Imperial College
London

Distributed analysis for LHCb

22 June 2004

.

What is analysis in LHCb?

- The aim of LHCb is to extract results on *CP* violation from rare *B*-meson decays.
  - Detector will see the full event rate of 40 MHz from LHC collisions.
  - When data leaves the detector the rate should be around 200 Hz.
    - This is the job of the trigger and not for discussion today.
  - Output data in DST format will be divided into streams each containing maybe 10^7 events per year.
    - These are the events that the end user will access for analysis.
    - Size for DST is about 150 kB per event.
  - For analysis we also need access to simulated data.
    - 10 times higher statistics.
    - 2 times larger size.

Logistics of analysis

- Data is produced at CERN and streamed DST is distributed to 5 Tier 1 centres.
- Simulated data is produced at Tier 2 and Tier 1 centres. Data again transferred to Tier 1 centres.
  - Only small overlap in datasets between different T1s.
  - Tier 1 centres will be main focus point for analysis.

A user view of an analysis – create algorithm

- An analysis will define a work flow of algorithms to run.
  - The algorithms will be written mainly in C++ and run inside the Gaudi framework.
    - Possibility to write algorithms in Python as well but no widespread use.
- Code for analysis will be a mixture of
  - Standard LHCb algorithms
    - Under version control
  - User modified standard algorithms
    - This is typically for development of the standard tools
  - User specific algorithms
    - These might be different from job to job.
    - No version control.
    - This is a major difference to production style jobs.

A user view of an analysis – configure job

- LHCb jobs are configured through a set of options files.
  - These files have no state and can be pre-processed in a static way.
- For analysis these options might be different from one job to the next.

```
ApplicationMgr.TopAlg += { "PreLoadParticles" };

// Set to use the CombinedParticleMaker
PreLoadParticles.PhysDesktop.ParticleMakerType = "CombinedParticleMaker";

// Default values for particle types to be made
// Note that when exclusive mode is selected the selection is done in
// the order set below
PreLoadParticles.PhysDesktop.CombinedParticleMaker.Particles = {"kaon", "pion" };

// Default value for particles to be selected exclusively or not
PreLoadParticles.PhysDesktop.CombinedParticleMaker.ExclusiveSelection = false;

// Default value for selection of kaon
//  PreLoadParticles.PhysDesktop.CombinedParticleMaker.KaonSelection = {
//    "det='RICH' k-pi='2.0' k-p='-2.0'" };

PreLoadParticles.PhysDesktop.CombinedParticleMaker.KaonSelection = {
   "det='RICH' k-pi='-5.0'" };
PreLoadParticles.PhysDesktop.CombinedParticleMaker.PionSelection = {  };
//   "det='RICH' pi-k='-5.0'" };
```

A user view of an analysis – define dataset

- Dataset for analysis needs specification.
- Selection can happen in many ways
  - Fixed list taken from static web page.
  - Selection in a database
    - "*The B to D_sK background calibration set*".
  - "*Same data as I used yesterday*".
  - Specific event
    - Event 2134539 from run 3473.
  - "*My usual test data*"
- Notice that none of these cases contain anything about LFN, PFN or other file specific information.
- User will often be blind to if they read the data directly or read an event summary file that simply points to the events.

A user view of an analysis – test and run algorithm

- For testing and running analysis the average physicist will always do what seems the easiest way to get a result tomorrow!
  - Interaction with complicated systems that might bring long term time savings never gain wide acceptance.
  - So the default behaviour for testing analysis is to fork jobs from the command line.
- For running larger scale jobs the incentive will always be to use what worked yesterday.
  - For larger scale analysis jobs this means the PBS batch system at CERN.
- To change the behaviour to be for GRID jobs we can.
  - Make it easier to use than LSF
    - Touch but should be our final goal.
  - Limit the available CPU and data resources at CERN.
    - If (usable) alternatives not in place we will get very unpopular!

A user view of an analysis – keeping track and merging

- For production of simulated events we know the importance of very good bookkeeping.
- In an analysis situation users love it if it comes for free.
  - The default way of working is to sort out in retrospect which jobs failed, which produced corrupt output etc.
- The last stage of an analysis is to pull all the results together.
  - Merge all histogram files.
  - Chain ROOT output files.
  - Normalise to analysed luminosity.
  - Search stdout for specific patterns.
    - Not nice but seen a lot.

User requirements for distributed analysis system

- For all users:
  - Responsive
  - Robust
  - Reliable
- In addition for new users
  - Intuitive.
  - Clear error reporting.
  - Transparent to changes in middleware.
- For more powerful users
  - Ability to deal with Event data collections.
  - Scripting language for job control.

LHCb requirements

- Ability to set priorities
  - Within LHCb we need the ability to control how resources allocated to LHCb are used.
- Accounting
  - From our current production we have seen the efficiency of monitoring to track down errors and wrong configurations in hardware/LCG/jobs.
  - Monitoring at the job level required of:
    - CPU usage
    - Memory usage
    - Data requests
    - Efficiency
    - Users

Some numbers

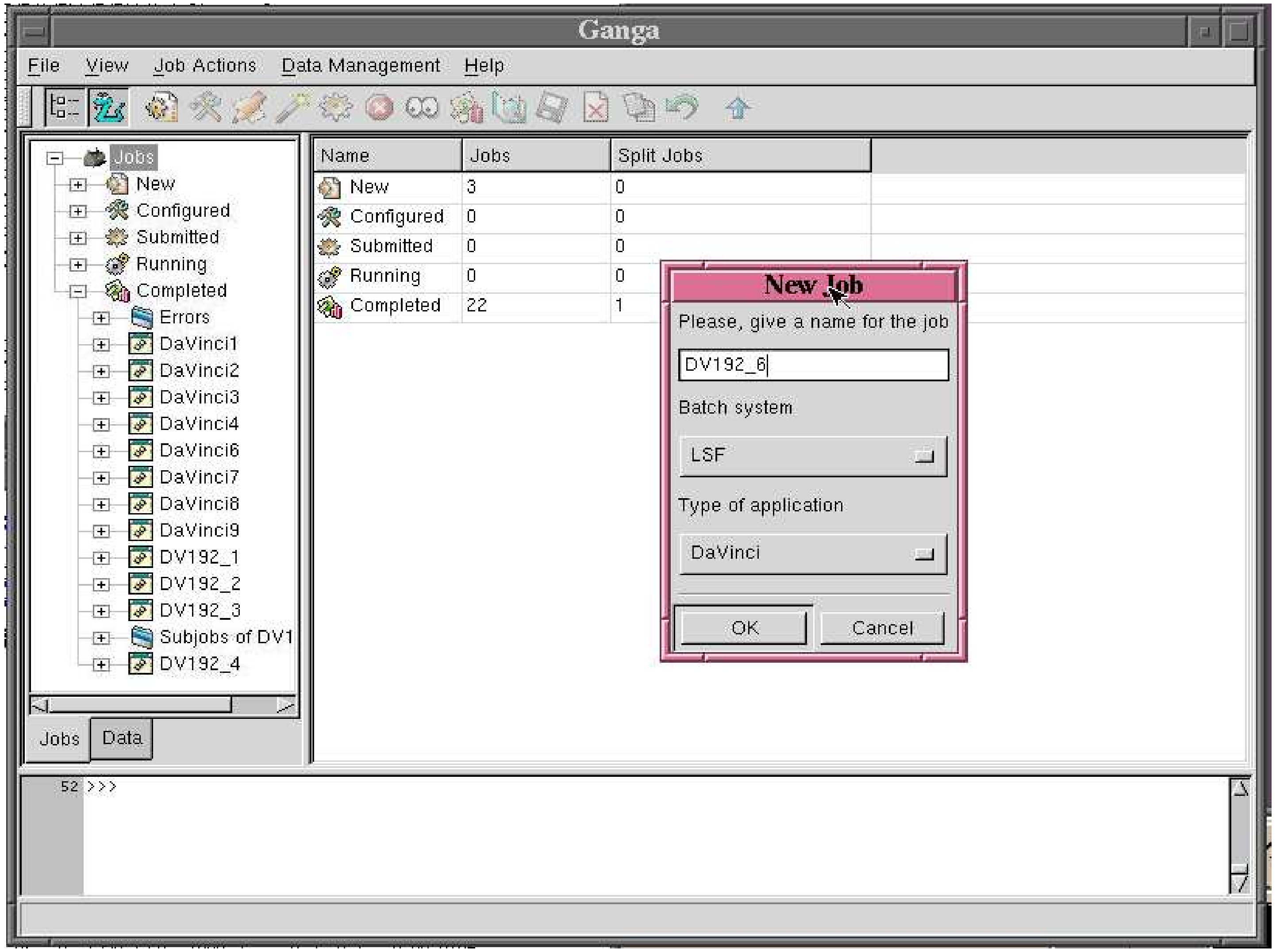
- Projected situation for 2008
  - 140 physicist performing some kind of analysis in LHCb
  - Each will have a number of analysis jobs running with of the order 10^7 events.
  - Quite frequent reruns over this data producing reduced datasets.
    - Space required for these of order 1 PB.
  - A typical Tier 1 centre should have
    - 1900 kSI2k yr of CPU
    - 500 TB disk space

The GANGA project

- The purpose of GANGA is to work as a wizard for LHCb users running GAUDI applications.
  - Mainly we have running C++ analysis applications (DaVinci) in mind.
- We need to support the following behaviour:
  - writing new Gaudi algorithms, either in C++ or Python
  - modifying existing algorithms;
  - modifying job options of existing/new algorithms.
- DC04 data will now (and in the future) be distributed mainly at Tier 1 centres but we should support Tier 2 and local data as well.
  - The input data will be data in POOL format.
  - Output from distributed analysis will be a combination of data in POOL format, ROOT/HBOOK files and stdout/stderr.

How it looks like ...

The GUI is fully functional.



Name	Jobs	Split Jobs
New	3	0
Configured	0	0
Submitted	0	0
Running	0	0
Completed	22	1

New Job

Please, give a name for the job

DV192_6

Batch system

LSF

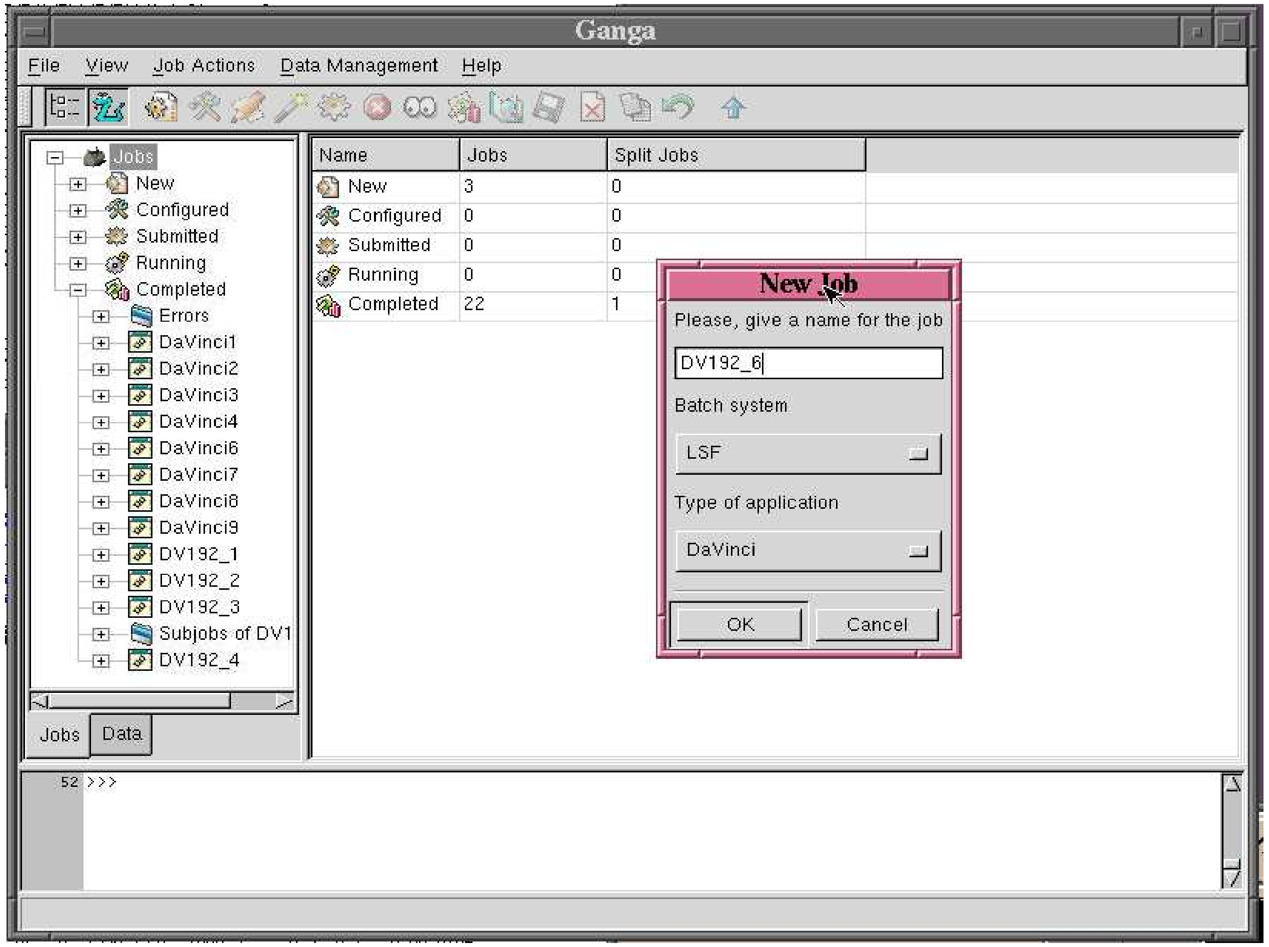
Type of application

DaVinci

```
52 >>>
```

Features

- Jobs in Ganga moves between different states.
  - This gives the bookkeeping
- The user can monitor jobs in the states:
  - New
  - Configured
  - Submitted
  - Running
  - Completed
  - Error – if reported!
- Jobs are preserved between invocations.
  - Status updated at regular intervals.



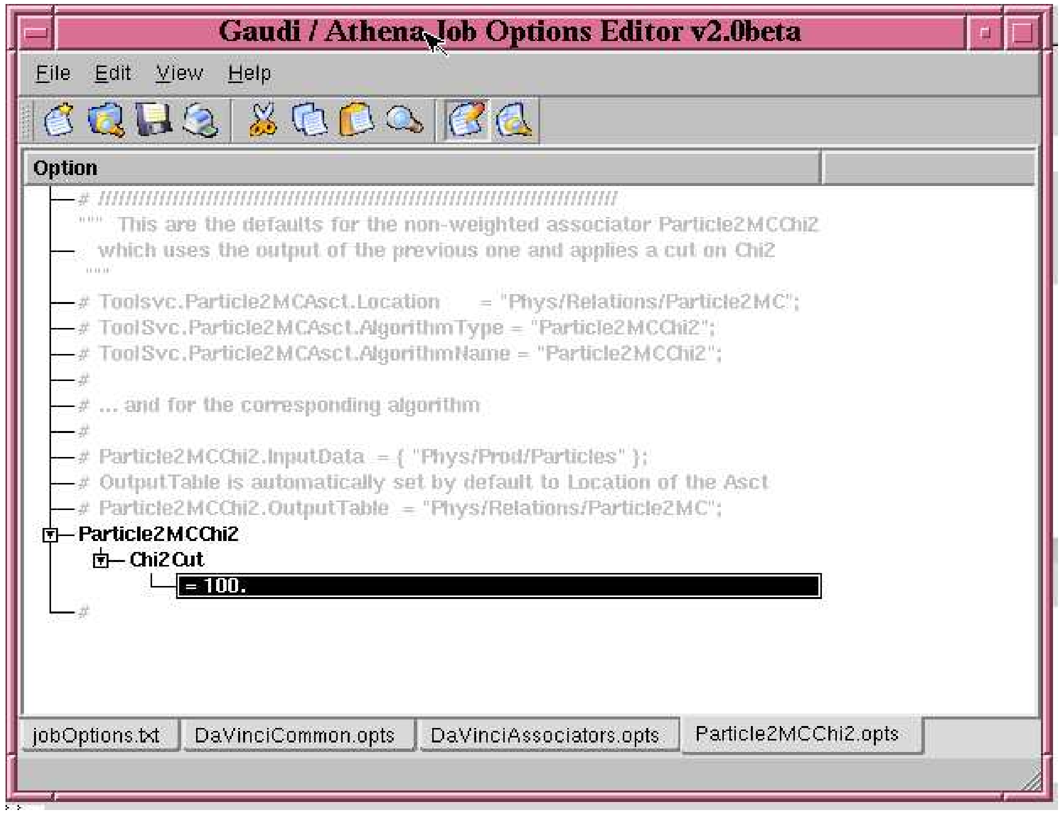
Name	Jobs	Split Jobs
New	3	0
Configured	0	0
Submitted	0	0
Running	0	0
Completed	22	1

Please, give a name for the job: DV192_6
Batch system: LSF
Type of application: DaVinci

Features

A Job Options Editor (JOE) to simplify the configuration of jobs

- Job options is one of the biggest sources of errors in LHCb analysis jobs.
- Saves a single expanded options file.
  - Important for GRID jobs.
- Allows easy navigation of options file.
- Editor should protect against common mistakes and offer the user choices.
  - Lots of potential in this area shown by first prototype.



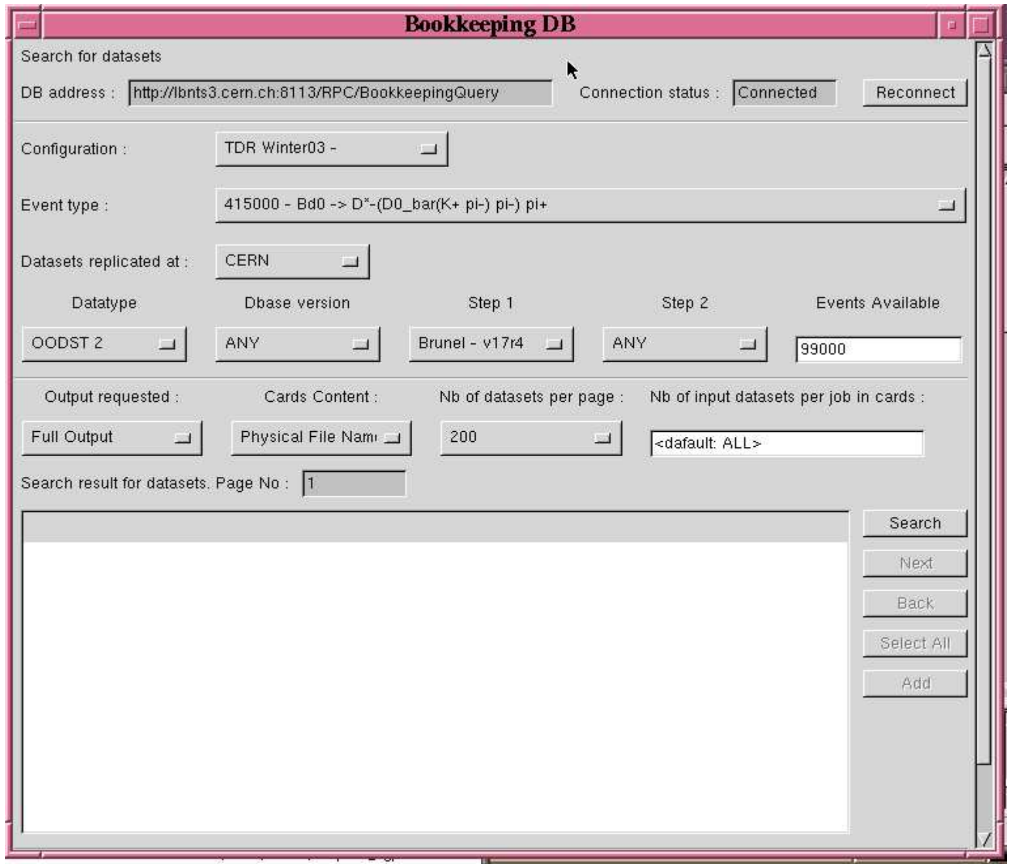
```
# ////////////////////////////////////////////////////////////////////////
"""  This are the defaults for the non-weighted associator Particle2MCChi2
  which uses the output of the previous one and applies a cut on Chi2
 """
# Toolsvc.Particle2MCAsct.Location      = "Phys/Relations/Particle2MC";
# ToolSvc.Particle2MCAsct.AlgorithmType = "Particle2MCChi2";
# ToolSvc.Particle2MCAsct.AlgorithmName = "Particle2MCChi2";
#
# ... and for the corresponding algorithm
#
# Particle2MCChi2.InputData  = { "Phys/Prod/Particles" };
# OutputTable is automatically set by default to Location of the Asct
# Particle2MCChi2.OutputTable  = "Phys/Relations/Particle2MC";
Particle2MCChi2
  Chi2Cut
    = 100.
#
```

Features

Interaction with LHCb bookkeeping database

- This will allow you to pick the data directly from within GANGA.

GANGA should in the future allow user to save dataset selection as templates.



DB address : http://lbnts3.cern.ch:8113/RPC/BookkeepingQuery

Connection status : Connected

Configuration : TDR Winter03 -

Event type : 415000 - Bd0 -> D*-(D0_bar(K+ pi-) pi-) pi+

Datasets replicated at : CERN

Datatype	Dbase version	Step 1	Step 2	Events Available
OODST 2	ANY	Brunel - v17r4	ANY	99000

Output requested :	Cards Content :	Nb of datasets per page :	Nb of input datasets per job in cards :
Full Output	Physical File Nam	200	<dafault: ALL>

Search result for datasets. Page No : 1

Getting the analysis job to the Grid

- Currently we are focused on getting LHCb analysis to run on LCG resources.
- Obvious solution is to submit jobs directly to resource broker.
  - For several reasons we do not pursue this direction.
    - LCG UI impossible to install.
    - No way for LHCb to keep track of analysis.
  - Solution is to use infrastructure of production system (DIRAC) as an intermediate step.
    - We get most monitoring for free.
    - Only trivial DIRAC UI to install (one click).
  - Status
    - Proof of concept analysis job has passed through system.
    - Still need to get LCG job to run with users certificate.
      - Ideas there but not implemented yet.

Status of GANGA

Submission	Local	Working
	LSF	Working
	DIRAC	In Progress
	LCG	Not working
	LCG proxy transfer	Deferred
Job Options	Selection	Working
	Editing	Working
	Expansion	In Progress
	Logical view	Deferred
Job handlers	Generic	Working
	DaVinci	Working
	User defined templates	Deferred
Data selection	Connection to bookkeeping database	In Progress
Installation	AFS	Working
	Local	Working
Job management	Splitting of input data	In Progress
	Merging of output	Deferred
Roaming	Job profile	Deferred
	Options and DLL packaging	In Progress

Internals

The analysis user is dependent on many features that are not directly visible.

- Connection between bookkeeping database and Alien File Catalogue.
- The monitoring facilities of DIRAC.
- Transfer of shared libraries and options files to where jobs run.
- Matching of sites where job runs to sites where data is available.
- ...

What are the goals for this year

- The goals for 2004 were defined at workshop in October 2003.
  - Only minor changes in plans since then.
- The idea is to demonstrate a working model of distributed analysis.
- Demonstration requires:
  - Many users of system.
  - Scope set correctly to have a product in place.
- Distinct data will be kept at Tier 1 centres (and not Tier 2).
  - No user accounts at the different Tier 1 centres
    - LCG (through DIRAC) will provide us with access.