STAR Scheduling status

Gabriele Carcassi

9 September 2002

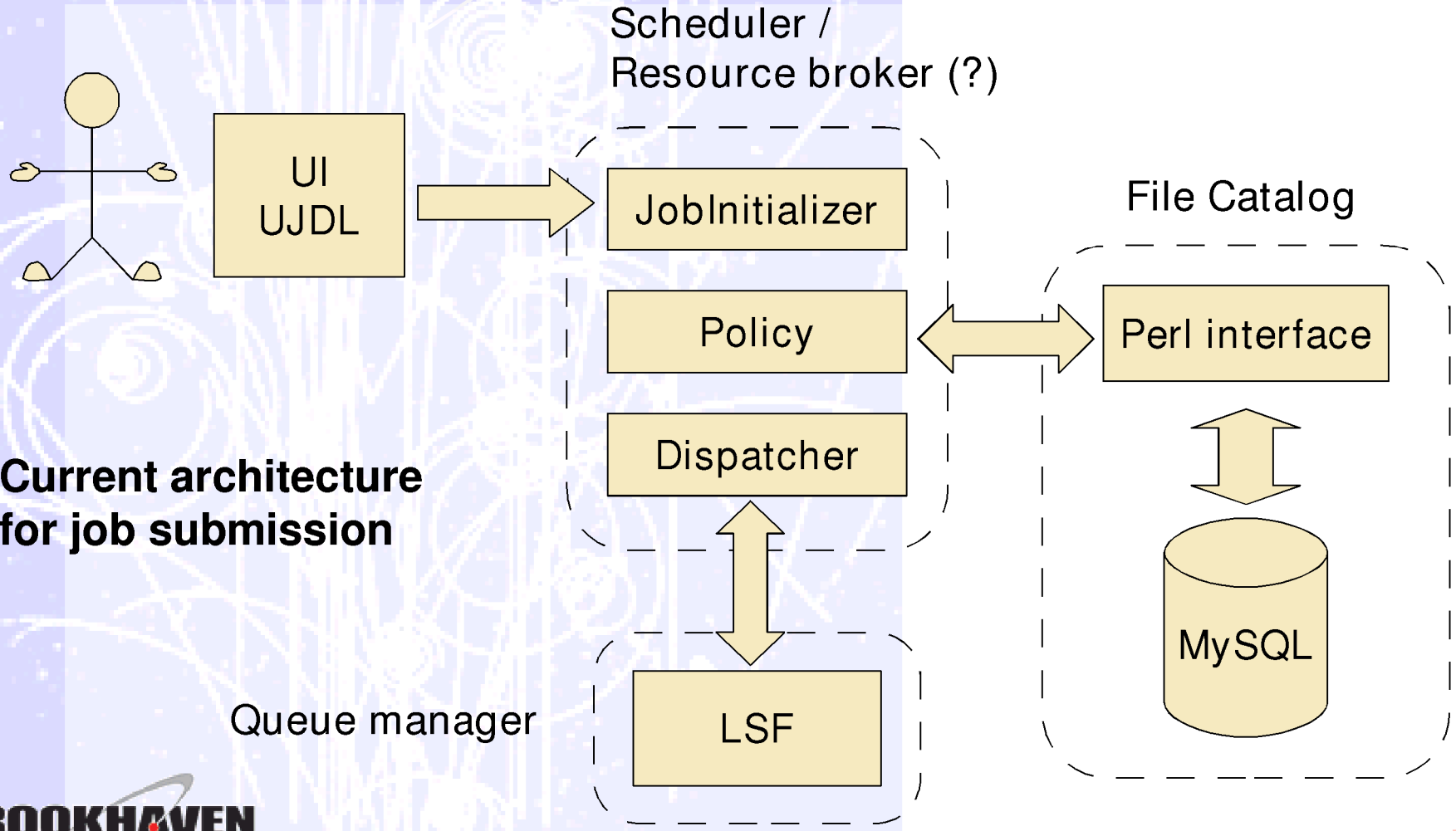
BROOKHAVEN
NATIONAL LABORATORY

STAR

Objectives

- Have something for September
- Stabilize the user interface used to submit jobs, based on the user perspective
- Provide an architecture that allow easy change
- Provide a way for the administrator to change the behavior of the system

STAR Scheduling architecture



Current architecture for job submission

BROOKHAVEN NATIONAL LABORATORY

STAR

User interface

- Driven by use cases, and not by the tools used to implement it
  - User basically gives the job and the list of input files, which can also be a catalog query
- User specify what he wants to do, and not how to do it
  - Simpler to use
  - Gives the administrator more flexibility in the implementation

User interface

- User job description in XML
  - Scheduler developed at Wayne State uses XML
  - Easy to extend:
    - Ex. multiple ways to describe the input

```
<input URL=?..?/>
<input filename=?..?/>
```

  - Parsers already available

Job Initializer

- Parses the xml job request
- Checks the request to see if it is valid
  - Checks for elements outside specification (typically errors)
  - Checks for consistency (existence of input files on disk, ...)
  - Checks for requirements (require the output file, ...)
- Creates the Java objects representing the request (JobRequest)

Job Initializer

- Current implementation
  - Strict parser: any keyword outside the specification stops the process
  - Checks for the existence of the stdin file and the stdout directory
  - Forces the stdout to prevent side effects (such as LSF would accidentally send the output by mail)

Policy

- From one request, creates a series of processes to fulfill that request
- Processes are created according to farm administrator's decisions
- The policy may query the file catalog, the queues or other middleware to make an optimal decision

Policy

- We anticipate a lot of the work in finding an optimal policy
- Policy is easily changeable, to allow the administrator to change the behavior of the system

BROOKHAVEN NATIONAL LABORATORY

STAR

Policy

- Current policy
  - The query is resolved by simply querying the catalog
  - Divide the job into several processes, according to where the input file is located
  - No more than 10 input files per job

File Catalog integration

- In the job description a user can specify one or more queries
- Depending on how these queries are resolved, the farm can be more or less efficient
- Mechanism to execute the query is separate from the query description
  - Easy to change catalog implementation

File Catalog integration

- Current implementation:
  - Very simple to allow fast implementation
  - Forwards the query as it is to the perl script interface of STAR catalog
    - Main advantage: same syntax for the user
  - No “smart” selection is made
    - No effort is done for selecting those files that would optimize the use of the farm

Dispatcher

- Talks to the underlying queue system
- Takes care of creating the script that will be executed
- Creates environment variables and the file list

Dispatcher

- Current implementation:
  - Creates file list and script in the directory where the job was submitted from
  - Creates environment variables containing the job id, the list of files and all the files in the list.
  - Creates a command line for LSF
  - Submits job to LSF

Other functionalities

- Log
  - The logging services provided in Java 1.4 are used to create a detailed log
  - Each entry has a level attribute (FINEST, FINER, FINE, INFO, WARNING, SEVERE), and the log can be selected to produce output only starting from one level
  - We will use FINEST during beta, INFO during the first months of production, and WARNING after that
  - Logging goes on behind the back of the user, providing full information about usage essential to trace bugs and problems associated with the policy.

Conclusion

- The tool is available and working
  - Beta quality: works reliably, some small feature might be needed, QA test still required.
- Allows the use of local disks
- Architecture is open to allow changes
  - Catalog implementation (MAGDA, RLS, GDMP, ... ?)
  - Dispatcher implementation (Condor, Condor-g – Globus, ...)