Experience in integrating PanDA components in CMS

Daniele Spiga
On behalf of the common analysis framework team

Outline

- History and status
- Schema of the implemented testbed and components description
- Experience by component
- Summary

History of the project

The activity started on March 2012 having three major milestones

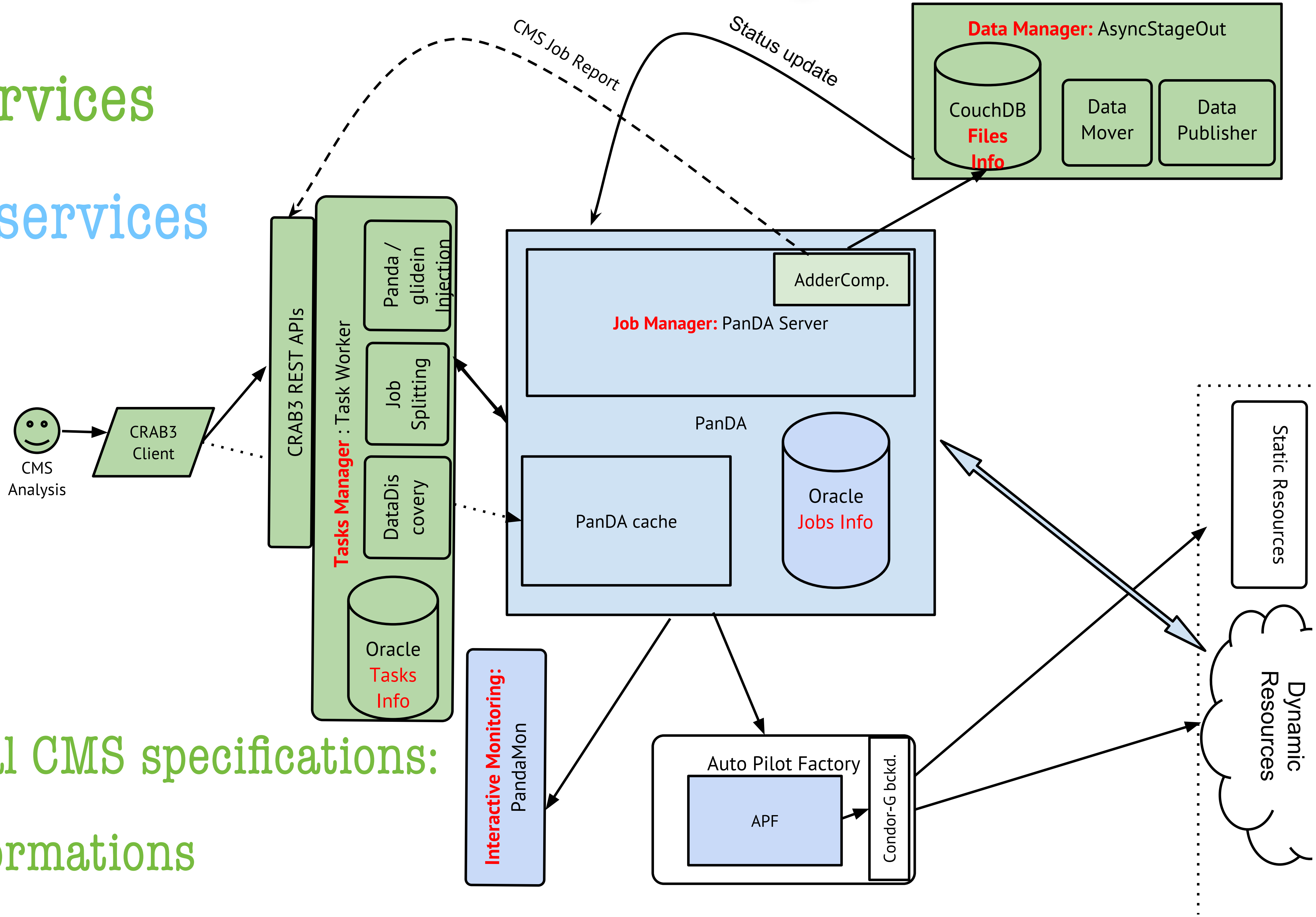
- May 2012: Feasibility study
  - Delivered document
- Dec. 2012: Prof-of-concept prototype development and testing
  - Presented results FNAL WorkShop
- Aug. 2013: Consolidation in a testbed

Where we are today

- Development phase ended, delivering a functioning system end of July
- CMS Testbed upgrade and validation happened in August
  - 11 beta users including Integration/Ops/Developers
- Currently starting functional test together with CMS Integration and Ops teams

Schema of the system

- CMS Services
- PanDA services



- Additional CMS specifications:
  - Transformations
  - Plugins for Pilot FW

CMS Components: roles

CRAB client / server

- CLI interface for task/job management and REST API interfacing the rest of the system
  - Async communication with TaskManager through DB

Task Manager

- Implements the concept of CMS Task (as group of jobDefIDs belonging to the same jobSetID)
  - grabs user requests from DB, interacts with CMS Data Management, splits jobs and injects specs into panda
  - Handles kill and resubmission requests

AsyncStageOut

- Handles user produced files
  - pushed by PanDA, transfers files thought FTS and notifies PanDA for job status update
  - Interacts with CMS Data Management (write mode)

Experience with PanDA server

Submitting 'hello word' (prun & runGen) is easy with ~no extra effort

Submitting a real CMS analysis job wouldn't be possible w/o Tadashi support and fixes

- job/file spec configuration, schedconfig configuration, skip server/dq2 interaction, ASO PanDA messaging and error reporting...
  - I still feel like we need to improve the spec configuration (e.g. do we abuse of jobParams)
- nota: PanDA DB is untouched

Server Documentation = (Tadashi + Code)

- Available for many F2F discussions, extremely responsive and yes, also patient :)
  - ✳ great experience of collaboration

Changes and fixes

- 90% done by Tadashi, with fast turnaround
  - ✳ details mostly 'hidden' to outsider but things work
- 10% Hassen Riahi fully developed the CMS plugin for the Adder component

Experience with APF

APF installation and configuration didn't require any expert

- Fernando did it the first time and wrote a small how-to
- Some initial effort required for debugging site (not APF dependent)
- We did a patch to allow round robin usage of a pool of certificates for pilots
  - cms need it to compete at site

Experience with Pilot Framework

Probably the most tricky part of the stack to make CMS compliant

- The initial version was almost ATLAS specific
  - Build a customized version on top of it insulating experiment specific methods
    - stage out / software checks / LFC / job report / LFN vs SURL / CMS error codes handling
  - Evolving now towards a version ATLAS free, with experiment plugins
  - Issue such as the error message propagation still work in progress

Here Paul is the documentation :)

- Extremely responsive, supporting all the needed changes and providing guidance for customizations
- Some patches developed by us and blessed by Paul
  - much appreciated approach!

Experience with AGIS

Introduced only during the last phase and just to configure APF queues

- Not trivial w/o guidance
  - After the Alessandro hands on everything went smoothly
    - 59 CMS Sites registered
- Schedconfig part done by hand and again, it is all to be discussed
  - It is almost empty because CMS has the concept of site catalogs (TFC)...

What is missing

and thus not reported here

- PanDA rebrokerage for CMS
  - Not tested, not expect to work for free
- gLExec integration not yet finished
- Scale tests
  - We have a plan for the HammerCloud integration with CAF but not yet done

Summary

We succeed in providing a testbed based on crab3-panda to run CMS workflows on it

- Room for improvements
- No scale test done yet, would be interesting to learn more

No show stopper encountered during the process

- Obviously many things are ATLAS specific and sometime have been enforced
  - job/file spec, schedconfig, cloud/queue/resources...

Lack of documentation doesn't help

- Probably impacts on the support required