Workflow Manager for HCS

The Specific Problem(s) Of High Content Screening

For The SyBIT Tech Day, Bern 03.10.2012
Specifics Of High Content Screening

- Very large datasets: ~20GB per dataset
- Long processing time: ~80 hrs per dataset
- Many datasets: ~2000 datasets currently acquired
- Workflows consist of ~10 - 20 processing steps
Specifics Of High Content Screening

- Very large datasets: ~20GB per dataset
- Long processing time: ~80 hrs per dataset
- Many datasets: ~2000 datasets currently acquired
- Workflows consist of ~10 - 20 processing steps
Specifics Of High Content Screening

- Very large datasets: ~20 GB per dataset, ~10,000 files
  - Network copy of data is expensive
  - Infrastructure comes to its limits (slow access, random problems)
- Long processing time: ~80 hrs per dataset
  - Desktop processing impossible
  - Long iteration cycles for testing and recovery
- Many datasets: ~2000 datasets currently acquired
  - Hard to keep track of individual status of a dataset
- Workflows consist of ~10 - 20 processing steps
  - Errors often leave half-finished results behind
Typical workflows have ~10 – 20 steps (parallel, with a merge-step)
Failed and incomplete processings are hard to handle:

- Which plates failed to complete the workflow?
- Are the missing steps even essential?
- If missing steps are essential, how to avoid recomputation?
- If recomputing from scratch, how to purge duplicate results?

These problems (though seemingly simple) take a huge amount of time from a highly-qualified facility head or technician.
Method updates are hard to handle:

- Which datasets are from which method in which version/settings?
- Is an update of a method version/settings even essential?
- If an update of results is essential, which dependent datasets require recomputation as well? How to avoid full recomputation?
- How to infer the "status" of the full system?

These problems (though seemingly simple) take a huge amount of time, and/or waste enormous resources on the cluster.
Missing User Story (2/3)

Screens analyzed

- Shigella K2
- Bartonella Test
- Listeria K1, K2
- Listeria GW1
- Listeria GW2
- Bartonella K1
- Brucella K2
- Shigella GW1
- Brucella K1
- Shigella GW2
- Listeria K1, K2
- Rhino Test
- Brucella GW1
- Brucella GW2

How to match?

Rhino Test

Shigella K2

Listeria K1, K2

Listeria GW1

Listeria GW2

Bartonella K1

Brucella K2

Shigella GW1

Brucella K1

Shigella GW2

Listeria K1, K2

Rhino Test

Brucella GW1

Brucella GW2
Other (smaller) missing user storys:

- Intelligent resource handling (order staging and processing, avoid overbooking and flooding, ...)
- Tight acquisition integration (early sanity check, email notification)
- Data management (*)
- Easy search, resume, recover, remove (*)
- Good cluster integration: pause, stop, kill, resume, cleanup, ...
- Prioritization (by job type, job size, user, group, ...)
- CHROOT-like cluster environment

(*) covered by making use of storage provider capabilities(?)
Acknowledgements

This work is based on prior work done by:

Berend Snijder, Pauli Rämö, Lucas Pelkmans, Vincent Rouilly, Eva Pujadas, Bela Hullar, Michael Podvinec, and Peter Kunszt.

Thanks!
Questions, Discussion?