SRB in the BioEmergences project

Dominique de Waleffe
dominique.dewaleffe@denali.be

Denali SA

CC-IN2P3 – Feb 2, 2009

The project
The architecture
SRB usage
iRODS?

Partners

- Framework Program 6 project
- Consortium:
  - CNRS – Centre De Recherche en Epistémologie Appliquée (CREA) (FRANCE)
  - Institut Curie (France)
  - Slovenska Technicka Univerzita V Bratislave (Slovakia)
  - Universidad de Málaga (Spain)
  - Denali Consulting S.A. (Belgium)
  - European Molecular Biology Laboratory (Germany)
  - University of Bologna (Italy)
  - CNRS – CC-IN2P3 (France)
- Project fact sheet on CORDIS:`http://tinyurl.com/5yc42k`

Project goals

What?

With the BioEMERGENCES project, we aim at providing an **experimental platform** to observe **in vivo** emergent patterns at various scales and **measure their variability between different individuals** of the same species. This is a strategy towards the measurement of the individual susceptibility to genetic diseases or response to treatments.

...

The main result expected from BioEMERGENCES is the **specification of a European platform to achieve high throughput measurement** of individual differences and screening of drugs combinations such as bi or tri-therapies.

Goals

Team

- Multi-disciplinary team : biologists, mathematicians, engineers, computer scientists

Research

- **Observe:** Using high definition microscopes, capture 4D sets of images of living embryos (Zebra Fish, Sea Urchin,...)
- **Transform:** Invent methods to go from images to symbolic representations (lineage trees, contours)
- **Compare:** Invent methods for efficient and meaningful comparisons

Industrialize

- Platform for high throughput execution of the processes

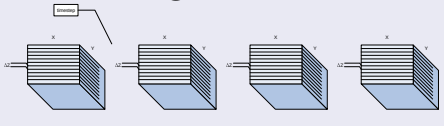
Some details

Gather observations

- Biologists place an embryo under microscope for a number of hours
- a stack of horizontal images of size $x * y$, separated in time by δt and space by δz are captured
- a new stack is captured every ΔT
- Repeated for many individuals under different conditions

Output

- A large set of large files containing raw images:



- A set of metadata describing the experiment

Some details

Reconstruct cell lineage tree

- Invent different algorithms to:
  - filter images (remove noise)
  - detect centers of cell nuclei ((x, y, z) position)
  - determine membrane contours (set of 3-D polygons)
  - determine nucleus contours (set of 3-D polygons)
  - identify mytosis (cell divisions)
  - track individual cell from step T_i to step T_{i+1} and build lineage tree
  - compare lineage trees , infer new results
- visualize reconstructions
- correct and annotate datasets

Some figures

Image sizes: $512 * 512 * 8$ to $1024 * 1024 * 8$ pixels, $0.5\mu < \delta x, \delta y < 1.5\mu$, but soon: $2048 * 2048 * 24$,

Number of images in stack: between 50 and 200,

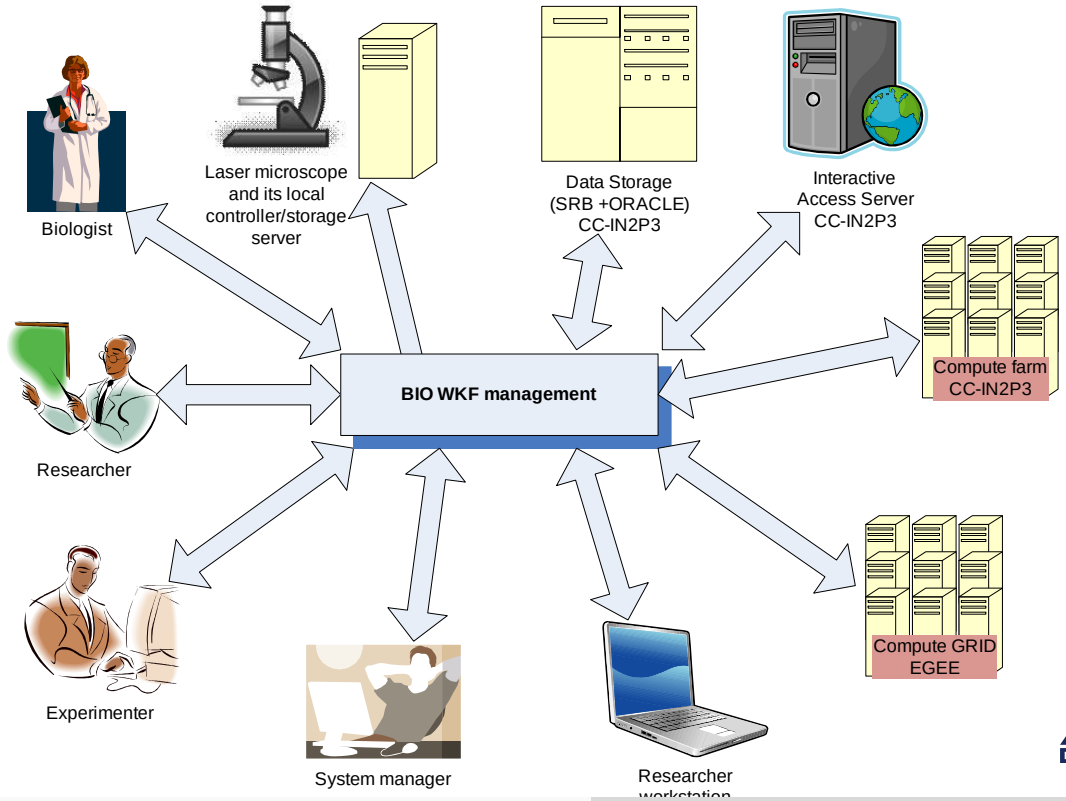
Number of time steps: ΔT typically between 1 and 10 minutes, a few tens to a few hundreds of time intervals captured.

Raw data volumes: 50 to 60 *Gigabytes* of raw image files per experiment (size: 512) but will soon be $1/2$ *Terabytes* with new microscope.

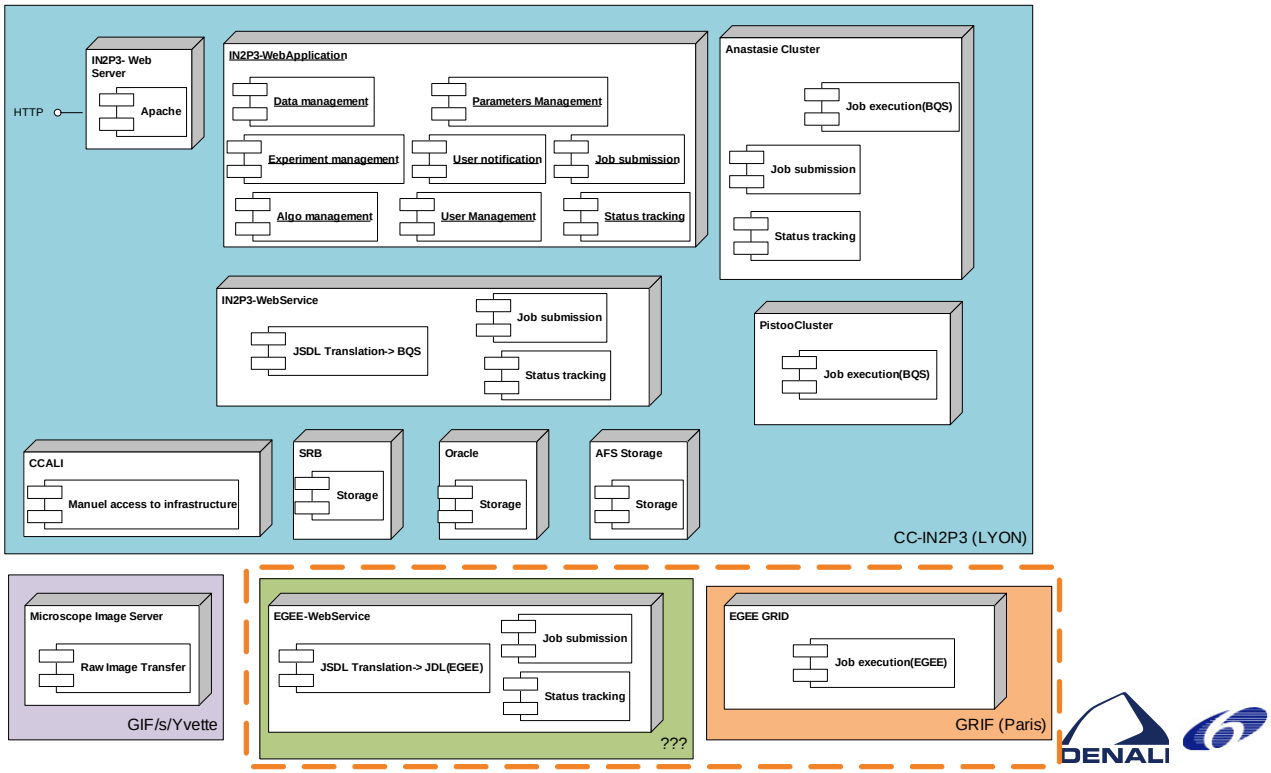
Number of cells: lineage trees contains several million cells.

Current storage used (SRB): in excess of 8 TB.

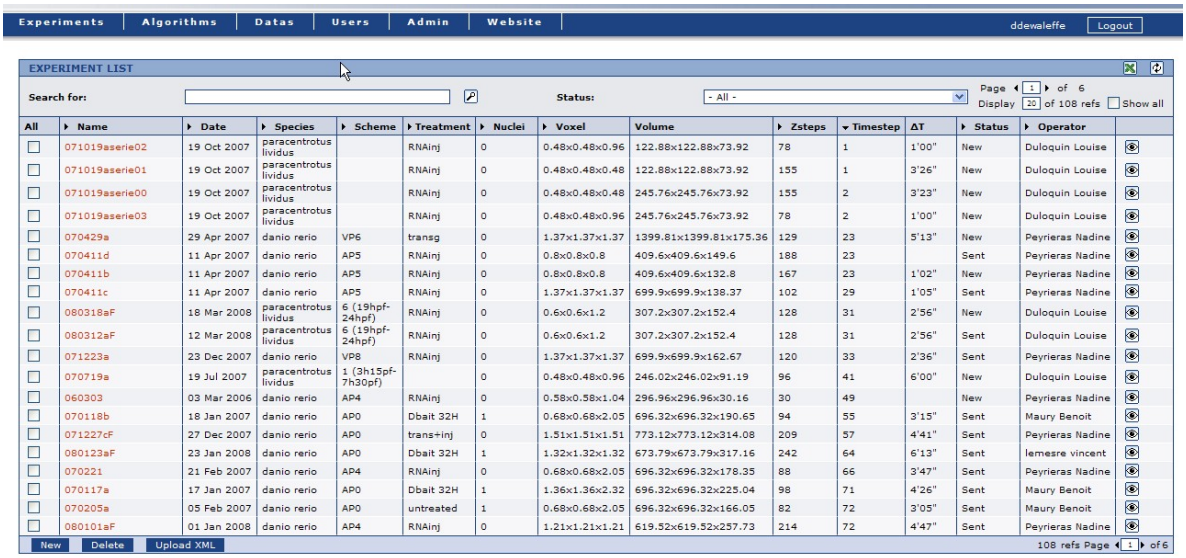
Context diagram



Deployment viewpoint

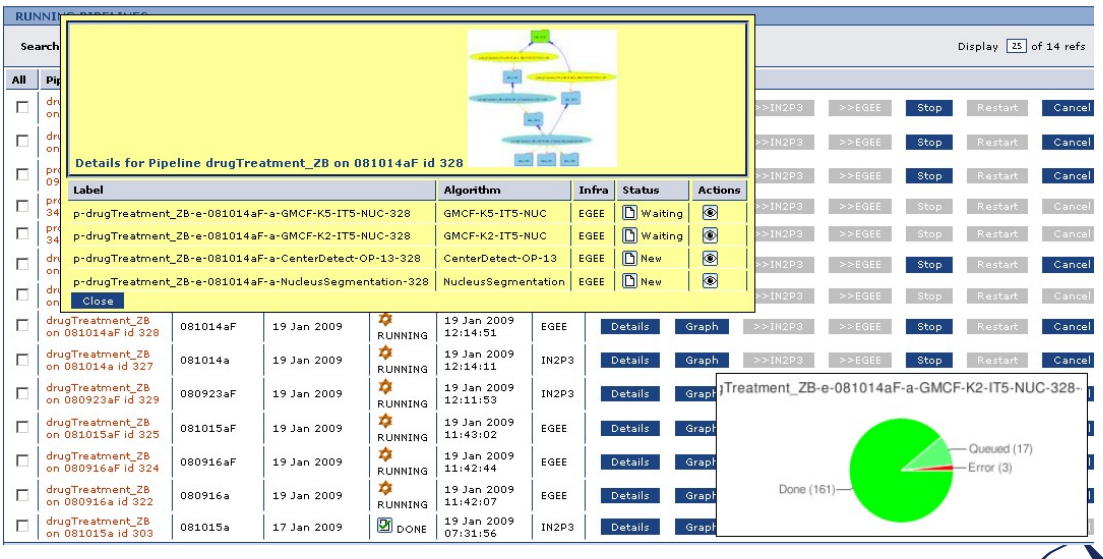


Application: experiment list

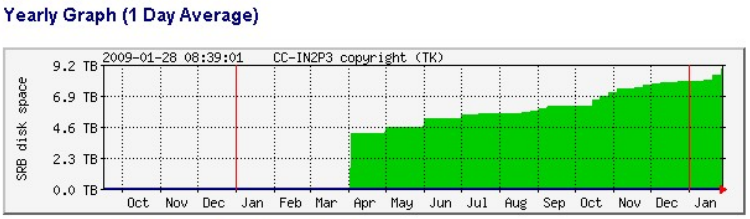


Application: processing pipelines

Details button: brings view below:



SRB usage: What

- Main repository: cc-in2p3 Lyon.
- Raw data storage. Data captured in Paris, format standardized, then copied to SRB. `Srsync` is used.
- Derived data.
  - Kept for reuse in further processing.
  - Data history kept in application's DB.
- Some additionnal files (movies of reconstructions,...)
- Stores algorithms (scripts, sources, builds procedures, executables)
- 

	Max	Average	Current
space used:	9059.0 GB	3747.0 GB	9059.0 GB

SRB usage: how

- Used in identical manner from both farms.
- Mostly used as a file system
- Command line: `Smkdir`, `Scd`, `Sls`, `Sput`, `Sget`, `Srm`, `Srsync`
- Jargon library used for
  - displaying raw data file lists
  - Streaming movies [todo]
- Not used in the project:
  - user meta data
  - web based browser

SRB usage: issues

- Slow access when doing operations requiring catalog access (e.g. `Sls -l`).
- Bizarre error messages, or no exit codes (makes it difficult in scripts)
- Locking bugs (multiple e.g. Smkdir X) which impact the whole group!
- deleted stuff is not always fully deleted

Why iRODS?

- Post-processing on ingestion. Could trigger raw data format changes on upload.
- Workflows. Probably redundant with what we already have or can have.
- Looks like iRODS would be a good fit but:
  - we have no extra budget in current project
  - team is currently looking at using ROOT as storage framework
  - taking advantage of iRODS imply large re-architecting effort
- sources available.

Why iRODS , some risks ?

- maintainability of complex rule base?
  - rule syntax (one liners, readability, choices of operators, comments?)

```
myRule|foo==1|action1(...);action2(...);...|action3(...);action4(...);...
```

  - Why not sligthly more verbose

```
rule myRule { //this rule is triggered when foo and does bar
   when ( foo == 1)
   do {
     /* watch that this action has side-effects */
     action1(...);
     action2(...);...
   }
   on failure {
     action3(...);
     action4(...);...
   }
}
```

- can I define new microServices as complex jobs (e.g submit job(s) to farm) without going to C programming?

Conclusion

- BioEmergences has complex distributed data/processing needs
- Could make use of iRODS if risks are shown to be a non issue

Questions?