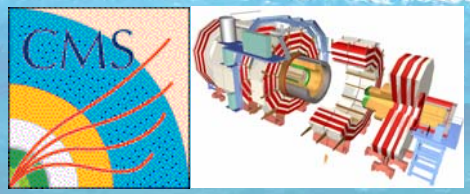
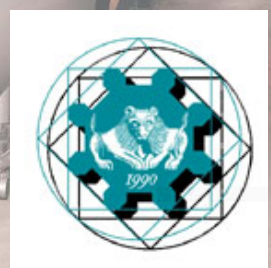


The CMS analysis chain in a distributed environment

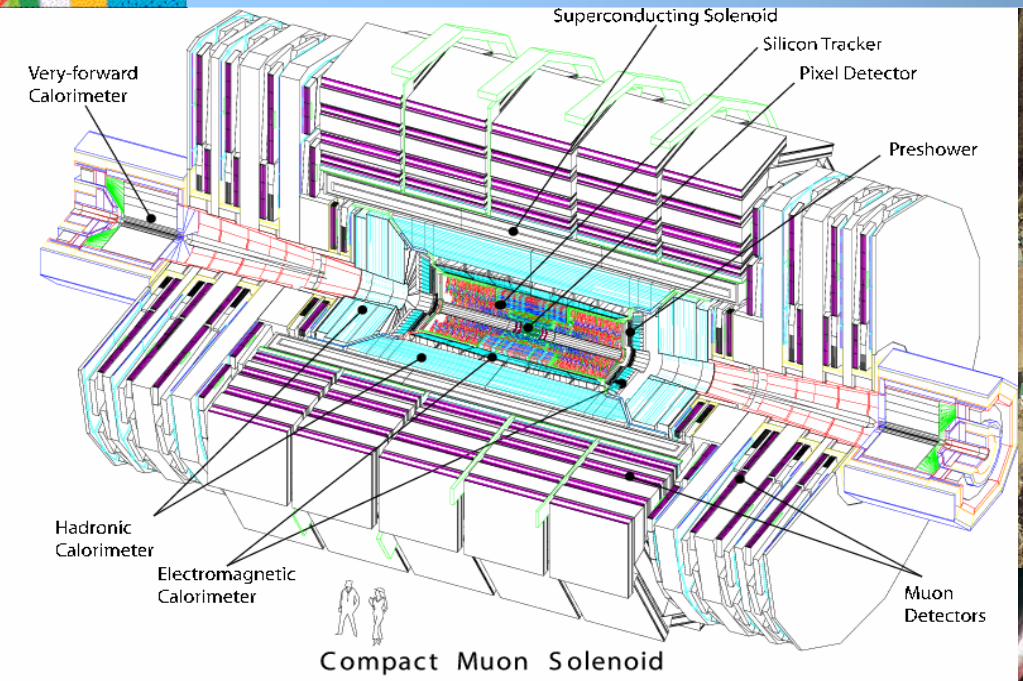
Nicola De Filippis

on behalf of the
CMS collaboration

ACAT 2005
DESY, Zeuthen, Germany
22nd – 27th May, 2005



The CMS experiment



The CMS Computing Model (1)

The CMS collaboration is making a big effort:

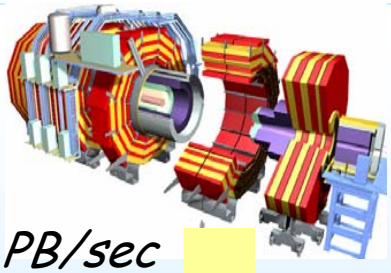
- to define the analysis model
- to develop software tools with the purpose of analyzing
  - several millions of simulated data
  - PetaBytes of real data per year
  - by a large number of people in many geografically distributed sites.

Problems to be faced out:

- large scale distributed computing and data access
- efficient data movement and validation chain
- reliable batch analysis processing
- definition of local and global policies to use the resources

The distributed architecture is one possible solution

The CMS Computing Model (2)



~ PB/sec

~ 100MB/sec

Online system

Offline farm

1PC* → PIII 1GHz

recorded data

Tier 0

CERN Computer center
~10K PCs*

- Filter→*raw* data
- Data Reconstruction
- Data Recording
- Distribution to Tier-1

Tier 1

France Regional Center ~2K PCs

Italy Regional Center

Fermilab Regional Center

~ 2.4 Gbits/sec

- Permannet data storage and management
- Data-heavy analysis
- re-processing
- Simulation
- ,Regional support

Tier 2

Tier2 Center ~500 PCs

Tier2 Center

Tier2 Center

~ 0.6 – 2. Gbits/sec

- Well-managed disk storage
- Massive Simulation
- End-user analysis

Tier 3

InstituteA

InstituteB

workstation

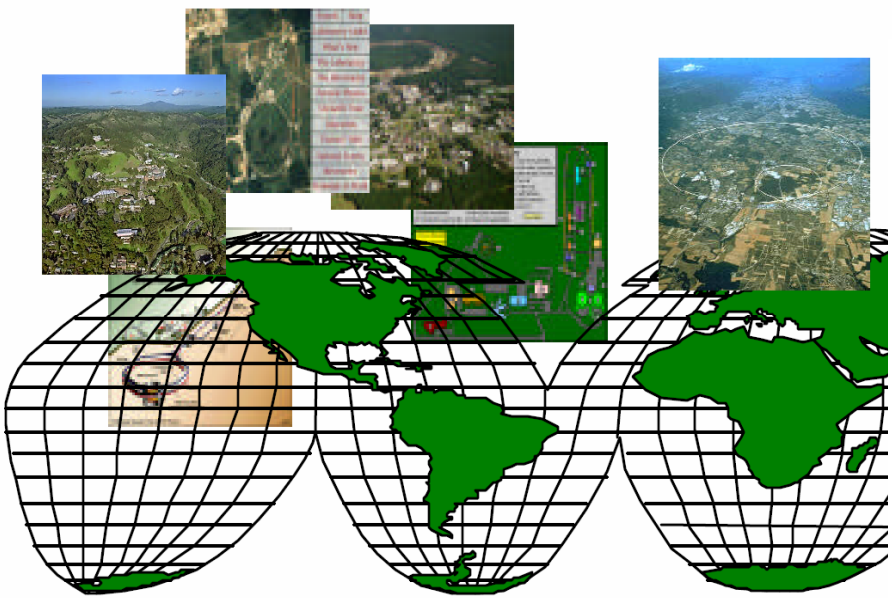
~ 100-1000 Mbits/sec

- End-user analysis

The CMS Computing Model (3)

CMS is collaborating with many Grid projects to explore the maturity and availability of middleware:

- *LHC Computing Grid (LCG)*, based on EU middleware
- *Open Science Grid (OSG)*, Grid infrastructure in the US



Main components of the LCG Middleware:

- Virtual Organizations (cms,atlas,ecc.)
- Resource Broker (RB)
- Replica Manager (RLS)
- Computing Elements (CEs)
- Storage Elements (SEs)
- Worker nodes (WNs)
- User Interfaces (UIs)

The CMS analysis tools

Overview:

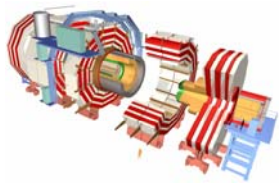
- **Data management**
  - Data Transfer service: **PHEDEX**
  - Data Validation stuff: **ValidationTools**
  - Data Publication service: **RefDB/PubDB**

- **Analysis Strategy**
  - Distributed Software installation: **XCMSI**
  - Analysis job submission tool: **CRAB**

- **Job Monitoring**
  - System monitoring: **BOSS**
  - application job monitoring: **JAM**

The data workflow

❑ Data are moved from Tier 0 to Tier 1 and Tier 2 sites via **PhEDEx**



❑ Data validation is triggered at the end of the transfer via **ValidationTools**

❑ Data are published in the local **PubDB**

PhEDEx (Physics Experiment Data Export)

PhEDEx is the CMS official tool for data movement/transfer:

Goals:

- Manage the prioritized transfer of files from multiple sources to multiple sinks
- Provide information on cost- latency and rate- of any transfer to enable scheduling

Features:

- Enables CMS to manage the distribution of data at dataset level rather than at file level
- Bridges the gap between “traditional” and “Grid” data distribution models
  - Traditional ⇒ large-scale transfers between large sites, often managed by hand
  - Grid ⇒replication of data in response to user demand

Strategy for data flow:

- Detector data flows to Tier 1 sites
  - Stored safely to tape and undergoes large-scale processing and analysis
- Processed data flows to Tier 2 sites
  - Undergoes small-scale analysis
- Simulation and analysis results flow from Tier 2 sites
  - Cached at Tier 1s

PhEDEx is a stable service at Tier 0, Tier 1 and Tier 2 sites all over the world

Data Validation Tools

CMS planned to implement a validation hierarchy with multiple steps:

a) Technical validation for production: which should make file integrity guarantees (checksum, size) and ensure the matching of the information stored in the central database RefDB and that extracted from local files;

b) Validation of data transfer: transfer validation to ensure the file integrity at the end of a transfer. It is covered up to now by PhEDEx

c) Validation for analysis at remote sites: which should ensure readyness of data for analysis in remote site. At this step data should be published in local file catalogs or database.

d) Physics validation: validation of physics content done by Physics groups. It should cover also calibrations validation.

Step c): Validation for analysis

Main features:

- building of data catalogs for local existing files
- validation of Monte Carlo event samples via CMS analysis codes
- processing of output histograms
- publishing of the validated information and the file catalogs in a local database, PubDB.

Use cases supported:

- the validation of official or "private" datasets in various data formats with different level of consistency check
- remote validation via grid

Implementation: bash and perl scripts with a configuration file + a GUI in perl-Tk

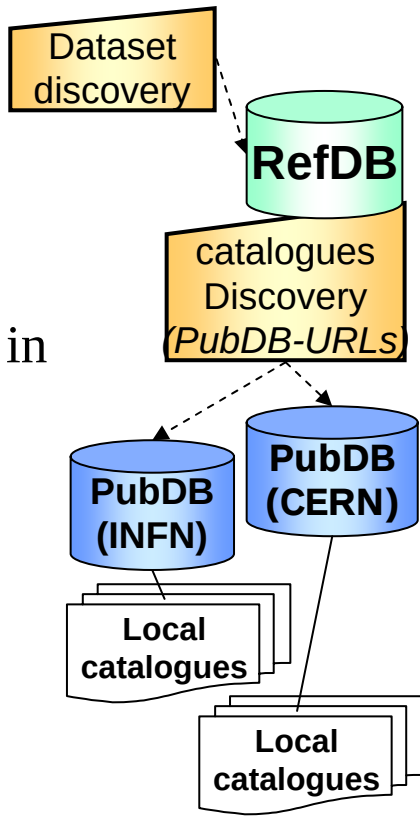
Result: automatization of the technical procedures to be performed by a site manager in a remote site to make data available for analysis users

In production: at Italian Tier-1 (CNAF) and at many Tier 2/3s in Europe.

PubDB (publication database)

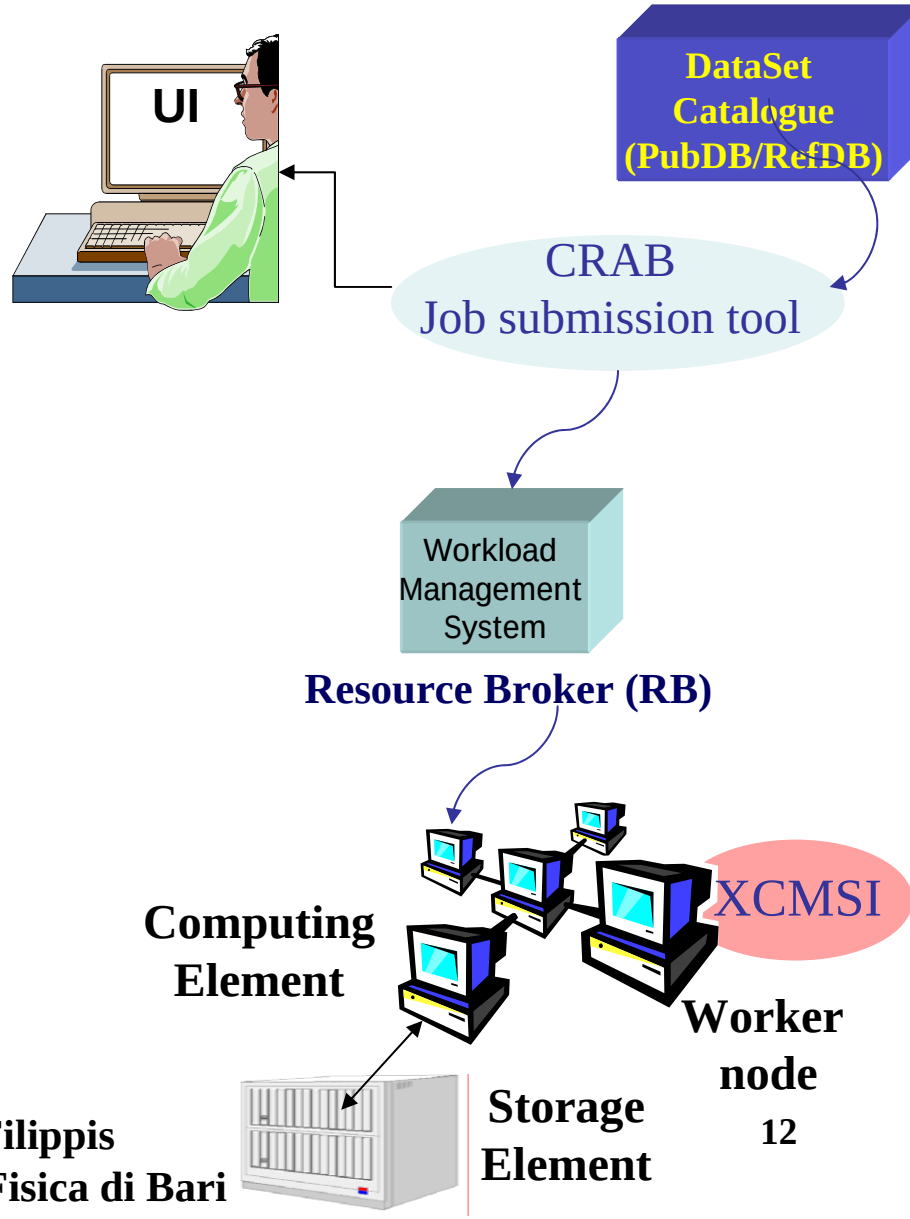
PubDB is a database for the publishing of data available for analysis in a site

- Local file catalogues are stored in PubDB together with dataset specific information, the validation status, the total number of validated events, the first and last run…
- Dataset Location: a global map of all datasets catalogues is held in the central database RefDB at CERN through the links to the various PubDBs
- PubDB/RefDB plays the main role in the data discovery system.
- Evolving to a new system with a
  - Dataset Bookkeeping System which will answer the basic question "Which data exist?"
  - Data Location Service which will allow a CMS user to find replicas of a given set of data in the distributed computing system.



The end-user analysis wokflow

- The user provides:
  - Dataset (runs,#event,..)
  - private code
- CRAB discoveries data and sites hosting them by querying RefDB/ PubDB
- CRAB prepares, splits and submits jobs to the Resource Broker
- The RB sends jobs at sites hosting the data provided the CMS software was installed
- CRAB retrieves automatically the output files of the the job



The CMS software installation: XCMSI

Goal:

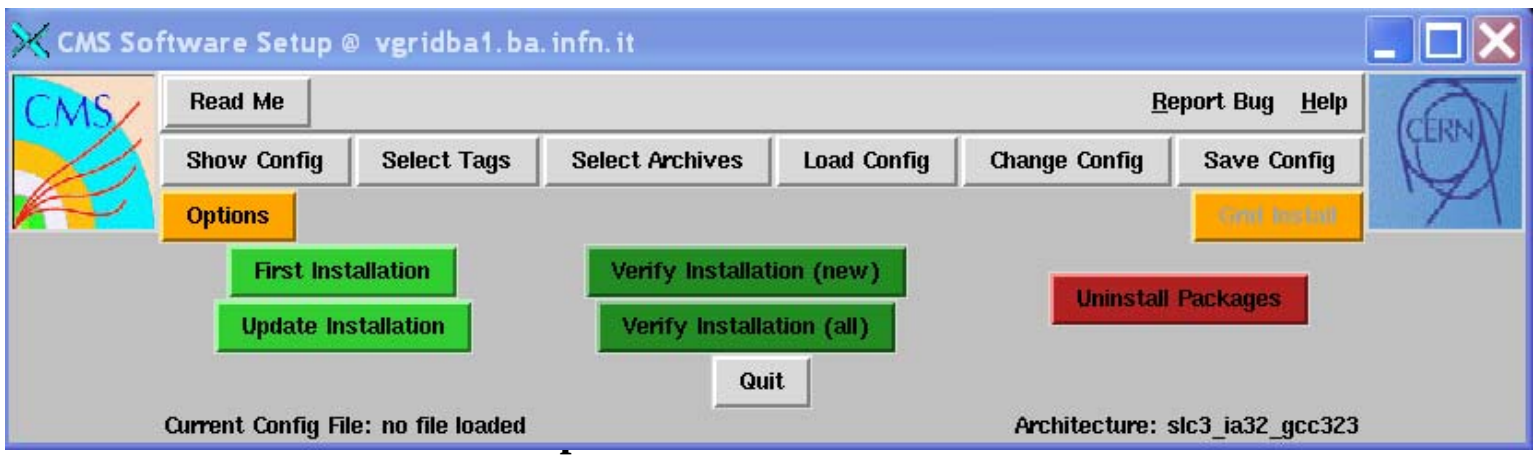
To provide complete CMS software environment for development and data analysis

Features:

- Relocatable packages
- Optional network download
- Save-able and reusable set-up
- Multi-platform support
- No root privileges required
- Batch mode installable
- Included validation procedure
- Multiple installations possible

In a GRID environment:

Installation done via *ad-hoc* job run by the cms SoftwareManager with privilegies;



CRAB (CMS remote analysis builder)

CRAB is a python user-friendly tool for:

- job preparation, splitting and submission of CMS analysis jobs
- analysing data available at remote sites by using the GRID infrastructure

Features:

- User Settings provided via a configuration file (dataset, data type)
- Data discovery querying RefDB and PubDB of remote sites
- Job splitting performed per events
- GRID details mostly hidden to the user
- status monitoring, job tracking and output management of the submitted jobs

Use cases supported:

- Official and private code analysis of published remote data

CRAB in production

- Actively used by **tens** of CMS users**,** with little or no Grid knowledge
- Already several physics presentation based on data accessed using CRAB
- Successfully used to access from any UI data at Tiers-1 (and some T2s):
  - FNAL (US)
  - CNAF (Italy)
  - PIC (Spain)
  - CERN
  - FZK (Germany)
  - IN2P3 (France)
  - RAL (UK): still working
  - Tiers-2: Legnaro, Bari, Perugia (Italy)
- Estimated total **O(10^7)** events analysed via CRAB on a distributed infrastructure

Job monitoring via BOSS

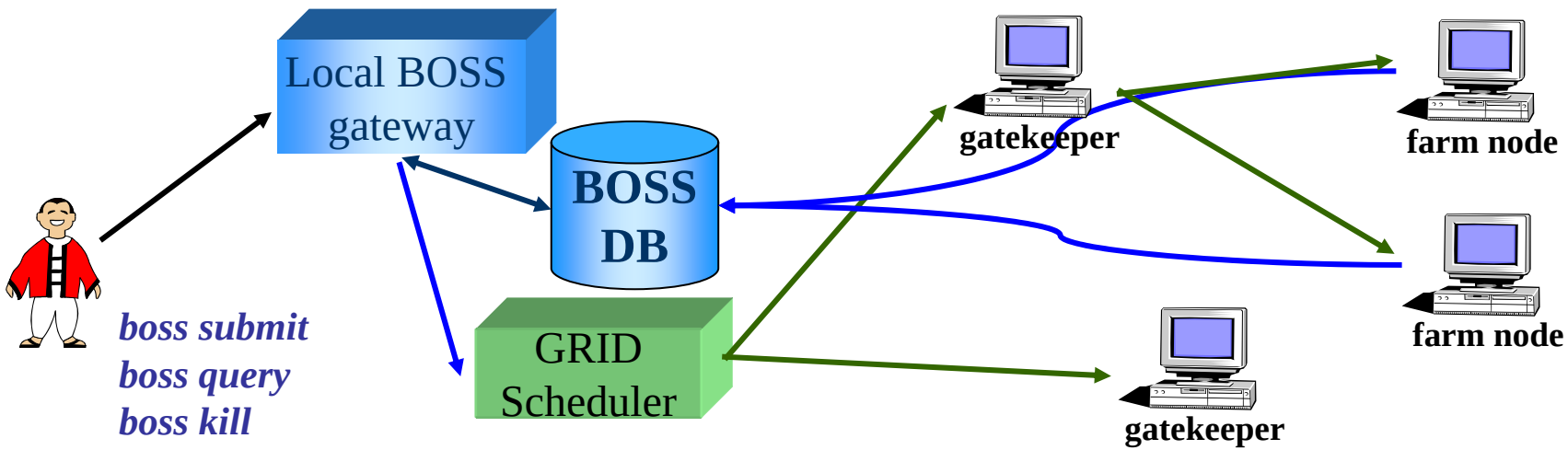
BOSS is a tool for job monitoring, logging and book-keeping

Features:

- Allows to deal with job-specific information (#events, run number, host, data)
- Stores info about jobs in a DB (MySQL server)
- Is not a job scheduler, but can be interfaced with most schedulers: LSF, PBS, Condor and to the GRID scheduler

In production:

- Used for Monte Carlo production and real-time analysis during *data challenge 2004*. A new workflow is going to be implemented and to be integrated in CRAB.



Application monitoring via JAM

JAM is a tool for job application monitoring

Goals:

- Monitoring, logging and bookkeeping, quality assurance and interactive control of an application for the very-end user

Features:

- Describe the application with tags
- Organize the split applications in sets
- Retrieve the values of the tags online (Client-Server based on gSoap)
- Store the infos in a pseudo-filesystem (real storage with MySQL)
- Define rules on tags to classify the application as good/maybe/bad
- Peek any remote file online
- Data transfer via basic C++ API

In production: first production release just ready to be used from generic users.

Conclusions

- CMS first working **prototype for Distributed User Analysis** is available and used by **real** users
- **Phedex, PubDB, ValidationTools, XCMSI, CRAB, BOSS, JAM** under development, deployment and in production in many sites
- CMS is using **Grid infrastructure** for physics analyses and Monte Carlo production
- **tens** of users, **10 million** of analysed data, **10000** jobs submitted
- CMS is designing a **new architecture** for the analysis workflow