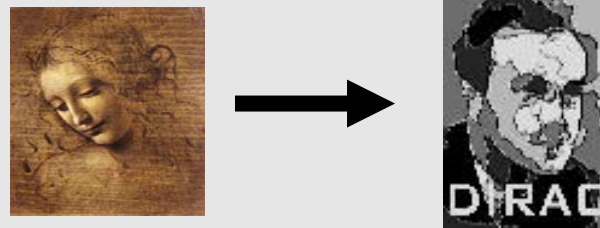
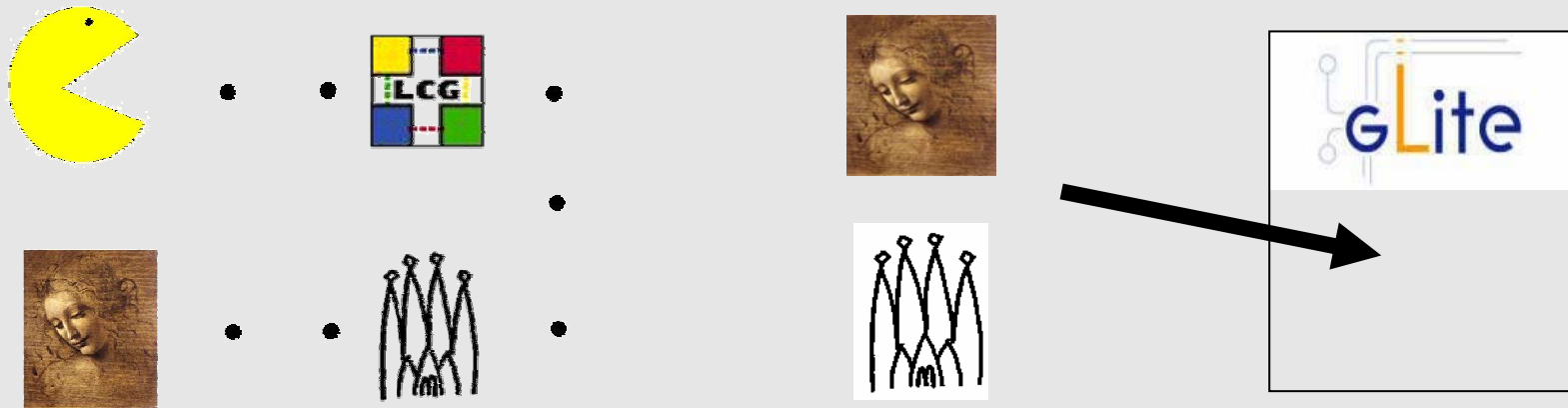


Interfacing DaVinci To The Grid



• Project Overview:- Interfacing DaVinci to the Grid

- DaVinci is the physics analysis package in LHCb

- Need to carry out analysis on the grid because resources and data (both experimental and Monte Carlo) are distributed

• **Aim of Project**

- To develop a distributed analysis model for the LHCb by interfacing DaVinci to the grid.

• Project Outline:- Interfacing DaVinci to the Grid

- 1) Distributed installation of software using Pacman. Necessary if not all software installed on grid nodes prior to running analysis
- 2) First attempt: first use of DaVinci physics analysis on the GLite framework. Use realistic physics analysis to test GLite framework for distributed analysis
- 3) Adaptation of DIRAC for physics analysis on the grid. DIRAC is Monte Carlo production system for LHCb, relying on Remote Agent Control. Research project to embed DaVinci physics analysis into DIRAC
- 4) Comparisons and conclusions on strategy to be adopted in LHCb for distributed analysis

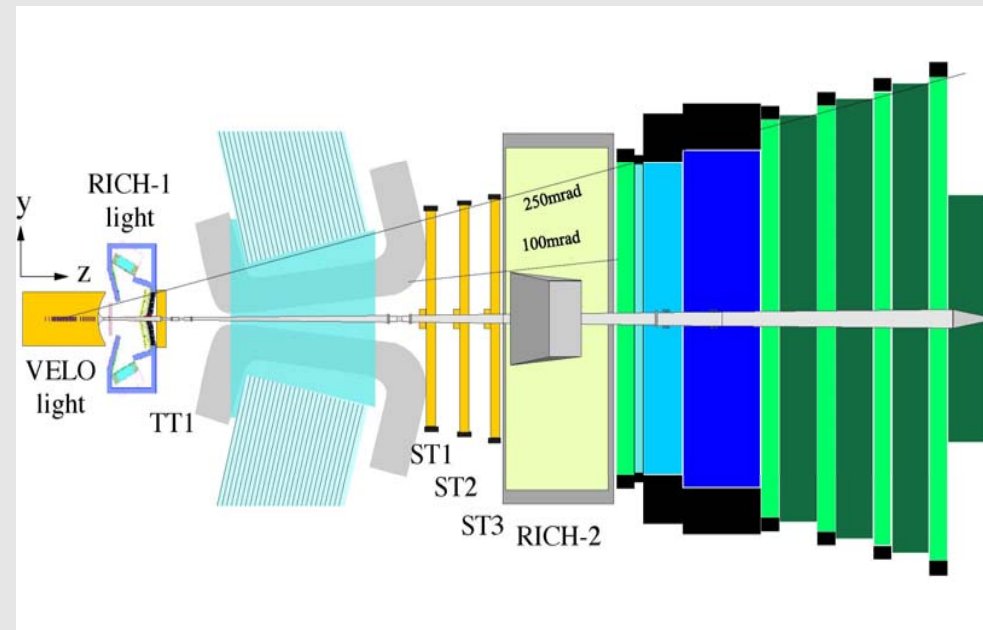
• Presentation Outline

- Introduction to LHCb and the Grid
- Software Distribution for LHCb Using Pacman
- Distributed Analysis Using DaVinci in the gLite Framework
- Enabling analysis for LHCb using DIRAC

- The LHCb Experiment

- Motivation

- Principally looking at CP violation
- Will search for physics within and beyond the Standard Model



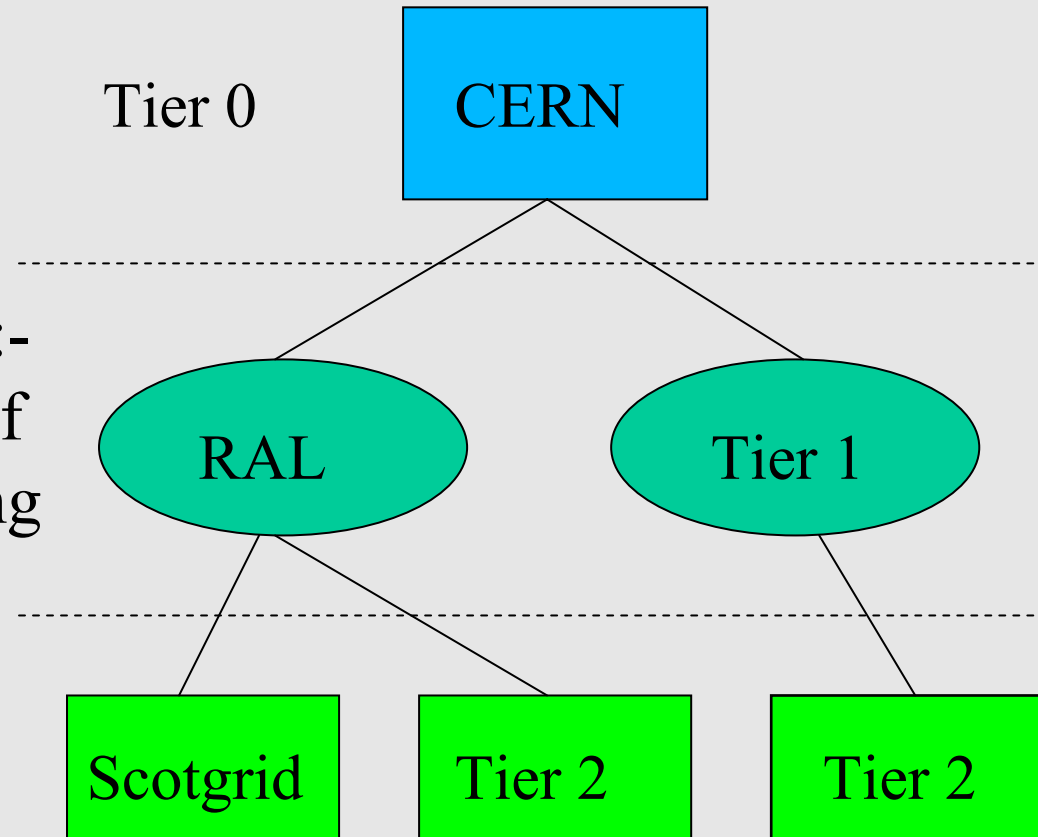
- Computing Requirements

- LHCb will generate around a Petabyte of data per year
- Approximately 100 Petabytes of storage is required to fully analyse the data
- No single institute has the necessary computing power

- What Is The Grid?

- Infrastructure

- Hierarchical organisation:-
- Maximises contribution of physicists without requiring presence at CERN
- Seamless access to available resources

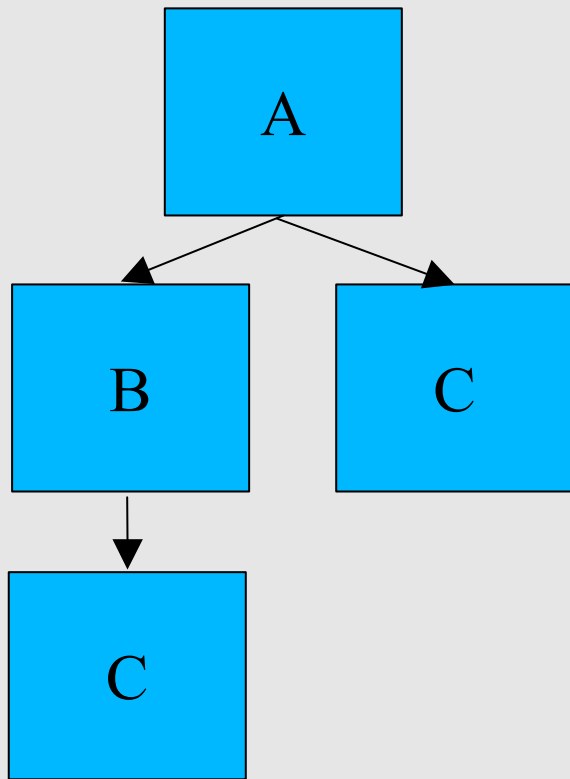


- Many computing nodes across the world

- Different platforms
- Grid enabling LHCb software is of the utmost importance

- Software Distribution for LHCb Using Pacman
- What does Pacman do?
 - Can transparently fetch, install, and manage software packages
 - Handles dependencies
 - Updates are automatically searched for
 - Can recognise natively installed software
 - Is very robust
- How does Pacman do this?
 - Package 'tarballs' are kept in a central web based cache
 - Each package has a corresponding '**.pacman**' file associated with it

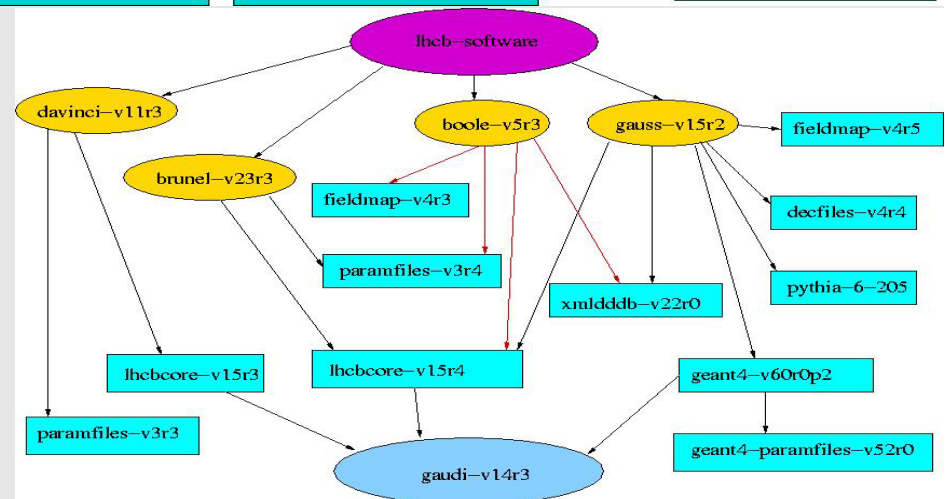
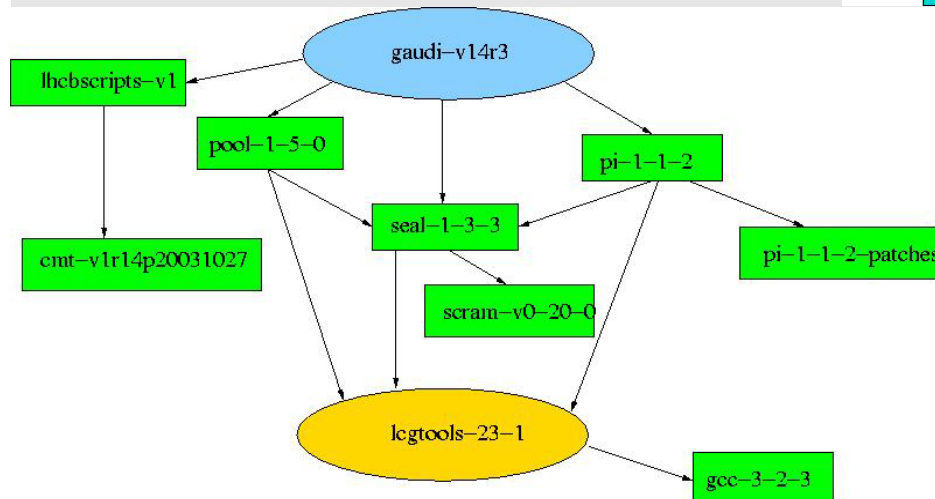
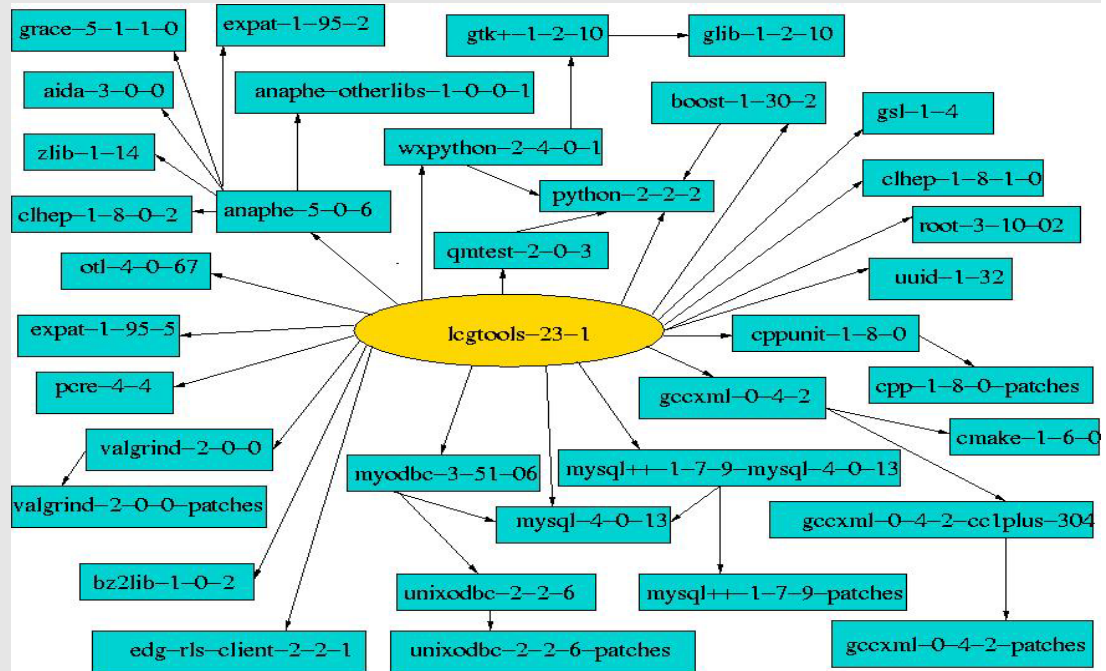
- How Pacman handles dependencies
- LHCb software ~ 60 packages with intricate relationships



- If package A depends on B and C
- With package B also depending on C
- Pacman knows only to install one version of C and to have it installed before A and B

• Overview of the LHCb Software

• Installations covered everything from compiler upwards



• Other Advantages Presented by Pacman

- Problems are solved once by the cache manager, leaving one point of contact if issues arise
- Maintenance of LHCb software between versions is greatly simplified
- Automatically generated index page allows for easy package management
- Much time is saved for others, especially those at external institutes

Pacman Local Installation

- Caches in Use -

- [Pacman](#)
- <http://ppewww.ph.gla.ac.uk/~stuart/localcache/gaudiProper/>

- Local Packages -

Package	Description	Installed?	Update?	Depends on	Daemons	Location	Date fetched
aida-2-2-0	AIDA 2.2.0 Test 2.00	yes	-	-	-	-	Wed Mar 10 16:27:26 2004
boost-1-30-2	Boost 1.30.2 Test 2.01	yes	-	python-2-2-2	-	-	Wed Mar 10 16:27:28 2004
cernlib-2002	Cernlib 2002 Test 2.00	yes	-	-	-	-	Wed Mar 10 16:27:37 2004
chep	CLHEP 1.8.0.0 Test 2.16	yes	-	-	-	-	Wed Mar 10 16:27:38 2004
cmv1r12r1	CMT v1r12 Test 2.00	yes	-	-	-	-	Wed Mar 10 16:27:03 2004
gaudi-v12-r0	Gaudi v12 r0 Test 2.00	yes	-	cmv1r12r1 python-2-2-2 aida-2-2-0 boost-1-30-2 cernlib-2002 chep gsl-1-3 htl-1-3-2-1 rootv0305042 xercesc-2-4-0	-	-	Wed Mar 10 16:27:47 2004
gaudi-v12-r0-installer	Gaudi v12 r0 Installer Test 2.11	yes	-	gaudi-v12-r0	-	-	Wed Mar 10 16:27:47 2004

• Summary Of Progress Made With Pacman

- A web visible cache for all the LHCb software was set up
- Full installations from source were performed on RH7.3 and RH9 at Scotgrid and Liverpool
- Installation & compilation of all LHCb software took around 23 hours, highlighting the need for an automated mechanism
- For more information, please see:-

S.K. Paterson: *Software Distribution for LHCb using Pacman*,
LHCb Note 2004-066 (August 2004)

.Distributed Analysis Using DaVinci in the gLite Framework

.Overview of DaVinci

- .Rests on the Gaudi Framework
- .OO-based, programmed in C++
- .Steered through options files
- .User specific libraries can be added in a simple way



- .To run DaVinci for a generic channel one needs:-

Standard options files

User channel specific algorithms

User options files

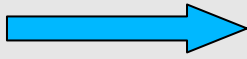
- .Output primarily comes in three forms:-

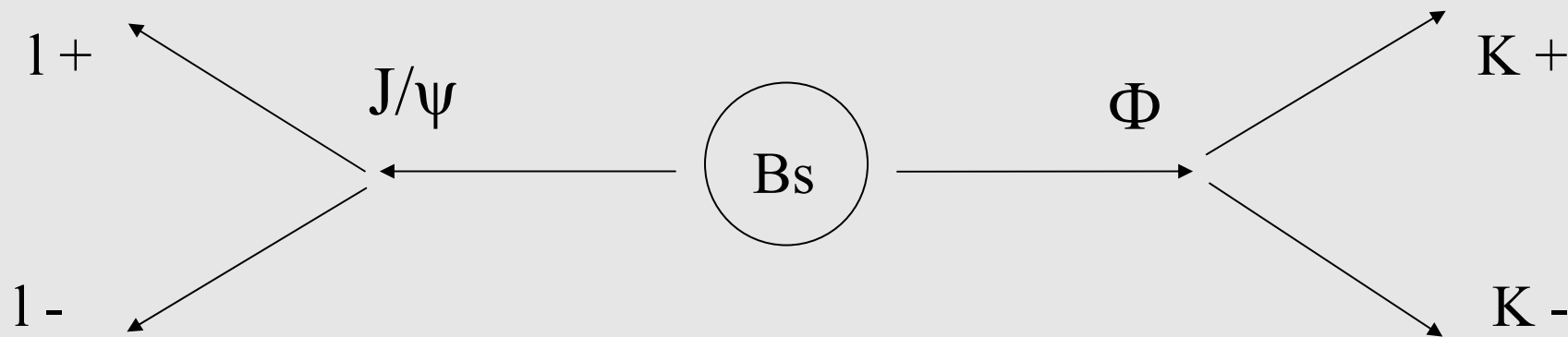
Histograms

Ntuples

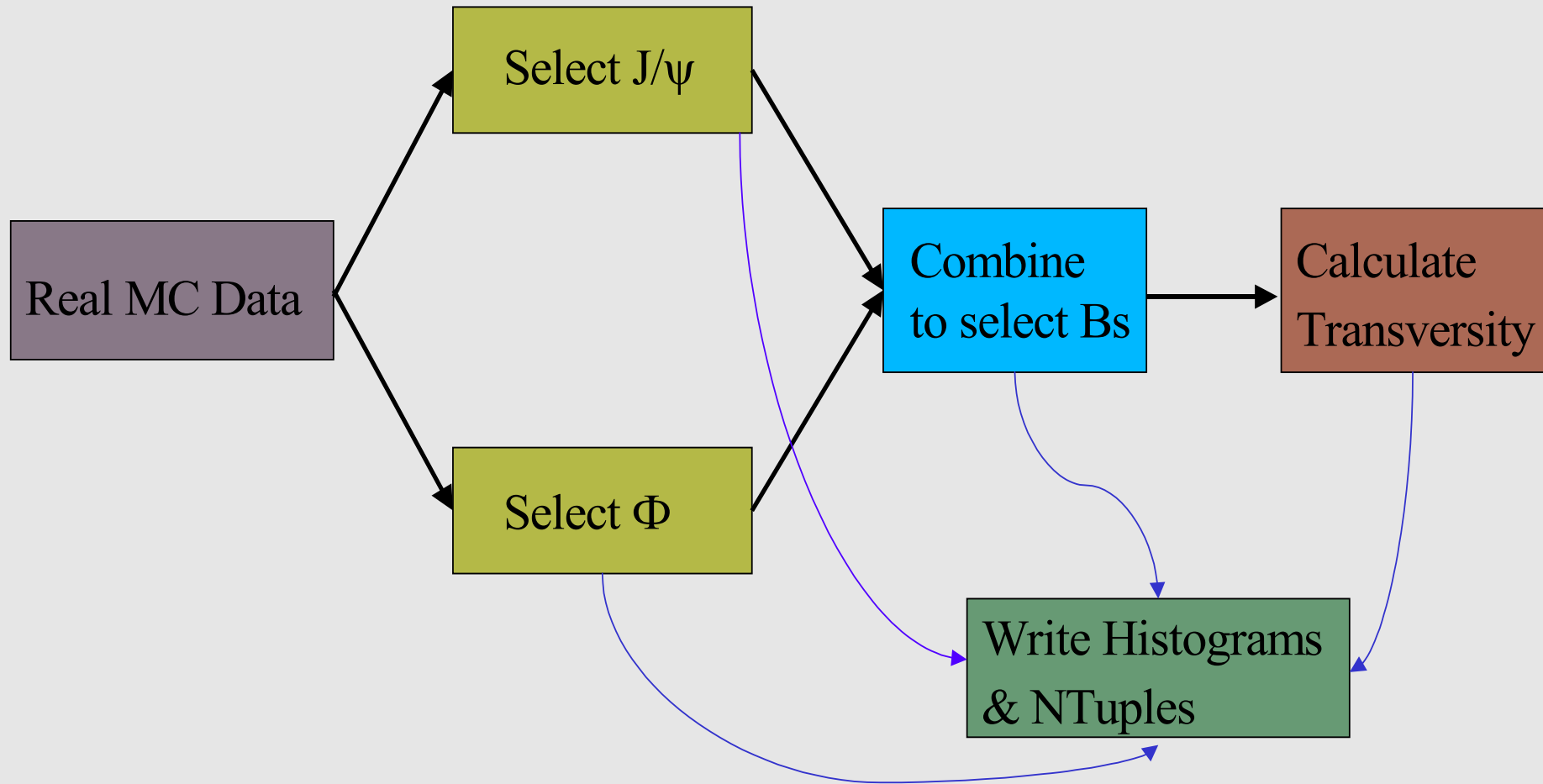
DaVinci Standard Output

• Bs to J/ψ Φ Channel

- Chosen as a ‘typical’ analysis
- Annual signal yield of ~ 100k events
- Possible to extract CKM property $\delta\gamma$
- If measured $\delta\gamma$ is large  new physics beyond SM
- Performed a realistic analysis as a physics user would carry out



• DaVinci – Algorithm Overview



• GLite Prototype

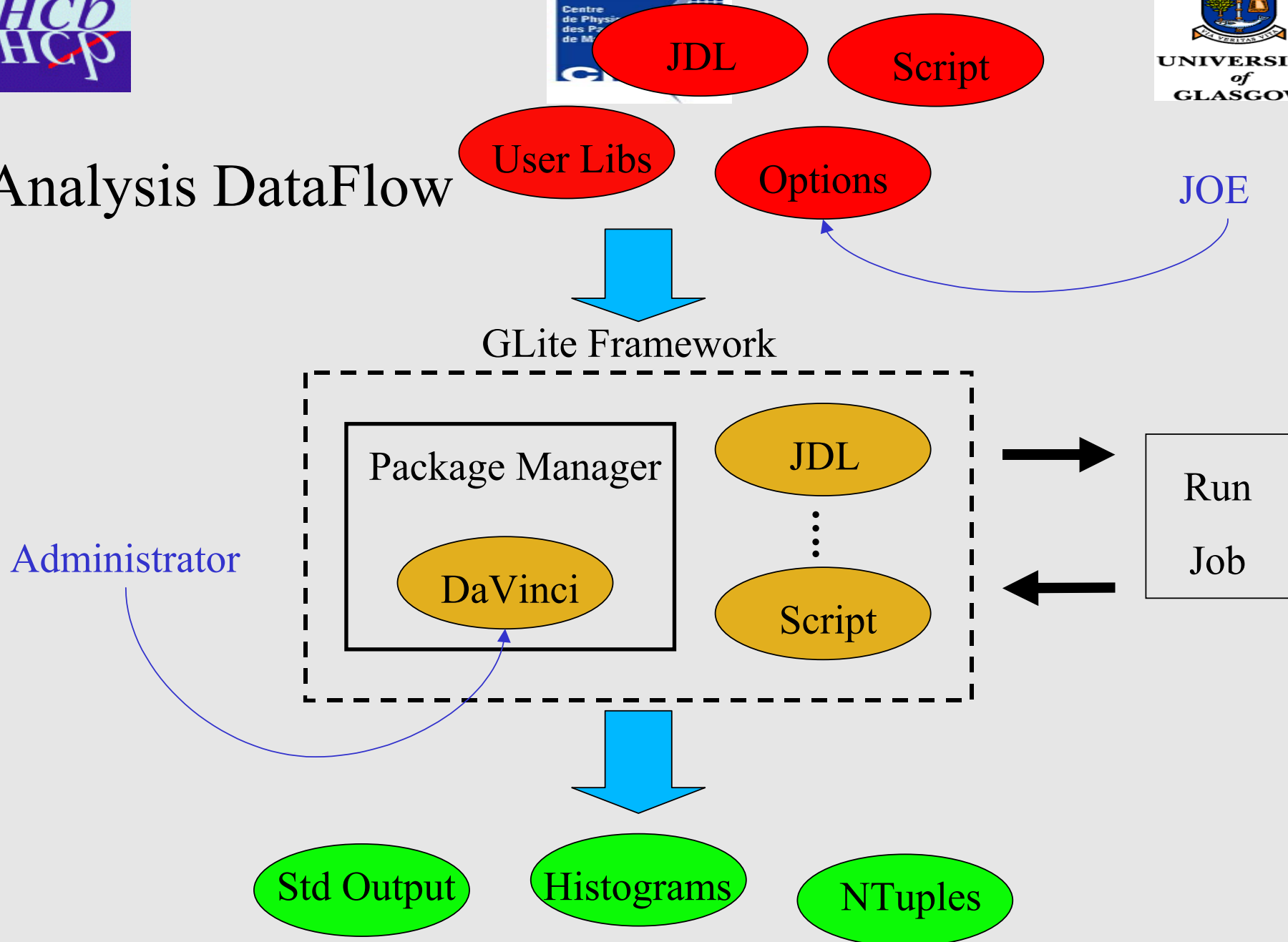
- Based on the AliEn file catalogue
- To run a typical job one needs:-

JDL file script to run application user specific files

- Must decide to use one tarball of desired package directly, or the GLite package manager
- The procedure for running a job is:-



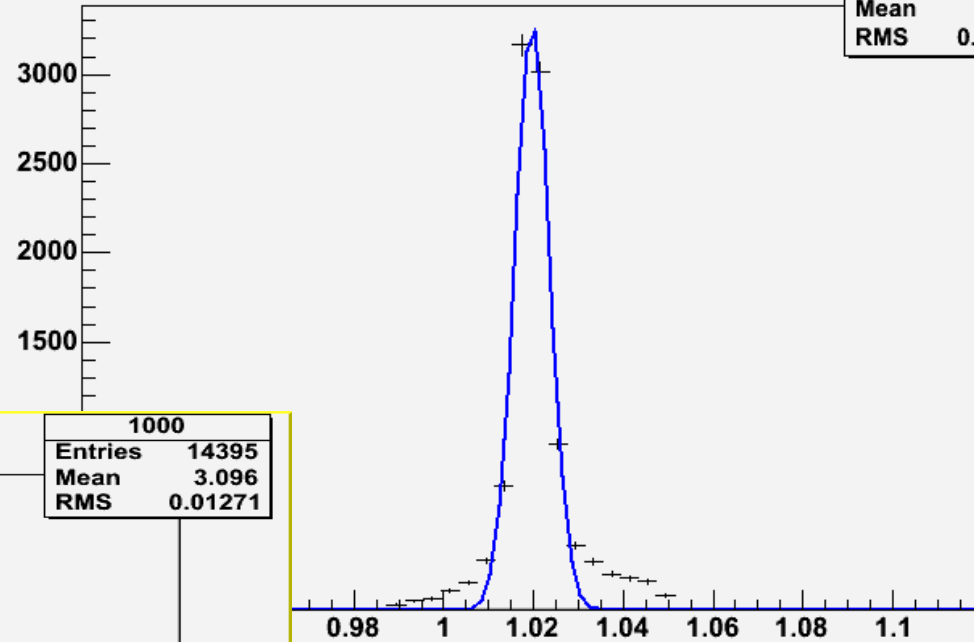
• Analysis DataFlow



• Analysis Results

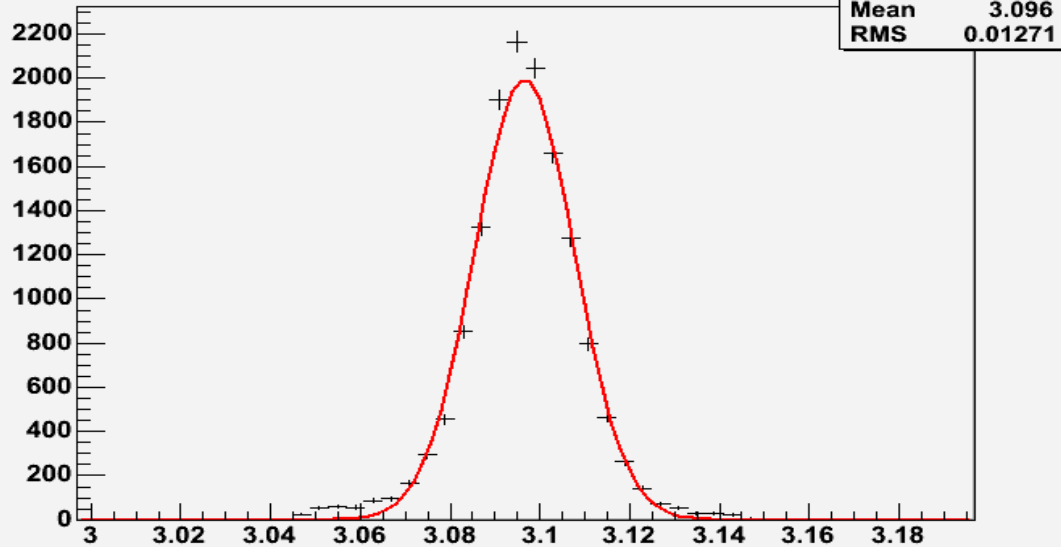
- Ran over 100,000 Events
- J/ψ Efficiency $\sim 14.4\%$
- Φ Efficiency $\sim 9.7\%$

SELECTED $\phi(1020)$ mass (GeV)



1100	
Entries	9703
Mean	1.021
RMS	0.008237

SELECTED $J/\psi(1S)$ mass (GeV)

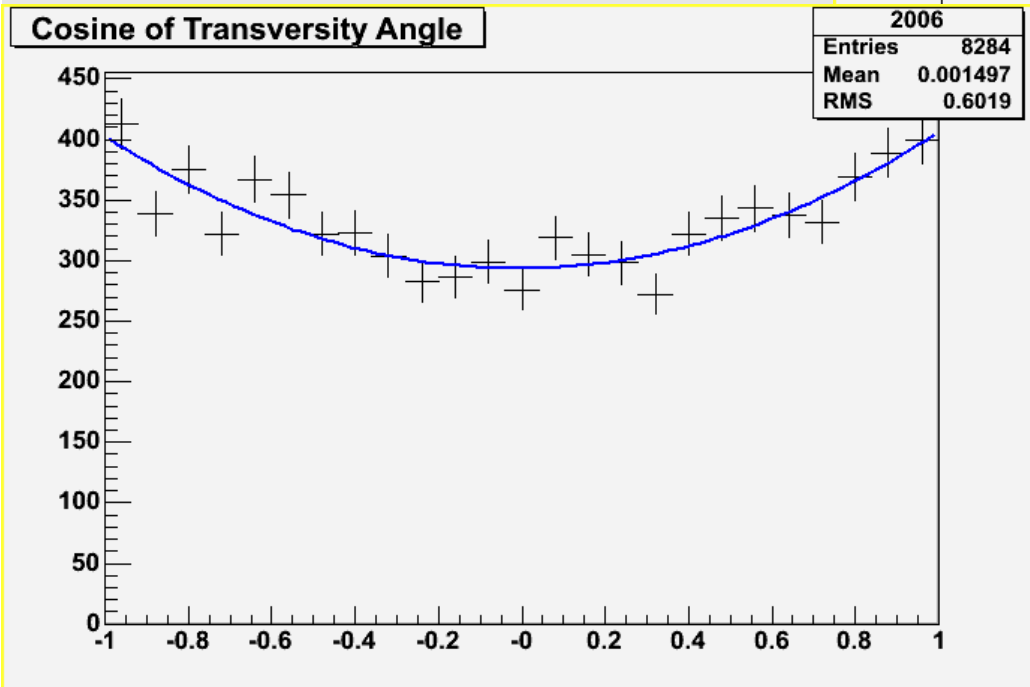
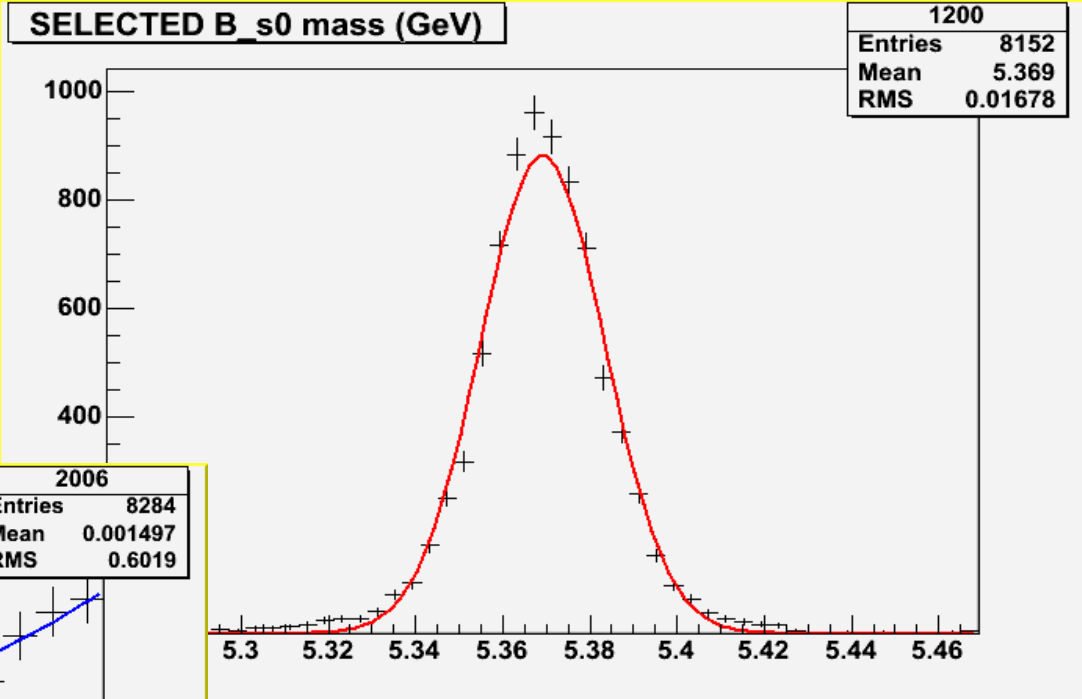


1000	
Entries	14395
Mean	3.096
RMS	0.01271

- Typical 1000 event job takes ~ 9 minutes

• Analysis Results 2

- Bs Efficiency $\sim 8.2\%$
- Transversity distribution promising



- Contributions from CP odd ($l=1$) & CP even ($l=0,2$) states
- Need to investigate helicity states in generation (EvtGen)
- Event selection is quite clean

• Problems Encountered

- Issues with reliability for individual jobs and commands
- File manipulation was cumbersome
- Some problems with datasets
- System was prone to hanging, often needed to be rebooted
- When problems occurred was often unclear whether user or the prototype itself was at fault

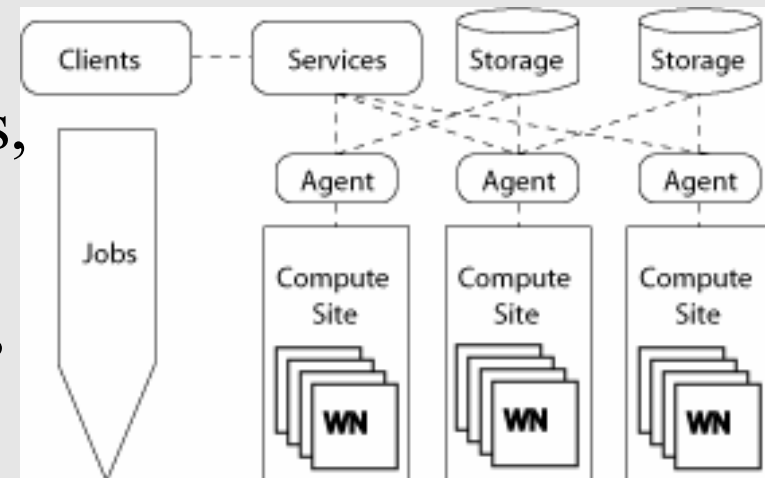
- Summary Of Progress Made With GLite
 - DaVinci was successfully introduced to GLite
 - Job splitting was possible
 - Use of the GLite package manager resulted in tarballs from release area being put straight to work
 - Large jobs over 100000 events led to the exploitation of Grid resources
 - The mechanism in place is scalable and could become autonomous for other versions of DaVinci
 - As it stands, LHCb users could utilise the GLite Prototype

- Enabling analysis for LHCb using DIRAC
- DIRAC – Distributed Infrastructure with Remote Agent Control

- LHCb grid system for Monte Carlo simulation, data production and analysis
- Integrates resources at LHCb production sites as well as on LCG



- DIRAC can be divided into services, agents and clients
- Service-oriented architecture makes DIRAC lightweight, robust and scalable



- Software Distribution in DIRAC
 - Decided to adopt LHCb system of software packaging by Florence Ranjard
 - Exports CMT package structure found at CERN
 - Relies on CMT for application setup and execution
 - This has been integrated into DIRAC for all of the LHCb production software as well as DaVinci

• Current & Future Work

- Am currently developing a new API for submitting jobs to the DIRAC Workload Management System (WMS)

- Issues with datasets

- Have to allow user to provide DLLs and options files

- Must allow real time monitoring etc.

- In the future, this will be extended to rewriting the DIRAC WMS for analysis

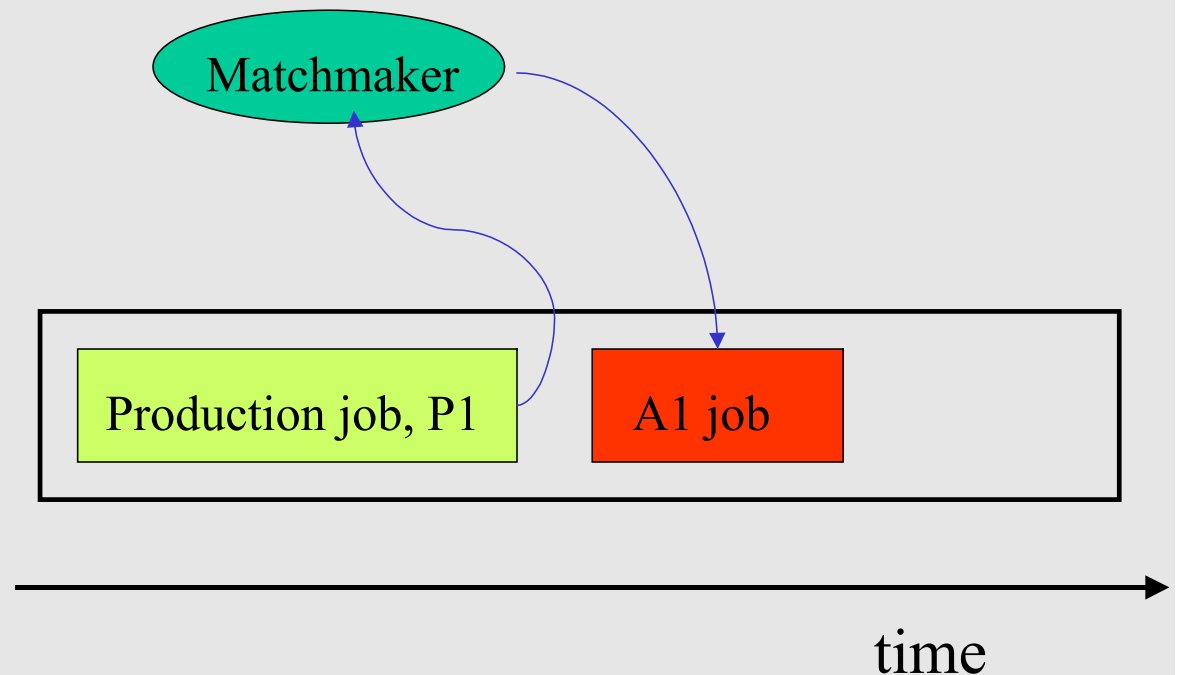
- Aim is to have concurrent production and analysis jobs

•Current & Future Work Contd.

•Analysis jobs will have priority over production jobs and sites will wait to execute analysis jobs before starting on the production jobs

•If no analysis jobs in queue then P1 will execute immediately

•Otherwise, analysis jobs will be executed until the time left at a site is equal to the time P1 will take.



• Conclusions

- Successfully automated Software Distribution for LHCb Using Pacman
- Distributed analysis was performed using DaVinci in the gLite Framework
- Future work will be to enable analysis for LHCb using DIRAC