Report from MC Production

Claire Gwenlan
Wolfgang Ehrenfeld
MC Production coordinators

ATLAS TIM
16.5.2013

ProdSys II Features

some PRODSYS II features:

- variable number of events per task optimised for running time, file size or site capability
- merge output at production site before consolidating at T1

- What are the boundary conditions from the MC work flows?

MC Workflows

- mc12 production steps
  - full simulation:
    evgen (5k) → simul (0.1k) → merge (1k) → digi+reco (0.5k) → merge (5k) → TAG (25k)
  - fast simulation:
    evgen (5k) → simul (1k) → digi+reco (0.5k) → merge (5k)
- optimisation:
  - many choices are historic
  - simul/digi+reco: CPU time for good turn around (~10h)
  - merging: file size for TAPE/transfer/bookkeeping
  - hardly any adjustment within one production step
    → one size fits all requests (does not work always)
- requirements:
  - store HITS on TAPE for possible reprocessing (about to drop this for AF-II simulation)
- (near) future production steps:
  - (fast) fast simulation: evgen (5k) → simul+digi+reco(0.2k) → merge (5k)

Variable Number of Events

- simulation:
  - simulation time per event constant to first order, some dependence on number of jets, not pre-tested per request
  - single particles can run much faster, but to not need too → electrons with high ET
  - significant overhead from G4/geometry setup
- digi+reco:
  - no pileup: constant time per event, runs quickly
  - fixed pileup: constant time per event, time per event scales worse than linear (my guess)
  - variable pileup (default mode): sampling of mu distribution done over 5000 events, in consecutive chunks → digi part needs currently fixed number of events per job in one task → running time from scouts are not representative
  - moderate overhead from initialisation
- merging:
  - HITS: running quickly, hardly any initialisation
  - AOD: running even faster (fast merge)

Variable Number of Events - Event Generation

- This is the most complex part of MC production!!!
  - fast LO generator ↔ multi-leg generators
  - generators without input ↔ with integration grids ↔ with events
  - no/medium/heavy filtering to populate interesting phase space
- running time per event is constant per task → ranges from a few seconds to a few hours
- initialisation time depends on generator and if integration grids are used → ranges from zero to half a day
- some use cases
  - LO generators, no filter or external events → no initialisation, a few seconds per event
  - LO generator with filtering → no initialisation, up to a few hours
  - multi-leg generators with integration files → moderate to long initialisation, moderate to long running time
- number of events per job from requester
  - boundary conditions are 5k events per job and maximum 1 day running time

Variable Number of Events - Some Remarks I

Good first estimate of number of events needed!

- event generation: default is 5k and can be overwritten in JO (specified in task definition)
  - jobs can run quick → optimise also on file size (can be true for any prod step)
- for the other steps rules are needed for either number of events or file size:
  - simul: 100 or 1000 depending on full or fast sim (will scale for single particles)
  - digi+reco:
    scale number of events by average mu value → who to do; how to discover average mu
    how to solve the mu sampling problem in the scouts
  - merging: file size per step (2GB for HITS and AOD?)
- specify number of events/file size in task definition
  - do not forget to improve on LIST definition parsing
- collect initialisation time and time per event from athena?
  → might be useful for better optimisation

Variable Number of Events - Some Remarks II

- a word on merging:
  - the merging transforms are decorated with some arguments and a release
    → use config tag for bookkeeping
  - easy evgen merging is a nice by-product
  - digi+reco (and all other steps) can write out more than one format, here AOD; it might be beneficial to also merge them (with a different transform and to a different file size) and move them to different locations (RDO,ESD mostly belong to groups)
  - merging will wait for sensible number of output files at site → limit number of sites for small tasks
  - direct T2 → T1 TAPE transfers (done for mc12 AF-II HITS)
  - no explicit merging step needed (besides TAPE archival)? → no transient datasets needed?
- a word on splitting:
  - number of events for HITS merging is related to number of events in digi+reco (same for evgen files as input and simulation)
    → will jobs using the same input go to the same site?

    example: single muons → 10k events per merged HITS → 20 jobs for one input file

Variable Number of Events - Some Remarks III

- filtering
  - in some rare cases (simulation) we are filtering, eg the number of output events is much smaller than the number of input events → this is not handled in the current system

- log files
  - how to handle log files for merging? merge with payload or separately?

Lost Files

- ProdSys II functionality for lost files recovery
- use cases
  - lost files after all production done → regenerate
  - lost files/failed registration during production, eg missing files for merging
    currently this problem is badly monitored, time intensive to understand and recover and not handled very well by current job definition

Reminder

- Do not forget about fair share between MC production, group production, user production!
- If nobody complains it does not mean that everything is working as expected. The grid might not be busy.
- There are many deadlines and challenges planned during LS1. ProdSys II testing and validation should fit into this.

Accounting - Long Term Monitoring I

- physics priority
  - requests are given a physics priority which is translated into prodsys priorities for the tasks involved
  - accounting on the number of events produced for PC is done by physics priority
  - proposal: decorate task with physics priority in task definition → store in ETASK/T_TASK_REQUEST → aggregate and displayed by historic view
  - can this be done now?
- fast and full sim task type
  - reporting of MC campaigns to PC is separated into full and fast sim
  - there will be a fast fast simulation in the (near) future
  - separation done by the simulation configuration tag (s or a) → not aggregated for historic view
  - proposal: decorate tasks with a simulation type (plain integer) in task definition → store in ETASK/T_TASK_REQUEST → aggregate and display by historic view
  - can this be done now?

Accounting - Long Term Monitoring II

- failure/error types (statistics)
  - What is the failure rate in simulation/reconstruction? Broken down by error type!
  - status: simple script to query t_task_request table
    → extend to query ejobexe for failed jobs to get error codes/error messages
  - when tested/used → give it to users
  - statistics
  - find and catalog reproducible errors → improve performance/job retries (needs athena/transform flagging)

Monitoring - Short Term Monitoring

- unified request identifier
  - users (group contacts/requesters) would like to have easy access to their requests → task tracking (some overlab with group production requirement)
  - proposal: decorate tasks from same request with a unique identifier and store is in ETASK/T_TASK_REQUEST table → panda monitoring and dashboard task monitoring can filter on ID
  - future: extend this concept with notifications (mails, user subscriptions, …)
- easier task/job controlling
  - add buttons to dashboard task/job monitoring for task finish, task abort, job kill
  - easier handling for small problems → scripts good for large processing
  - needs ACLs
  - provided for user tasks/jobs?
- improved task views
  - all monitoring is based on individual tasks - jobs are displayed per task
  - monitoring should display task chain

Request Handling Service

- service to handle production requests
  - model request preparation, approval and monitoring
  - similar request from / tool implemented in group production
- some requirements:
  - group contacts prepare request (MC JO or input datasets, number of events, priority, steps, config tags, …)
  - production team checks/modifies/approves/submit/restart/...
  - monitoring requests

Obsolete Tasks/Data

- MC production coordinators can obsolete tasks, including group production via a script
- use cases:
  - clean up of unmerged HITS → obsolete
  - free space on TAPE → obsolete merged AF-II HITS from mc11
  - buggy samples (evgen or simul or recon problems) → obsolete tasks and all other children
- the last use case touches group production → automatic notification
- procedure for discussion:
  - obsolete parent task via script
  - automatic aggregation of all children tasks
  - inform all affected users
  - delete datasets after X days

Input/Output Data Distribution

- evgen input distribution
  - current: stored in CERN-PROD_PHYS-GENER and replicated to five fixed T1s
  - proposal: store additional copy in another PHYS-GENER and replicate with restricted lifetime to five random T1s based on DSID
- output distribution for tasks with more than one payload
  - example: pile task with RDO, ESD, AOD
  - current: all into DATADISK, group contact replicate RDO, ESD into group space
  - limitation: only one payload output destination
  - proposal: define output destination per payload type
- input signal replication
  - task is run in input data cloud
  - problematic for digi+reco tasks with pileup samples in only a few clouds
  - can signal inputs be automatically replicated into pileup cloud?

Task Submission

- DeFT - we have tested DeFT for LIST submission
  - allows for LIST chaining
  - 50% failure rate (out of 2)
  - DeFT LISTs should be accessible via AFS for manual manipulation
  - DeFT should be able to get LIST location via URL
  - should go into production soon (remove manual intervention)
- command line interface status (should have been ready last year)
- closely follow up in ADC development meetings

Configuration Tag Definition

- the new transform framework started from scratch to overcome the problems and hacks from the old
  - available arguments can be queried/given in a file (avoid ListOfDefaultPositionalKeys)
- this is an opportunity to do the same with the tag definition interface and task definition script
  - many manual interventions are needed to the tag definition interface and task definition script to support a new transform
  - all the necessary information should be stored within the transform but Graeme needs to know what is important for prodsys
  - arguments: input/output data, plain arguments, python like argument (correct quoting), per job arguments (random numbers), per task arguments, …
  - more new transforms will come, some to overcome prodsys short commings: sim+digi+reco in one task (no need to store AF-II HITS if they do not go to TAPE)
- some old and outstanding request
  - add a comment string to the configuration tag
  - store and list the creator
  - transform information from CVMFS (instead of AFS)

Clone Interface

- request for a working task clone interface
  - sometimes it is easier to modify an existing task definition to create a new tasks than preparing a LIST definition file or a task from scratch
  - reminder: panda had/has this functionality but the group setting is broken → new clone interface
  - after many iterations with Valeri the new interface only works for event generation as output dataset name generation is non trivial

Summary

- ProdSys II
  - real life examples for variable number of events feature and merging
  - lost data regeneration
- request handling service
  - will help all production teams a lot but needs commitment from ADC
- all the rest
  - propose to follow up in biweekly ADC development meetings to discuss priorities, implementation details and progress