

CCR workshop – Stato e prospettive del calcolo scientifico  
Legnaro, 16-18 Febbraio 2011

# Problematiche di sottomissione job (per gli esperimenti di LHC)

David Rebatto – INFN Milano

- Job submission issues
  - interface consistency and transparency
  - security
- Job management issues
  - interface consistency and transparency
  - accuracy
  - failure management
- Solutions adopted by LHC experiments
- Future plans

# Job submission issues (I)

- Hiding technicalities
  - LHC experiments are huge communities, involving several institutes in many different countries
  - Their computing grids are heterogeneous in terms of middleware stacks
    - Their users need to submit jobs via gLite WMS, ARC, Condor, CREAM, or directly to LRMS
  - This complexity must be hidden from the users
    - They must be provided with a consistent interface, no matter where their jobs are going to run
- **Ideally, the whole grid should act as a big batch system**

# Job submission issues (II)

- Security (AAA)
  - Authentication
    - By means of X.509 credentials
    - The VO must confirm the affiliation
  - Authorization
    - The VO may grant specific groups and roles to a user
    - The services may grant specific level of access according to the groups and roles
  - Accounting
    - Every action performed on the grid should be ascribed to a person, with non-repudiation

- Security (continued)
  - Consistency
    - Each service has to implement AAA on top of shared libraries (currently LCAS, LCMAPS)
    - Inconsistency in the local mapping among different services on the same node are likely to occur
      - e.g. GridFTP running on StoRM and CREAM nodes requires a different configuration file...
  - A centralized AAA service is needed
    - gLite developed ARGUS

# Job management issues (I)

- Monitoring jobs
  - Transparency wrt middleware
    - Every middleware stack provides its own tools to check the status of the submitted jobs, but users should be exposed to a common interface
  - Accuracy
    - The jobs' status are evaluated by event driven state machines
    - A failure in the delivery of some events can lead to wrong status reports, and the jobs appear to be stuck in some intermediate status



# Job management issues (II)

- Managing failures
  - A failed job needs to be resubmitted
    - it has to go through a site's queue again
  - A stuck job is even trickier, because before being resubmitted it needs to
    - be identified as “stuck” (some heuristic needed)
    - be removed from the system (and usually the regular cancellation commands don't work)

# Job management issues (III)

- Managing jobs' execution order
  - In a generic batch system
    - jobs have a priority, and they are executed accordingly
    - the priorities can be altered while the job is queued
  - This is usually not possible with grid middleware
    - the job execution order cannot be predicted
    - the user cannot “boost” a job to have it executed before the others (even his own ones)
    - common use case: 99% completed tasks with few jobs lagging behind
  - **Ideally, the whole grid should act as a big batch system (again)**



# Job submission frameworks

- To cope with some or all the previous issues, the experiments developed their own job submission frameworks
  - AliEn, DIRAC, CRAB, Ganga, Panda, etc.
  - Nobody writes JDL, Condor submit files, etc.
- Two different approaches
  - Pure **pilot based** frameworks
    - AliEn, DIRAC, Panda
  - **Direct submission capable** frameworks
    - CRAB, Ganga, Production Agent

# Direct submission frameworks

- Ganga
  - A Python shell with a set of predefined classes to create, submit and manage jobs
  - Plug-in's for gLite-WMS, local schedulers, DIRAC, Panda
- CRAB and Production Agent
  - Both use BossLite as interface to the job submission system
  - Plug-in's for gLite-WMS, glidein-WMS, Condor-G, ARC and local schedulers

- DIRAC WMS (LHCb)
  - The DIRAC WMS pioneered the use of a central task queue and generic pilot agents
  - Pilot submission through WLCG WMS as well as direct submission to CREAM are supported
  - A central web portal offers unique entry point for all user and production manager interactions
    - Production requests
    - Production monitoring
    - User job monitoring
    - Accounting

# Pilot based frameworks

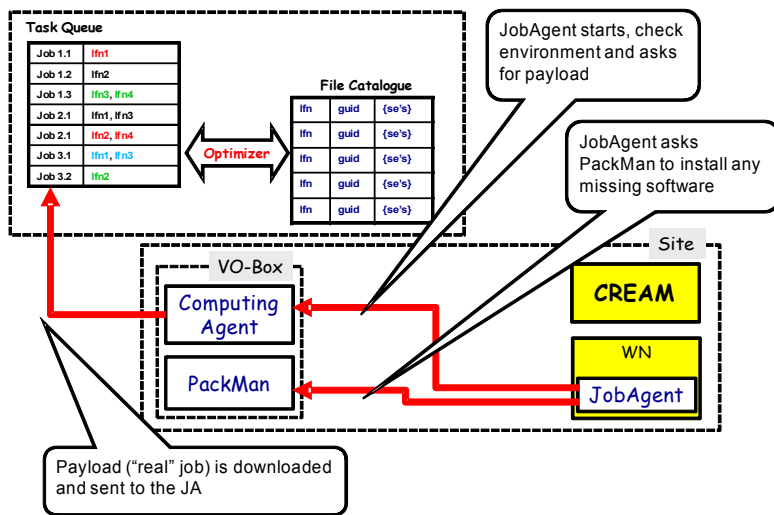
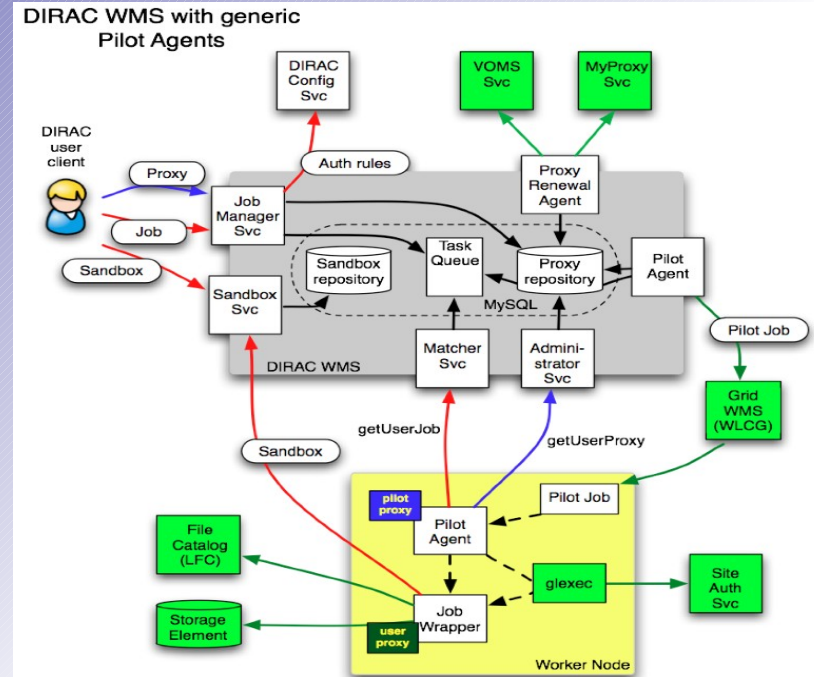
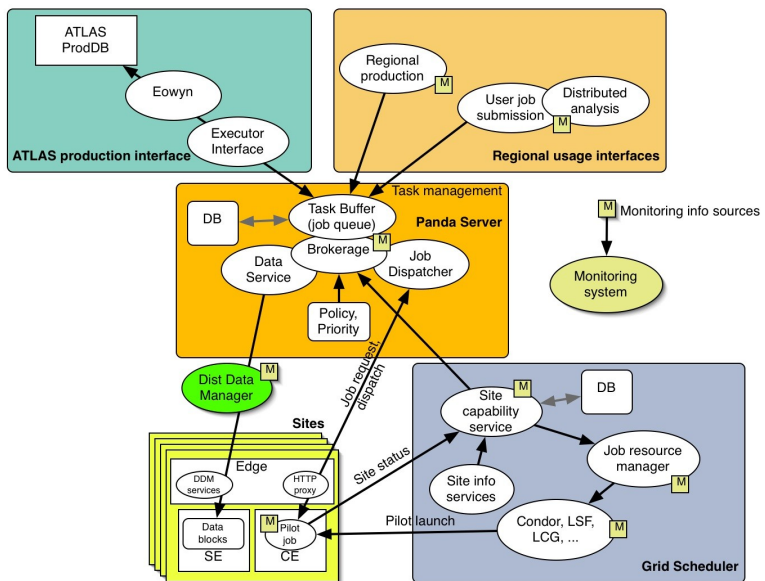
- AliEn
  - JobAgents are submitted directly to CREAM, ARC and OSG from the site's local VO BOX
  - The same Computing Agent contacts the central service and dispatch jobs to the JobAgents

# Pilot based frameworks

- Panda (ATLAS)
  - Pilots are submitted by lightweight pilot factories via CondorG, gLite WMS or directly to the LRMS
  - Pilots are “fire and forget”: the bookkeeping is provided by the central server via direct communication with the running job



# Pilot based frameworks



“Convergent evolution”



# To pilot or not to pilot?

- Hot and long standing topic with many supporters on both sides
- Main arguments by **users** and **site admin's**:
  - Pros
    - Pilots are disposable
    - More robust against middleware failures
    - The centralized DB of jobs acts as big batch system's queue, under full control of the VO
    - Highest priority jobs are granted to run first
  - Cons
    - Accounting and security issues
    - Any local “intelligent scheduling” disrupted (all the jobs look the same)
    - Possible waste of resources
    - Single point of failure
    - Scalability

# Pilot jobs identity issue

- With pilots all the jobs run under the same identity
  - who is liable for a potential abuse?
  - how can a site ban a single user?
- A solution using GLExec and ARGUS under test now
  - Testbed at CNAF
  - GLExec must be deployed on all the WNs

# Sites survey about MUPJ

- Different sites have different feelings about Multi User Pilot Jobs (MUPJ)
- Accepted, but many *caveats*
  - The most common one is that the pilot submitter or his VO must take liability for any abuse

site/region	MUPJ support	general	general	general
ASAC	no	no	no	no
BE-Brno-DTL	no	no	no	no
CE-IL-Torino	no	no	no	no
CE-IL-SP	no	no	no	no
CE-IL	depends: responsibility must be reported	no	no	no
CE-IL-2	depends: submitter always change VO name/username responsibility	no	no	no
DE-DESY-ITVM	no	no	no	no
DE-DESY	no	no	no	no
DE-DESY-IT	no	no	no	no
DE-DESY-IT2	no	no	no	no
DE-DESY-IT3	no	no	no	no
DE-DESY-IT4	no	no	no	no
DE-DESY-IT5	no	no	no	no
DE-DESY-IT6	no	no	no	no
DE-DESY-IT7	no	no	no	no
DE-DESY-IT8	no	no	no	no
DE-DESY-IT9	no	no	no	no
DE-DESY-IT10	no	no	no	no
DE-DESY-IT11	no	no	no	no
DE-DESY-IT12	no	no	no	no
DE-DESY-IT13	no	no	no	no
DE-DESY-IT14	no	no	no	no
DE-DESY-IT15	no	no	no	no
DE-DESY-IT16	no	no	no	no
DE-DESY-IT17	no	no	no	no
DE-DESY-IT18	no	no	no	no
DE-DESY-IT19	no	no	no	no
DE-DESY-IT20	no	no	no	no
DE-DESY-IT21	no	no	no	no
DE-DESY-IT22	no	no	no	no
DE-DESY-IT23	no	no	no	no
DE-DESY-IT24	no	no	no	no
DE-DESY-IT25	no	no	no	no
DE-DESY-IT26	no	no	no	no
DE-DESY-IT27	no	no	no	no
DE-DESY-IT28	no	no	no	no
DE-DESY-IT29	no	no	no	no
DE-DESY-IT30	no	no	no	no
DE-DESY-IT31	no	no	no	no
DE-DESY-IT32	no	no	no	no
DE-DESY-IT33	no	no	no	no
DE-DESY-IT34	no	no	no	no
DE-DESY-IT35	no	no	no	no
DE-DESY-IT36	no	no	no	no
DE-DESY-IT37	no	no	no	no
DE-DESY-IT38	no	no	no	no
DE-DESY-IT39	no	no	no	no
DE-DESY-IT40	no	no	no	no
DE-DESY-IT41	no	no	no	no
DE-DESY-IT42	no	no	no	no
DE-DESY-IT43	no	no	no	no
DE-DESY-IT44	no	no	no	no
DE-DESY-IT45	no	no	no	no
DE-DESY-IT46	no	no	no	no
DE-DESY-IT47	no	no	no	no
DE-DESY-IT48	no	no	no	no
DE-DESY-IT49	no	no	no	no
DE-DESY-IT50	no	no	no	no
DE-DESY-IT51	no	no	no	no
DE-DESY-IT52	no	no	no	no
DE-DESY-IT53	no	no	no	no
DE-DESY-IT54	no	no	no	no
DE-DESY-IT55	no	no	no	no
DE-DESY-IT56	no	no	no	no
DE-DESY-IT57	no	no	no	no
DE-DESY-IT58	no	no	no	no
DE-DESY-IT59	no	no	no	no
DE-DESY-IT60	no	no	no	no
DE-DESY-IT61	no	no	no	no
DE-DESY-IT62	no	no	no	no
DE-DESY-IT63	no	no	no	no
DE-DESY-IT64	no	no	no	no
DE-DESY-IT65	no	no	no	no
DE-DESY-IT66	no	no	no	no
DE-DESY-IT67	no	no	no	no
DE-DESY-IT68	no	no	no	no
DE-DESY-IT69	no	no	no	no
DE-DESY-IT70	no	no	no	no
DE-DESY-IT71	no	no	no	no
DE-DESY-IT72	no	no	no	no
DE-DESY-IT73	no	no	no	no
DE-DESY-IT74	no	no	no	no
DE-DESY-IT75	no	no	no	no
DE-DESY-IT76	no	no	no	no
DE-DESY-IT77	no	no	no	no
DE-DESY-IT78	no	no	no	no
DE-DESY-IT79	no	no	no	no
DE-DESY-IT80	no	no	no	no
DE-DESY-IT81	no	no	no	no
DE-DESY-IT82	no	no	no	no
DE-DESY-IT83	no	no	no	no
DE-DESY-IT84	no	no	no	no
DE-DESY-IT85	no	no	no	no
DE-DESY-IT86	no	no	no	no
DE-DESY-IT87	no	no	no	no
DE-DESY-IT88	no	no	no	no
DE-DESY-IT89	no	no	no	no
DE-DESY-IT90	no	no	no	no
DE-DESY-IT91	no	no	no	no
DE-DESY-IT92	no	no	no	no
DE-DESY-IT93	no	no	no	no
DE-DESY-IT94	no	no	no	no
DE-DESY-IT95	no	no	no	no
DE-DESY-IT96	no	no	no	no
DE-DESY-IT97	no	no	no	no
DE-DESY-IT98	no	no	no	no
DE-DESY-IT99	no	no	no	no
DE-DESY-IT100	no	no	no	no

Survey conducted by Maarten Litmath - 17/02/2010

# Pilot jobs LRMS scheduling issue

- Different classes of job need different resources
  - MC production & reco
    - Low I/O, Max CPU efficiency
  - Analysis train
    - Optimized I/O, managed, scheduled
  - User jobs
    - Lowest CPU efficiency, unpredictable time and RAM
- Panda submits pilots using different roles
  - Sites can direct different job to different queues
  - ...but it reduces flexibility in Panda's scheduling

# The future of job submission

- Non-pilot submission is going to disappear?
  - GANGA is used 90% as a front-end toward DIRAC and PANDA
  - In CRAB, the new WMAgent is optimized for working with glidein-WMS. It is probable that this will become the preferred 'scheduler'
  - ATLAS still uses WMS submission for the software installation jobs, but is considering to adopt CERNVM-FS



- CREAM is welcomed by all the experiments (and it's the only option on SL5...)
  - Allows both direct and middleware mediated submission
  - Will offer whole-nodes allocation semantics
    - Already under testing for PBS and LSF
    - Useful for those experiments which can exploit multi-core machines
  - ALICE deployed it in production long time ago, and reports remarkable stability



- All the experiments are increasing CREAM usage and plan to phase out the LCG-CE in the next few weeks
  - ALICE is already using CREAM in all WLCG sites
  - CMS will enable CREAM direct submission in the WMAgent, either using the web service interface or embedding ICE
  - ATLAS already uses the pilot factories to submit to CREAM both via WMS and Condor
  - LHCb has several sites (e.g. CNAF Tier-2) already working with direct CREAM submission

- Thanks to
  - Stefano Bagnasco (ALICE)
  - Claudio Grandi (CMS)
  - Vincenzo Vagnoni (LHCb)