Workflow Manager for HCS

The Specific Problem(s) Of High Content Screening

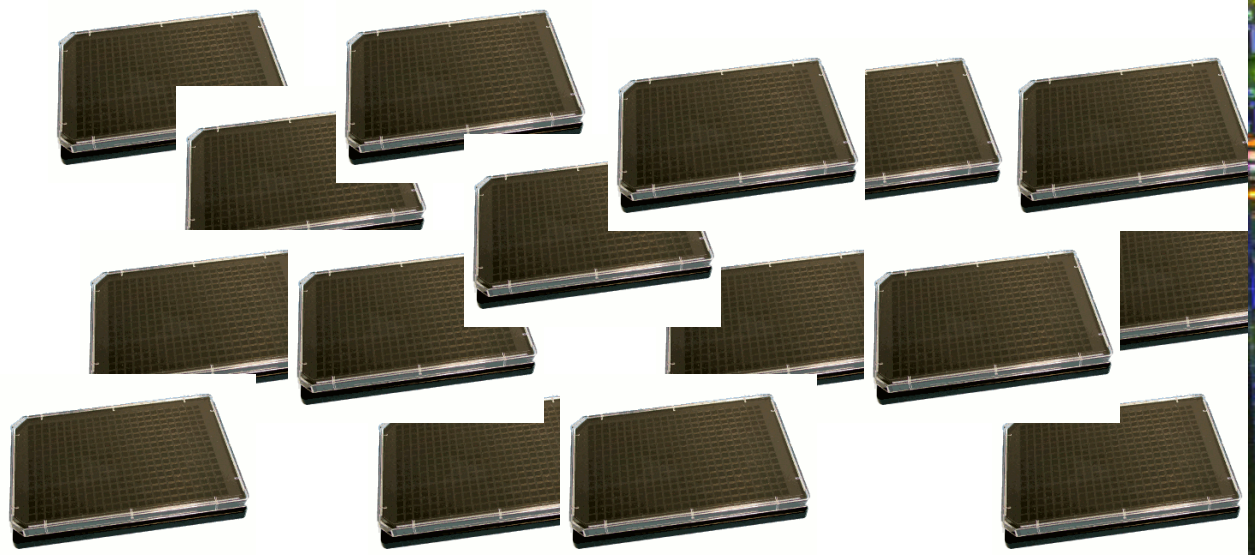
For The SyBIT Tech Day, Bern 03.10.2012

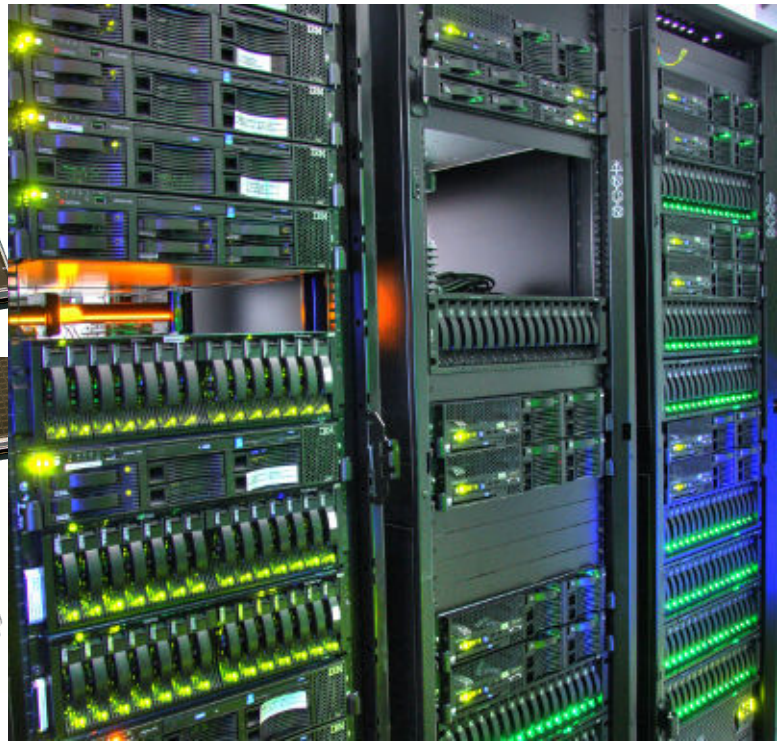
Specifics Of High Content Screening

- 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



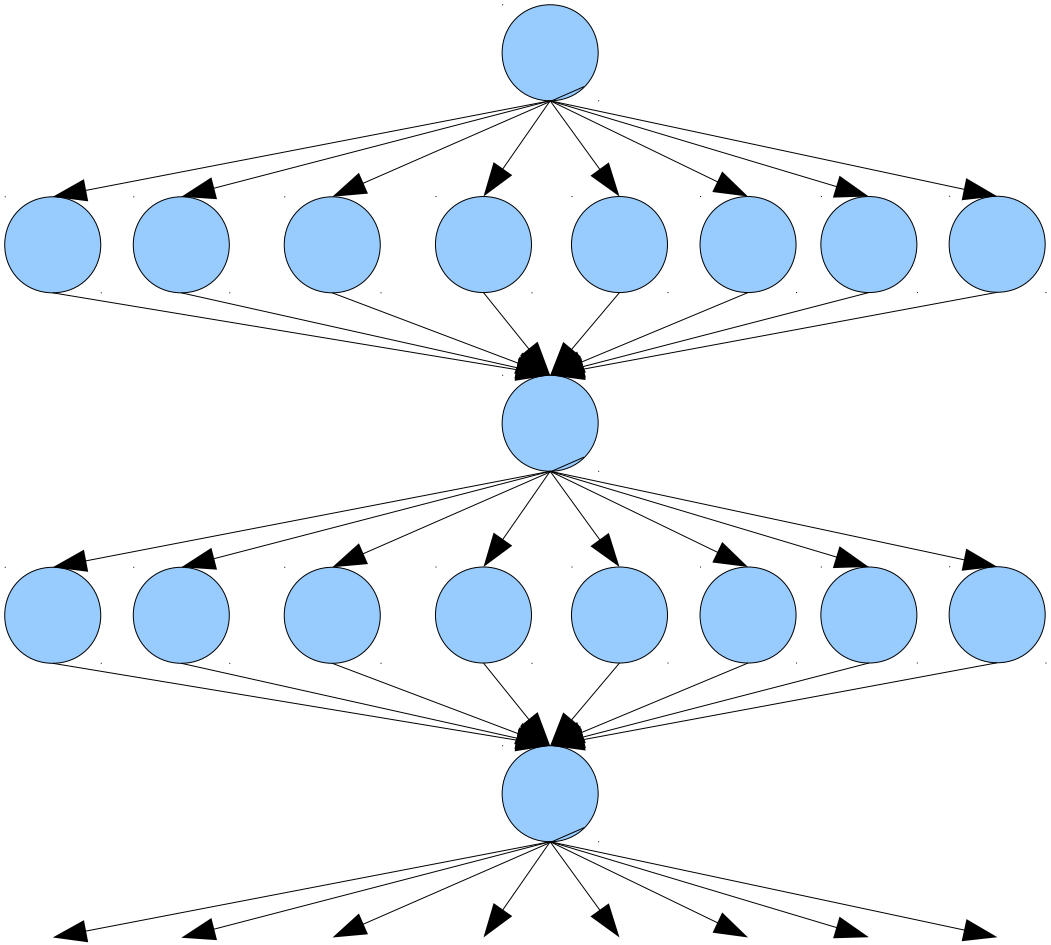


Specifics Of High Content Screening

- 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

Workflows

- Typical workflows have ~10 – 20 steps (parallel, with a merge-step)



...

Missing User Story (1/3)

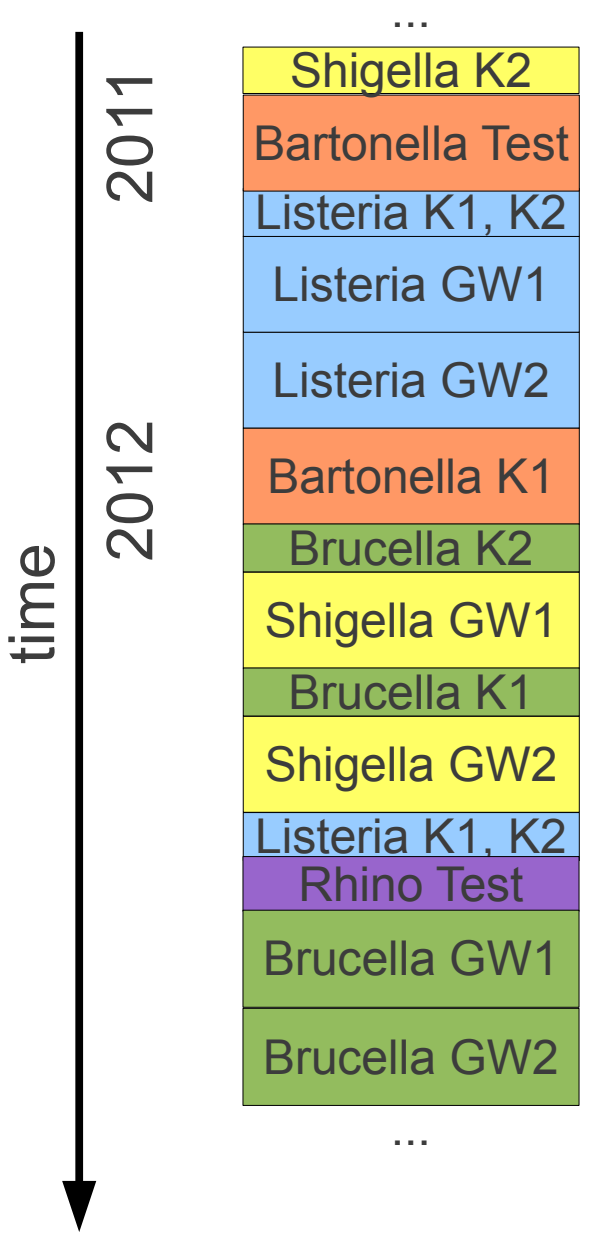
- Failed and incomplete processings are hard to handle:
  - Which plates failed to complete the workflow?
  - Are the missing steps even essential?
  - If missing steps are essential, how to avoid recomputation?
  - If recomputing from scratch, how to purge duplicate results?
- These problems (though seemingly simple) take a huge amount of time from a highly-qualified facility head or technician

Missing User Story (2/3)

- 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 dependend 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 ressources on the cluster

Missing User Story (2/3)

Screens analyzed



How to match?

QualityAssessment/QualityAssessment_rev725_R2009b-original
PlateSummary/PlateSummary_rev712_R2009b
FeatureZScoring/FeatureZScoring_rev712_R2009b
DataRefactoring/DataRefactoring_rev712_R2009b
ShadingCorrection/ShadingCorrection_rev712_R2009b
QualityAssessment/QualityAssessment_rev725_R2009b
PreviewThumbnails/PreviewThumbnails_rev756_R2009b
PlateMetaData/PlateMetaData_rev756_R2009b
ObjectClassification/ObjectClassification_rev756_R2010b
MetaDataExtraction/MetaDataExtraction_rev756_ome7235
InvasomeInfectionScoring/InvasomeInfectionScoring_rev756_R2009b
ExportData/ExportData_rev756_R2009b
ChannelSeparation/ChannelSeparation_rev756_R2009b
QualityAssessment/QualityAssessment_rev756_R2009b
ShadingCorrection/ShadingCorrection_rev756_R2009b
DataRefactoring/DataRefactoring_rev756_R2009b
FeatureZScoring/FeatureZScoring_rev756_R2009b
PlateSummary/PlateSummary_rev756_R2009b
MergePreClusterCellProfiler/MergePreClusterCellProfiler_rev756
QualityAssessment/QualityAssessment_rev770_R2009b
ShadingCorrection/ShadingCorrection_rev770_R2009b
MergePreClusterCellProfiler/MergePreClusterCellProfiler_rev770
DataRefactoring/DataRefactoring_rev770_R2009b
PlateSummary/PlateSummary_rev770_R2009b
FeatureZScoring/FeatureZScoring_rev770_R2009b
ChannelSeparation/ChannelSeparation_rev770_R2009b
ObjectClassification/ObjectClassification_rev770_R2009b
CP1/CP1_rev689_R2010b_10946
CP1/CP1_rev773_R2009b_10946
MetaDataExtraction/MetaDataExtraction_rev733_ome7235
MetaDataExtraction/MetaDataExtraction_rev770_ome7235
ChannelSeparation/ChannelSeparation_rev779_R2009b
CP1/CP1_rev802_R2009b_10946
CP1/CP1_rev807_R2009b_10946
ChannelSeparation/ChannelSeparation_rev807_R2009b
DataRefactoring/DataRefactoring_rev807_R2009b
ExportData/ExportData_rev807_R2009b
FeatureZScoring/FeatureZScoring_rev807_R2009b
MergePreClusterCellProfiler/MergePreClusterCellProfiler_rev807
InvasomeInfectionScoring/InvasomeInfectionScoring_rev807_R2009b
ObjectClassification/ObjectClassification_rev807_R2009b
PlateMetaData/PlateMetaData_rev807_R2009b
PlateSummary/PlateSummary_rev807_R2009b
PreviewThumbnails/PreviewThumbnails_rev807_R2009b
MetaDataExtraction/MetaDataExtraction_rev807_ome7235
ShadingCorrection/ShadingCorrection_rev821_R2009b
QualityAssessment/QualityAssessment_rev821_R2009b
PreviewThumbnails/PreviewThumbnails_rev821_R2009b
PlateSummary/PlateSummary_rev821_R2009b
PlateMetaData/PlateMetaData_rev821_R2009b
ObjectClassification/ObjectClassification_rev821_R2009b
MetaDataExtraction/MetaDataExtraction_rev821_ome7235
MergePreClusterCellProfiler/MergePreClusterCellProfiler_rev821
InvasomeInfectionScoring/InvasomeInfectionScoring_rev821_R2009b
FeatureZScoring/FeatureZScoring_rev821_R2009b
ExportData/ExportData_rev821_R2009b
DataRefactoring/DataRefactoring_rev821_R2009b
CP1/CP1_rev821_R2009b_10946
ChannelSeparation/ChannelSeparation_rev821_R2009b
ChannelSeparation/ChannelSeparation_rev840_R2009b
DataRefactoring/DataRefactoring_rev840_R2009b
ExportData/ExportData_rev840_R2009b

Missing User Story (3/3)

- Other (smaller) missing user storys:
  - Intelligent ressource handling (order staging and processing, avoid overbooking and flooding, ...)
  - Thight acquisition integration (early sanity check, email notificate)
  - 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!

End of Presentation

- Questions, Discussion?