Galaxy Pasteur

Patchwork of experiences and improvements

Olivia Doppelt-Azeroual, Sophie Créno et Fabien Mareuil
CIB, Institut Pasteur, Paris
Summary

Part 0 : Galaxy Pasteur

Part 1 : Adaptations to the Pasteur infrastructure
   ● Module
   ● “Libraries” automation
   ● Galaxy reporting

Part 2 : Problems and corrections
   ● I/O Problems
   ● Purged User Problem

Part 3 : Future improvements
   ● Upload submitted for remote execution
   ● Statistics on Galaxy reporting
   ● SynBioWatch Project
At Institut Pasteur, Galaxy consists in:

- An instance used by 133 users and administrated by 1.5 fte administrators.
- A ToolShed with 89 repositories containing 289 tools.
- An average of 1783 jobs per month since February 2013.
- Trainings on Galaxy, twice a year.
- Trainings using Galaxy, twice a year.
Adaptations to the Pasteur infrastructure

"Libraries" automation

● Why automate library creation?
  ○ Private data to deal with,
  ○ Only 1.5 fte administrators,
  ○ Big data upload and export.

● New API script `scripts/api/automate_library.py`, how it works:
  ○ Root cron script execution with admin API key (launched every 10 minutes)
  ○ Retrieves the users list using the API
  ○ Checks if the exchange directories exist for each user
  ○ If not, creates a Galaxy library named “login” using the API and creates 2 directories `export/“login” upload/“login”` with the right linux permissions
  ○ Sends an email to admins who modify the library permission (by hand)

● Clarifications:
  ○ Deals also with linux permissions of exported files.
  ○ User `cp/scp` the data in `upload/“login”` and upload in Galaxy through the interface.
Adaptations to the Pasteur infrastructure

Module

- Module provides a way to dynamically modify of a user’s environment.
  - Uses modulefiles
  - Allows the management of several packages/software versions on the same instance

- Patches on `/lib/galaxy/` directory:
  - `config.py`
  - `jobs/__init__.py`
  - `jobs/runners/__init__.py`

- How it works:
  - A `module_conf.xml` file lists tool ids and their associated modules:
    `<tool id="tophat2" version="2.0.7" module="tophat/2.0.7" />`
  - When the tool is launched, Galaxy uses the tool id to retrieve the list of modules
  - Then, Galaxy creates a `module.sh` script to load the modules
Adaptations to the Pasteur infrastructure

Galaxy reporting

- Natively in Galaxy
- Setup equivalent to ToolShed
- Provides many interesting metrics
  - Jobs per month
  - Jobs per user
  - Jobs per tool
  - User disk usage
  - ...
Part 0: Galaxy Pasteur

Part 1: Adaptations to the Pasteur infrastructure
- Module
- “Libraries” automation
- Galaxy reporting

Part 2: Problems and corrections
- I/O Problems
- Purged User Problem
Problems and corrections

I/O Problems

- **Identification of two I/O intensive process**
  - Galaxy renaming step for output files (output --> dataset)
  - Execution of `set_metadata.sh` script which collects metadata information

- **Patches on:**
  - `lib/galaxy/jobs/runners/__init__.py`
  - `set_metadata.sh`

- **How it works:**
  - `cp` and `rm` commands are replaced by `mv` command (faster on the same file system)
  - `set_metadata.py` script executed on cluster nodes
Problems and corrections

Purged User problem

- There and back again at Pasteur (No way to unpurge a user)
  - A user left the Institut Pasteur (purged) and got back a month later.
  - Impossible to unpurge the user

- Modified API script:
  - scripts/cleanup_datasets/pgcleanup.py

- How it works:
  - New function, operating directly on the Galaxy database
  - Purged and Deleted attributes for that user are changed from true to false

update galaxy_user set purged='f', deleted='f' where id in (select id from galaxy_user where email='%s');
Status

Part 0 : Galaxy Pasteur

Part 1 : Adaptations to the Pasteur infrastructure
  ● Module
  ● “Libraries” automation
  ● Galaxy reporting

Part 2 : Problems and corrections
  ● I/O Problems
  ● Purged User Problem

Part 3 : Future improvements
  ● Upload submitted for remote execution
  ● Statistics on Galaxy reporting
  ● SynBioWatch Project
Galaxy mainly used for NGS analysis.

Another I/O intensive process is the upload of big files
- For the moment, the process is run on the web server (head)
- Upload is handled like a Galaxy tool, xml + script

Idea: patch tools/data_source/upload.py, tools/data_source/upload.xml
- We need to differentiate the upload possibilities (http and cp from upload/"login")
- Identify file system uploads and remotely execute them on the cluster
- We are testing this solution
Future improvements

● Galaxy reporting
  ○ More statistics are needed
    ■ automation of data retrieval from Galaxy reporting
    ■ graphics generation
  ○ Project:
    ■ scripts development to automate it
    ■ use of Galaxy API to retrieve data

● Tool ID with ToolShed
  ○ too long name (full path of ToolShed directory)
    galaxy.web.pasteur.fr/toolshed-pasteur/repos/fmareuil/gatk2/gatk2_base_recalibrator/0.0.4
Future improvements

Collaborative project at Institut Pasteur

- The PGP (Pôle de Génotypage des Pathogènes) group is implementing a specific web interface to facilitate the management of their analysis to detect pathogens within NGS sequences samples:
  - It contains a LIMS database and a result exploratory interface
  - It is launched on a web server linked to Institut Pasteur infrastructure.
  - It communicates remotely with Galaxy to execute pre-built analysis workflows.

- Our contribution is to help building the remote communication with Galaxy API. They need to:
  - Copy (big) data within Galaxy environment, (ok)
  - Upload those data into Galaxy libraries (ok)
  - Import those libraries content into Galaxy histories (almost ok)
  - Launch the workflows (fixed or tunable options) (not yet)
  - Export the results (not yet)
Acknowledgments

Yes for an unified Galaxy WIKI!!