.infn.it/superbwiki/index.php/Distributed_Computing/Dirac_testbed
- **DIRAC testbed webportals:**
 - bbrbuild01.cr.cnaf.infn.it
 - <https://bbrbuild01.cr.cnaf.infn.it:8443/DIRAC/>
 - sb-serv04.cr.cnaf.infn.it
 - <https://sb-serv04.cr.cnaf.infn.it:8443/DIRAC/>
- **Production WebUI**
 - https://bbr-serv09.cr.cnaf.infn.it:8443/~webui_sbk5/
- **SuperB resources (google Doc)**
 - <https://docs.google.com/spreadsheet/ccc?key=0AsjxRpEZ2zEIIdHU2cjh4YTdrajdYdlNmajdBVGZWLWc#gid=0>
- **SuperB Production Shifter Guide wiki page**
 - http://mailman.fe.infn.it/superbwiki/index.php/Distributed_Computing/Production_Shift_Guide

WebUI workflow



