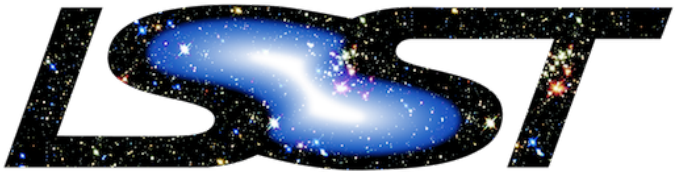

LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)

QA Strategy Working Group Report

Swinbank, J.D, Bellm, E.C., Chiang, H.-F., Fausti, A., Krughoff, K.S., MacArthur, L.A., Morton, T.D. and Roby, T.

DMTN-085

Latest Revision: 2018-07-05

DRAFT

Abstract

Abstract.

Change Record

Version	Date	Description	Owner name
5c96f08	2018-07-05	Unreleased draft.	Bellm et al.

Contents

QA Strategy Working Group Report

1 Introduction

Assignee

John

This report constitutes the primary artefact produced by the DM QA Strategy Working Group (QAWG), addressing its charge as defined in LDM-622.

2 Approach to the Problem

Assignee

John

2.1 Pipeline debugging

Assignee

John

What tools do we need to help pipeline developers with their everyday work?

- How do you go about debugging a `Task` that is crashing?
- Is `lsstDebug` adequate?
- Do we need an afwFigure, for generating plots, to go alongside `afwDisplay`, for showing images?
- What additional capabilities are needed for developers running and debugging at scale, e.g. log collection, identification of failed jobs, etc.
- What's needed from an image viewer for pipeline developers? Is DS9 or Firefly adequate? Is there value to the afwDisplay abstraction layer, or does it simply make it harder for us to use Firefly's advanced features?

- How do we view images which don't fit in memory on a single node?
- How do we handle fake sources? Is this a provenance issue?

2.2 Drill down

Assignee
John

2.3 Datasets and test infrastructure

Assignee
John

3 Design sketch

3.1 Pipeline debugging

Assignee
John

3.2 Drill down

Assignee
Tim

3.3 Datasets and test infrastructure

Assignee
John

4 Core components

4.1 Updated pipeline debugging system

Assignee
Simon

i.e. redesigned `lsstDebug`.

4.2 Logging

Assignee
Simon

4.3 Capability for developers to run pipelines at scale

Assignee
Lauren

4.4 Guidance on visualization

Assignee
Lauren

We're requesting a set of guidelines for developers here, not a new framework — but that's still a concrete deliverable (it's just documentation, rather than code). We might suggest that these guidelines be developed by a new WG, per Simon's suggestion[1].

4.5 Image viewer

Assignee
Trey

As of 2018-06-12 we haven't converged on a solid recommendation here.

Key considerations:

[1] `https://confluence.lsstcorp.org/display/DM/Pipeline+Debugging+Design`

- Firefly is the annointed solution being provided by DM to external stakeholders (commissioning, operations, etc). It feels right to everybody that we should be dogfooding it, and also benefitting from development being carried out for those stakeholders.
- Currently, Firefly is unappealing to developers (primarily, I think, because of slowness of user interface, and perhaps also due to installation issues). Can we resolve these issues?
- We'd want to support visualization in a number of different environments, for e.g.:
  - Inside a Jupyter notebook;
  - As a standalone tool, à la DS9;
  - Embedded in a dashboard, à la JS9, Aladin-Lite, etc.
- Do we lose flexibility by mandating the use of a backend-agnostic API (`afwDisplay`) rather than going "all-in" on e.g. a custom Firefly interface?
- We'll need to do full focal plane visualization, which none(?) of the current tools support well.

Options include:

- Do nothing; continue as we are, which means most people will use DS9 and a few will drift to Firefly as commissioning ramps up.
- Issue some sort of edict that pipelines developers have to use Firefly.
- Encourage the use of some other tool (Ginga?) instead of or as well as some of the above.
- Probably others.

Sounds like we need somebody from the QAWG to actually write some requirements — or a wishlist set of features we want — here.

4.6 Catalog visualization tools

Assignee
Lauren

For visualizing bigger-than-memory catalogs. May include e.g. the capability to spin up Dask clusters on demand, combined with Holoviews/Datashader/whatever. Somebody who knows about this stuff needs to write a summary...

4.7 Provenance

Assignee
Hsin-Fang

This section should note:

- That provenance is an immediate issue impacting QA work, so a solution is a priority;
- Some requirements as to the granularity at which provenance tracking is necessary for QA.

A Glossary

aggregate metric An aggregation of multiple point metrics. For example, the overall photometric repeatability for a particular tract given multiple observations of each star.

aggregation A single result—e.g., a metric value—computed from a collection of input values. For example, we can sum or average a metric computed over patches to produce an aggregate metric at tract level.

dashboard A visual display of the most important information needed to achieve one or more objectives, consolidated and arranged on a single screen so that the information can be monitored at a glance (Few, 2013).

drill down Move from a higher level aggregation of data to its inputs. For example, given data describing a tract, we might drill down to constituent patches and then to objects; given a visit, we might drill down to CCD and then source. In the context of this document, it refers to the act of identifying an issue in a high-level summary of the data (e.g. an aberrant metric value) and interactively investigating its inputs to find the source of the problem.

KPM Key Performance Metric.

metric We follow the SQR-019 definition of a metric as a measurable quantities which may be tracked. A metric has a name, description, unit, references, and tags (which are used

for grouping). A metric is a scalar by definition. We consider multiple types of metric in this document; see aggregate metric, model metric, point metric.

metric value The result of computing a particular metric on some given data. Note that we *compute*, rather than measure, metric values.

model metric A metric describing a model related to the data. For example, the coefficients of a 2D polynomial fit to the background of a single CCD exposure.

monitoring The process of collecting, storing, aggregating and visualizing metrics.

point metric A metric that is associated with a single entry in a catalog. Examples include the shape of a source, the standard deviation of the flux of an object detected on a coadd, the flux of an source detected on a difference image.

QAWG QA Strategy Working Group.

releaseable product A software package or other component of the DM system which is expected to be included in the next tagged release of the system. At time of writing, this implies inclusion in a standard top-level package (e.g. lsst_distrib), but we note that future changes to the release procedure may render that definition obsolete.

tidy data Tidy datasets have a specific structure: each variable is a column, each observation is a row, and each type of observational unit is a table (Wickham, 2014).

References

Few, S., 2013, *Information Dashboard Design*, Analytics Press, 2 edn.

[SQR-019], Sick, J., Fausti, A., 2018, *LSST Verification Framework API Demonstration*, SQR-019, URL `https://sqr-019.lsst.io`

[LDM-622], Swinbank, J., 2018, *Data Management QA Strategy Working Group Charge*, LDM-622, URL `https://ls.st/LDM-622`

Wickham, H., 2014, Journal of Statistical Software, Articles, 59, 1, URL `https://www.jstatsoft.org/v059/i10`, doi:10.18637/jss.v059.i10