

How to succeed using grid in High Energy Physics experiments.

James Cunha Werner
High Energy Physics, University of Manchester
Schuster Laboratory, Brunswick Street
Manchester M13 9PL
jamwer2000@hotmail.com

Abstract

This paper discusses the use of an eScience Grid (LCG) in providing computational resources for a modern particle physics experiment. A job submission system was written to provide users with easy access to grid resources to run their analysis programs over huge datasets integrated with experiment software packages and bookkeeping system. Programming Techniques were proposed to overcome grid and experiment software problems in an effective way. The proof of evidence was the study of tau decays in $N \text{ Pi}^0$ reconstruction from pairs of photons using a 2-layered recursive algorithm. Following techniques described in this paper, PhD students and researchers will be able to select events to test their hypothesis in less than 5 weeks.

1. Introduction

This paper presents how e-Science approaches can contribute to High Energy Physics (HEP) data analysis by providing access to large scale computing resources.

Conventional clusters require users have account and they are not integrated with other clusters to balance the workload. Integration comes with Internet connection and certificates that allows users a wide access. Grid infrastructure ^{[1][2]} was organized in a middleware layer, installed in dedicated computers (described in section 2), that provide resources access functionality for experiment software (section 3).

Integration of different catalogues and resources in a transparent way comes with the job submission system (termed EasyGrid, section 4). Easygrid provides users with commands to execute all tasks required, recovering results and searching information to support further analysis when something goes wrong. Users can increase grid efficiency adopting some programming techniques in their event analysis programs (section 5).

Section 6 shows a proof of case where a complete experiment project was developed. There was 5 million Monte Carlo events generated and 5 hundreds millions events searched from the real experiment using grid technology, which allowed the selection of 3 million events of interest, requiring 5 weeks of work.

2. Grid Architecture.

Grid computing provides the solution to HEP requirements. Grid middleware, working over the Internet, provides the necessary hardware infrastructure grouped in functional modules.

LCG grid ^{[3][4]} implementation has been used in HEP experiments due its characteristics of independent parallel processing. There are many worker nodes (WN), managed by Compute Elements (CE), running jobs directed by resource brokers (RB).

Users submit their jobs using the User Interface software (UI) installed in their laptops and desktops, which provides all infrastructures ^[5] to send a job to the RB, checking for certification and permissions automatically. RB matches CEs that can perform the task with information from general catalogues, and send the jobs to its queues for processing.

Catalogues with the location of each data file (Metadata) and its replicas are available centralized for each experiment (called virtual organization, VO).

Metadata management is performed by the Replica Location Service (RLS) ^[6].

Files and replicas can be kept on storage elements (SE) in several sites, to increase network perform and backup.

Users send parameter files and scripts for processing through the Input sandbox. Results come back through Output sandbox.

Jobs were run at Rutherford-Appleton Laboratory (800 CPUs) and University of Manchester (80 CPUs) grid facilities.

2.1 Standards definition.

The reason and benefits of standard definitions for all sites in one experiment are:

- It reduces time in software installation and configuration. Automation tools can be adopted.
- Centralized procedures allow immediate update and easy access.
- User's software arriving at any WN will be able to run, because all have the same configuration script, in any catalogued heterogeneous computer.
- Integrity and consistency check through well know benchmarks.

Some Standards are built in LCG and experiment software installation to implement an efficient environment.

The definition of closest SE to WN reduces network traffic, because analysis software binaries are too big to be deployed in the input sandbox. Experiment software sometimes do not need to be installed in a NFS resource, it can also be downloaded from SE in tar bowl format.

File names must be unique, and allow track back. Easygrid adopts filename stored in SE as a string with username, hostname, time stamp in the format HHMMSSDDMMYYYY, and file name. System managers can send emails using username@hostname because the certificate track back is too complicated.

The link between WN local software and user's application can be achieved very easily through the use of LCG standard software location (VO_experiment_SW_DIR). The initialization script should be called by the user's script to define all local variables and grant access to libraries and packages. In BaBar experiment ^[7] is:

```
. $VO_BABAR_SW_DIR/babar-grid-setup-env-rel.sh
```

This script initializes BaBar environment to run user's analysis for one specific release (defined by rel). Does not matter where system managers installed BaBar packages and condition and configuration database. After run this script, users are able to run

analysis program (termed BetaMiniApp) without any problem or further variable definition.

This allows system managers implement their software and hardware architectures in their site, independently from experiment software and users.

Publishing information about sites is implemented by LCG in 2 levels. LCG provides a publishing schema using Glue Schema ^[5] for general information about hardware resources (CPU, memory, maximum time queue, etc). LCG tags allow experiment system managers to define specific tags for their sites, such as available software releases, available datasets, etc. Each experiment should have its own tags standards to allow an efficient use of grid resources.

Glue Schema and tags are matched with information provided by EasyGrid in Job Definition Language (JDL) requirement clause. JDL files are generated automatically by EasyGrid to fulfill users requirements using those standards.

2.2 Upgrade policy.

Changes in operating system, LCG, and software configuration can be very disruptive. Guarantee compatibility between different library versions, packages, and users software can even more difficult. There are several packages involved (850 in BaBar experiment), from grid itself to compilers and operating system, and its relationship is not documented or even defined somewhere.

Upgrades can be a source of permanent or intermittent errors. Some incompatible libraries can generate dumps which allow trace the problem to some memory leakage, consequence of different number of parameters in the heap for example. Run benchmarks are a wise procedure to forecast and diagnostic errors.

Users must feel everything they achieve is guaranteed. Return to mark zero drive users crazy not only because it waist their time, but also because always before the upgrade things looks better.

The central point is: why should system managers upgrade, and when?

Every new software release brings some improvement, some times fixing errors or improving performance. Dealing with so many packages and products means every day there are something new. However, every upgrade should follow a period of maturation.

Some companies use to upgrade a once several products after season, or in maintenance stops. There is a meeting and strategic planning is draw to guarantee that most will continue without disruption. A contingency plan is also developed, just in case.

And never a non-return upgrade is done. Always be prepared to return to the old configuration.

However, working with grid technology is easy to upgrade reducing risks through the use of testbeds.

Testbeds are 10 computers running the production environment, without users, and available for tests and studies of system managers and developers.

Using a testbed independent from production, upgrades could be done in the following way:

- Plan all upgrades every few months. Nothing is so important that justifies stop production or put your reputation in risk.
- During the time between upgrades, evaluate impacts and benefits of each upgrade. Verify possible incompatibilities and interferences of new releases and different packages.
- When benefits are relevant, plan a date to upgrade only the testbed.
- Download all new packages in one directory called /work/upgrades/AAAAMMDD. Some packages take long to download.
- Implement upgrade in the testbed. Use "script /work/upgrades/AAAAMMDD/testbed.log" before login to have a log of all work in the terminal.
- Run benchmark software and evaluate real impact and benefits.
- ONLY IF BENEFIT IS EVIDENT, plan a date to upgrade production. Use "script /work/upgrades/AAAAMMDD/prod.log" before login to have a log of all work.
- Upgrade production and wait a week to verify no unpredictable errors occur in production. Now test is broad than testbed benchmark, and some problems can occurs.
- If the problem is fixed fast, fine. Otherwise, restore previous version in production and study the problem in the testbed. You should be able to repeat the situation in the testbed, as far both have the same configuration and software installed. The error can become part of the benchmark.
- After one week of no problems, start system tuning and optimization. Sometimes, optimization can generate problems in software configuration.

The point is to reduce the number of problems and complains, always present in a research environment, evolving to follow users requirements under new circumstances.

2.3 Users Production support.

Users need a permanent support to avoid waste of time. PhD students and Researchers have fixed term

contract, and their goals are develop an outstanding research and test hypothesis. It is not install software or understand how to use it. Most users will have the same problems as well.

To overcome these difficulties, web pages documentation ^[8] is an efficacious solution. Users can access 24 hours a day, get sample codes, study benchmarks and learn fast with pre-defined modules.

Modules necessary to select events from raw data are obtain histograms, bi-parametric histograms, column wise ntuples, row wise ntuples, listing data to comma delimited text file (csv format for integration with excel), Monte Carlo truth analysis, particle identification, and event reconstruction from raw data. A skilful physicist is able to develop them in 1 month or less and save a lot of time for users, system managers, support people, and colleagues.

Another important tool is the page with error messages and actions. Same errors occurs several times for different users, and a FAQ page can reduce the number of questions in the user support center.

Support center will keep web pages update and have time to deal with new problems or pointing users to FAQ pages. In any condition, users will fill supported by a helpful and care environment.

3. BaBar High Energy Experiment.

The BaBar experiment ^{[7][9]} studies the differences between matter and antimatter, to throw light on the problem, posed by Sakharov, of how the matter-antimatter symmetric Big Bang can have given rise to today's matter-dominated universe. High energy collisions between electrons (matter) and positrons (antimatter) produces other elementary particles (tau, pions, kaons, etc), giving tracks and clusters which are recorded by several high granularity detectors ^[10] and from which the properties of the short-lived particles can be deduced, allowing a clear understanding of universe origin and evolution.

3.1 Metadata: The Bookkeeper system ^[11].

Each recorded collision, called an event, comprises a large volume of data, and thousand of millions of events are recorded, giving a total dataset size of hundreds of thousands of Gigabytes (or hundreds of Terabytes).

The total event sample is separated into different *datasets* according to simple criteria, and a physicist will be interested in performing an analysis on a selected dataset.

Metadata information can be defined in different levels. Skims implementation are very restrictive

because most skims occupy terabytes of memory, and most sites do not have the complete skims, only some topologies. File implementation is very demanding because there are 700,000 files in BaBar experiment, and will require several queries per job submission. Dataset implementation is the best approach, because it requires only one access per job, requires reasonable amount of space to store physical data, and is how physicists think about their tasks.

To manage the files of each dataset, there is the bookkeeping system. Physicists can recover datasets file names that matches requirements.

These requirements, from grid point of view, are:

- Select the necessary number of data files that contains some number of events. The number of events defines the processing time and can be optimized to each grid environment.
- Select data files from some date. Users want to update their selected events dataset since the last processing, without have to do all over again.
- Some systems provide a remote query, which allows users know what datasets are available in a remote site, for remote job submission.
- Define the initial number of sequence for data files. Users will be able to have a history directory with all dataset files that already were processed.

Bookkeeper system is updated every night to guarantee synchronization between all sites.

3.2 Analysis framework ^[12].

Experiment software contains all necessary packages to perform all users tasks. However, users are required to develop their own analysis software that uses experiment packages in a specific way to select events of interest for a specific decay.

Analysis is computationally intensive owing to the size and number of the event files involved; a single dataset may contain many thousand event files, each containing thousands of events. Using a single node would take a prohibitively long time.

Each event file is processed by the same analysis program so several CPUs can be used in parallel to reduce the elapsed time. However the limits of single clusters are now being reached, and a single analysis job needs the power provided by many clusters distributed across the world.

3.3 Monte Carlo Generation.

Monte Carlo is the technique to test the hypotheses and understand the data from experiment. Usually, it contains 3 stages. The first stage generates events

based in branch rates, physical model parameters and conservation laws. The second stage introduces detector transfer function, producing all the data in a “simulated detector”. The last stage reconstructs the simulated event providing all physical information.

Match Monte Carlo events with experimental results require a good understanding of physics processes and detector. Fit data with Monte Carlo events means a complete understanding of all of it.

4. EasyGrid: layer between grid middleware and experiment software.

The EasyGrid Job Submission software is an intermediate layer between the Grid and user’s analysis software (written in C++).

EasyGrid’s first task is to find what event files are in the dataset, and where are they available. There are 3 ways to perform the task: using Bookkeeper (if a centralized catalogue is available), RLS or VO tags.

RLS option has for each dataset a metadata file containing the names of the event files. These physical files are registered with the RLS, with several logical file names in the format *datasetname_CEJobQueue* assigned to them as aliases, showing the CEs which contain copies of that dataset. If a CE holds any of the files in a dataset it holds all of them. Searching all the aliases for a dataset name provides a list of CEs to which jobs can be submitted. One interesting advantage is this method can be used to store file handlers of files stored in dCache or other file system, providing the link between the logical file name and its physical storage.

VO tags can publish, for each site, which datasets are available. In this case, the requirement clause will provide all necessary information to distribute the jobs in the organization sites, without need for any search.

The list of CEs defines the SEs/NFS that will store analysis software binary and large parametric files to minimize network traffic. EasyGrid provides all necessary procedures to store files remotely.

The next stage is generation of all necessary information to submit the jobs on the Grid. This is done by the GEnerator of Resources Available (GERA) which produces the Job Description Language (JDL) files, the script with all necessary tasks to run the analysis remotely at a WN, and some grid dependent analysis parameters. The JDL files define the input sandbox with all necessary files to be transferred, and a WN balance load algorithm matches requirements to perform the task optimally. When the task is delivered in the WN, scripts start running to initialize the specific BaBar

environment, and the analysis software binary is downloaded from SE and run.

Users can follow up the process querying job status. If the job is done, a task recovering results in the user's directory is performed automatically. If the job was aborted in the process, the diagnostic listing is stored in the history file for further analysis.

EasyGrid was developed using the RAP (Rapid Application Prototyping) methodology. Several versions were developed, covering different approaches and functionalities, and two applications were used as examples, which allowed us to evaluate each strategy and acquire information to write the production system specification.

5. Developing analysis software in grid environment.

The use of analysis framework provide all classes and variables inherit from the framework to allow users access to selectors to identify particles and analysis object classes to access the physical information from the event. Users develop several modules for different tasks, and can run more than one at same time.

There are 3 main functions in each user program:

- Initialization (or constructor), which performs all initial tasks, such as initialize histograms and ntuples objects.
- Events procedure runs for each event performing calculus and evaluation. This procedure will verify if the event fulfils users requirements to be selected and counted in the histograms.
- Destructors, which performs all finalization procedures, such as generate the histograms output for later analysis.

5.1 Programming Techniques.

Framework structure does not allow the definition of entry points and recovery, because in case grid crashes, there is no easy way to store state files for later reprocessing.

However, the framework can print a message every 1,000 events and can restart jumping any initial number of events.

The idea is to develop software in two stages. The first stage selects events and stores them in NFS or SE. The second stage will evaluate histograms and perform analysis, and will not require grid resources because it reads only pertinent information for analysis. The second stage can be run in 10-20 minutes in the user's desktop/laptop.

In case grid, framework or file servers have crashed, users can recover the process jumping to the last event

done properly. If the problems are events themselves (sometimes selectors abort, or there are inconsistencies in the data), the user can jump 1,000 events and resume processing.

5.2 Reconstruction algorithm using recursive techniques.

The most important algorithm in the event selection is the one to reconstruct particles and resonance from other tracks and gammas. For example, Pi^0 is reconstructed from 2 gammas that have invariant mass between 130 and 150 MeV.

Invariant mass is defined as:

$$M^2 = \left(\sum E_i\right)^2 - \left(\sum P_i\right)^2$$

where M is the invariant mass, E is the energy and P is the momentum of each particle considered in the reconstruction.

The algorithm must take in account:

- Every particle should be used only once. One gamma cannot come from 2 different Pi^0 's.
- The combination should minimize error in the invariant mass. If a pair of gammas is an invariant mass candidate, there would not be a better pair of available tracks/gammas with lower invariant mass error.
- Users can be able to combine any number of particles and gammas.

Implementation of recursive algorithm for reconstruction follows the algorithm described in the web page^[8].

The advantage of the algorithm is does not evaluate all possibilities (a exponential explosion problem). This algorithm was used in the case study to reconstruct Pi^0 's.

6. Case study: tau decaying to N Pi^0 .

Electron and positron collide in BaBar detector and can produce events with a pair of tau particles^{[13][14]} (one positive and other negative). Both can decay to intermediate resonance and finally to N Pi^0 , and there will be difficult to reconstruct the event because it will be difficult know from which tau the particle come from. To overcome this difficulty, we use to select tag decays where there are not any of the particles we want to study. The tag decay was tau decaying in $\mu \nu_\mu \nu_{\text{tau}}$ (no pions at all).

The first condition (cut) for event selection was only one muon in the event. BaBar particle identification package provides several muon selectors. The choice

of the selector depends how pure sample are required and the compromise with number of events (more tight the selector, less events will be selected, but the sample will be more pure).

Monte Carlo simulation shows the selector's quality. Table I, column A shows selector's accuracy of 1 muon events. It is correct in 81% of events. Column B shows the result for events that have also 1 charged pion (only 63% are correct). The events with one muon and one charged pion can decay in a resonance and later in 0,1, 2, 3, or 4 pi's (column C, 70% correct).

Selected number of events from experimental data (BaBar Run1, 2, 3, and 4) is shown in Table II. There are few 4 pi0s decays because its branch rate is too little.

Decays can contaminate other decays if one or more gammas are lost or background is mingled with a real gamma. Table I shows the contamination for each decay.

The next step is to reconstruct each resonance's decay using the recursive algorithm. Resonances are Rho-770MeV for 1 Pi0 (Fig.1), A1-1260MeV for 2 Pi0s (Fig.2), Rho-1450MeV for 3 Pi0s (Fig.3), and Rho-1450MeV+Pion for 4 Pi0s (Fig.4).

Statistical error difference between Monte Carlo and experimental data are quite visible. Monte Carlo data was multiply by 18 (instead 482/4=98) to be in the same scale as experimental data due efficiency of trigger and skims classification are not considered in Monte Carlo simulation.

Despite resonance peak have been fitted quite well by Monte Carlo events, background is not so precise. The reason can be the limited decays defined for MC generation – only tau+ tau-. This can limit the contribution to the background because due the selection errors, events from different channels can be contaminating the experimental data. Asymmetry peaks can be explained by Landau distribution^[15].

Adherence between matter and antimatter results in all plots are a clear proof that software, grid, catalogues, and everything else are running without error. It is also a demonstration that there is not CP violation in this tau decay mode, and there is not a way to explain how the matter-antimatter symmetric Big Bang can give rise to today's matter-dominated universe considering only these decays.

Invariant mass is the start point to test models, hypothesis or just feelings. The next step is to adjust parametric models to the shape of invariant mass to verify if particle characteristics predicted in the models are correct. Table III shows the results for Breit-Wigner invariant mass modeling given by:

$$F(m) = \frac{1}{2\pi} \frac{\Gamma(m_R)}{(m_R - m)^2 + \Gamma(m_R)^2/4}$$

The further development will be to obtain spectral functions to understand QCD/QED^{[16][17][18][19]}. It is where physicists' jobs begin.

Table I Monte Carlo simulation for tau+ tau- decays. Bold values are the correct selection.

	Particle	#	Forecast					Total events
			0	1	2	3	4	
A	Muon	1	6	81	13	0		4,890,000
B	Pion	1	21	63	11	4	1	350,523
C	Pi ⁰	1	2	72	23	3		115,894
		2	1	6	74	17	2	22,552
		3	0	2	11	68	17	2,072
		4	0	0	3	15	68	160

Table II Experimental data for tau+ tau- decay. Percentage of each possible particle.

Partic	#	% Number of particles					Total events
		0	1	2	3	4	
Muon	1	71	21	8	0		482,303,947
Pion	1	29	54	11	15	2	101,928,861
Pi ⁰	1		82	15	2	0.3	3,487,564

Table III Breit-Wigner mass distribution modeling.

	1Pi ⁰	2Pi ⁰	3Pi ⁰
Normal	4.5 10 ⁶	1.0 10 ⁶	7.2 10 ⁴
Mass	770	1110	1410
Width	160	380	380

7. Conclusion.

This paper presents a successful use of grid technology in High Energy Physics by one physicist without HEP skills. I have worked all my life developing industrial modeling, optimization and automation using distributed technologies and artificial intelligence.

Select events with some particular characteristic in common are a quite demanding process. The case study searched 500 million real data events and generated 5 million Monte Carlo events. Proposed reconstruction algorithm using recursive techniques

was very fast and flexible. It was used in reconstruction of $Pi0$ s and resonances.

EasyGrid job submission system and grid provide the resources to do it in reasonable time, despite grid still in development and implementations are experimental.

EasyGrid framework was efficient, fast and user friendly dealing with crashes and aborts. Programming techniques help users save time in case something goes wrong. At the end of the day, develop research as PhD or fixed term contracted requires things done fast, to have enough time to learn about HEP and produce thesis and papers.

The important contributions of Grid use in HEP are:

- Fast results dealing with enormous databases (Terabytes). This allows researchers to test different approaches and have results faster allowing a more efficient use of time.
- Generation of automatic procedures such as EasyGrid (users have time to think).
- Sharing costs and balancing available resources. Researchers are not submitting everyday. There is a cycle process of submission, analysis, discussion, and resubmission. Grid allows user access to resources throughout the world when they need them, and makes them available to others when they do not. With enough users the load will balance to a large extent.

7. Acknowledgements.

The author would like to acknowledge GridPP/UK for funding this project. I also would like to thank Professor Roger Barlow, Grid UK people and BaBar collaboration colleagues for interesting discussions.

References

- [1] Foster, I.; Kesselman, C.; Tuecke, S.; "The anatomy of the Grid" International J. Supercomputer Applications, 15(3), 2001, available online from www.globus.org/alliance/publications/papers/anatomy.pdf
- [2] I. Foster, C. Kesselman, J. Nick, S. Tuecke; "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration" Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002 available online from <http://www.globus.org/alliance/publications/papers/ogsa.pdf>
- [3] CERN, "LHC Computing Grid"; <http://lcg.web.cern.ch/LCG/>
- [4] CERN, "LHC Computing Grid Technical Design Report" available online from http://lcg.web.cern.ch/LCG/tdr/LCG_TDR_v1_04.pdf
- [5] EIS team, LCG-2 User Guide <https://edms.cern.ch/file/454439/LCG-2-UserGuide.pdf>
- [6] Data Grid, "Replica Location Service" at <http://edg-wp2.web.cern.ch/edg-wp2/replication/replica-location-service/>
- [7] BaBar Collaboration, "The BaBar experiment home page", <http://www.slac.stanford.edu/BFROOT/>
- [8] Werner, J.C.; "HEP analysis, Grid and EasyGrid Job Submission Prototype: Babar/CM2 showcase" at <http://www.hep.man.ac.uk/u/jamwer/>
- [9] Harrison, P.F.; Quinn, H.R.; "The BaBar Physics Book" SLAC Report 504, October 1998, available at <http://www.slac.stanford.edu/pubs/slacreports/slac-r-504.html>
- [10] BaBar Collaboration; "The BaBar detector", Nuclear Instruments and Methods in Physics Research A479(2002) 1-116 available at <http://www.hep.man.ac.uk/u/jamwer/babarnucl.pdf>
- [11] BaBar Collaboration; "BaBar Bookkeeping - a distributed meta-data catalog of the BaBar event store." CHEP04 available at <http://indico.cern.ch/contributionDisplay.py?contribId=338&sessionId=7&confId=0>
- [12] BaBar Collaboration; "Workbook for BaBar Offline Users" available at <http://www.slac.stanford.edu/BFROOT/www/doc/workbook/workbook.html>
- [13] Karlsruhe, J.K. et al.; "Physics with Tau Leptons" Springer Tracts in Modern Physics 160, Springer, 1962.
- [14] Particle Data Group, "Particle Physics Booklet", July 2004, available at <http://pdg.lbl.gov/pdg.html>
- [15] Perkins, D.H.; "Introduction to High Energy Physics" 3rd edition, Addison-Wesley Publishing Co, Inc, 1987, pg 40.
- [16] Davier, M.; Hocker, A.; Zhang, Z.; "The physics of hadronic tau decays", submitted to Reviews of Modern Physics, available at High Energy Physics – Phenomenology, <http://arxiv.org/pdf/hep-ph/0507078>
- [17] Davier, M.; "Branching ratios and spectral functions of tau decays: final ALEPH measurements and physics implications" submitted to Physics Reports available at <http://arxiv.org/pdf/hep-ex/0506072>
- [18] Anderson, S.; "Hadronic Structure in the Decay tau-to pi-pi0 nu_tau" Phys.Rev. D61 (2000) 112002 available at <http://arxiv.org/pdf/hep-ex/9910046>
- [19] Browder, T.E. et al.; "Structure Functions in the Decay tau -> pi+- pi0 nu_tau" Phys.Rev. D61 (2000) 052004 available at <http://arxiv.org/pdf/hep-ex/9908030>

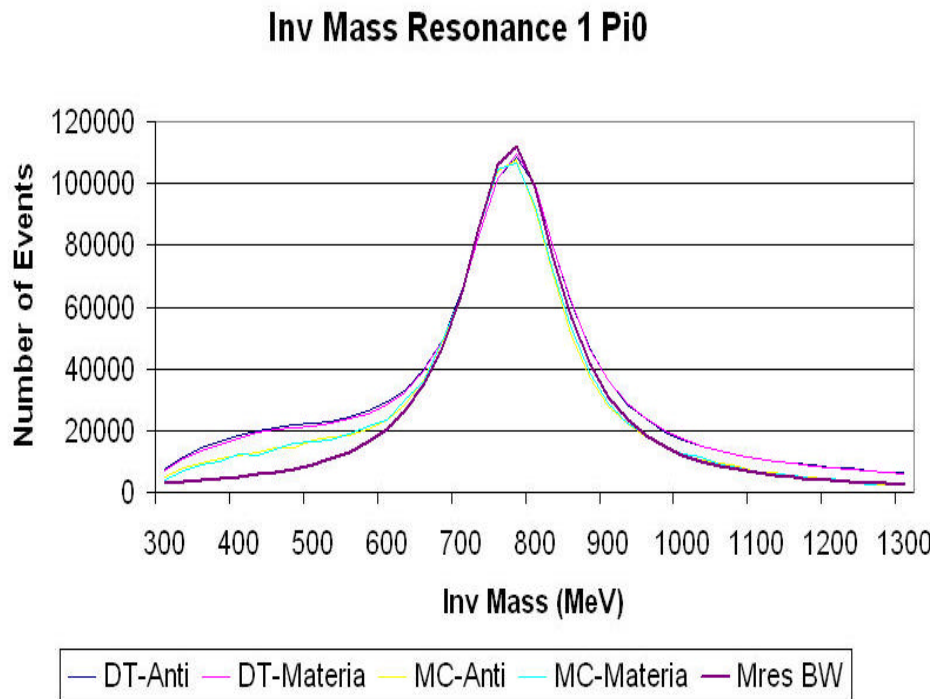


Fig. 1 Rho 770 Mev resonance invariant mass, 1 Pi⁰ decav.

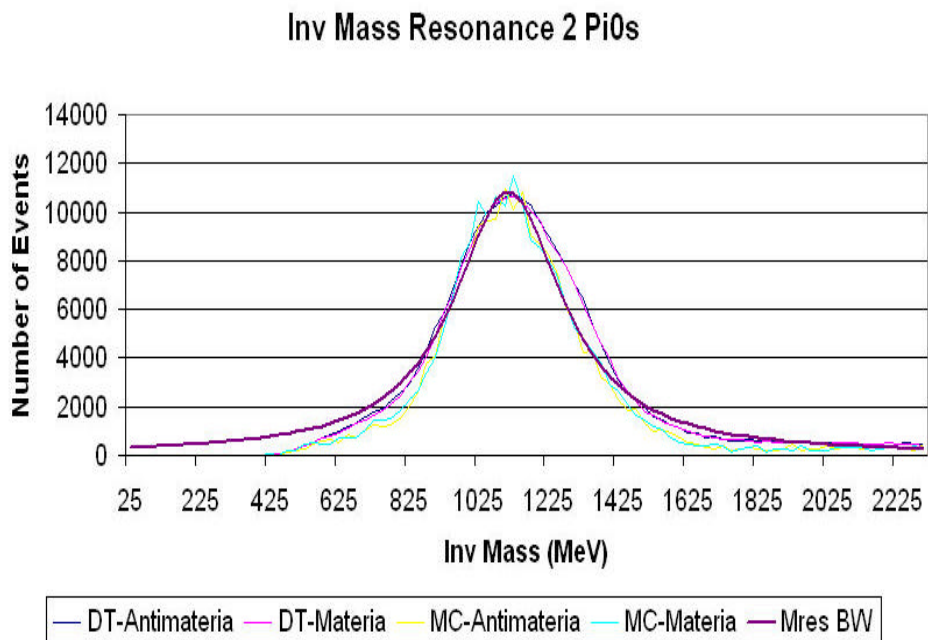


Fig.2 A1 1260 Mev resonance invariant mass, 2 Pi⁰s decay.

Inv Mass Resonance - 3 Pi0s

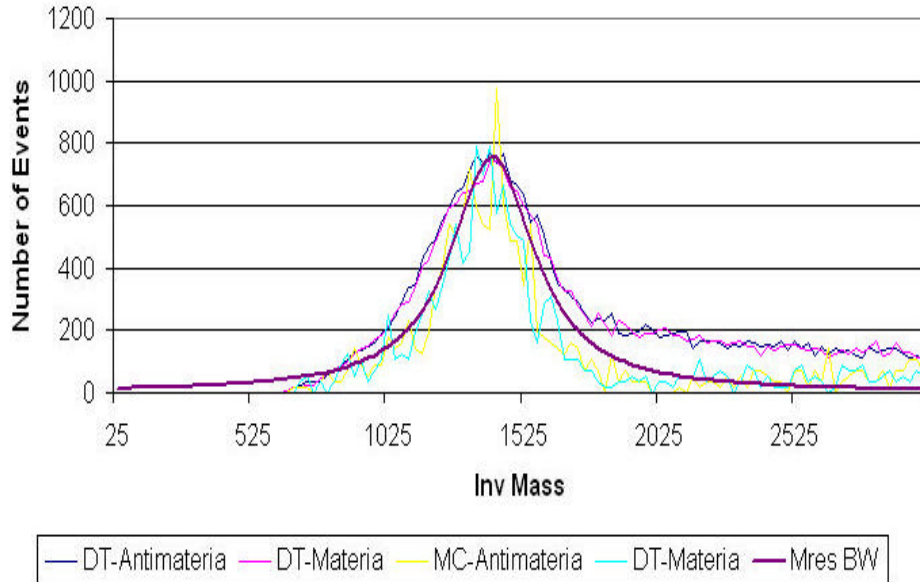


Fig.3 Rho 1450MeV resonance invariant mass, 3 Pi⁰s decay.

Inv Mass 4 Pi0s

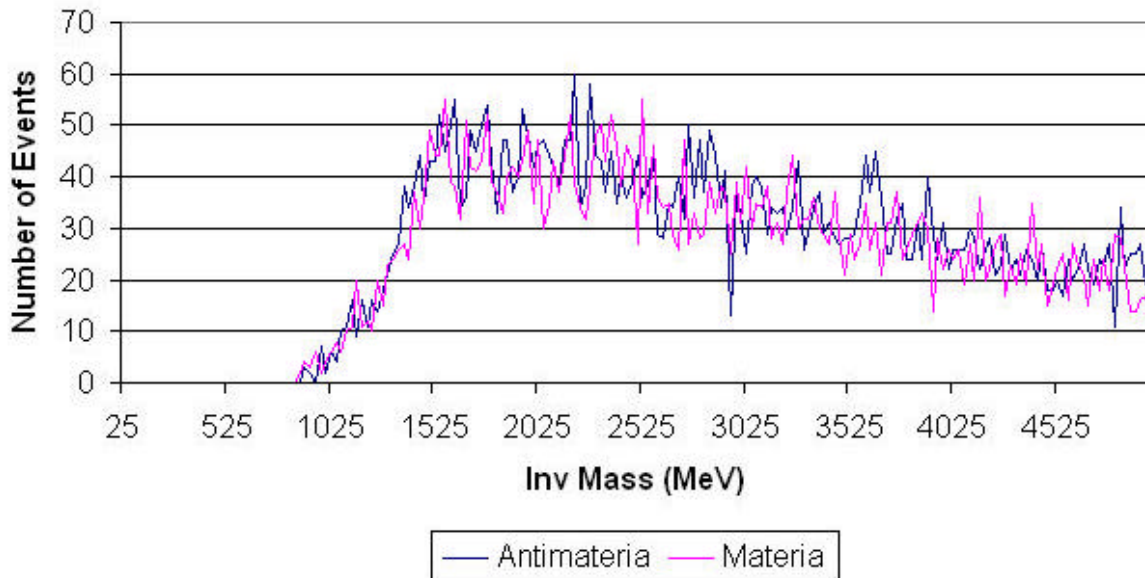


Fig.4 Rho 1450+Pion resonance invariant mass, 4 Pi⁰s decay.