CMS Workflows

Dr Christopher Jones
CCE IOS Meeting
8 April 2020

What Does CMS Call a Workflow?

A workflow

- Starts from a related set of input
- Applies a transformation to the input
- Create an output

We only use “workflow” to describe a multi-node processing

- Often it is also multi-site processing

Categories of Workflows

Production

- Done centrally on behalf of the collaboration

Analysis

- Done by individual or small groups of physicist

We will only discuss Production workflows

Production Workflow Groups

Group based on the starting point of the workflow

Data

- Processing of the ‘RAW’ data recorded from the detector

Monte Carlo

- Creating and processing of simulated collisions

Data Workflows

Input is the ‘RAW’ data recorded from the detector

Output consists of objects useful for physics analysis

Only 1 processing step: **Reconstruction *(RECO)***

- Calibrates/Aligns RAW data to create DIGIs
  - Corresponds to a physical response of the detector to
  - E.g. position at which a particle intersected a detector plane
- Applies pattern recognition algorithms to create physics objects
  - E.g. collection of intersection points used to calculate a particle’s trajectory

Monte Carlo Workflows

Processing of simulated collisions

Done using multiple steps

- **GEN**
  - Generates the underlying physics particles using theoretical models
- **SIM**
  - Simulates the detector response to the generated physics particles
  - This creates a RAW data format
- **RECO**
  - Apply same step as used by Data workflows

- Some steps may be combined within the same workflow
  - E.g. GEN and SIM may be run within the same application, or
  - E.g. GEN may be run and followed immediately on the same node by SIM

Characteristics of Data/Monte Carlo

Events

- Corresponds to an actual or simulated collision in the detector
- The smallest fundamental unit of physics
- Each event is statistically independent
  - Gives ‘perfect parallelization’

LuminosityBlock

- A set of consecutive Events
  - Data: 23 seconds of data taking
  - Monte Carlo: An arbitrary number of events to make processing easier
  - every Event is in only 1 LuminosityBlock
- Needed for doing accounting for physics analysis
- Fundamental unit of processing for the workflow system
  - A workflow job always processes an integral number of LuminosityBlocks
  - If a workflow job fails, all events in incomplete LuminosityBlocks will be dropped

Characteristics of Data/Monte Carlo (2)

Run

- Groups consecutive LuminosityBlocks
  - Every LuminosityBlock is in only 1 Run
  - Data: denotes a running time period of the detector (usually several hours)
  - Monte Carlo: Not used (assign all simulations to Run 1)
- The configuration of the detector is unchanging during a Run

DataSet

- Grouping of events by some criteria
  - Data: based on criteria used when deciding to record the event from the detector
    - multiple DataSets can share the same Run/LuminosityBlock and even Events
  - Monte Carlo: based on the configuration of the generator and simulated detector
    - No sharing of Events
- DataSets contain from 10s of thousands to 10s of billions of events (10GB to 1PB)

Files

- Files hold Events for a given LuminosityBlock
- LuminosityBlocks are never split across files
- A file belongs to only one DataSet

cmsRun Application

All production workflows run the cmsRun application

Characteristics of cmsRun

- Uses a python based configuration file
- Dynamically loads components
  - sources: read inputs from data files, databases, etc
    - what inputs to use (e.g. file names) are set in the configuration
  - producers: transform data
  - sinks: writes data to output
- Which processing step a cmsRun job is doing is dependent on what components were loaded
  - A job can have anywhere from 10s to 1000s of components loaded
- The application is multi-threaded
  - Number of threads to use are set in the configuration
  - Can process multiple Events concurrently
  - Can run multiple components for the same Event concurrently

Workflow Wrapper

Jobs run by the Workflow system use a Wrapper

Wrapper is responsible for handling interactions with the cmsRun application

- Handles setting up environment needed for cmsRun
- Handles reporting status of the job
- Deals with all outputs generated by the job

Workflow Setup

Decide on input

- Data: a DataSet is chosen
  - this forms a set of RAW data files to use
- Monte Carlo: a particular generator configuration and # events are chosen

Choose which CMS software release to use

- E.g. simulating 2018 data requires using the software used in 2018

Decide which Steps are part of the workflow

- E.g. GEN or GEN+SIM
- This sets which cmsRun configuration template is to be used
  - template has everything except names of input and output files

Decide if use a chain of Steps

- Within one job can run multiple cmsRun jobs in series
- E.g. 1st cmsRun creates GEN writing to local disk

  2nd cmsRun reads GEN and creates SIM file

Workflow Construction

Decide which computing sites will participate

- Usually dependent on which sites have files for the input data set

Decide how many LuminosityBlocks each job will process

- Based on
  - how many Events could be processed in 48 hours with 1 thread
  - average number of Events in a LuminosityBlock from the input

Set name of the DataSet to generate

- The result of every workflow is a new DataSet

Running a Workflow

Parameters describing workflow are uploaded to the system

cmsRun configurations for each job are generated

- Specify which input files to read and which LuminosityBlocks within those files to use
- Specify unique output file names for each job

For each cmsRun configuration a workflow task is created and registered

For each task, an additional job pilot is scheduled to the workflow system

- These pilots are scheduled to site batch systems

When a job pilot runs on a site

- It gets a specific task from the workflow system
- It executes the Wrapper which is assigned to the task
- Once finished the Wrapper
  - reports the status of the task to the workflow system
  - moves the output of the task to the site assigned for permanent storage
    - usually the permanent storage site IS just the site at which it ran

Finishing a Workflow

A task that failed is retried several times

- Often a failure is caused by a transient site problem

Once enough tasks have finished, file merge tasks are added to the system

- CMS would like all output files to be about 10GB in size since good for tape storage
- merge tasks take files with consecutive LuminosityBlocks and merge them
- only the merged files are kept

Once the vast majority of tasks succeed a workflow is marked as done

- The generated DataSet may then be used for other workflows or for analysis