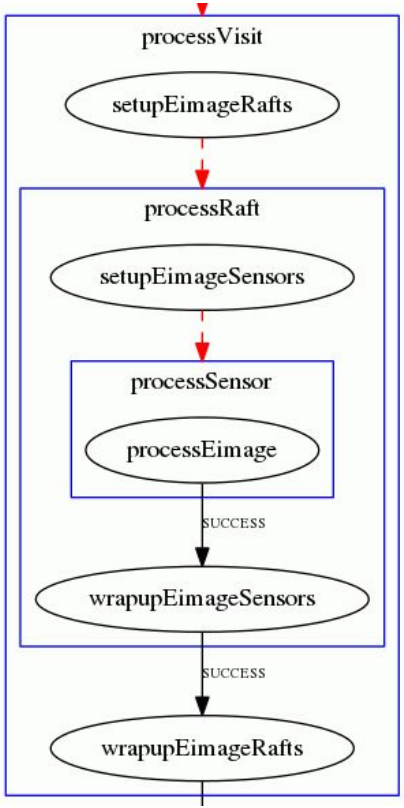
DESC Data Challenge 1 (DC1) DM Processing workflow

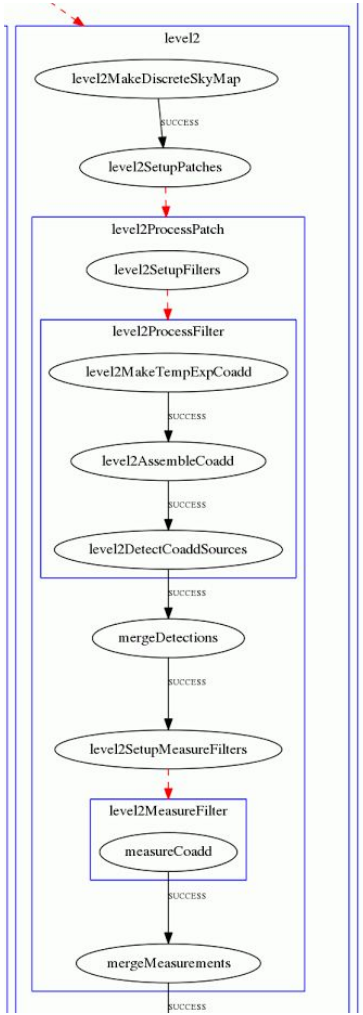
Tony Johnson (SLAC)

LSST All Hands -- August 2017

Background

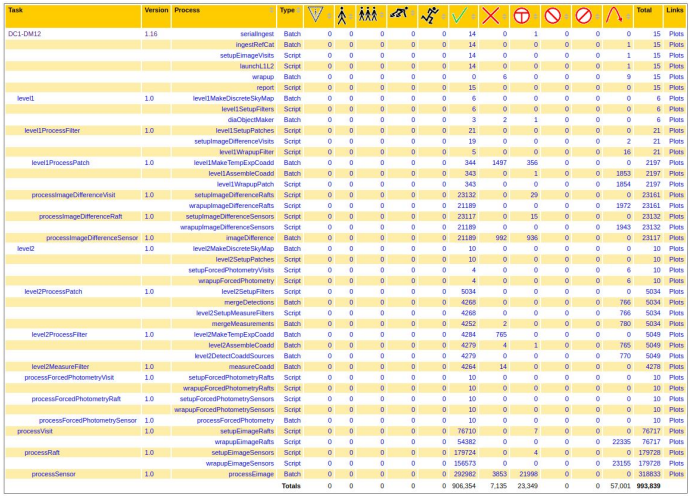
- DESC has developed a single workflow which can be used for Twinkles and DC1
  - Runs image differencing and/or full level 2 processing
  - This has been setup to run using SLAC workflow engine with jobs running at NERSC
- Run multiple times
    - Twinkles vs DC1
    - Imsim vs Phosim
    - Dithered vs Undithered
    - Bugs vs no-bugs (perhaps)
  - Total cost at NERSC: ~125k core/hours, 30k jobs
  - Elapsed time typically 2 weeks
- For twinkles we realize that what we really need is a workflow which simulates “year-by-year” processing
  - Work in progress



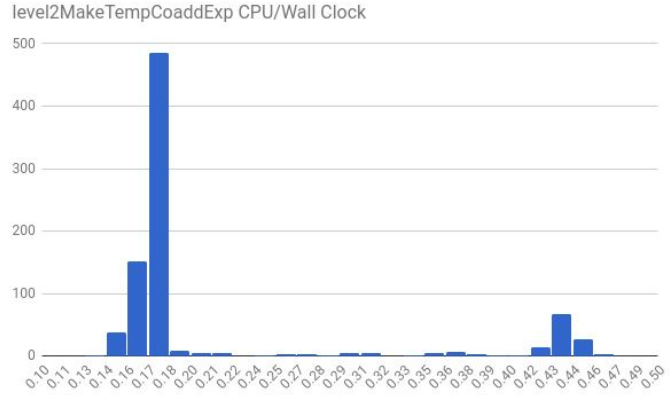


Workflow essential features

- Centralized workflow able to submit to multiple distributed inhomogeneous clusters
- Complete (permanent) history of what was done
  - Able to run multiple workflows at once (multi-tenant)
- Ability to rerun subsets of processing which failed
  - Or things which appeared to succeed
- Good diagnostics, in particular ability to summarize cpu/wallclock/memory usage etc
  - Broken down by process/stream/etc



Task	Version	Process	Type												Total	Links
DC1-DM12	1.16	serialIngest	Batch	0	0	0	0	0	14	0	1	0	0	0	15	Plots
		ingestRefCat	Batch	0	0	0	0	0	14	0	0	0	0	1	15	Plots
		setupEimageVisits	Script	0	0	0	0	0	14	0	0	0	0	0	15	Plots
		launchL1L2	Script	0	0	0	0	0	14	0	0	0	0	1	15	Plots
		wrapup	Batch	0	0	0	0	0	0	6	0	0	0	9	15	Plots
		report	Script	0	0	0	0	0	15	0	0	0	0	0	15	Plots
level1	1.0	level1MakeDiscreteSkyMap	Batch	0	0	0	0	0	6	0	0	0	0	0	6	Plots
		level1SetupFilters	Script	0	0	0	0	0	6	0	0	0	0	0	6	Plots
		diaObjectMaker	Batch	0	0	0	0	0	3	2	1	0	0	0	6	Plots
level1ProcessFilter	1.0	level1SetupPatches	Script	0	0	0	0	0	21	0	0	0	0	0	21	Plots
		setupImageDifferenceVisits	Script	0	0	0	0	0	19	0	0	0	0	2	21	Plots
		level1WrapupFilter	Script	0	0	0	0	0	5	0	0	0	0	16	21	Plots
level1ProcessPatch	1.0	level1MakeTempExpCoadd	Batch	0	0	0	0	0	344	1497	356	0	0	0	2197	Plots
		level1AssembleCoadd	Batch	0	0	0	0	0	343	0	1	0	0	1853	2197	Plots
		level1WrapupPatch	Script	0	0	0	0	0	343	0	0	0	0	1854	2197	Plots
processImageDifferenceVisit	1.0	setupImageDifferenceRafts	Script	0	0	0	0	0	23132	0	29	0	0	0	23161	Plots
		wrapupImageDifferenceRafts	Script	0	0	0	0	0	21189	0	0	0	0	1972	23161	Plots
processImageDifferenceRaft	1.0	setupImageDifferenceSensors	Script	0	0	0	0	0	23117	0	15	0	0	0	23132	Plots
		wrapupImageDifferenceSensors	Script	0	0	0	0	0	21189	0	0	0	0	1943	23132	Plots
processImageDifferenceSensor	1.0	imageDifference	Batch	0	0	0	0	0	21189	992	936	0	0	0	23117	Plots
level2	1.0	level2MakeDiscreteSkyMap	Batch	0	0	0	0	0	10	0	0	0	0	0	10	Plots
		level2SetupPatches	Script	0	0	0	0	0	10	0	0	0	0	0	10	Plots
		setupForcedPhotometryVisits	Script	0	0	0	0	0	4	0	0	0	0	6	10	Plots
		wrapupForcedPhotometry	Script	0	0	0	0	0	4	0	0	0	0	6	10	Plots
level2ProcessPatch	1.0	level2SetupFilters	Script	0	0	0	0	0	5034	0	0	0	0	0	5034	Plots
		mergeDetections	Batch	0	0	0	0	0	4268	0	0	0	0	766	5034	Plots
		level2SetupMeasureFilters	Script	0	0	0	0	0	4268	0	0	0	0	766	5034	Plots
		mergeMeasurements	Batch	0	0	0	0	0	4252	2	0	0	0	780	5034	Plots
level2ProcessFilter	1.0	level2MakeTempExpCoadd	Batch	0	0	0	0	0	4284	785	0	0	0	0	5049	Plots
		level2AssembleCoadd	Batch	0	0	0	0	0	4279	4	1	0	0	765	5049	Plots
		level2DetectCoaddSources	Batch	0	0	0	0	0	4279	0	0	0	0	770	5049	Plots
level2MeasureFilter	1.0	measureCoadd	Batch	0	0	0	0	0	4284	14	0	0	0	0	4278	Plots
processForcedPhotometryVisit	1.0	setupForcedPhotometryRafts	Script	0	0	0	0	0	10	0	0	0	0	0	10	Plots
		wrapupForcedPhotometryRafts	Script	0	0	0	0	0	10	0	0	0	0	0	10	Plots
processForcedPhotometryRaft	1.0	setupForcedPhotometrySensors	Script	0	0	0	0	0	10	0	0	0	0	0	10	Plots
		wrapupForcedPhotometrySensors	Script	0	0	0	0	0	10	0	0	0	0	0	10	Plots
processForcedPhotometrySensor	1.0	processForcedPhotometry	Batch	0	0	0	0	0	10	0	0	0	0	0	10	Plots
processVisit	1.0	setupEimageRafts	Script	0	0	0	0	0	76710	0	7	0	0	0	76717	Plots
		wrapupEimageRafts	Script	0	0	0	0	0	54382	0	0	0	0	22335	76717	Plots
processRaft	1.0	setupEimageSensors	Script	0	0	0	0	0	179724	0	4	0	0	0	179728	Plots
		wrapupEimageSensors	Script	0	0	0	0	0	156573	0	0	0	0	23155	179728	Plots
processSensor	1.0	processEimage	Batch	0	0	0	0	0	292982	3853	21998	0	0	0	318833	Plots
		Totals		0	0	0	0	0	906,354	7,135	23,349	0	0	57,001	**993,839**	



Problems and solutions

- Problem: Understanding how to optimally use DM
  - Solution: Use the excellent cookbook produced by Simon Krughoff
- Problem: Understanding how to parallelize workflow
  - Solution: Interrogate Simon about details of cookbook
  - Future: Make use of supertasks?
- Problem: Ingest phosim/imsim data
  - Official solution from cookbook insanely slow and non-parallelizable
  - Unofficial solution: Temporary DESC rewrite
- Problem: Obscure DM error messages
  - Solution: Ignore them and hope they were not important (some of them were)
  - Future: Need to incorporate meaningful diagnostics into workflow
- Problem: Getting DM expertise to help solve problems/bugs
  - Solution: Get help from Paul Price and Robert Kupton (priceless)
  - Future:
    - Need to develop closer relationship between DM and DESC
    - Spread DM expertise within DESC, including understanding future roadmap
    - Make DM more aware of desc specific issues (e.g. NERSC)

Problems and solutions (continued)

- Problem: NERSC (and similar super-computer centers)
  - Problem: Limited slots in serial queues
    - Solution: Use “pilot” like functionality to suck jobs into slurm jobs running on multiple complete nodes
  - Problem: Limited time limit in queues
    - Solution: Rerun jobs which run out of time
    - Future: Support for checkpointing (possible with DM?)
  - Problem: Limited IO capabilities, especially for meta-data heavy python
    - Solution: Install code into contrib area (limited benefit)
    - Solution: Use shifter/docker to encapsulate code into “in-memory” docker image
    - Future: Eliminate python for production running
  - Problem: Memory Usage
    - Some coadd jobs take > 20GB -- NERSC has 2GB (or less) per core
  - Problem: Shifter (memory usage and single point of failure)
    - Solution: Bug report filed, and transfer of knowledge in process
  - Problem: KNL
    - Solution: Try it
  - Problem: Downtime and long queue times
    - Solution: Suck it up
    - Future: Need mechanism to give DESC input to NERSC

Conclusions

- Getting DM processing to work at DC1 scale at NERSC was not trivial
  - Not as smooth (=automated) as I would like, but it does work
- We need to scale this up for DC2 (and DC3)
  - Ideally we would standardize on the same tools that DM is using
  - But the challenges of running at NERSC, and at large scale, may present unique challenges
    - We should work together to fix them