LARGE SYNOPTIC SURVEY TELESCOPE

Large Synoptic Survey Telescope (LSST)
Data Management

Batch Production Services Design

Michelle Gower and Kian-Tat Lim

DMTN-123

Latest Revision: 2019-08-06

Abstract

The LSST DM Batch Production Services are designed to allow large-scale workflows, including the Data Release and Calibration Product Productions, to execute in well-managed fashion, potentially in multiple environments. They ensure that provenance is captured and recorded to understand what environment and parameters were used to produce any dataset.

Change Record

Version	Date	Description	Owner name
1	2019-07-08	Initial version extracted from LDM-152.	Kian-Tat Lim

Document curator: Michelle Gower

Document source location: `https://github.com/lsst-dm/dmtn-123`

Contents

Batch Production Services Design

1 Introduction

This document describes the baseline design of the LSST battch production services, including the following components:

- Workload/Workflow Management
- Processing Control

Workload and Workflow Management interfaces with the Task Framework from the Data Management Middleware [LDM-152] to sequence the execution of dataflow graphs across one or more distributed computational environments. Processing Control uses the Workload and Workflow Management tools to execute campaigns (applications of pipelines with specific configurations to sets of data) in an efficient, fault-tolerant manner while monitoring the state of execution.

The substantial computational and bandwidth requirements of the LSST Data Management System (DMS) force the designs to be conscious of performance, scalability, and fault tolerance.

Use cases for the Batch Production Services are given in LDM-633 and requirements are defined in LDM-636.

Figure 1 illustrates how various parts of the middleware interact with each other.

2 Workload/Workflow Management

The Workload and Workflow Management component provides management of the execution of science payloads ranging from a single pipeline to a series of “campaigns”, each consisting of multiple pipelines. Its services are able to handle massively distributed computing, executing jobs when their inputs become available and tracking their status and outputs. They ensure that the data needed for a job is accessible to it and that outputs (including log files, if any) are preserved. They can allocate work across multiple computing environments, in particular between NCSA and the Satellite Computing Facility at CC-IN2P3.

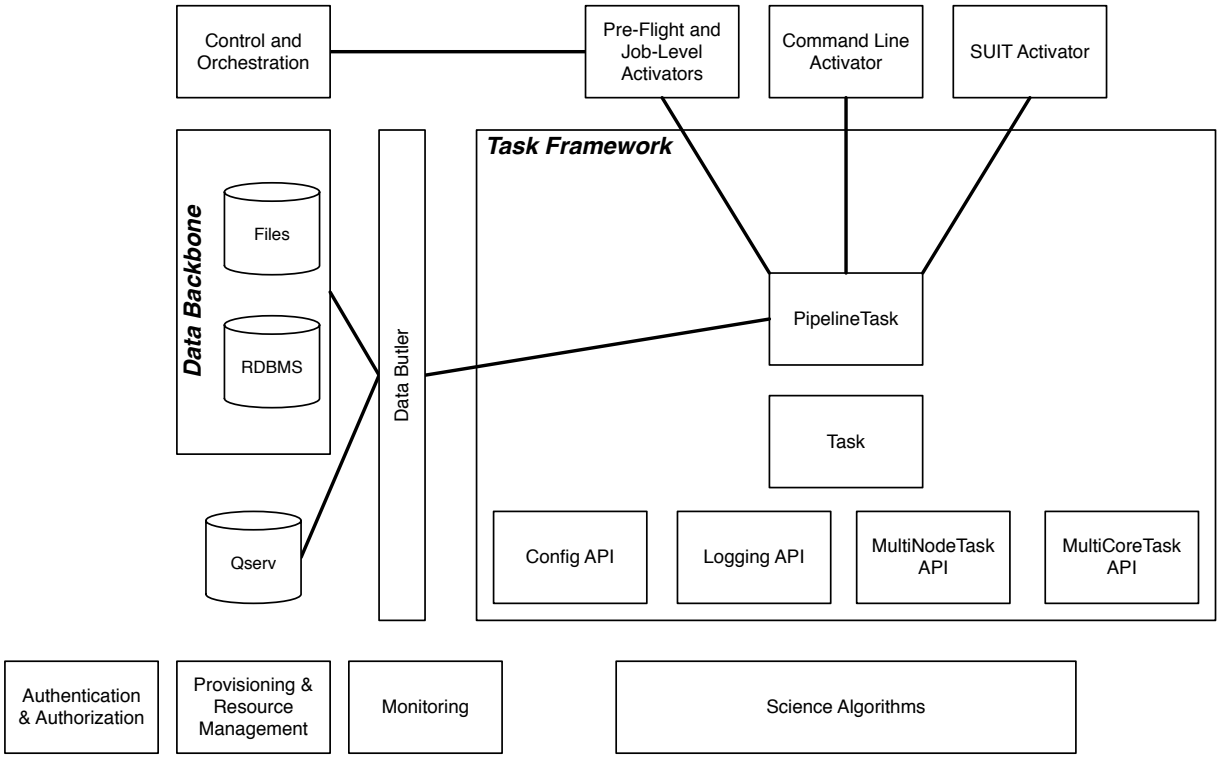

FIGURE 1: Data Management Middleware and Infrastructure

This component invokes SuperTasks (section **??**) via Activators to sequence the execution of dataflow graphs across one or more distributed computational environments, in particular compute resources allocated via a batch computing service.

2.1 Batch Computing

Batch computing occurs on LSST dedicated computing platforms at NCSA and CC-IN2P3 and potentially on other platforms. Resources other than local CPU and storage for computation, such as curation storage to hold final data products (the Data Backbone, section **??**) and network connectivity, are also needed to completely execute a pipeline and completely realize the data handling scheme for input and output datasets; the Workload and Workflow component utilizes these as well.

Computing resources are physical items which are not always fit for use. They have scheduled and unscheduled downtimes and may have scheduled availability. The management of campaigns requires the detection of unscheduled downtimes of resources, recovery of executing pipelines affected by unscheduled downtimes, and arranging for the best use of available resources.

One class of potential resources are opportunistic resources which may be very capacious but may not guarantee that jobs run to completion. These resources may be needed in contingency circumstances. The workload management system is capable of differentiating "kills" from other failures so as to enable use of these resources.

2.2 Workflow Management

The Workflow Management service provides for orchestration and execution of pipelines. Its basic functionality is as follows:

- Pre-job context:
  - Supports pre-handling of any input pipeline data sets when in-job context for input data is not required.
  - Pre-stages input data into a platform's storage system, if available.
  - Produces condensed versions of database tables into portable lightweight format when required.
  - Deals with platform-specific edge services.

  - Handles identities and provides for local identity on the computing platforms.
  - Provides credentials and end-point information for any needed LSST services.
- In-job context:
  - Provides stage-in for any in-job pipeline input data sets.
  - Provides any Butler configurations necessarily provided from in-job context.
  - Invokes the pipeline and collects pipeline output status and other operational data.
  - Provides stage-out for pipeline output data sets when stage-out requires job context.
- Post-job context:
  - Ingests any designated data into database tables.
  - Arranges for any post-job stage out from cluster file systems.
  - Arranges for detailed ingest into custodial data systems.
  - Transmits job status to workload management.

The baseline design for the Workflow Management service uses a workflow tool such as Pegasus (Pegasus) together with HTCondor (HTCondor), custom Activators, and Data Backbone interface scripts.

2.3 Workload Management

Workload Management considers the ensemble of available compute resources and the ensemble of campaigns to be executed and dispatches pipeline invocations to the Workload orchestration system based on resource availability and campaign priority. It considers pipeline failures reported by the Workload orchestration system, distinguishing errors from computing resources and computational errors where possible, and arranges for incident reports and retrying failed invocations when appropriate. It exposes the progress of each campaign to operations staff and monitoring systems, providing appropriate logging and events.

The prototype for the software implementing this service is in GitHub repository `lsst-dm/ctrl_bps`.

A References

References

HTCondor, HTCondor, URL https://research.cs.wisc.edu/htcondor/index.html

[LDM-633], Kowalik, M., Gower, M., Kooper, R., 2019, *Offline Batch Production Services Use Cases*, LDM-633, URL https://ls.st/LDM-633

[LDM-636], Kowalik, M., Gower, M., Kooper, R., 2019, *Batch Production Service Requirements*, LDM-636, URL https://ls.st/LDM-636

[LDM-152], Lim, K.T., Dubois-Felsmann, G., Johnson, M., Jurić, M., Petravick, D., 2017, *Data Management Middleware Design*, LDM-152, URL https://ls.st/LDM-152

Pegasus, Pegasus WMS, URL https://pegasus.isi.edu/

B Acronyms