



DIRAC testbed

SuperBDIRAC development

Bruno Santeramo

27/09/2012



Summary



- Simulation production use case
 - Using Severus via DIRAC
 - Porting WebUI functionalities into DIRAC webportal
 - SuperBDIRAC module
 - Next steps
- Summary from Pisa meeting
- DIRAC status overview
- Upgrade plan
- Milestone proposal
- Next meeting reminder



Simulation production use case



- **Goal: manage all aspects related to Simulation Production (FastSim and FullSim) via DIRAC**
 - User management (role and permissions)
 - Site Management per Session
 - Production creation and monitoring
 - Requests creation and monitoring
 - Bunch jobs submission and monitoring

DIRAC need to interact with Bookkeeping Database (SBK5)



Using Severus via DIRAC



- Severus: python wrapper to perform SuperB MonteCarlo productions
 - Severus take care about stagein, stageout, Simulation program execution, SBK5 update
- Severus must be launched via DIRAC



Launching Severus via DIRAC



- Status: proof of concept
- mc_production.py
 - A python script powered by DIRAC API
 - Get php file generated by WebUI as input
 - Parse php and severus conf file
 - Retrieve necessary parameters
 - Launch DIRAC jobs using a modified version of Severus wrapper

- Parameters to create production:
 - Production: `prod_series`
 - Session: `session_name`
 - Sites: `submission_site`
 - Start runnum: `Minrunnum`
 - End runnum: `Maxrunnum`
 - Number of events: `nevents`
 - Mode: `[test|normal]`
 - User: `uid`
 - Timestamp: `ts_prepared`
 - Analysys – Generator – Geometry - Background

- Test results:
 - Severus works properly via DIRAC
 - SBK5 updated
 - Submission properly visible from WebUI
 - Stagein and stageout via lcg commands
 - Stagein works properly
 - Stageout works properly
 - LFC properly updated
 - Tested jobs running up to 80.000 events
 - MC production must be performed by sb_productionmanager users
 - sb_productionmanager DIRAC group maps user with VOMSRole = ProductionManager



Issues using Severus



- Issues:
 - Submission must be done via DIRAC webportal
 - Task assignement to SuperBDIRAC/ProductionSystem development
 - WebUI check selected runnumbers before job submission to avoid duplicate works
 - Need a query to SBK5
 - A Python script used by WebUI updating SBK5 for each job
 - Status = submitted
 - ts_submitted = timestamp
 - Need a query SBK5



Severus test next steps



- Future steps proposal:
 - Solve issues
 - Test DIRAC and LFC
 - Perform Stagein and Stageout via DIRAC on LFC



Porting WebUI functionalities into DIRAC webportal



- Session → Site Management
 - Retrieve site's info from SBK5
 - SEs and CEs data configured in DIRAC
 - Resource management (enable and support site per Request) to be performed via DIRAC
- FastStim → Production
 - Production list:
 - Display all data; add `soft_lfn`, `exe_path`, `working_dir` from `SBK5.fastsim_soft_ref` table
 - Create Production:
 - Enable functionality limited to production manager users



Porting WebUI functionalities into DIRAC webportal



- FastSim → Production Requests
 - _ Replicate Request monitoring functionalities
 - _ Create Request supporting templates, setting and output type definition
- FastSim → Expert Init
 - _ Replicate all functionalities
 - _ Page restricted to enabled users
 - _ Site selection linked to Session Management
 - _ Production submission via DIRAC webportal
- FastSim → Shift Init
 - _ Replicate all functionalities
 - _ Requests monitoring functionalities
 - _ Available sites (data from SBK5 via SQLAlchemy)
 - _ Support for production and test jobs submission
- FastSim → Job Monitor
 - _ Monitoring criteria (data from SBK5 via SQLAlchemy)
 - _ Detailed jobs monitoring
- FastSim → Submission Monitor
 - _ Monitoring criteria (data from SBK5 via SQLAlchemy)
 - _ Detailed jobs monitoring



SuperB DIRAC module



- DIRAC need customization to fulfil SuperB requirements
- DIRAC module is a container for VO specific code
- SuperB DIRAC ↔ DIRAC module for SuperB
 - _ Interaction with BookKeeping DB (SBK5)
 - Testing SQLAlchemy
 - _ Simulation Production use case
 - Porting Severus under DIRAC
 - _ Extending DIRAC Webportal to perform Production related actions
 - Porting WebUI functionalities under SuperB DIRAC
 - _ Production Workflow definition
 - Production System must be implemented
 - _ DIRAC Transformation System is a “framework” for workflow and repetitive job/data management tasks
 - _ Interesting paper on Transformation System (LHCb experience)
<http://iopscience.iop.org/1742-6596/368/1/012010>



Next steps



- Working version of SuperBDIRAC
- Workload monitoring for Simulation Production
- Complete job submission stack
 - Ganga jobs submission via DIRAC backend
- Complete Mass Data Transfer test
 - Transfer an entire dataset via FTS



Summary from Pisa Meeting



- <http://agenda.infn.it/conferenceDisplay.py?confId=4437>
- Production must move to distributed environment asap
 - Distributed job submission
 - Distributed data storage
- Milestone:
 - Working SuperB DIRAC module within December 2012
 - ?? Live demonstration at December SuperB meeting @ Frascati ??



DIRAC status overview



- Stable version: v6r5p3
 - Bug fix
 - Included File Catalog browser in webportal
 - Not production ready
 - Should be really useful – need testing
- Development version:
 - Enhancements
 - Bug fixes
 - API polishing



Upgrade plan



- `bbrbuild01.cr.cnaf.infn.it`
 - `v6r5` $\rightarrow$ `v6r5p3` (upgrade)
 - Jobs and data management test
- `sb-serv04.cr.cnaf.infn.it`
 - `v633p17` $\rightarrow$ `v6r5p3` (from scratch)
 - SBK5-DIRAC testing
- `gridtest-05.ba.infn.it`
 - `v6r5p1` $\rightarrow$ `v6r5p3` (from scratch)
 - SuperBDIRAC installation test



Milestone proposal



- First October week
 - _ SuperB DIRAC installation
 - _ SBK5 insert query via SQLAlchemy
 - Sbkprodrhas only read permission
 - Write and update permission necessary to replicate Site Management page
- October
 - _ Transformation System testing
 - Should be useful for workflow and Massive Data Transfers
 - _ Site Management porting under SuperB DIRAC
 - _ First FastSim job submission via SuperB DIRAC
 - Prototype (Expert or Shift) Init page
- November
 - _ Manage permission rules for Operations via SuperB DIRAC web interface
 - _ Production Requests page
 - _ Monitoring functionalities in SuperB DIRAC
- December
 - _ SuperB DIRAC able to perform complete MC production stack



Next meeting reminder



- Third DIRAC User Meeting
 - 29-31 October 2012
 - <http://diracgrid.org/?p=183>
 - Bruno Santeramo should participate
 - Someone else interested ?
- SuperB Collaboration Meeting @ Frascati
 - December 2012



BACKUP SLIDES



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, especially to
Andrei Tsaregorodtsev, Federico Stagni, Matvey Sapunov,
Krzysztof Daniel Ciba, Ricardo Graciani



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)

- 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

- 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

- 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

