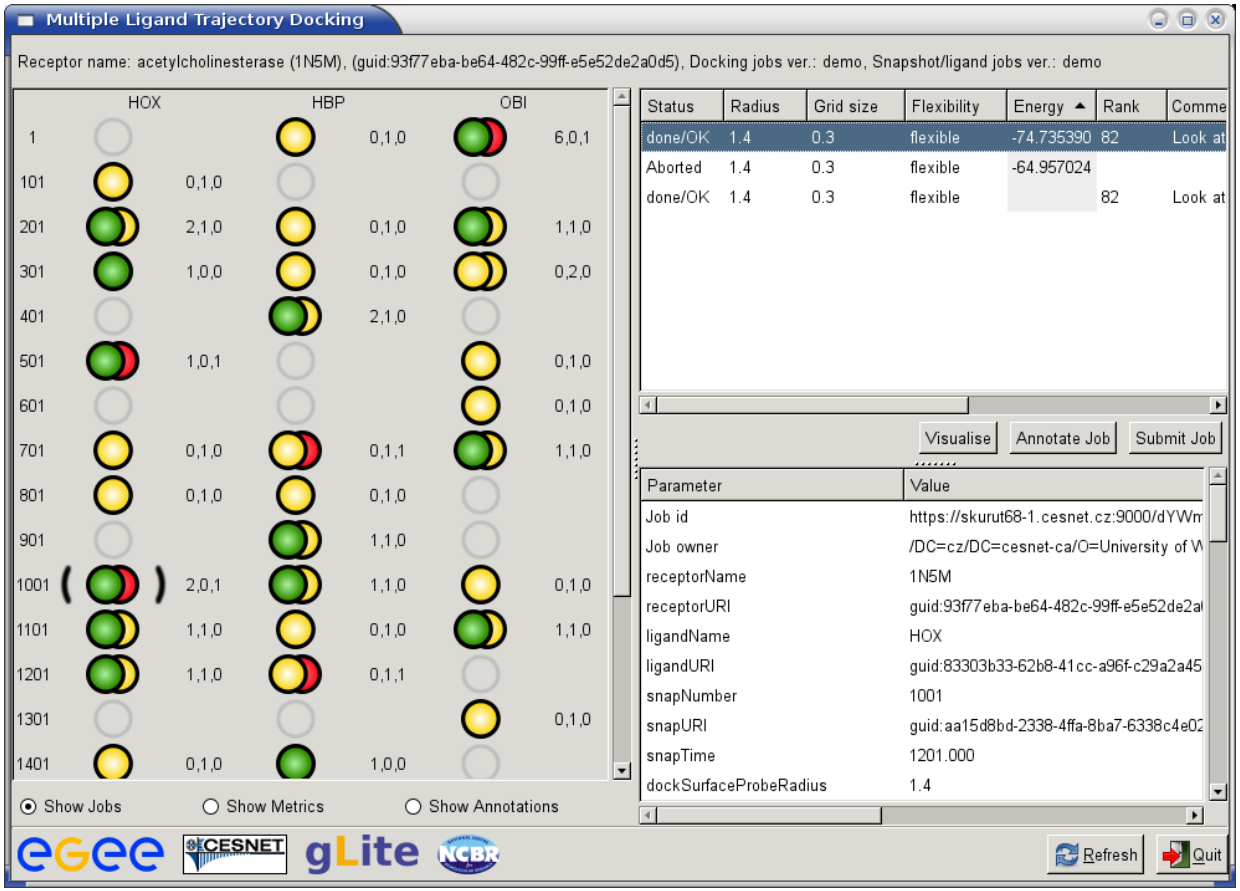
Job Provenance – Status and Plans

Aleš Křenek, on behalf of CESNET JRA1 team

User Forum Demo

- computational chemistry application (search for drug)
- logical "2D array" of jobs (candidate combinations)



User Forum Demo (2)

- custom graphical application (GTK)
- different display views
  - middleware focused – running, finished, and failed jobs
  - application focused – result ranking
- shared view on results of multiple users
- support for adding annotations and job submission (clonning)
- dozens of jobs submitted and completed during the demo
- services were queried continously
- **no failure occured**
- `http://egee.cesnet.cz/mediawiki/index.php/Job_Provenance_Demo`

User Forum Demo (2)

- custom graphical application (GTK)
- different display views
  - middleware focused – running, finished, and failed jobs
  - application focused – result ranking
- shared view on results of multiple users
- support for adding annotations and job submission (clonning)
- dozens of jobs submitted and completed during the demo
- services were queried continously
- **no failure occured**
- `http://egee.cesnet.cz/mediawiki/index.php/Job_Provenance_Demo`

Status

- service functionality is stable
- but still room for improvements
  - no complex performance tests done – bottlenecks expected
  - client integration in User Interface
  - reliable store-and-forward JPPS → JPIS communication (new interlogger)
  - define deployment scenarios, documentation, . . .

Status

- service functionality is stable
- but still room for improvements
  - no complex performance tests done – bottlenecks expected
  - client integration in User Interface
  - reliable store-and-forward JPPS → JPIS communication (new interlogger)
  - define deployment scenarios, documentation, ...

Plans

Immediate

- User forum followup – support the application community
- 2nd Provenance Challenge

Medium term

- Experimental deployment on part of Atlas production
  - poster at CHEP'07 (September)
- Define, test and support **elmentary production deployment**
  - start: during summer
  - finished: end of 2007
- Listed issues will be addressed

Longer term

- Actively look for other application groups and support them
- **Job browser**

Elementary deployment

Purge L&B data

- L&B database can't grow forever
  - regular problems in production
- dump raw events into compressed file
- purge all events and job status details from L&B
- preserve tiny record (zombie)
  - "this job was here once upon a time"
  - stored aside, queriable only with specific jobid
- timeframe for keeping the data
  - successfull jobs: several days
  - failed jobs: few weeks (at most)
  - zombies: months – years

Elementary deployment (2)

Preserve the data

- dumped events uploaded to JPPS
- JPPS endpoint kept with the zombie
- stored in compact form virtually forever
- may be augmented with further data
  - annotations
  - sandbox files
  - application specific
- only few, well known and permanent JPPS instances
  - specific for very large VO's (Atlas) only
  - "catch-all" otherwise, e.g. per ROC

Elementary deployment (3)

Access the data

- suitable JPIS instances
  - frequent JPPS queries should be avoided for performance reasons
- foreseen final solution – find JPIS via Service Discovery
  - SD query
    - "I want to know about Bob & Alice's jobs from August 2005"
    - returns list of JPIS's to query
  - see presentation from JRA1 All-hands, November 2006
  - should be doable with the current SAGA SD API proposal
- for the time being ...
  - default JPIS instance on L&B server box
  - jobs of this L&B, e.g. for one year
  - keep only fraction of job data (1/10 – 1/1000 wrt. L&B)
    - owner, VO, final state, submission and completition time, CE, ...
  - 1 kB per job $\times$ 100 kjobs/day $=$ 35 GB/year

Elementary deployment (4)

Integration in clients

- current CLI UI – reasonably transparent to the users
- additional options to query also JPIS/JPPS when L&B returns zombie
- options for implementation
  - legacy python UI?
  - wrap JPIS in L&B client (bad solution)
  - legacy L&B consumer API + similar API for JPIS @ L&B
  - legacy L&B + WS JPIS client
  - 100 % native WS implementation
- any other (graphical) clients to support?

Job browser

Motivation

- Claudio's idea, inspired by JP demo at EGEE UF apparently
- *"Let's have a nice generic graphical browser to show user's jobs"*

Basic requirements

- keep it simple
  - usecases are complex and application specific, can't foresee them all
  - user groups like to develop their own tools either
- both L&B and JP client
- support off-line mode
  - download data on even large set of jobs for analysis on journey
- reasonable portability, technology reuse

Job browser

Motivation

- Claudio's idea, inspired by JP demo at EGEE UF apparently
- *"Let's have a nice generic graphical browser to show user's jobs"*

Basic requirements

- keep it simple
  - usecases are complex and application specific, can't foresee them all
  - user groups like to develop their own tools either
- both L&B and JP client
- support off-line mode
  - download data on even large set of jobs for analysis on journey
- reasonable portability, technology reuse

Job browser (2)

Supported views

- middleware vs. application oriented
  - job state, # of resubmissions, used CE, ...
  - specific annotations and results
- aggregate vs. detailed
  - coloured array or other visualization of aggregate state, patterns etc.
  - pretty-printed glite-job-status/logging-info output

Job browser (3)

Considered implementation

- web browser
  - aggregate views
  - platform-independent plugin (firefox)
  - 3-tier web application (universal thin client)
  - we need survey of existing job monitoring tools
- email client
  - job maps to email message
  - accessed with IMAPS – off-line for free
  - HTML message body to pretty-print or specific client plugin
  - email headers to allow client-side filtering and search
  - server-side filters (queries)
    - other interface (web) to create filtered folders
    - control folder with editable "messages"
  - job status and events folders

Job browser (3)

Considered implementation

- web browser
  - aggregate views
  - platform-independent plugin (firefox)
  - 3-tier web application (universal thin client)
  - we need survey of existing job monitoring tools
- email client
  - job maps to email message
  - accessed with IMAPS – off-line for free
  - HTML message body to pretty-print or specific client plugin
  - email headers to allow client-side filtering and search
  - server-side filters (queries)
    - other interface (web) to create filtered folders
    - control folder with editable "messages"
  - job status and events folders