KIPAC KAVLI INSTITUTE FOR PARTICLE ASTROPHYSICS & COSMOLOGY

Campaign Management

Eric Charles
DMLTVF2F

Outline

- Scope of the work
  - Some terminology
  - What CM tools will look like eventually
    - Examples from Fermi-LAT data processing web interface
  - Aside, what they won't look like:
    - Basically, this is everything that Robert cares about
- How to get there from where we are
- Next steps

Scope of the Work

Some Terminology

- In DMTN-181 K-T applied several synonyms for “task” to different levels of data processing:
  - Task: as per DM, one quanta of Pipetask
  - Job: a bunch of tasks that get run together
  - Workflow: a bunch of job that are defined in a BPS workflow file
  - Campaign: a bunch of workflows that represent a complete processing of some set of data
  - Production: a bunch of related campaigns

Some More Terminology

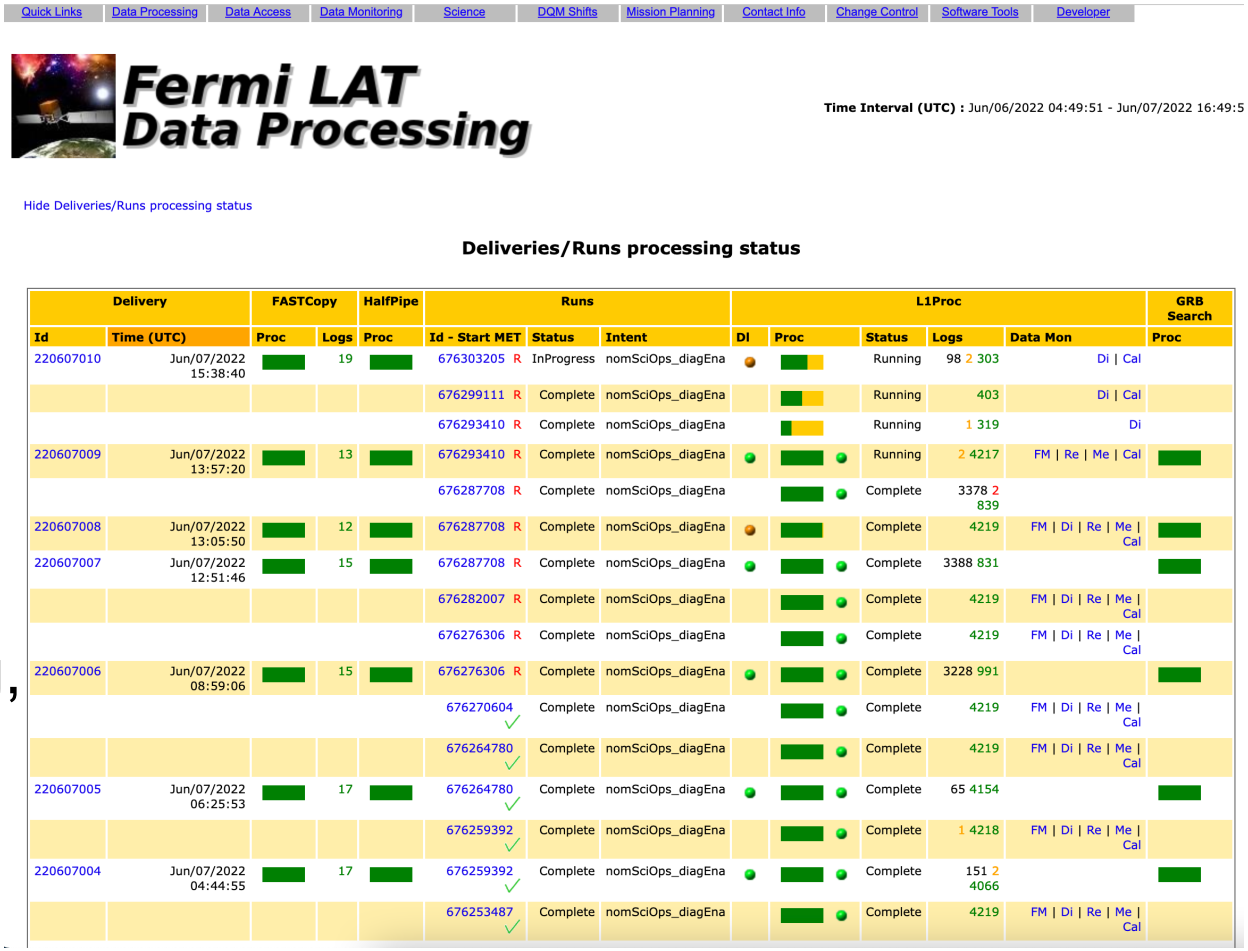
- In practice DP0.2 has played about a bit differently
  - Task: as per DM, one quanta of Pipetask
  - Job: a bunch of tasks that get run together
    - These are defined by the BPS clustering mechanism
  - Workflow: a bunch of jobs that are defined in a BPS workflow file
  - Group: a subset of data and one or more workflows that it takes to process them
  - Step: a part of a campaign
    - These are defined by the pipeline yaml files
  - Campaign: a bunch of workflows that represent a complete processing of some set of data
  - Production: a bunch of related campaigns
    - E.g., DP0.2, which includes DP0.2 testing, DP0.2 production
  - SomethingSuitablyEpic: all the campaigns of a particular type
    - E.g. DataProc, SimProc, CalibProc

Example end-user interface

This is the top-level view of the Fermi-LAT data processing

It provides links to:

Processing status,
Data quality monitoring,
Automated alarms



A table like this could be the “step to group” level view, showing which groups are currently being processed for a given step.

Automated Alarm Drill Down

Alarms for run 676223709

Mode	Type	Error	Warning	Undefined	Clean
Recon	Hist	0	1	0	3454

WARNING Status

Severity	Mode	Type	Variable Name	Algorithm	Value	Limits	Details
5	Recon	Hist	ReconAcd2PhaMipsCorrectedAngle_PMTB_Zoom_TH1_AcdTile_46	gauss_mean	0.884 +- 0.015	[0.7 | 0.9 | --- | 1.3 | 1.8]	View

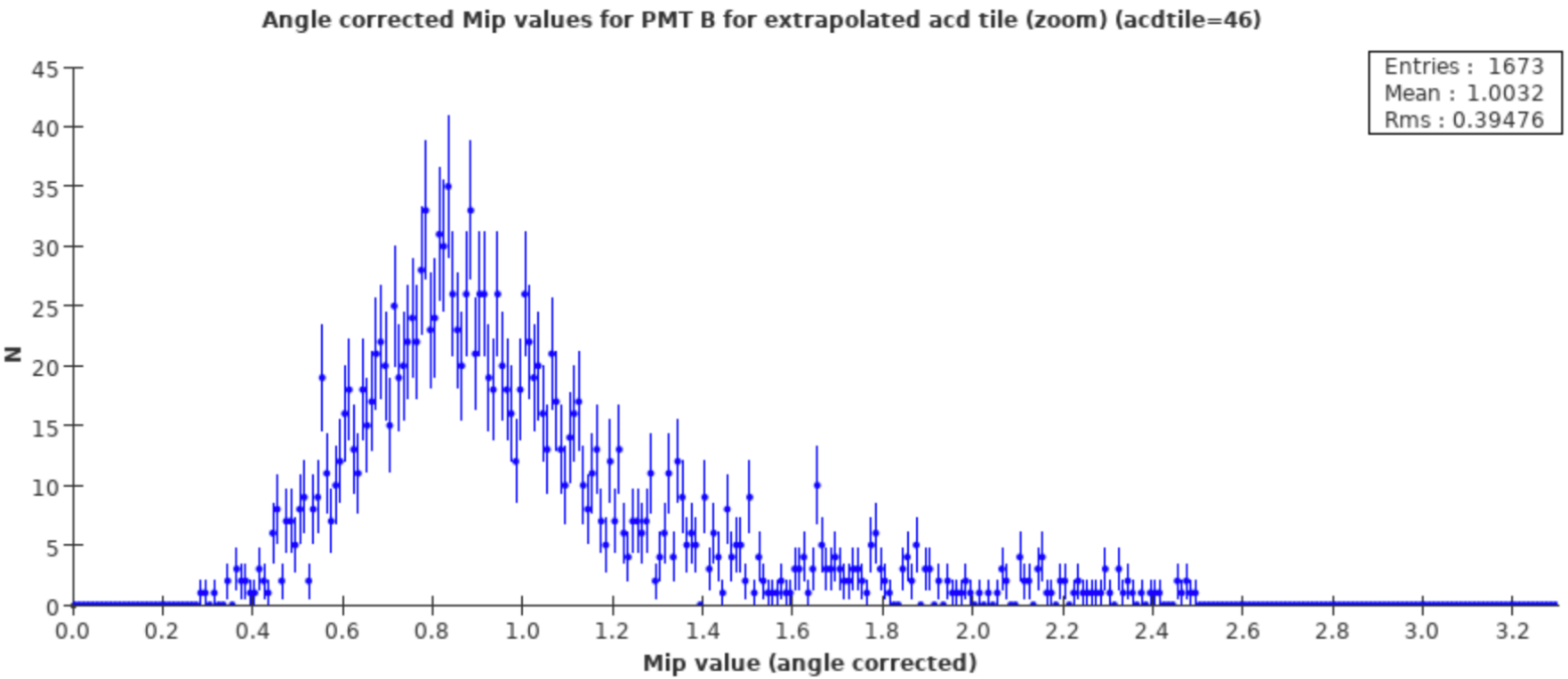
This is what is presented when the user clicks on a “group” where automated warnings had been generated.

In this case the warning is pretty minor, a derived quantity is just slightly outside the nominal ranges

Automated Alarm Associated Plot

Level **AcdTile**

Variable: **Recon_ReconAcd2PhaMipsCorrectedAngle_PMTB_Zoom_TH1_AcdTile**



Users can also click through to the plot underlying the derived quantity, showing that data for the channel generating the alarm are reasonable

Data quality monitoring

Quick Links | Data Processing | Data Access | Data Monitoring | Science | DQM Shifts | Mission Planning | Contact Info | Change Control | Software Tools | Developer

Fermi LAT Data Quality Monitoring

Time Interval (UTC) : Jun/06/2022 15:54:52.668 - Jun/06/2022 17:06:48.087
RunId : 676223709
Mode: Recon (Select All)
For the selected runs: **Intent:** nomSciOps_diagEna | **Moot Key:** 3053
Refresh Data

Version: 2.8-SNAPSHOT
User: echarles . (Switch|Logout) | Help | Jira
Mode: **Prod** | Dev | Test | ProdTest
Table | **Plots** | Alarms | Errors | Images
Selection | Bad Intervals | Data Info

Filter on|**off** Sort **Alph** **Expert**

Root
- **Shift Plots**
  - **Arrival Times**
  - **Event Rates to SSR**
  - **FastMonErrors**
  - **GEM-Filter Rates**
  - **GPS**
  - **Hardware Trigger**
  - **Navigation**
  - **Occupancies**
    - CAL Hit map (tower,layer)
    - CAL_HI map (tower)
    - CAL_LO map (tower)
    - CNO map (GARC)
    - Crate Number
    - Distribution of ACD Digis
    - Num. strips hit (GTFE)
    - ROI map (tower)
    - TKR Hit map (tower, plane)
    - TKR Trg map (tower)
    - Veto map (tile)
  - **Physics**
    - Clean Evts Rate
    - Source Evts Rate
    - Transient Evts Rate
    - UltraClean Evts Rate
  - **Recon**
  - **Solar Flares**

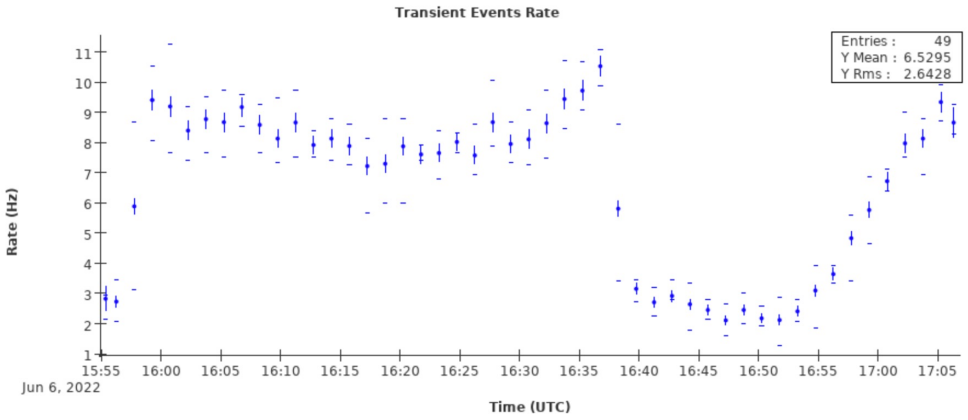
/Shift Plots/Occupancies

Show Selection Form | Parameters Selection: **Partial** | Select completely: acdtile | Hide Data Description

Level **LAT**

Variable: **Merit_Trend_Rate_TransientEvts**

Rate of events after application of Transient gamma cuts from Pass 8.
You can also check out the Solar Flare Report, the Space Weather Website and the Space Weather Plots.



Download: (*vector*) eps svg pdf swf ps , (*bitmap*) jpg png ppm gif
Cross Trend|View All Data Points

Y Axis limits: Min 0.966 Max 11.57
X Axis limits: Min Jun/06/2022 15:54:52.668 Max Jun/06/2022 17:06:48.087

This is not part of CM, but it is a view into the “shifter” plots available in Fermi DQM. These are what the data shifters use to mark data as good & ready for export.

Data processing monitoring

Details for Run 676264780

Latest Delivery 220607006

Run Summary

SCID	RunID	Intent	Type	Analysis	Status	Datagrams	Num of Events	Datagram Start	Datagram End	Event Begin	Event End	Moot Key
77	676264780	nomSciOps_diagEna	LPA		Complete	20136	2406381	2022-06-07 03:19:39.095695	2022-06-07 04:56:29.086459	2022-06-07 03:19:38.669676	2022-06-07 04:56:29.085426	3053

Run Quality

	ACQ Status	Status	Quality	Burst Advocate	L1Proc Status
✓	Complete	Reviewed	Good	Unset	Complete

Run Data Sets

First Processing

Create Date (UTC)	Name	Type	Format	No. Events	Filesize	Links	Up To Date
07-Jun-2022 11:11:21	r0676264780	RECON	root	2,406,381	32.6 GB	Data Cat|Dnld	✓
07-Jun-2022 10:55:28	r0676264780	TKRTRENDALARM	xml	0	3.3 kB	Data Cat|Dnld	✓
07-Jun-2022 10:54:58	r0676264780	TKRTREND	root	0	61.5 kB	Data Cat|Dnld	✓
07-Jun-2022 10:54:18	r0676264780	TKRREPORT	tar	0	600.0 kB	Data Cat|Dnld	✓
07-Jun-2022 10:54:18	r0676264780	TKRMONITOR	root	0	2.5 MB	Data Cat|Dnld	✓
07-Jun-2022 10:52:17	r0676264780	TKRANALYSIS	root	0	13.5 MB	Data Cat|Dnld	✓
07-Jun-2022 10:48:26	r0676264780	VERIFYERRORALARM	xml	0	6.9 kB	Data Cat|Dnld	✓
07-Jun-2022 10:48:06	r0676264780	VERIFYLOG	xml	0	216 B	Data Cat|Dnld	✓
07-Jun-2022 10:48:06	r0676264780	VERIFYHISTO	root	0	5.6 kB	Data Cat|Dnld	✓
07-Jun-2022 10:45:25	r0676264780	RECONHISTALARMDIST	root	0	7.4 kB	Data Cat|Dnld	✓
07-Jun-2022 10:44:35	r0676264780	RECONHISTALARM	xml	0	2.1 MB	Data Cat|Dnld	✓
07-Jun-2022 10:43:54	r0676264780	RECONHIST	root	0	9.4 MB	Data Cat|Dnld	✓

This is the drill down menu for data processing monitoring, it shows which data are available (and their location) from a particular “run”

For us “run” would be “workflow”

What CM is not!

- CM is not a tool to “analyze all the data from last night in some new way”
  - That is a question of setting up a yaml pipeline to chain together the right tasks and to figure out the right data queries to define the input data etc…
  - That is important, but _NOT_ what we are talking about here.
  - Here we are assuming that we already have more or less working pipeline yaml files, and some way to define the inputs that we want to run them on.
  - Once you’ve done that, then we can talk about if it makes sense to turn those pipelines in ”Campaigns” for CM

How to get there from where we are

How things are done now

- Task: defined by ***pipetask***
- Job: defined by ***BPS*** (clustering), executed by ***panda***
- Workflow: defined by a ***prodstatus*** executable / executed by ***panda***
- Group: managed in ***git*** using yaml files from ***prodstatus***
- Step: managed as ***JIRA*** issues
- Campaign: managed as an ”epic” ***JIRA*** issue
- Production: also managed as an “epic” ***JIRA*** issue (maybe?)

Proposal for changes

- Task: defined by ***pipetask***
- Job: defined by BPS (clustering), executed by ***panda***
- Workflow: defined by a ***prodstatus*** executable / executed by *panda*
- Group: defined / managed by ***prodstatus*** using yaml & sql DB
- Step: defined / managed by ***prodstatus*** using yaml & sql DB
- Campaign: defined / managed by ***prodstatus*** using yaml & sql DB
- Production: defined / managed by ***prodstatus*** using yaml & sql DB

Keep in mind: workflows are executed by panda, what we need from CM are tools to

generate workflows & prepare input collections
track workflow execution
provide book-keeping for layers above workflows
provide a nice user interface

Tagged collections for “group” definition

- Currently ***prodstatus*** scripts:
  - break each campaign Into “steps” (following the pipeline yaml)
  - and then divides each ”step” into groups (using data selection queries) and produces yaml files for each which are then tracked in git / JIRA

- Proposal:
  - break campaign into ”steps” as now
  - break data into subsets using “Tagged” collections and use collection naming as a bookkeeping tools
    - All the BPS workflows in a “step” are identical, except for input and output collection names
    - ”group” collects the one or more workflows needed to process all the data in that input ”Tagged” collection (allowing for rescuing crashed submissions)
    - Track execution using sql DB tables

CM Elements

- ***prodstatus*** package (https://github.com/lsst-dm/prodstatus)
  - CM scripts
- Production launching area
  - Local version of scripts and configuration files, where the prodstatus scripts are run from, this is editable.
- YAML archive
  - A location for saving “as run” versions of configurations files, file are copied here and made read-only.
- PANDA archive
  - A location for saving log & monitoring files from PANDA
- SQL DB
  - A simple database for bookkeeping
- Web interface package
  - Where we develop and keep the tools needed for the web-interface
- CM web server
  - User-facing web interface

Proposed *prodstatus* scripts

- create{XXX} (i.e., createWorkflow, createGroup, createStep, …}
  - Takes a combination of python and yaml as input
  - Generates relevant DB entries
  - Does relevant butler collection management tasks (i.e., butler associate, making chained collections)
- launchGroup
  - Launches current workflow of a particular ”group” in panda
  - Updates relevant DB entries
- checkWorkflowStatus
  - Checks on status of workflows in panda
  - Updated relevant DM entries
- check{XXX}Status {i.e., checkGroupStaus, checkStepStatus, …}
  - Does internal bookkeeping in SQL DB, updating which “Groups”, ”Steps”, etc.. are completed and updating links to data products

Proposed web hooks

- gen{XXX}Table (i.e., genStepTable, genCampaignTable, …)
  - Generate the high-level tables that are the outward facing interface
    - Could generate static HTML at first for testing by eventually should be dynamic
- validate{XXX} (i.e., validateWorkflow, validateGroup, validateStep…)
  - Marks data as good & ready to be used in subsequent processing
- invalidate{XXX} (i.e., validateWorkflow, validateGroup, validateStep…)
  - Marks data as bad & cleans up
- launchGroup
  - Launches processing for a particular group
- retryGroup
  - Creates a new workflow for a particular group and launches it

Next Steps

Next Steps

- Set up a working example on some test data
  - Work locally on SDF:
    - Butler repos, SQL DB, yaml archive, production area, web interface can all be local
    - Generate web interface as static html
- Port example to cloud:
  - Production area is local
  - Butler repos, SQL DB, yaml archive and web interface are remote

Splitting up the work

- To be discussed