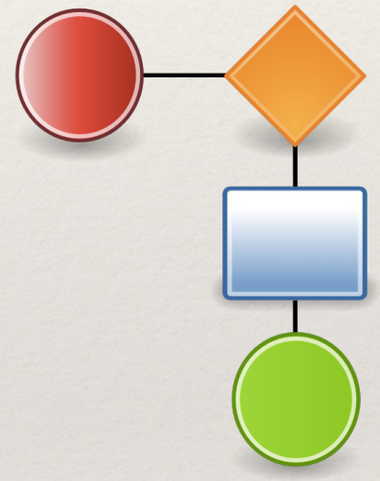
S. Zimmer (Geneva)

DmpWorkflow: production MC system



DAMPE-CNAF Meeting

May 11, 2016

Rationale

- probably ~2 experts running production MC (Gargano & Zimmer)
- hard limit at CNAF: 24 hrs / job
  - need O(10000s) of 'instances' to complete large statistics
- typical chain: generation, digitization & reconstruction
  - each step produces at least 1 ROOT file & 1 xml file
- would like to avoid having to shepherd each job individually -> use workflow system
- starting point: no existing scripts for LSF @ CNAF, so may as well develop our own…

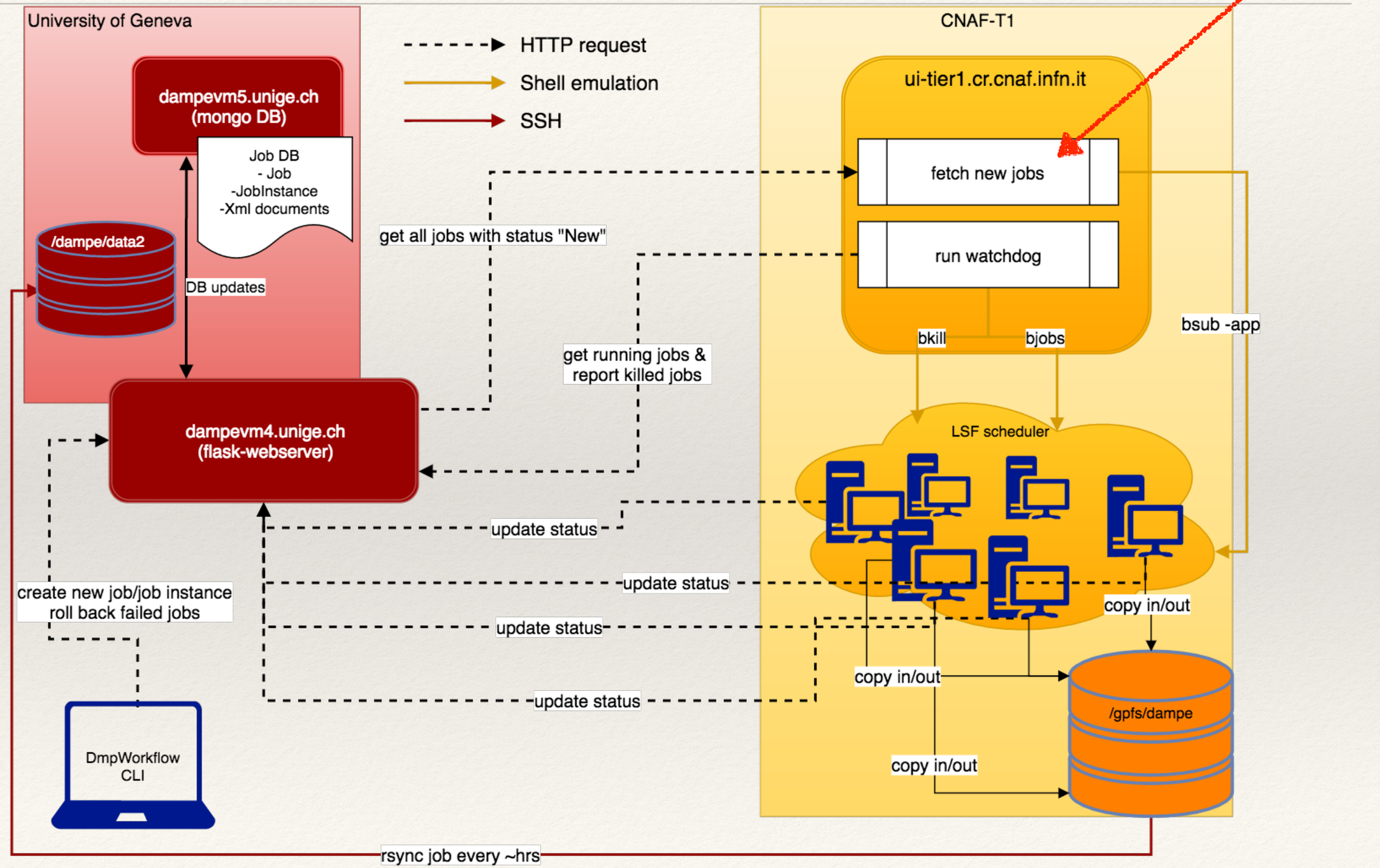
need/ want dedicated workflow management system

DmpWorkflow

- Web-framework based on flask
- DB: NoSQL MongoDB
- simple client-scripts for interacting with the web-server
- web-frontend for monitoring
- pure python code base
- https://github.com/zimmerst/DmpWorkflow

mongoDB

Implementation



Workflow Job Steps (executed in a batch worker)

- make scratch dir on worker node
- for each file in InputFiles, attempt to copy to scratch => “PreparingInputData”
- dump environment settings
- run payload => “ExecutingApplication”
- for each file in OutputFiles, attempt to copy to output directory => “PreparingOutputData”
- cleanup & report final status “Done, ApplicationComplete”

Current Status

DAMPE Workflow Management pipeline submission system

Admin Interface

Job Summary

Major Status

Name	Site	Type	Release Tag	#events	New	Submitted	Running	Suspended	Done	Failed	Terminated	Total
allProton-trunk-r3716_1GeV_100GeV	CNAF	Generation	build-r3716	0	3670	57	51	0	1222	0	0	5000

DAMPE Workflow Management pipeline submission system

Admin Interface

Details on Job

allProton-trunk-r3716_1GeV_100GeV

type: Generation **Release Tag:** build-r3716 **Number of Events on File:** 0 **Dependencies on other Jobs:** None

MetaData

name	value
DAMPE_COMMON	/storage/gpfs_ams/dampe/users/dampe_prod
DAMPE_LOGLEVEL	WARNING
TASKDIR_ROOT	/storage/gpfs_ams/dampe/users/dampe_prod/config/allProton-trunk-r3716_1GeV_100GeV
EMIN	1e3
EMAX	1e5
NEVENTS	10000
BATCH_OVERRIDE_CPUTIME	04:00

InputFiles

source	target	file_type
/storage/gpfs_ams/dampe/users/dampe_prod/config/allProton-trunk-r3716_1GeV_100GeV/allProton.mac	G4macro.mac	mac

dampevm4.unige.ch:5000

Job Wrapper

"Payload"

Executable: /opt/exp_software/dampe/externals/virtualEnvs/DAMPE/bin/python

Application Script:

```
#! /usr/bin/python
import DMPSW
import libDmpSimu as DmpSim
import os, sys
#-------------------------------------------------------------------
# IOSvc options
DMPSW.IOSvc.Set("OutData/FileName","mc.sim.root")
DMPSW.IOSvc.Set("OutData/Tag","Sim")

DMPSW.Core.Set("LogLevel",os.getenv("DAMPE_LOGLEVEL","INFO")) #@ Possible options are: NONE, ERROR, WARNING, INFO, DEBUG
DMPSW.Core.Set("EventNumber",os.getenv("NEVENTS","10000"))
SimAlg = DmpSim.DmpSimAlg() #@ Get instance of DAMPE simulation tool (interface to geant4)
SimAlg.Set("StoreSteps","false") # can turn on steps later on.

# set energy range
SimAlg.Set("EnergyMin",os.getenv("EMIN","1.e2"))
SimAlg.Set("EnergyMax",os.getenv("EMAX","1.e3"))

#@ Geant4 macro
SimAlg.Set("MacFile", "G4macro.mac")
seed = int(os.getenv("DWF_SIXDIGIT","1234"))
SimAlg.Set("RandomSeed","%i"%seed)
```

Instance Summary

Job Instances (5000 total)

id	batchId	created at	last sign of life	status	minor_status	host
000001	None	2016-05-09 00:10:19	2016-05-09 00:10:19	New	AwaitingBatchSubmission	None
000002	None	2016-05-09 00:10:19	2016-05-09 00:10:19	New	AwaitingBatchSubmission	None

not yet submitted

...

004804	34335010	2016-05-09 08:04:12	2016-05-09 11:00:28	Submitted	WaitingForExecution	None
004805	34335008	2016-05-09 08:04:12	2016-05-09 11:00:27	Submitted	WaitingForExecution	None
004806	34335007	2016-05-09 08:04:12	2016-05-09 11:00:26	Submitted	WaitingForExecution	None
004807	34335006	2016-05-09 08:04:12	2016-05-09 11:57:16	Done	ApplicationComplete	wn-200-13-21-02-a.cr.cnaf.infn.it
004808	34335005	2016-05-09 08:04:12	2016-05-09 11:54:54	Done	ApplicationComplete	wn-200-13-11-09-a.cr.cnaf.infn.it
004809	34335003	2016-05-09 08:04:12	2016-05-09 11:29:00	Running	ExecutingApplication	wn-206-03-27-01-a.cr.cnaf.infn.it
004810	34335000	2016-05-09 08:04:12	2016-05-09 11:26:25	Running	ExecutingApplication	wn-200-11-01-13-a.cr.cnaf.infn.it
004811	34334997	2016-05-09 08:04:12	2016-05-09 11:26:25	Running	ExecutingApplication	wn-200-10-31-11-a.cr.cnaf.infn.it
004812	34334995	2016-05-09 08:04:12	2016-05-09 11:25:57	Running	ExecutingApplication	wn-206-03-27-01-a.cr.cnaf.infn.it

pending

ExecutingApplication = Running Payload

Additional Information in DB


```
],
 "minor_status": "ApplicationComplete",
 "site": "CNAF",
 "status": "Done",
 "status_history": [
   {
     "status": "New",
     "minor_status": "AwaitingBatchSubmission",
     "update": ISODate("2016-05-09T08:04:24.124Z")
   },
   {
     "status": "Submitted",
     "minor_status": "WaitingForExecution",
     "update": ISODate("2016-05-09T08:16:31.343Z")
   },
   {
     "status": "Running",
     "minor_status": "PreparingInputData",
     "update": ISODate("2016-05-09T08:34:31.18Z")
   },
   {
     "status": "Running",
     "minor_status": "ExecutingApplication",
     "update": ISODate("2016-05-09T08:34:31.429Z")
   },
   {
     "status": "Running",
     "minor_status": "PreparingOutputData",
     "update": ISODate("2016-05-09T08:54:05.47Z")
   },
   {
     "status": "Done",
     "minor_status": "ApplicationComplete",
     "update": ISODate("2016-05-09T08:54:21.181Z")
   }
```


Retain full status history


```
 "cpu": [
   {
     "value": 0,
     "time": ISODate("2016-05-02T12:50:10.931Z")
   },
   {
     "value": 594,
     "time": ISODate("2016-05-02T13:05:45.178Z")
   },
   {
     "value": 726,
     "time": ISODate("2016-05-02T13:08:22.885Z")
   }
 ],
```

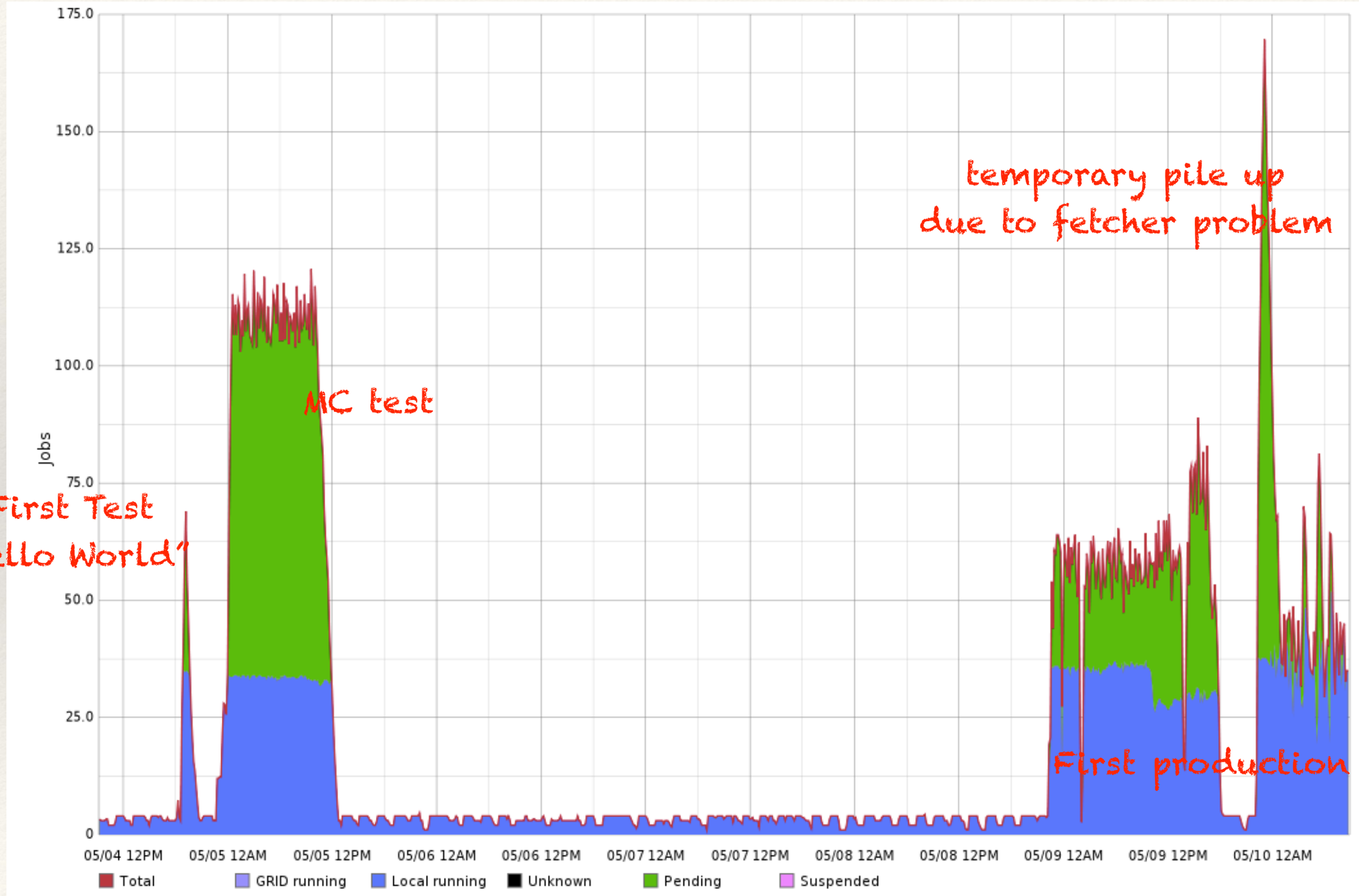


```
 "memory": [
   {
     "value": 0,
     "time": ISODate("2016-05-02T12:50:10.917Z")
   },
   {
     "value": 540.80859375,
     "time": ISODate("2016-05-02T13:05:45.170Z")
   },
   {
     "value": 540.92578125,
     "time": ISODate("2016-05-02T13:08:22.876Z")
   }
 ],
```


watchdog checks if memory/cpu is above threshold and triggers kill signal "bkill" -> trap failed jobs

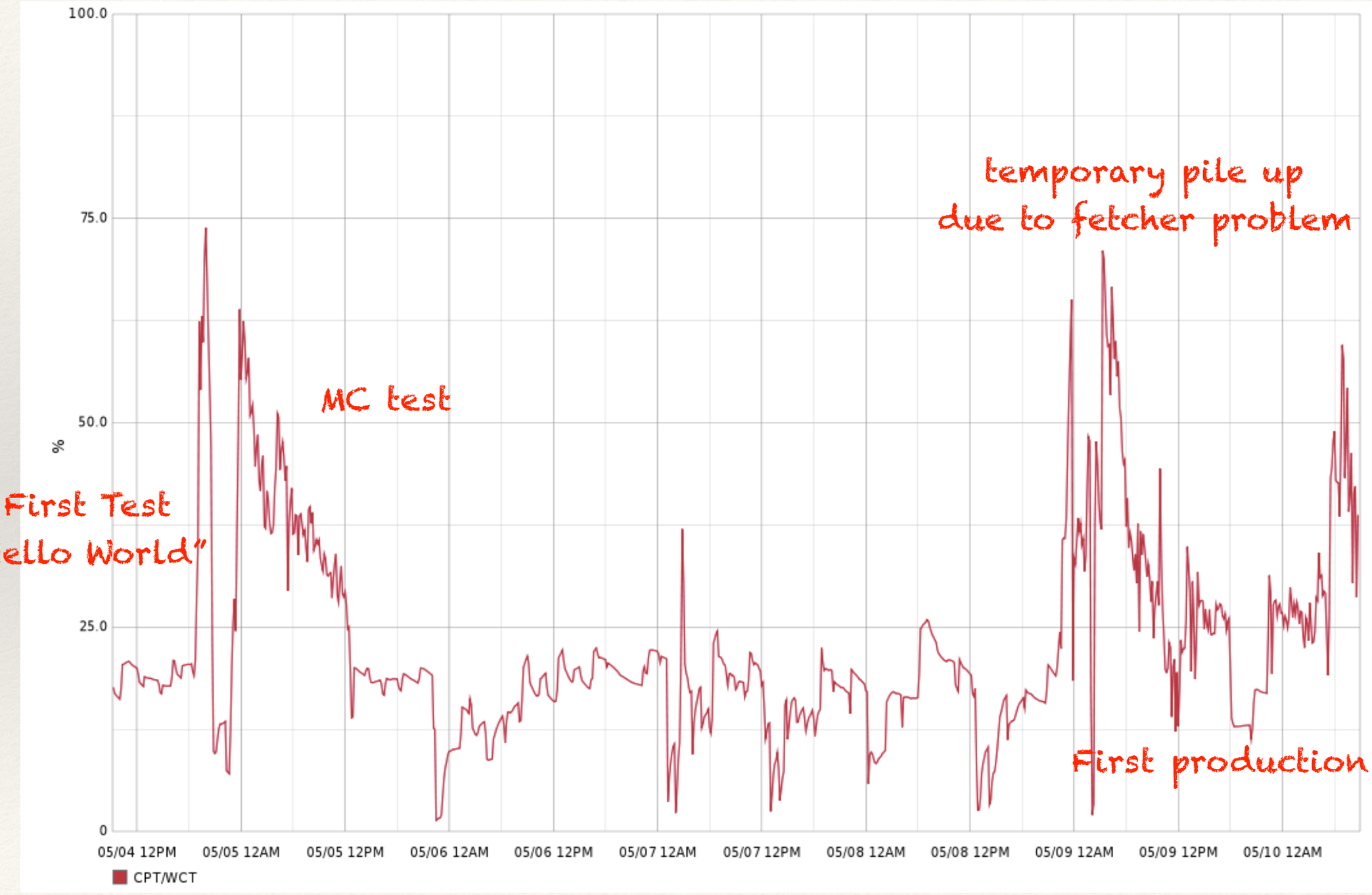
Usage Statistics from LSF

- running jobs -



Usage Statistics from LSF

- job efficiency -



Summary & Conclusions

- completed first tests at CNAF
  - prototypical setup in place, sort of works…
  - efficiency around ~25% -> jobs too short?
  - only 30 slots so far - would like to expand this
  - probably not using batch efficiently yet
- next steps:
  - further improvements to workflow system (bugs, access control, accounting tools)
  - deploy scripts inside cron job rather than as manual loop (dedicated UI?)
  - store data on XROOTd to scale up production & facilitate easy replication
  - stress-test of infrastructure in near future