



Clarens Grid Portal Toolkit

Conrad Steenberg, Julian Bunn, Frank van Lingen, Michael Thomas, Harvey Newman, Faisal Khan





Overview

◆ **What is Clarens? (CGPTK?)**

- ➔ **Architecture**
- ➔ **Web services**
- ➔ **Web portals**
- ➔ **Clients**

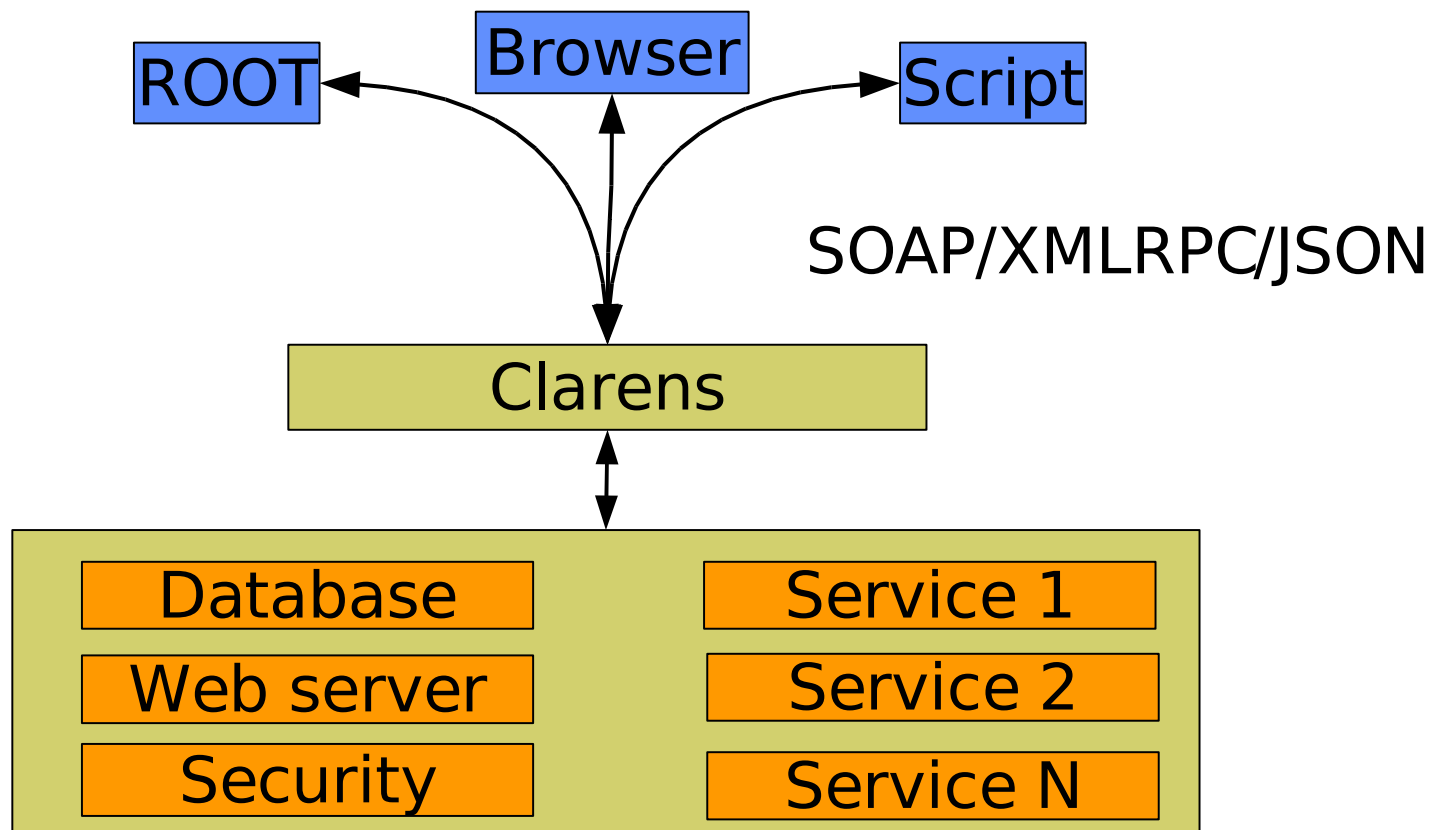
◆ **Demonstrations**

- ➔ **AJAX interfaces**
- ➔ **ROOT analyses**
- ➔ **Python scripts**
- ➔ **Java GUI**



What is Clarens?

- ◆ **Set of APIs for building loosely coupled distributed systems**
- ◆ **Currently implemented as Web services**
 - ➔ **3 implementations: Apache/Python, Java/Tomcat, pure Python**

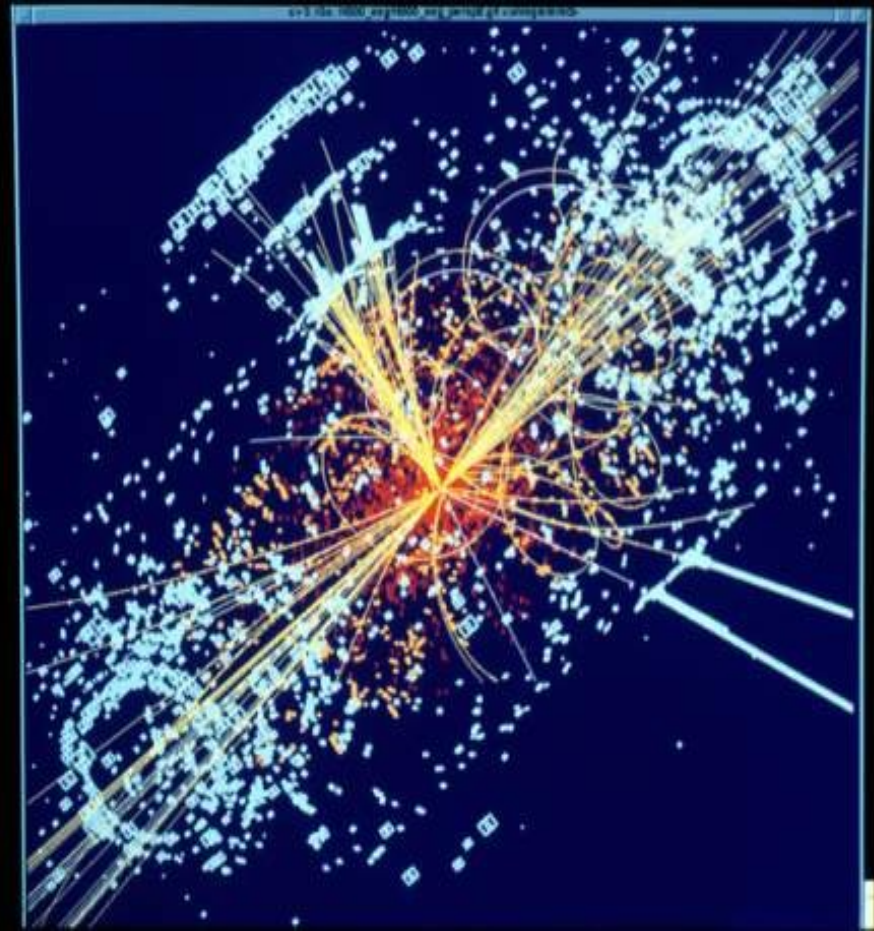
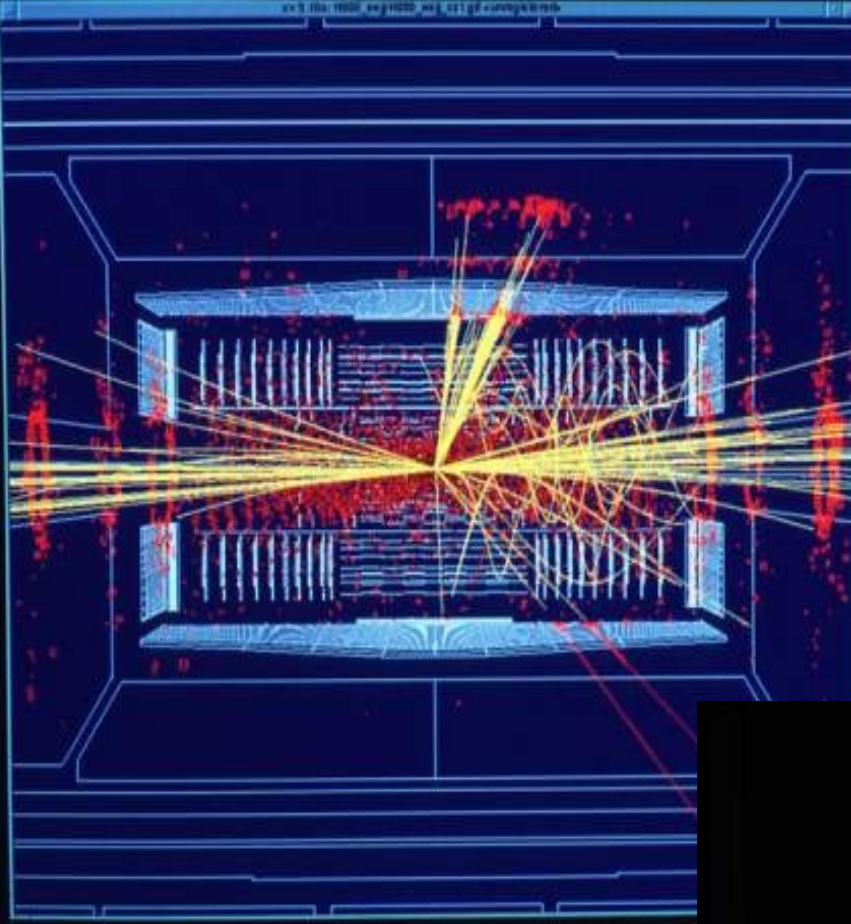




What is Clarens? (Historical)

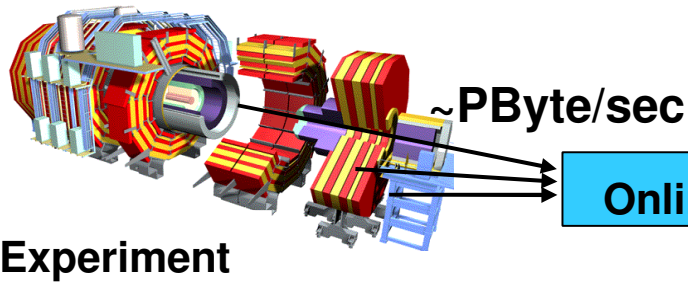
- ◆ **Work started in early 2001**
- ◆ **Motivation was to build simple, high-performance system to provide access to CMS data/compute resources – Grid Analysis Environment (GAE)**
- ◆ **Implemented Python layer on top of Apache web server using mod_python**
- ◆ **Used XMLRPC as serialization mechanism with Java Analysis Studio (JAS) as client**
- ◆ **First data access was JAS accessing CMS data in Objectivity OODB**

Physics analysis



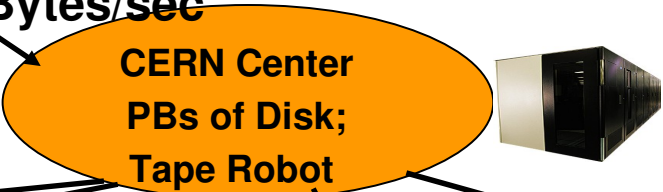


LHC Data Grid Hierarchy:



CERN/Outside Resource Ratio ~1:2
 Tier0/(Σ Tier1)/(Σ Tier2) ~1:1:1

Tier 0 +1

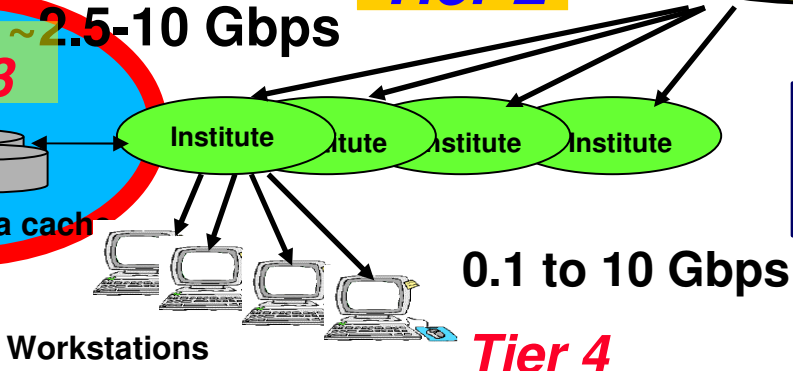
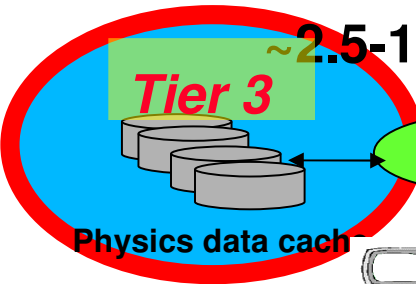
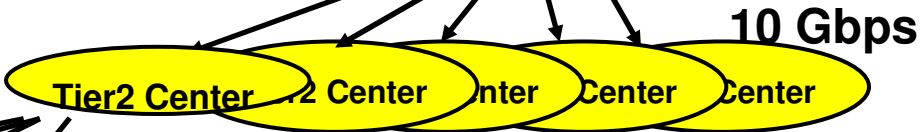


Tier 1

~10 Gbps



Tier 2



Tens of Petabytes by 2007-8.
 An Exabyte by ~2015.

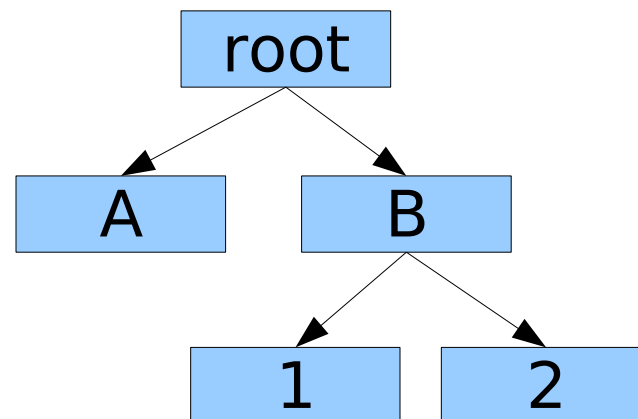


Web Services



Security I

- ◆ **Any distributed system exposed to the internet needs strong security!**
- ◆ **Authentication based on X509 certificates/private keys**
 - ➔ **Allows easy access using any browser over HTTPS**
- ◆ **Can use proxies generated by GTK 2+**
- ◆ **Built-in hierarchy of groups and subgroups based on X509 Dns**
 - ➔ **Virtual organization (VO)**





Security II

- ◆ **Authentication done for all web service calls based on VO**
 - ➔ **Specified in file, analogous to .htaccess in Apache**
- ◆ **File access authentication**
- ◆ **Authentication machinery available for use by service writers to protect server-side resources**

```
access=[("file",[ORDER_DENY_ALLOW,          # Order
        [],          # Allow individuals
        ["admin"],  # Allow group
        [],          # Deny indiv
        ["unc-chapel-hill"], # Deny groups
        [None, None, None]])] # modtime, start_time,end_time
```



Standard services (classes)

- ◆ **System (method ACL admin, reflection, documentation)**
- ◆ **Group (VO administration: create, edit, delete groups)**
- ◆ **File access (Read, list, ACL admin, write)**
 - ➔ **File upload presents a problem in web services: don't want to serialize large files**
 - **Use HTTP put**
 - **HTTP POST with multipart-mixed content type 1st part is service call, 2nd part is file**
- ◆ **Proxy service: store, retrieve proxies with username/password**
 - ➔ **Can also authenticate using username/password**
- ◆ **Shell service: execute shell commands in sandbox as different user**
 - ➔ **Uses gridmap-file, or more complex VO-based authorization**



Web Portals



Browser support I

- ◆ **Since we have a web server and have a file service, why not respond to HTTP GET requests?**
 - ➔ **HTTP GET /clarens/dir/file.txt**
 - ➔ **file.read("/dir/file.txt",0,-1)**
 - ➔ **Apply access control to all file access**
- ◆ **Also support file upload**
 - ➔ **Using HTML forms**
 - ➔ **file.write("/clarens/dir/file.txt",0,size)**
- ◆ **Put together, this makes for a very powerful AJAX portal toolkit!**
 - ➔ **Serve Javascript code to be executed in a browser**
 - ➔ **Can use any AJAX GUI toolkit**
 - ➔ **Added Javascript Object Notation (JSON) serialization format**



Browser support II

- ◆ **Through `mod_python`, supports Python Server Pages (PSP)**
- ◆ **PSP scripts are python code executed on the server side**
- ◆ **Have full access to all Python libraries, including Clarens code**
 - ➔ **Observe standard server-side security precautions!**




Standard Clarens portal

Clarens: envoy.cacr.caltech.edu:8443 - Mozilla Firefox

File Edit View Go Bookmarks Tools Help

https://envoy.cacr.caltech.edu:8443/clarens/



Service Index

Session
Login
Logout
Proxy Management
User tools
Information Service
File Access
Proxy Service (advanced)
Service Registry
Administration
VO Management
Method ACL Management
Hotgrid
New account
Validate account
Renew account
Login
Logout

Clarens Web Interface

How to use this page

Click on one of the service links in the navigation panel on the left. If the panel is empty, your browser may not be supported.

To return to this page, click on the words Service Index in the navigation panel.

File browsing

- Any directory can be browsed by appending `auto_index.html` to the URL
- If an `index.html` file exists in a directory, that will be loaded when a directory URL is specified
- If no `index.html` file exists, a directory listing will be displayed instead
- Try this out [now](#).

More Information

- Global service [repository](#) and search page.
- Clarens [Home](#)
- Mailing [list](#) for questions or comments.
- Project [page](#)



Pythia portal

Mozilla Firefox

File Edit View Go Bookmarks Tools Help

https://envoy.cacr.caltech.edu:8443/clarens/web/pythia/pythia.html

General News Travel Programming Employment Search HEP Bank & Invest Linux Motors Google

clarens Pythia Portal Prototype

This is a Portal to the [CMKIN](#) Pythia application. [Pythia](#) is a program for simulating the physics of particle collisions. The input data to Pythia are specifications of which type of collision is required, and the number of collisions (events) that are to be generated. This Portal offers several pre-built input data "cards" files. For Portal users who enter the page with a sufficiently strong certificate, the Portal allows a custom data cards file to be created by editing a base file, and an arbitrarily large number of events to be generated. Once the Pythia/CMKIN application is run, the output data (an ntuple in HEPEVT format) is made available for download from the Portal. The output data is typically used as input to a full HEP detector simulation program, which tracks the collision products through the detector volumes.

Portal for: Conrad Steenberg 178947

The procedure to use this Portal is:

1. Select what types of events are to be generated using the "Select Datacards" section
2. Modify the datacards (if required)
3. Run Pythia
4. View output, and download the produced ntuple to your system

1. Select Datacards

Select	Filename	Description
<input type="radio"/>	litvin_higgs.txt	Higgs to two photons in 14 TeV proton proton Collisions
<input type="radio"/>	beam_halo_gen.txt	Beam halo at 14 TeV

2. Modify Datacards (if required)

```
LIST
C
C----- Start of channel independent control cards -----
```



Clients

- ◆ **Want to maximize the number of ways to interact with distributed system**
 - ➔ **High level of resistance to having to install a GB+ blob of code just to submit a job or access a file**
- ◆ **Use of multiple protocols makes convenient access from variety of clients possible, most common being**
 - ➔ **Browser (Javascript, HTML)**
 - ➔ **Java client**
 - ➔ **C++ ROOT client**
 - ➔ **Python**



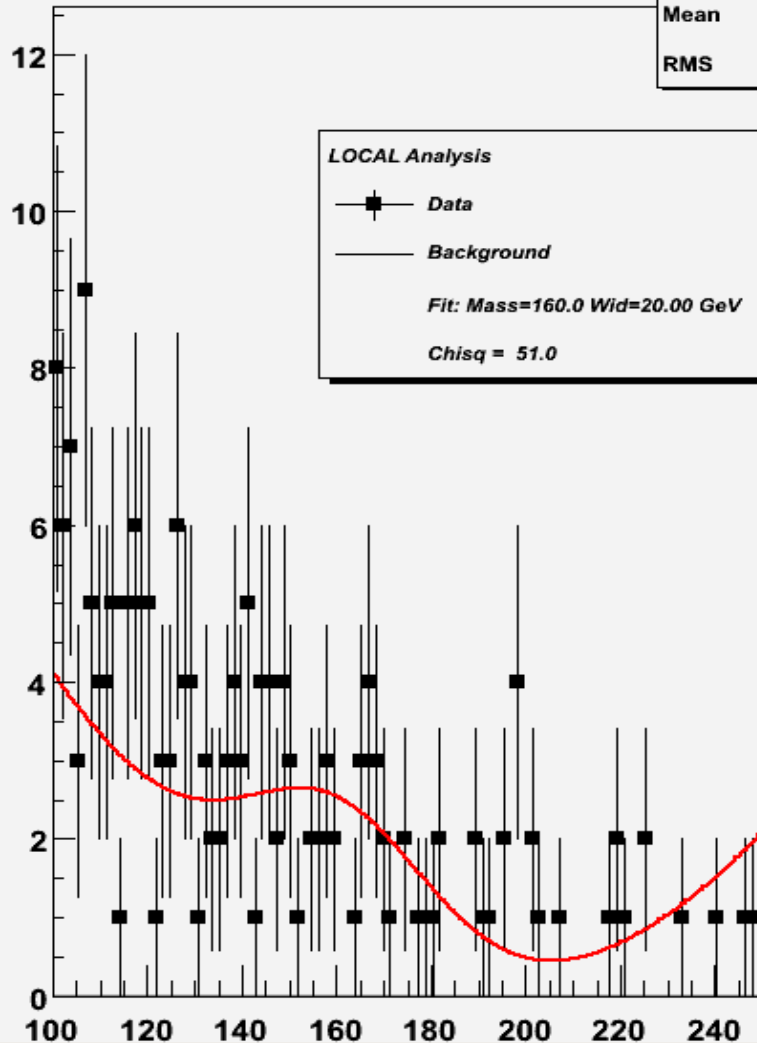
ROOT C++ Client

Analysis of Remote Data using Rootlets

Invariant Mass (Correction S25) of Supercluster Pairs (GeV/c²)

hRecMass1

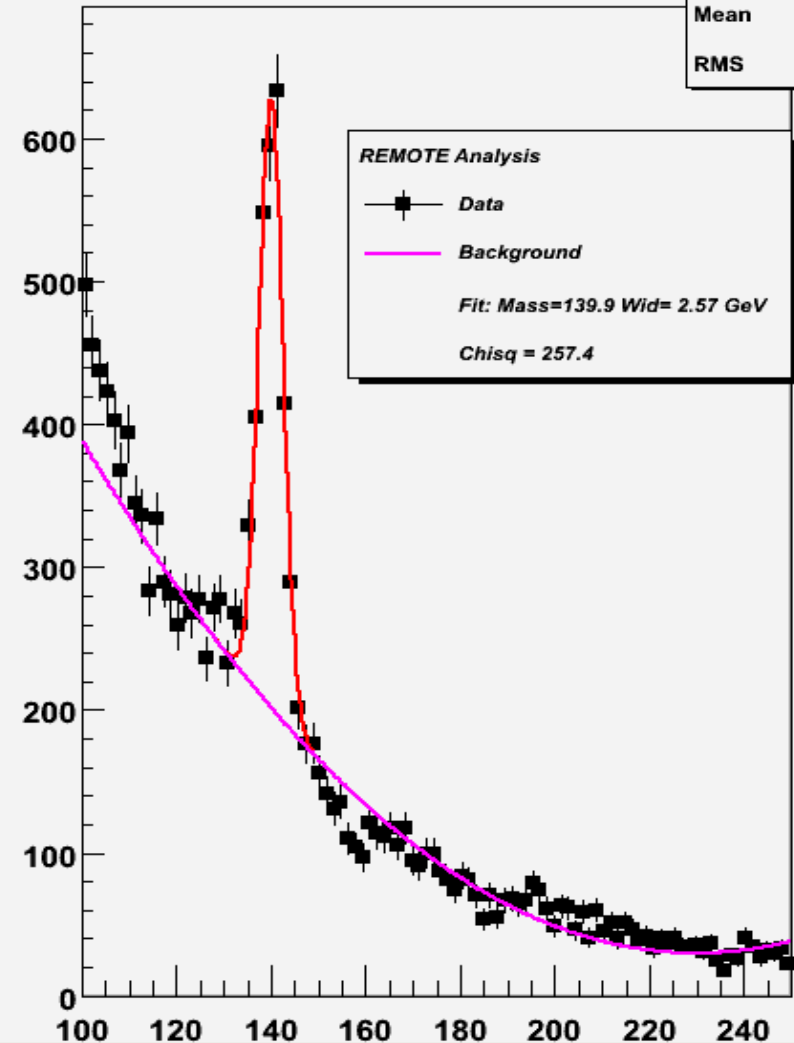
Entries	968
Mean	140.3
RMS	34.55



Invariant Mass (Correction S25) of Supercluster Pairs (GeV/c²)

hRecMass1

Entries	68714
Mean	141.3
RMS	33.88





Demonstrations



“Rootlet” services

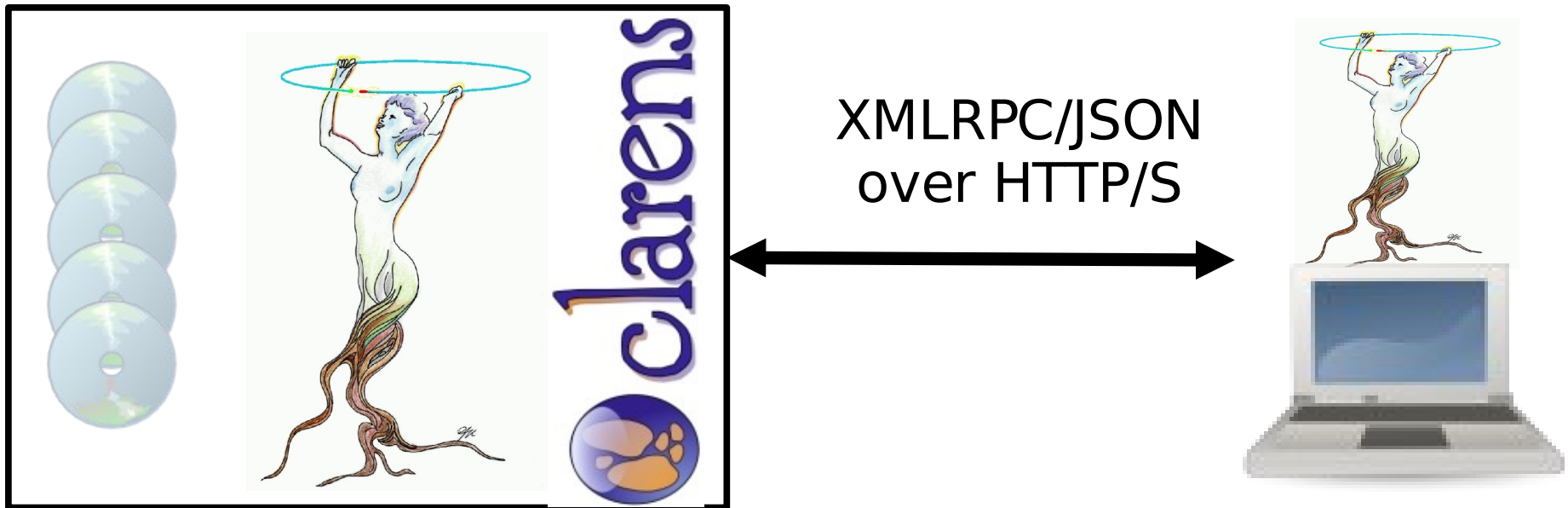
- ◆ **In analogy to a *servlet* a *Rootlet* is a dedicated instance of Root on the server**
- ◆ **Similar service also exists for Pythia**
- ◆ **Services allow parameters to be passed to server instances of Root and Pythia**
 - ➔ **Allow more controlled invocation of Root or Pythia**
 - ➔ **Implemented graduated authorization system in collaboration with NVO/Nessi**
 - **Guest, weakly authenticated, and strongly authenticated users receive different amounts of CPU time**
- ◆ **Browser-based “portal”**
- ◆ **Root client interfaces to services**
 - ➔ **Access remote files using clarens://host/~file URL format**



Rootlet Demo at SC'06

◆2 Parts:

- ➔ Local integration ROOT script (.C)
- ➔ Remote analysis ROOT script (.C/h)





Rootlet Demo at SC'06

◆2 Parts:

- ➔ Local integration ROOT script (.C)
- ➔ Remote analysis ROOT script (.C/h)

◆Rootlet services running on show floor, Caltech, and in Brazil

- ➔ Move to TeraGrid at SDSC after SC06

◆Root analysis script submitted to Rootlet service

- ➔ Runs on Clarens server machine or as batch job using batch scheduler
- ➔ Code moved to data
- ➔ Analysis results in the form of a Root histogram file on server
- ➔ Results streamed to Root client as they become available, or available for later retrieval

◆Continuously running demo to exercise all parts of the system



Pythia portal

- ◆ **Analogous to Rootlet**
- ◆ **Prototype browser “portal” to enable user-friendly parameter specification**
 - ➔ **But also access “power user” functionality**
- ◆ **Small jobs (few seconds) can be run on server**
- ◆ **Longer jobs run using batch scheduler**
- ◆ **Multiple job submission**
 - ➔ **Make sure scheduler can handle this!**



Summary

- ◆ **Clarens provides tools for to build scalable distributed systems**
 - ➔ **Relies on networking**
- ◆ **Web services used as ubiquitous means of data access and interaction**
- ◆ **Provides facilities to build web-based portals**



Clarens Pointers

- ◆ **Clarens web page: <http://clarens.sourceforge.net>**
- ◆ **GAE web page: <http://ultralight.caltech.edu/gaeweb/>**
- ◆ **Service discovery: <http://discover.gridservice.info>**
- ◆ **Example portal:
<https://envoy.cacr.caltech.edu:8443/clarens/>**