Analisi In CMS

Stefano Lacaprara

Department of Physics
INFN & University of Padova

II Riunione Nazionale CMS-INFN,
Bologna, 20 gennaio 2005

Outline

Outline

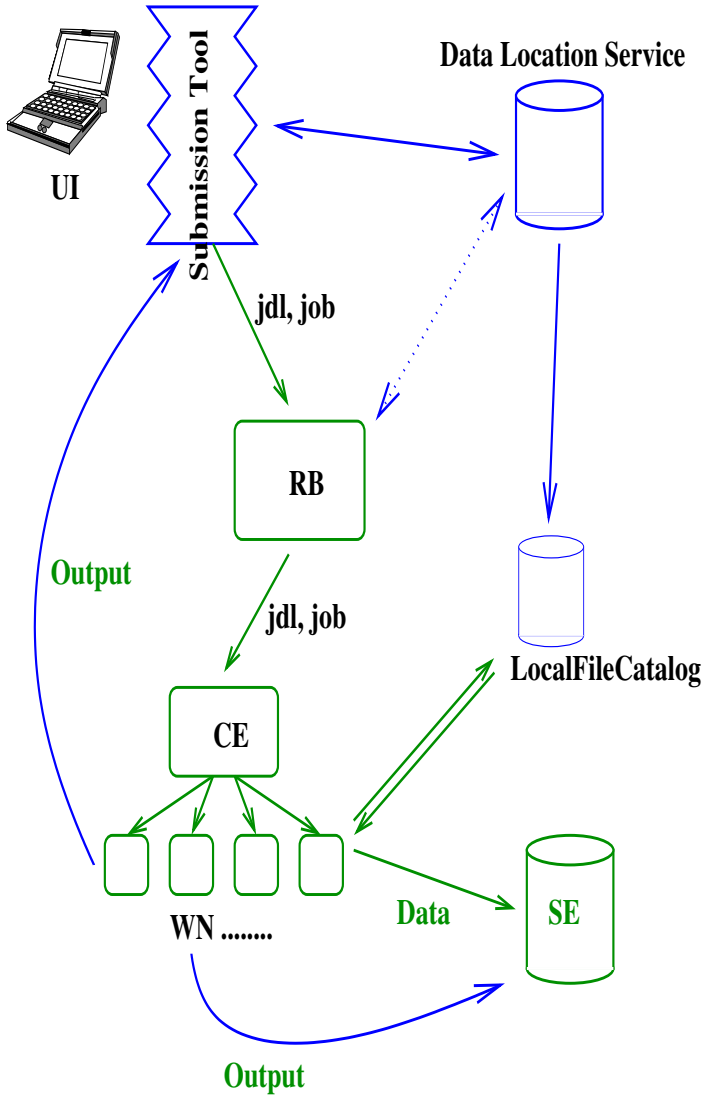
How To Do Analysis?

- CMS model foresee Distributed, Tier-based data deployment
- From computing point of view, the most complex issue when doing remote analysis is Locating and Accessing Data
- How to access Data?
  - Batch: large input data
  - Interactive: small input data (typically local)
  - Root based, interactive analysis in Luca's Talk
- Will concentrate on Batch access
- To access data first need to *locate data*
- General issue about Catalogs and Publication
- Then get access to resources
- End-user oriented tools needed for both

Access to Resources

- Batch analysis will be done near to Data
- Data will be in a Tier-1 (not necessarily the national one) where, in general, direct access will not be provided
- So access will be provide via GRID
- To access remote resources user needs an User Interface (UI)
- To allow remote access to local data
  - deploy GRID middleware wherever data is
  - In order to host data visible via Grid, mandatory to have a complete Grid enabled farm (Computing Element, Working Nodes and Storage Element)

Actual Workflow



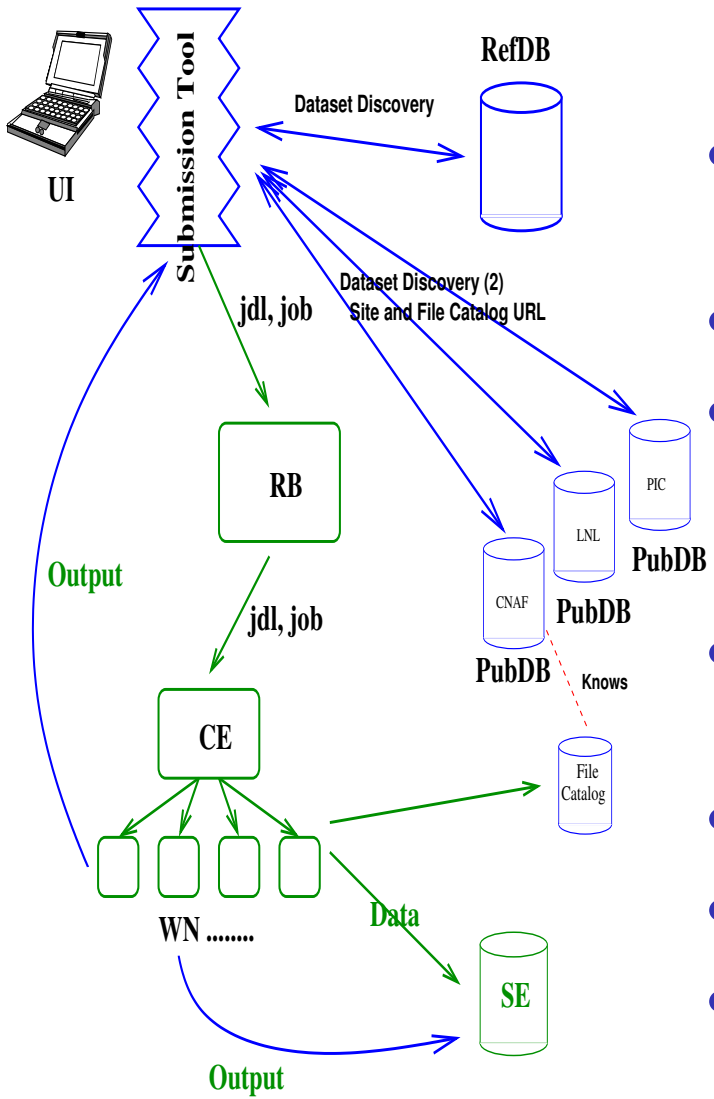
- User develops code on local UI
- Use CMS tool for Grid submission
  - Input is Data to be accessed, code
  - Job preparation (private code, splitting, submission, . . .)
  - Create wrapper job to be submitted to Grid
- RB (or tool) uses Data Location Service to find good Site
- Job arrives to Working Node and runs against local Data using a local FileCatalog
- Output is retrieved or stored on Storage Element

Outline

Data Publication: catalogs

- Data Transfer service (PhedEx) in Daniele's talk
- To access data for Analysis need to know where data is
  - Need a set of catalogs (aka DB, ...) in order to locate and access data
  - These sources of information should be few and with well define responsibilities and scopes
  - Today situation in CMS (IMO) is not very clean, lacking an overall view
  - Key issue to be defined
- Will present the current status and a proposal for future development

Present



- Dataset Discovery: RefDB (CERN) and PubDB (one per site)
- RefDB knows which PubDBs publishing data
- Each PubDBs publish site (CE, SE)
- Local PubDBs knows about Dataset details (# events, ...) and URL of local FileCatalog(s)
- Submission tool query RefDB & eligible PubDBs
- find Dataset location and tell the RB
- RB ship job to CE
- From WN, LocalFileCatalog (xml or mysql) knows file location (used by COBRA)

INFN contribution

- PhedEx (Bologna)
- Validation Tool (Bari)
  - Goal: be sure that if user job crashes is user's responsibility, not data's
  - Test data access with "standard" executable accessing all data (Hits, Digis, DST, etc ...)
  - Automatic publication on PubDB (Bari)
  - Under integration with Data Movement Tool (PhedEx)
- PubDB development (Bologna)
  - Adapt initial design (CERN) to needs of remote analysis
  - Development, test and deployment
  - Room for DB experts
- PubDB deployment and test at CNAF, BO, LNL, BA: very active!
- more activity on submission tool (later)

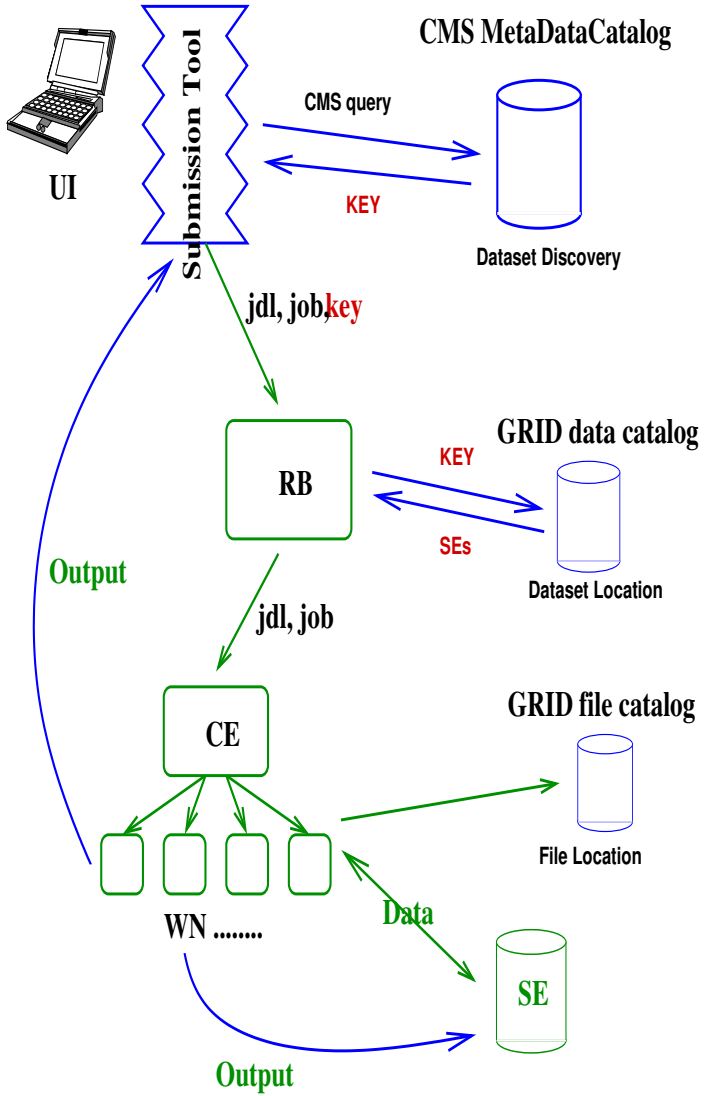
Actual problems

- Data Discovery and Data Location is handled by same service
- Not clear distinction between central service and local ones
- Need to contact several, remote PubDBs just to know that data is available
- Actual Data Discovery (SE or CE) done by CMS catalog!
- (ab)using the Grid just to get access to remote CE
- Not possible to contact local PubDB from local CE: need to do at beginning (and for all possible PubDBs)
- Need to build a complete set of catalogs according to user requirements (eg DST+MC, Digis, etc . . .): can be tricky if multiple catalogs!

With all these limitations

It works!

Data Location and Access: a proposal



- CMS specific MetaData Catalog
  - higher level, interface to physicist
  - provide query mechanism
  - output is a "key(s)", which is uniquely associated with Data chunk
  - Data Chunk is an unbreakable unit (Atom). the granularity is defined by CMS (today is Dataset . . .)
- Grid Data Catalog
  - Given key identifying DataChunk $\Rightarrow$ list of SE(s) $\Rightarrow$ RB get CE(s)
  - Use only *abstract Data*, not files!
- Grid local file catalog
  - Available at local sites
  - Files location for CMS framework **and** DataManagement system

Data Management

- Data is created by Production in a given site
  - Register new DataChunks on CMS MetaDataCatalog
  - Fill local file catalog with produced files
  - Register keys on Grid Catalogs
- Data Movement
  - Want to move DataChunk to Site B
  - Query on CMS MetaDataCatalog to get key for that DataChunk
  - Get list of SE(s) which holds DataChunks
  - (possible skip first 3 steps if push data from site A)
  - DM access local file catalog to get set of files and move them to site B
  - DM register that also Site B has DataChunk

ToDo's

- Replace multiple xml FileCatalog with unique, mysql-based
- Already possible and tested (Bari)
- Some complication from Cobra MetaData handling (must be improved in future)
- PhedEx V2.1 is going to use the same local mysql File Catalog for Data Movement
- Dataset is today an *atom*: too big, but a starting point
- First prototype for high level interface for Dataset discovery
  - Work extended on RB interfaced to RefDB-PubDBs via web service Clarence-based (PD-CERN)
  - Can be extended to use *any* DB
- Activity within "italian–APROM" to reach consensus about overall architecture
- Some services are GRID responsibility
- Need to provide clean and agreed requirements in order to drive GRID people work

Outline

Site Responsibilities

- Data related
- At present
  - PhedEx agents for data import/export (Data Management)
  - PubDB installation and maintenance
  - Publication of locally produced or imported data on local PubDB
  - Local File Catalogs (either xml or mysql)
  - Automatization of most of these process (mostly publication) underway
- Site responsibility will decrease when full integration with Grid will be achieved

CMS software installation

- To allow local development and test
- To guarantee usage by Grid user
  - Grid user must find on grid Working Node exactly the same environment she find interactively: *i.e.* `scram`-based
  - Tool (`xcmsi`) for installation of CMS software
  - Usage within Grid: remote installation via Grid job by *privileged user* on all Grid sites (Padova)
  - SW installation on Grid resources not a site responsibility
- Issues
  - Different tool (`DAR`) used by Production
  - Looking for automatic procedure for deploying
  - Not only stable release, but also developing code (*e.g.* night building . . .)

Outline

Remote Access Tool Functionalities

- User friendly interface for CMS user to grid services
- Should hide as much as possible of GRID technicalities to end user
- Running on Grid should be transparent to user
- Interact with Grid and non-Grid services
  - DataDiscovery
  - Job preparation
  - Job splitting
  - Monitoring and bookkeeping
  - Output Retrieval
  - ...

Status

- Many parallel projects started
- Obviously based on current catalogs infrastructure (need to perform a set of dirty tricks in order to run!)
- INFN (Padova) CMS Remote Analysis Builder
  - Initial test by Bari, Bologna
  - Working Prototype with all needed functionalities (Beta-Tester wanted!)
- UK GROSS (based on BOSS (Bologna))
- Other project focused on US-Grid
  - GAE: based on Clarence
  - RunJob: from McRunJob used by production

CRAB Functionalities

- What the user is supposed to do and know
- She is able to run interactively on local data
- Develop/test her code on a local machine (UI)
- Provide to CRAB (via configuration file):
  - Dataset/Owner she want to access
  - Type of data she needs (DST, Digis, ...)
  - Job splitting directives (# event per jobs)
  - Name of Executable
  - `.orcarc` cards: the one she used locally!
- CRAB provide
  - Dataset discovery
  - Changes of `.orcarc` to run on remote site (including catalogs, splitting, etc ...)
  - Job submission, tracking
  - Simple monitoring
  - Automatic output retrieval at the end
- Grid details are hidden to user

CRAB Future

- Integration CRAB and GROSS/BOSS
  - General agreement, need detailed workplan
  - Use CRAB for job preparation, splitting, data discovery, etc
  - Extend BOSS with GROSS functionalities and use it for submission, monitoring, tracking, logging, output retrieval, etc ...
  - Better handling of jobs (using Boss DB), including *task i.e.* multiple jobs seen as a whole
  - Use Boss for interaction with Grid scheduler
  - Much better logging and bookkeeping
- Other issues
  - Job splitting: today event-based, discussion for future (Run-based or *atom*-based)
  - Real-Time monitoring
    - Foreseen with extended BOSS
    - Using JAM (Job Application Monitoring) (Bari): already tested
  - Give to user, get feedback and use cases

Summary

- Data Location and Access is critical for Distributed Analysis
- Short term solution is under test, proven to work
- For longer term, need to agree on a more general architecture, involving Data and Workload Management
- Possible architecture, including integration with GRID service presented
- First set of automatic remote submission tool being developed and tested
- Users are starting to use them
- Plan for longer term

Acknowledgment

Many thanks to all these people for help and feedback

Lucia Silvestris, Claudio Grandi, Nicola de Filippis, Alessandra Fanfani, Daniele Bonacorsi, Federica Fanzago, Marco Corvo, Nikolay Smirnov