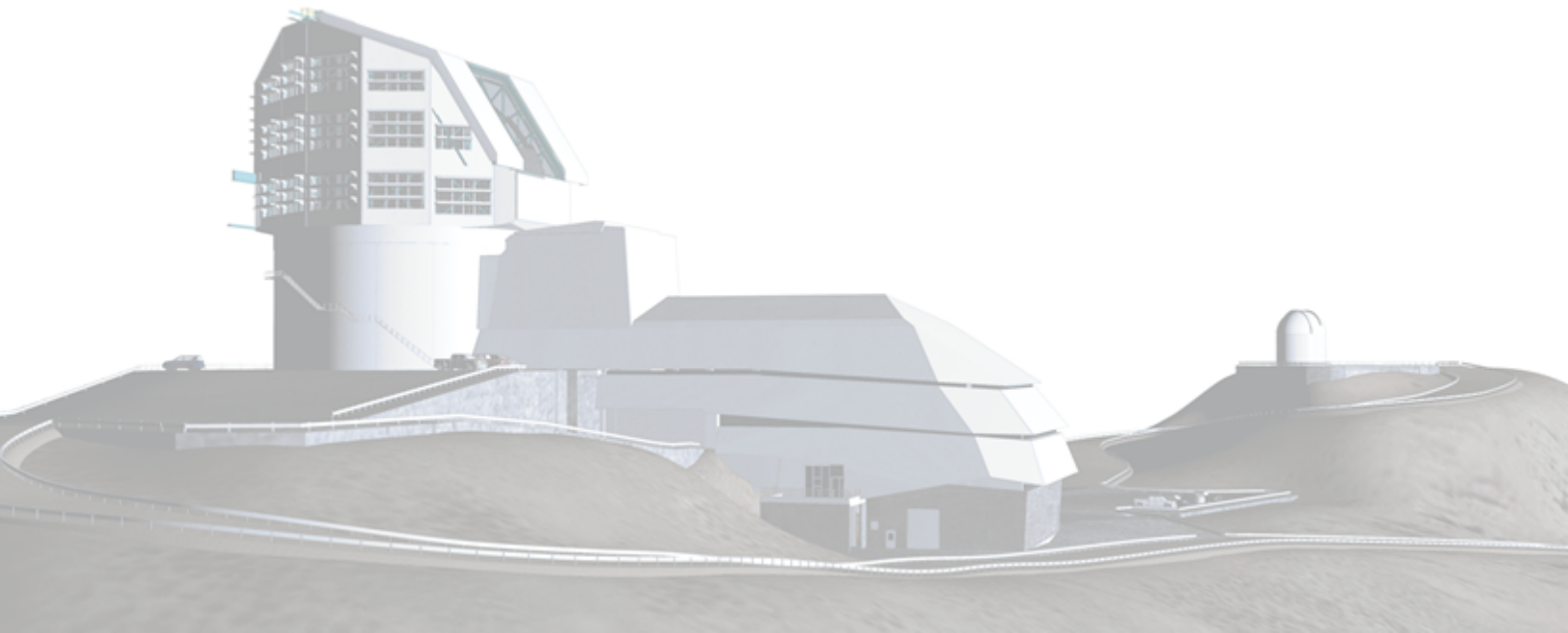
Vera C. Rubin Observatory
Rubin Observatory Operations

Compute Resource Usage of DP0.2 production run

Brian Yanny, Nikolay Kuropatkin, Huan Lin, Jennifer Adelman-McCarthy, Colin Slater, Hsin-Fang Chiang

RTN-039

Latest Revision: 2023-07-10



Abstract

We present a summary of wallclock and cpu time, memory and storage usage of the DP0.2 production run in the iDF. Over the course of 150 calendar days in 2021-2022, preliminary Data Release Production (DRP) pipelines ran on 20,000 exposures of simulated Rubin data, representing 10-20 full nights of images, covering 150 tracts (300 square degrees), to typically 5 year depth (50 returns to the same spot on the sky for each of 6 filters). This 'half-percent' survey used a compute cluster of approximately 4000 cores, each core with up to 14 GB/core RAM available (less typically used) with jobs taking typically a few seconds to a fraction of an hour. Small subsets (a few percent) of the processing pipetasks used nodes with up to 236 GB/core, and were allowed to run for up to 3 days. The input images were approximately 75TB in size, while the outputs, counting transient datasetTypes took up 3.4PB in an object store. After removing transient datasetTypes (e.g. warps), the final storage used by DP0.2 was approximately 2.5 PB. Of the 150 calendar days, approximately 60 days were running and the remainder used for debugging. Thus, to keep up with expected DRP during the course of the survey (6-12 days to process 10-20 nights of data), a factor of 5-10 speed up (or increased core counts, or some combination of both) are estimated to be needed.

Change Record

Version	Date	Description	Owner name
1	YYYY-MM-DD	Unreleased.	Brian Yanny

Document source location: `https://github.com/lsst/rtn-039`

Contents

Compute Resource Usage of DP0.2 production run

1 Introduction and Scope of DP0.2

DP0.2 (Data Preview 0.2) consisted of the (re)processing of a set of approximately 20,000 Rubin camera simulated exposures (each with up to 189 detectors per exposure) covering 157 tracts (300 square degrees) in 6 filters ($ugrizy$) to a typical depth of half that of the full survey. These 20K visits represent approximately 10 nights of data gathering, roughly 0.5% of the full survey.

Processing of the DESC-generated DC-2 simulated dataset (Collaboration, DESC) took place between Dec 18, 2021 and May 16, 2022. Processing took place in a Google Cloud environment, named the iDF (Interim Data Facility). In excess of 4,000 cores were available in a number of queues and memory/core configurations within the cloud production environment. The total cpu usage over the course of DP0.2 was approximately 2.5M core-hours. Storage was provided in the form of S3 addressable object storage space. At peak, the production used 3.3 PiB of object store. After removal of intermediate data products (Warp images and intermediate single epoch CCD images), the final storage in the S3 data store at the end of DP0.2 production amounted to approximately 2.55 PiB.

The scope of production consisted of running seven pipeline steps. Each step was in turn made up of one or more pipetasks. Before production started, the 20K raw exposures/visits were pre-loaded into S3 storage, along with dimension and URI metadata in a common butler registry, used for all processing.

A Skymap and Photometric solutions for the raw exposures were also provided ahead of time – these were not created as part of the DP0.2 processing.

After processing was complete, a copy of the large butler registry was made for end user access and HiPS (large-format zoomable, pannable) image maps of the tracts were generated. Also, the detected and measured output object and source tables were put into a qServ database for collaboration access. These Butler copy, HiPS creation and qServ loading operations are not included in the compute resources tabulated here.

2 Software Configuration

The versions of the LSST software environment used for production were based on **v23** of the lsst distribution. After the start of step 1 processing a number of updates to the software occured to address issues encountered during processing and this resulted in the small revision numbers of the **v23** distribution to change during DP0.2.

The workflow software system was PanDA (PanDA), and the BPS (Batch Processing Service) software (part of the LSST distribution) was used to interface between the Butler and PanDA using a set of **yaml** job submission files (one for each group of each step of processing, see below). In addition, the Google Cloud logging system was used to collect logging information output by the production pipelines and organize it into a searchable form sorted by date of production.

3 Processing Groups and clustering directives

Each step of the seven step processing flow for DP0.2 (modeled on the eventual Data Release Production (DRP) system planned for regular Rubin processing) consists of a well defined, ordered list of pipeline tasks (pipetasks) run on either a set of single or dual exposure visits (steps 1,2,4,6) or on coadd tracts (steps 3, 5, 7) (and their associated visits).

The BPS system takes as input a yaml file. The yaml files used during DP0.2 are archived at: `https://github.com/lsst-dm/dp02-processing/tree/main/full/production`. As step 1 production commenced it was determined that an arbitrarily large number of quanta could not be successfully processed with a single BPS submission. For example, step 1 production runs five pipetasks on 20K visits, where each visit consists of typically 150 detector images, thus resulting in 20K*150*5 = 15M individual quanta of processing for step 1.

A key component of the PanDA workflow system is the ”Intelligent Data Delivery Service” (IDDS) which manages the scheduling of individual processing quanta to a set of processing cores. The BPS and step description yaml configurations guide the creation of a ”quantum graph” which fully describes the job flow dependencies (what tasks depend on what other tasks being complete). Also created at the same time as the quantum graph for each ’BPS submit’, is a so-called ’execution butler’ which contains file metadata and URI information for all inputs needed to completely run a BPS processing submission.

While one can define a single large DP0.2 step 1 BPS configuration yaml file (specifying all 20K visits to be processed), time and memory restrictions on the Butler, PanDA and compute systems prevent successful completion of the implied 15M quanta processing from a single BPS submission.

One issue encountered was that the time to generate the quantum graph and execution butler was prohibitive, it would still take days and many hundreds of GB to generate such quantum graphs and execution butlers. Practically speaking, there is a limit of a few hours and a few GB for preparation of and size of the quantum graph and execution butler for a single BPS submission.

3.1 Clustering of pipetasks

A second limitation on how large a workflow can be accomodated in one BPS submission comes with the operation of the IDDS and how pipetasks are clustered (or not clustered). The IDDS system has within it an N^2 algorithm which matches jobs done to jobs-to-be-done and determines which jobs may be run next, having their inputs already produced as outputs from the previous step.

To use the Butler and PanDA workflow system in practice, some division of sets of pipetasks into groups is done. Also, clustering sets of related, one-after-the-other pipetasks (in step 1, for example the five pipetasks are run on each exposure/detector in this order: **isr** followed by **characterizeImage** followed by **calibrate** followed by **writeSourceTable** followed by **transformSourceTable**). Since this dependence structure is fixed and invariant it helps to reduce the load on IDDS if these five pipetasks are clustered into one pipetask, called **visit_focal_plane_1**. While IDDS generally checks all running jobs and tries to match up any done job and its outputs to any requested waiting-to-run job with inputs that match the outputs aleady completed, having some ordered lists of pipetasks combined into a clustered task allows IDDS to not have to match everything against everything in these cases. There are also savings in complexity in generating the quantum graph in these cases. Since the order and set of pipetasks is fixed and every output always flows to the next pipetask's input, IDDS need not try to match every possible pipetask **B** to follow on from pipetask **A**, thus greatly reducing the number of possible combinations panDA IDDS needs to examine to determine which quanta with which inputs to run next through which job. The named definition of clusters of pipetasks used in DP0.2 processing are given in **clustering.yaml** files in `https://github.com/lsst-dm/dp02-processing/tree/main/full/production` for each step.

4 PanDA queues

BPS submissions in DP0.2 ran in the Google Cloud, and a set of six panDA job queues weere established with different retry characteristics and memory limits. The cost to use computing in the cloud increases linearly with maximum memory allocated per job and also computing is a factor of several times cheaper if the queue could be 'preempted' (forcing the job to run again from the start). These queues are listed in Table 1. Due to this structure, the cheapest queue is the pre-emptable lowest max memory queue, and the most expensive is the non-preemptable highest max memory queue. The default queue for BPS submits is the **GOOGLE_TEST** (pre-emptable) queue and the vast majority of jobs completed successfully in this queue. The maximum number of retries allowed was set by default to be 5, and the usual reason for a job to be retried was preemption. If a job failed exactly the same way 5 times typically this indicated a pipeline problem, and these cases were fed back to the pipeline development team for investigation. If a long running job failed 5 times at different points in the processing, this often indicates rather a repeated preemption, and suggests a **rescue** submission may be needed running with a higher max memory or in a non-preempt queue.

The special queue, GOOGLE_MERGE, is used for merging output metadata and objects (image files, catalogs) back into the main Butler registry (for metadata) with links to these objects recorded in the object store portion of the Butler.

TABLE 1: PanDA queues used for DP0.2 processing

queue	maxMem(GB)	used by	Note
DOMA_LSST_GOOGLE_TEST	14	default	
DOMA_LSST_GOOGLE_MERGE	14	butler merge	
DOMA_LSST_GOOGLE_HIMEM	40		
DOMA_LSST_GOOGLE_HIMEM_NON_PREEMPT	40		
DOMA_LSST_GOOGLE_EXTRA_HIMEM	236		
DOMA_LSST_GOOGLE_EXTRA_HIMEM_NON_PREEMPT	236		

5 Memory usage

A few pipetasks required more than the usually amount of memory, and these tasks were assigned to a higher max memory queue. Also, a few related Q/A pipetasks occasionally re-

quired exceedingly long runtimes on selected quanta, these were handled by a rescue BPS submissions, each with their own bps submit yaml which indicated to the butler to only run on quanta which did not previously complete, and indicated to panDA to run in a special non-preemptable (NO PE) queue. The pipetasks which required extra memory or no-preempt status for at least some quanta are listed in Table 2. Also listed in Table 2 are typical maximum memory requirements for most quanta running through an indicated pipetask.

TABLE 2: Memory requirements for processing pipetasks

pipetask	reqMem(MB)	Notes
makeWarp	8192	
assembleCoadd	16384	
deblend	16384	
measure	16384	
mergeMeasurements	16384	
forcedPhotCoadd	4096	
writeObjectTable	16384	
matchCatalogsPatch	4096	
modelPhotRepGal4	4096	
imageDifference	4096	
matchCatalogsPatchMultiBand	120000	
matchCatalogsTract	120000	
wPerp	120000	
TE1	46000	
TE2	46000	
consolidateObjectTable	4096	
consolidateForcedSourceTable	120000	
consolidateForcedSourceOnDiaObjectTable	18000	
deblend	32768	step 3 rescue
measure	16384	40 GB NO PE
deblend	32768	step 3 faro rescue
deblend	32768	236GB NO PE
matchCatalogsPatchMultiBand	120000	236GB NO PE
matchCatalogsTract	120000	236GB NO PE
consolidateForcedSourceOnDiaObjectTable	18000	step 5 40 GB

6 Storage

Storage for DP0.2 inputs and outputs was done using a large S3 compatible (key= unique POSIX-style path+filename plus bucket prefix, value=pointer to bytes of a data file or table) object store in the Google Cloud. The datasetType names for the types of objects which used the bulk of the storage are listed in Table 3. Also listed for each datasetType are estimates of the number of DP0.2 objects of a given type, the typical size of a single object of that type (in MB), the total DP0.2 storage for objects of that type (in TB), and which step in the processing creates objects of that type. Also, if a datasetType is transient, and is not retained once the downstream pipetasks which make use of that datasetType are complete, the datasetType is marked with a 'D' in the 'Delete' column. The warp images are the largest example of datasetTypes that are transient. To illustrate, important retained datasetType are **calexp** objects, representing a single (exposure,detector) corrected and calibrated ccd image. Each **calexp** takes about 101 MB of space, and there are 2.8M in the DP0.2 dataset (20,000 exposures times 150 detectors/exposure, on average, gives 3M as a rough estimate of the number of CCD images in the DP0.2 raw input dataset). Note that which the full LSST camera has 189 detectors, for DP0.2, on average only 150 of these are present per exposure, though some exposures do have all 189 detectors present in DP0.2. Recall that in rough approximation for some subset of data production and data products, DP0.2 represents one-half of one percent (1/200th) of the full LSST.

7 Processing Summary

DP0.2 processing took place in 2021-2022, over the course of about 150 calendar days. Processing was done step-by-step, using the 7 pipetask sets defined by the pipeline team in the v23 release of the processing software distribution. Work was done to break the exposures or tracts needed to be processed into managable **groups**. The size of each group in terms of numbers of exposures or tracts was set so that no quantum graph or execution butler generation took more than about 1 hour per group, and so that no group's total run time exceeded approximately 1 day (24 hours). Additionally, the panDA directive: maxJobsPerTask: 30000 was added to some **step's** BPS submit yaml files to allow panDA to break groups with large number of quanta into sets of no more than 30,000. This allowed the IDDS job matching algo-

TABLE 3: Storage by datasetType

TB	Nobj	MB/obj	step	D	dataSetType
75	2.9M	26	input		raw
506	5.1M	104	step3	D	deepCoadd_directWarp
506	2.2M	230	step4		goodSeeingDiff_templateExp
389	5.1M	80	step3	D	deepCoadd_psfMatchedWarp
293	2.9M	102	step1	D	icExp
283	2.8M	101	step1		calexp
258	2.9M	90	step1	D	postISRCCD
224	2.2M	102	step4		goodSeeingDiff_differenceExp
200	3.4M	60	step4		forced_diff
184	4.6M	40	step5		forced_diff_diaObject
150	3.4M	44	step4		mergedForcedSource
39	0.6M	651	step3		deepCoadd_meas
18	4.6M	4	step5		forced_src_diaObject
12	7.5K	1600	step5		forcedSourceTable
12	0.6M	196	step3		deepCoadd
12	0.6M	196	step3		deepCoadd_calexp
10	2.6M	3.7	step6		calibratedSource
8.4	2.8M	3	step1		source
8.3	0.6M	141	step3		goodSeeingCoadd
5	4.6M	1.2	step5		mergedForcedSourceOnDiaObject
3.4	3.4M	1.0	step4		forced_src
3.4	2.8M	1.20	step1		sourceTable
2.4	2.8M	0.85	step1		icSrc
1.6	10K	170	step3		objectTable
1.3	2.6M	0.5	step6		calibratedSourceTable
1	20K	50	step6		calibratedSourceTable_visit
1.2	20K	58	step2		sourceTable_visit
1.0	156	6600	step3		objectTable_tract
0.5	60K	6.4	step3		deepCoadd_inputMap
0.4	2.2M	0.18	step4		goodSeeingDiff_diaSrc
0.3	2.8M	0.11	step1		calexpBackground
0.28	7.5K	36	step5		forcedSourceOnDiaObjectTable
0.2	150	1900	step5		forcedSourceOnDiaObjectTable_tract
0.2	2.9M	0.08	step1		icExpBackground
0.2	60K	4	step3		deepCoadd_det
0.2	60K	3	step3		deepCoadd_measMatchFull
0.12	2.2M	0.06	step4		goodSeeingDiff_diaSrcTable
0.1	60K	2.4	step3		deepCoadd_nImage
0.1	60K	1.4	step3		goodSeeingCoadd_nImage
0.02	20K	1	step6		diaSourceTable
0.02	2.8M	0.01	step1		srcMatch (src:7MB)
0.02	20K	0.10	step2		visitSummary
3.4PB(1.9PB)					Total in S3 without(with) Deletes

rithm, which determines what outputs are ready to be fed in as inputs to the next pipetask in a set, to avoid taking excessive memory, as the memory usage and compute time scale as N^2 where N is the number of jobs in the task. The limit of 30,000 allowed panDA IDDS to run on a machine with 12GB RAM ($30000^2 \sim 1G$). Some trial-and-error was required to determine the optimal group size, in visits (steps 1,2, 4) or tracts (steps 3,5,6,7), for each **step**. Most quanta in a group ran for similar lengths of time and used similar amounts of memory. There were, however, cases where a small percentage of quanta exceeded the default memory of a queue and hung, or were repeatedly pre-empted as they ran too long (sometimes for days), some of the Quality Assurance metric tasks fell into this category. For these cases, nearly all in step 3 or step 5 processing, rescue BPS workflow yamls were designed and submitted to higher max memory queues or to non-preempt queues to allow the subset of failed quanta to have the resources needed to complete.

7.1 DP0.2 Campaign tracking

The preliminary operations campaign tools available in the **prodstatus** repo `https://github.com/lsst-dm/prodstatus` were used to monitor DP0.2 processing and record compute resource metrics in a series of JIRA tickets, summarizing information about each group run for each of the seven DRP steps. The original ticket requesting DP0.2 DRP processing is `https://jira.lsstcorp.org/browse/PREOPS-905`, which contains extensive comments about issues that came up during processing. A document describing the DP0.2 campaign in some detail is available as `https://rtn-041.lsst.io/v/DM-17130/index.html`.

Table 4 shows, for each step, 1) whether the step is exposure(visit) or tract based, 2) the number of groups the exposures or tracts are broken into, 3) the starting and ending calendar dates for processing, 4) the approximate number of core-hours used, and 5) a software version stamp. In total, about 2.5M core-hours were used (on a cluster with about 4000 cores) over 150 calendar days. Of these 150 days, approximately 60 days were used for running, with most of the remainder used for debugging, rescue design, and group sizing trial-and-error. 2.5M/4000/24 = 26 days, so cores were used at less than 50% efficiency.

The input dataset represents data typically gathered over 10-20 observing nights. DRP is required to process (not considering cumulative reprocessing here) about 300 nights of data in 200 days, and so, 10-20 nights should be turned around in 6-12 calendar days. These very approximate figures can be used to estimate rough factors by which the processing needs to be sped-up or the factor by which the number (or efficiency) of cores needs to be increased

(or some combination of the two). Roughly a factor of 5-10 speed up or size up is estimated to be needed.

TABLE 4: Summary of DP0.2 cpu usage by step

step	group by	N groups	start	end	core-hr	software	Note
step1	visit/det	14x3	2022-12-18	2022-01-12	166K	v23_0_0_rc5	
step2	visit	14	2022-01-20	2022-01-24	22K	v23_0_0_rc5	
step3	tract	33	2022-02-18	2022-03-25	1100K	v23_0_1	
step4	visit	40	2022-04-01	2022-04-30	1100K	v23_0_1_rc4	
step7	all	1	2022-05-01	2022-05-01	10	v23_0_2_rc2	
step5	tract	52	2022-05-03	2022-05-12	66K	v23_0_2_rc2	
step6	visit	20	2022-05-12	2022-05-16	16K	v23_0_2_rc3	
purge	exp/warp	2	2022-04-01	2022-05-30	10	v23_0_2_rc2	
total			150d		2500K		

The JIRA issues which record the **BPS** submit yamls for each group within each of the **steps** of prodstatus tracking of DP0.2 processing are linked from JIRA issue `https://jira.lsstcorp.org/browse/DRP-473`. From this **campaign-level** issue, one can access the seven individual **step-level** issues, and from each of these in turn, the individual BPS submit workflow groups can be accessed, with one JIRA issue per BPS submission. Several hundred BPS submissions make up the complete DP0.2 campaign. For each of these, information on the number of quanta, the time-to-run, the max memory used per pipetask among other metrics are drawn from Butler and PanDA tables and are recorded along with a uniquely-identifying BPS processing timestamp ID, and links to the panDA workflow IDDS webpages.

8 DP0.2 wallclock estimates compared with DRP requirements

Over the course of 150 calendar days in 2021-2022, preliminary Data Release Production (DRP) processing ran in the iDF on about 20,000 exposures of simulated Rubin data, representing 10-20 full nights of images, covering 150 tracts (300 square degrees), to typically 5 year depth (50 returns to the same spot on the sky for each of 6 filters). This half-percent survey used 2.5M core-hours on a Google cloud compute cluster of approximately 4000 cores, each core with (normally) up to 14 GB/core RAM available (less typically used) with jobs taking typically a few seconds to a fraction of an hour per quanta. Small subsets (a few percent) of the processing pipetasks used nodes with up to 236 GB/core, and were allowed to run for up to 3 days.

The input images were approximately 75 TB in size, while the outputs, counting transient datasetTypes, expanded to 3.4 PB in an object store. After removing transient datasetTypes (e.g. warps), the final storage used by DP0.2 was approximately 2.5 PB. Of the 150 calendar days, approximately 60 days were running and the remainder used for debugging and development.

Thus, to keep up with expected DRP during the initial stages of the survey (6-12 days to process 10-20 nights of data), a factor of 5-10 speed up (or increased core counts or efficiency, or some combination of both) are estimated to be needed.

A References

[DESC], Collaboration, 2020, *The LSST DESC DC2 Simulated Sky Survey*, DESC, URL `https://arxiv.org/pdf/2010.05926.pdf`

[PanDA], PanDA, 2021, *Production and Distributed Analysis*, PanDA, URL `https://panda-wms.readthedocs.io/en/latest`

B Acronyms

Acronym	Description
B	Byte (8 bit)
BPS	Batch Production Service
CCD	Charge-Coupled Device
DC	Data Center
DESC	Dark Energy Science Collaboration
DM	Data Management
DP0	Data Preview 0
DRP	Data Release Production
GB	Gigabyte
LSST	Legacy Survey of Space and Time (formerly Large Synoptic Survey Telescope)

MB	MegaByte
OPS	Operations
PB	PetaByte
POSIX	Portable Operating System Interface
PanDA	Production ANd Distributed Analysis system
RAM	Random Access Memory
RTN	Rubin Technical Note
S3	(Amazon) Simple Storage Service
TB	TeraByte
bps	bit(s) per second