

DIAL

Distributed Interactive Analysis of Large datasets

LCG Analysis RTAG
CERN

David Adams
BNL
July 15, 2003



David Adams
BROOKHAVEN
NATIONAL LABORATORY



Contents

Goals of DIAL

What is DIAL?

Design

- Dataset
- Application
- Task
- Job
- Result
- Scheduler
- Exchange format

Implementation

Status

Development plans

Batch production

Interactivity issues

Other projects

RTAG architecture



David Adams

BROOKHAVEN
NATIONAL LABORATORY



Analysis RTAG

July 15, 2003 2

Goals of DIAL

1. Demonstrate the feasibility of interactive analysis of large datasets
 - How much data can we analyze interactively?
2. Set requirements for GRID services
 - Datasets, schedulers, jobs, results, resource discovery, authentication, allocation, ...
3. Provide ATLAS with a useful analysis tool
 - For current and upcoming data challenges
 - Real world deliverable
 - Like to add another experiment would show generality



What is DIAL?

Distributed

- Data and processing

Interactive

- Iterative processing with prompt response
 - (seconds rather than hours)

Analysis of

- Fill histograms, select events, ...

Large datasets

- Any event data (not just ntuples or tag)



What is DIAL? (cont)

DIAL provides a connection between

- Interactive analysis framework
 - Fitting, presentation graphics, ...
 - E.g. ROOT, JAS, ...
- and Data processing application
 - Natural to the data of interest
 - E.g. athena for ATLAS

DIAL distributes processing

- Among sites, farms, nodes
- To provide user with desired response time

Look to other projects to provide most infrastructure



David Adams

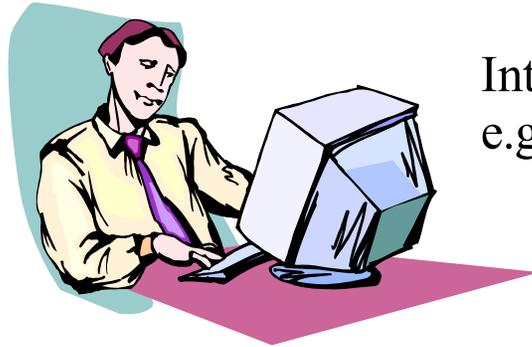
BROOKHAVEN
NATIONAL LABORATORY



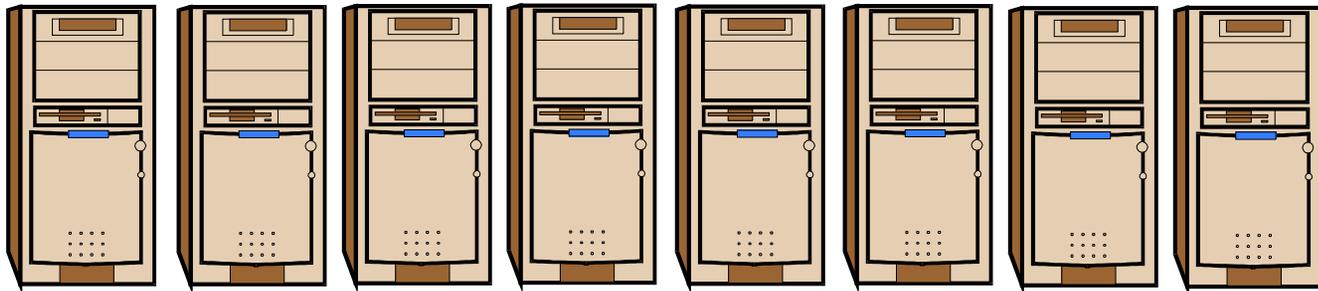
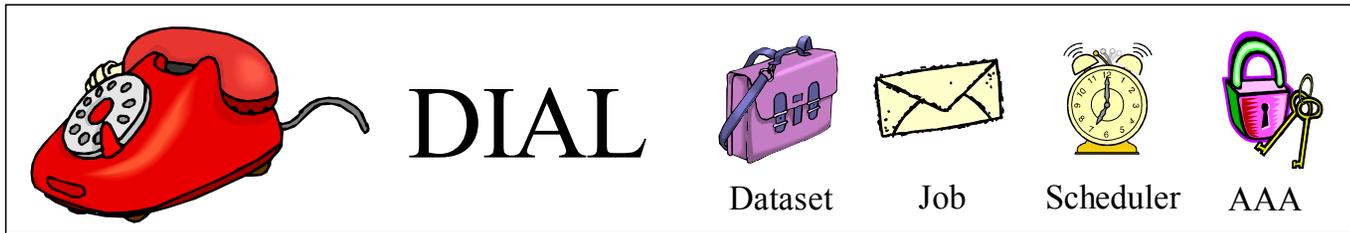
DIAL

Analysis RTAG

July 15, 2003 5



Interactive analysis
e.g. ROOT, JAS, ...



Distributed processing running data-specific application



Design

DIAL has the following major components

- **Dataset** describing the data of interest
- **Application** defined by experiment/site
- **Task** is user extension to the application
- **Job** uses application and task to process a dataset
- **Result** is the output of a job
- **Scheduler** creates and manages jobs

Together these define a high-level JDL

- (job definition language)

Figure shows how these components interact →



David Adams

BROOKHAVEN
NATIONAL LABORATORY

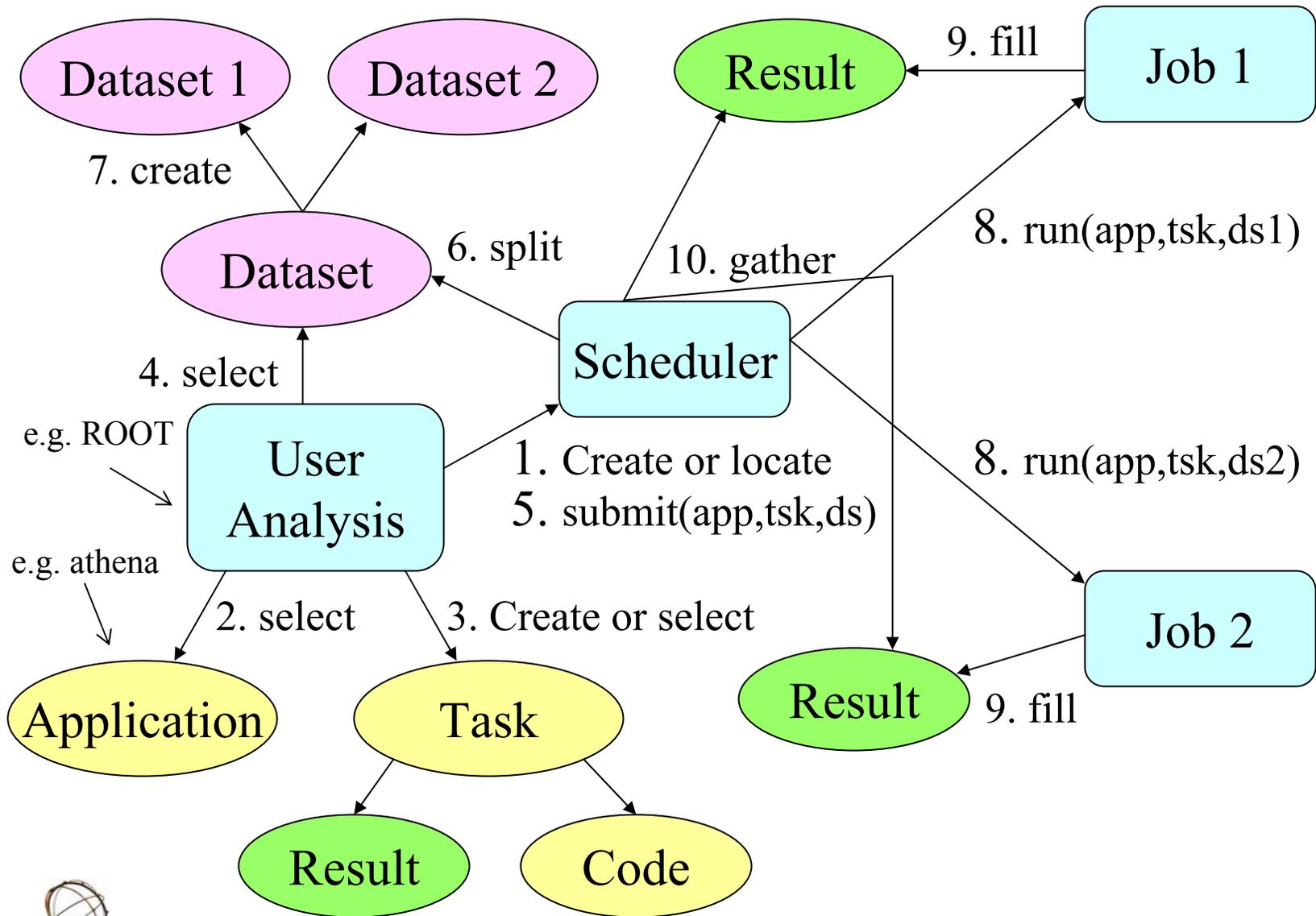


DIAL

Analysis RTAG

July 15, 2003

7



Dataset

Dataset specifies a collection of data

- Used as input for job
- Of special interest is event dataset
 - Each piece of data associated with one “event”
 - Events can be processed independently

Not just a collection of (logical files)

- Data may appear in multiple logical files,
- Some data in databases, ...
- However collection of LF's is most interesting case

Datasets can be split into sub-datasets

- Each can be processed independently
- Each used to define a sub-job for distributed processing



David Adams

BROOKHAVEN
NATIONAL LABORATORY



DIAL

Analysis RTAG

July 15, 2003 9

Dataset (cont)

Splitting datasets is the key to distributed processing

- Hard problem
 - Especially for interactive analysis
- Locate data that can be accessed quickly
- Match CPU and data
- Starting to identify splitting as a separate component

Datasets are subject for another talk



David Adams
BROOKHAVEN
NATIONAL LABORATORY



Analysis RTAG

July 15, 2003 10

Application

User specifies application with

- Name and version
 - E.g. dial_cbnt 0.3

Physical application

- Installed at sites where data is processed
- Receives task and dataset as input
- Creates result with output
- Typically a wrapper around an existing data-processing executable
 - PAW for PAW files (e.g. dial_cbnt)
 - ROOT for ROOT files
 - Athena for ATLAS event data



David Adams

BROOKHAVEN
NATIONAL LABORATORY



DIAL

Analysis RTAG

July 15, 2003 11

Task

Task allows users to extend an application

- Run time configuration
 - Parameters,
 - Sequence of algorithms, ...
- Code for processing the data
- Empty result to specify output
 - E.g. empty histograms (name, ID, min, max, # bins),
 - Policy for naming and placing data in output files, ...

Application

- Defines the task syntax
- Provides means to build and install task
 - E.g. compile and dynamic load



David Adams

BROOKHAVEN
NATIONAL LABORATORY



Analysis RTAG

July 15, 2003 12

Job

Job created by specifying

- Dataset, application and task
- Submitting these to a scheduler

Job has a user interface providing

- Status
 - Dataset, application and task
 - Running, done, failed, ...
 - Start and stop times
 - Result
- Means to update the status
 - Check schedulers, queues, etc...



David Adams

BROOKHAVEN
NATIONAL LABORATORY



Analysis RTAG

July 15, 2003 13

Result

Result is filled during processing

- Examples
 - Histogram
 - Event list
 - New dataset
 - Combination of the above
- Accessible to the user through job
- Partial results may be available during processing

Should be small

- Dataset or logical file identifiers rather than the data in those files

Scheduler

A DIAL scheduler provides means to

- Submit a job
- Terminate a job
- Monitor a job
 - Access user interface for the job
- Verify availability of an application
- Handle tasks
 - Installation
 - Verification of installation
 - Verify consistency with application

DIAL philosophy is scheduler should be generic

- I.e. not restricted to a single application



David Adams

BROOKHAVEN
NATIONAL LABORATORY



DIAL

Analysis RTAG

July 15, 2003 15

Scheduler (cont)

Schedulers may form a hierarchy

- Corresponding to that of compute nodes
 - Grid, site, farm, node
- Each scheduler splits job into sub-jobs and distributes these over lower-level schedulers
- Lowest level LocalScheduler starts processes to carry out the sub-jobs
- Scheduler concatenates results for its sub-jobs
- User may enter the hierarchy at any level
- Client-server communication



David Adams

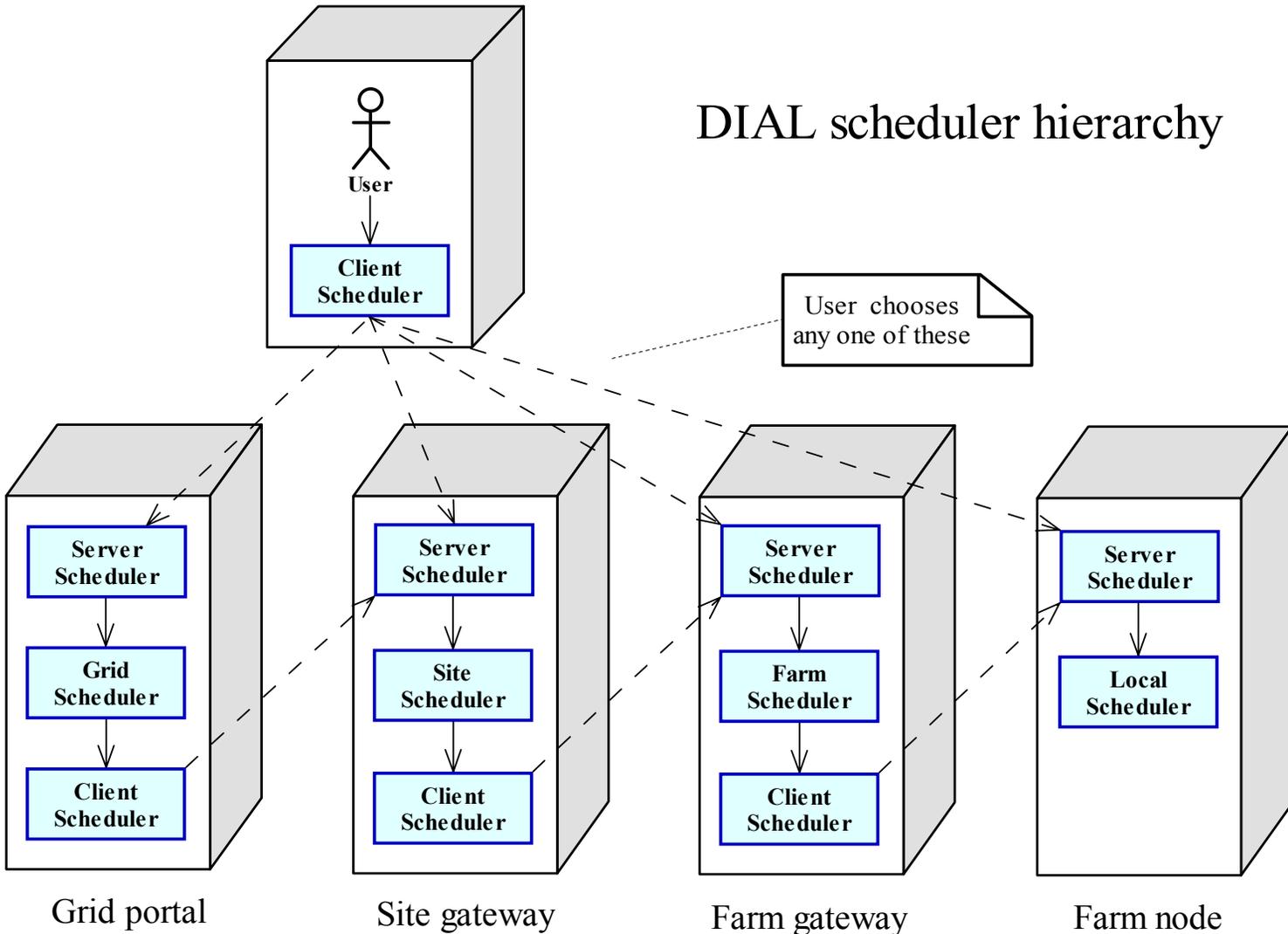
BROOKHAVEN
NATIONAL LABORATORY



Analysis RTAG

July 15, 2003 16

DIAL scheduler hierarchy



Exchange format

DIAL data components are exchanged

- (Data means all but scheduler)
- Between
 - User and scheduler
 - Scheduler and scheduler
 - Scheduler and application executable
- Exchange can be between
 - Different process
 - Different nodes or sites
 - Different languages (C++, Java, Python, ...)
- Each component has an XML representation
 - Should XML define the JDL?



Implementation

DIAL provides in C++

- Generic implementations for
 - Application, Task and Job
- Interfaces (base classes) for
 - Dataset, Result, Job and scheduler
- ATLAS implementations of these interfaces
 - Dataset: AthenaRoot file, combined ntuple (hbook file)
 - Result: hbook file
 - Job: child process, LSF (soon)
 - Scheduler: Single job, farm (soon)
- Utilities and registries to support these components
- Some of the implementation lags behind the design



David Adams

BROOKHAVEN
NATIONAL LABORATORY



DIAL

Analysis RTAG

July 15, 2003 19

Implementation (cont)

Applications

- dial_cbnt is wrapper around PAW for processing ATLAS combined ntuple files

ROOT can be used as front end to DIAL

- All classes imported with ACLiC
- Demo shows how to
 - Create application specification for dial_cbnt
 - Create empty result from an hbook file with empty histograms
 - Create a task with this result and code to fill histograms
 - Create a combined ntuple dataset from an XML file
 - Submit the application, task and dataset to a scheduler
 - Fetch the job description
 - Query the job for status and result



David Adams

BROOKHAVEN
NATIONAL LABORATORY



Analysis RTAG

July 15, 2003 20

Status

Preceding is in DIAL release 0.3 last month

- Demo runs at BNL (ACF and RCF) and CERN (lxplus)
- Scheduler is very simple (no job splitting)

Anticipate release 0.4 in August

- Magda files accessible
- Local distributed processing

More information

- DIAL home page
 - <http://www.usatlas.bnl.gov/~dladams/dial>
- CHEP paper
 - http://www.usatlas.bnl.gov/~dladams/dial/talks/dial_chep2003.pdf
 - Design and implementation have evolved



David Adams

BROOKHAVEN
NATIONAL LABORATORY



DIAL

Analysis RTAG

July 15, 2003 21

Development plans

Farm Scheduler

- Distribute processing over a single farm (BNL)
 - First LSF (BNL and CERN)
 - Then Condor (local, COD, master-worker, Condor-G?)
- Makes DIAL a useful tool for distributed processing

Remote access to scheduler

- Job submission from anywhere
- Web service

Add policy to scheduler interface

- Response time
- How to split dataset for distributed processing



David Adams

BROOKHAVEN
NATIONAL LABORATORY



DIAL

Analysis RTAG

July 15, 2003 22

Development plans (cont)

Grid schedulers

- Distribute data and processing over multiple sites
- Interact with dataset, file and replica catalogs
- Authentication, authorization, resource location and allocation, ...

ATLAS POOL dataset

- After ATLAS incorporates POOL

ATLAS athena as application



David Adams

BROOKHAVEN
NATIONAL LABORATORY



Analysis RTAG

July 15, 2003 23

Batch production

Original goal of DIAL was to provide means for interactive data analysis.

However interactive analysis and batch production are not so easily separated

- DIAL JDL is appropriate for either
- Schedulers will need to operate over a continuum of
 - Produced data size
 - Acceptable response time
- Some users will want to interactively browse a small sample and then submit larger sample to batch



Interactivity issues

Response time is critical

- Interactive system provides means for user to specify maximum acceptable response time
- All actions must take place within this time
 - Locate data and resources
 - Splitting and matchmaking
 - Job submission
 - Gathering of results
- Longer latency for first pass over a dataset
 - Record state for later passes
 - Still must be able to adjust to changing conditions



David Adams

BROOKHAVEN
NATIONAL LABORATORY



Analysis RTAG

July 15, 2003 25

Interactivity issues (cont)

Interactive and batch must share resources

- Share implies more available resources for both
- Interactive use varies significantly
 - Time of day
 - Time to the next conference
 - Discovery of interesting events
- Interactive request must be able to preempt long-running batch jobs
 - But allocation determined by sites, experiments, ...



David Adams

BROOKHAVEN
NATIONAL LABORATORY



Analysis RTAG

July 15, 2003 26

Other projects

JDL (job definition language)

- DIAL may be thought of as a proposal for a user-level (high-level) JDL
- Like to come to agreement on this JDL with other projects so components can be shared
- Language issues: C++, Java, Python, XML, ...

Scheduler

- Look to other projects to deliver robust and efficient implementations of the DIAL scheduler
- In the mean time DIAL will
 - Try to wrap analogous components from other projects
 - Make relatively simple implementations to meet our project goals



David Adams

BROOKHAVEN
NATIONAL LABORATORY



DIAL

Analysis RTAG

July 15, 2003 27

Other projects (cont)

Candidates to provide DIAL schedulers:

- Look to **Condor** to deliver a grid scheduler
- Scheduler can be implemented using **Chimera**
- Scheduler restricted to ROOT as an application could be implemented using **PROOF**
- Scheduler for athena jobs can come from **GANGA**
- Tech-X has SBIR to develop **JDAP** scheduler for use with JAS
- A java-based scheduler is being developed by **STAR**
- Last (but not least?), **DIAL** will provide schedulers



David Adams

BROOKHAVEN
NATIONAL LABORATORY



Analysis RTAG

July 15, 2003 28

Other projects (cont)

Build web services to create and access DIAL schedulers

- Using **Clarens** and/or **OGSA**

Environments that might use DIAL schedulers:

- A Python implementation of the JDL would enable **GANGA** and **PI/SEAL** to use DIAL schedulers
- DIAL is already imported into **ROOT**
- A Java version of the JDL would enable **JAS** to use DIAL schedulers
- Web or script based user-level production for **any experiment** could benefit from the scheduler interface



David Adams

BROOKHAVEN
NATIONAL LABORATORY



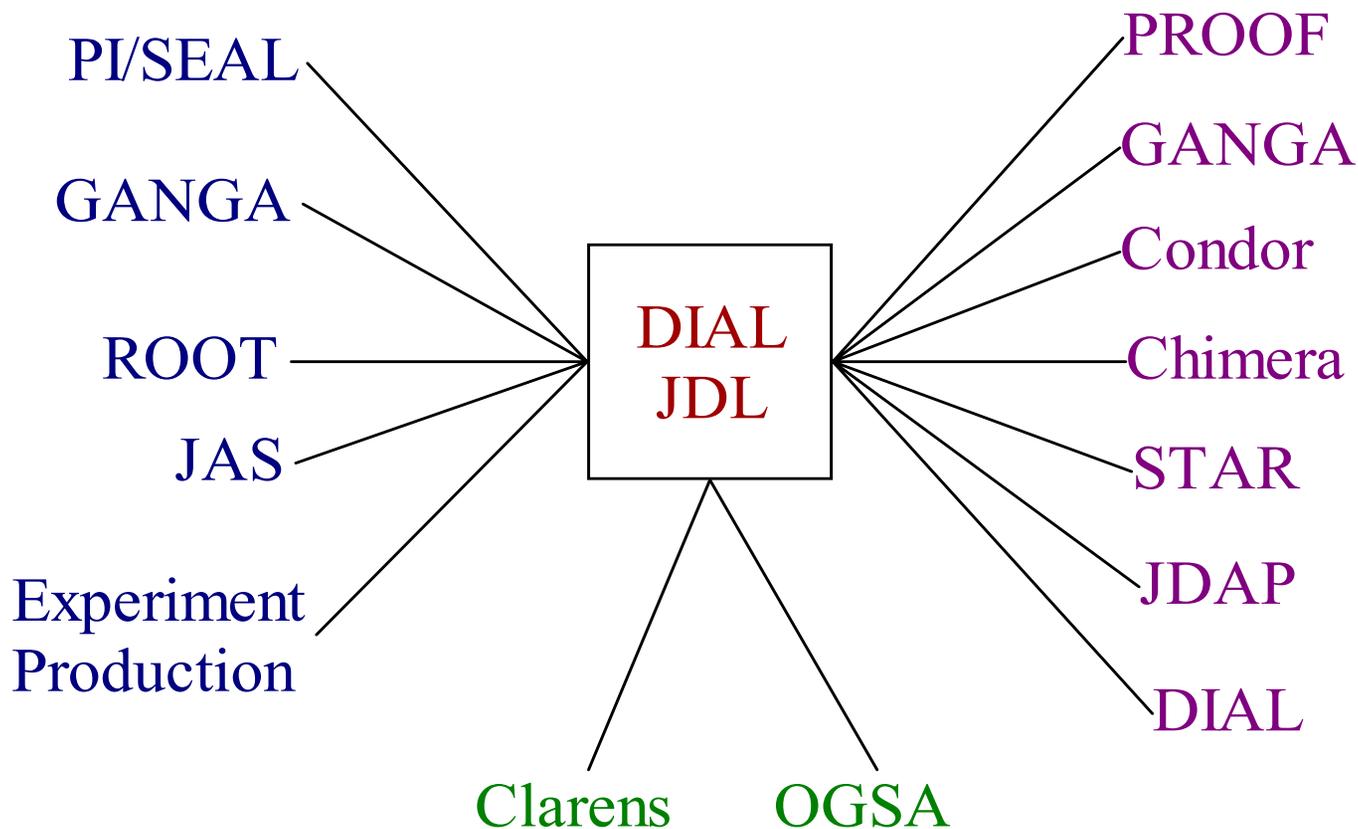
DIAL

Analysis RTAG

July 15, 2003 29

User interfaces

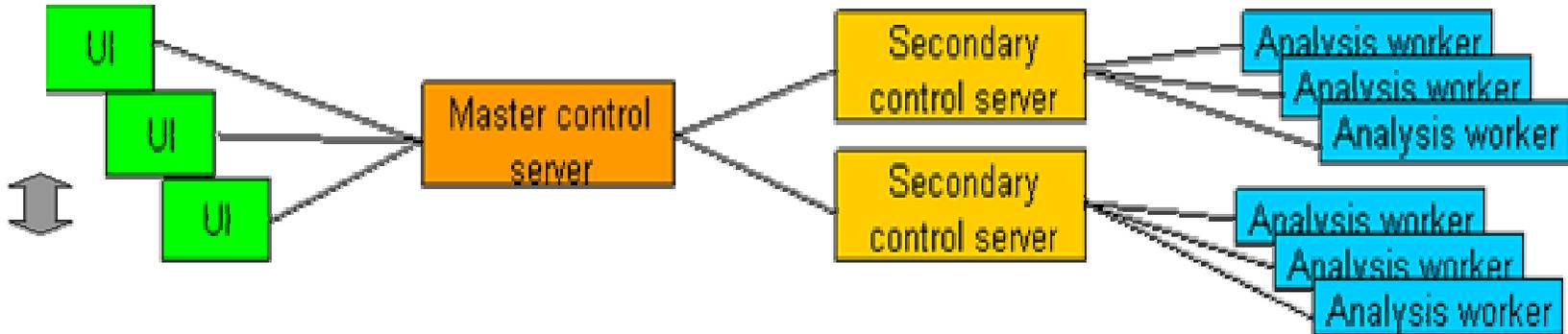
Schedulers



Web service infrastructure



RTAG architecture



How does DIAL connect to the RTAG architecture?

- Pictured above
- DIAL addresses the lines
 - What is exchanged (dataset, app, task, job, ...)
 - Interfaces
- DIAL neither requires nor excludes persistent workers
 - Choice made by the scheduler

