



WMAgent

Seangchan Ryu (FNAL), Alan Rodrigues (Nebraska),
Dirk Hufnagel (FNAL), Jean-Roch Vlimant (Caltech)

CMS WM Review Part1

May 10th 2018

Overview

WM System: ReqMgr2, Global WorkQueue, WMStats, WMAgent, (Unified)

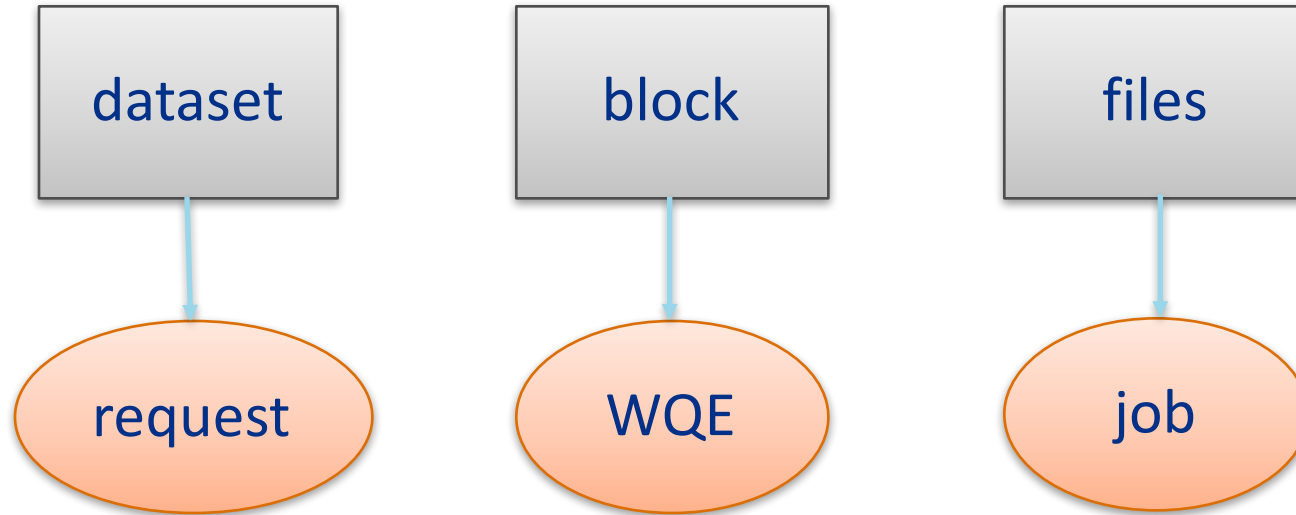
- Definitions (workflow, task, step, and job)
- System overview (interactions between components)
- Request/workflow life cycle – WM as state machine
- Job life cycle – WMAgent as state machine
- Handling of multiple requests and resources
- Tasks of each application/components
- Tier0 WMAgent
- Technology
- References

Definitions

Hierarchical structure

- CMS data: dataset, block, file;
- Workflow specification for WM: workflow, task, step;
- Work unit for WM: request/workflow, workqueue element (WQE), job;
- Job belongs to one workflow and one task. But contains multiple steps

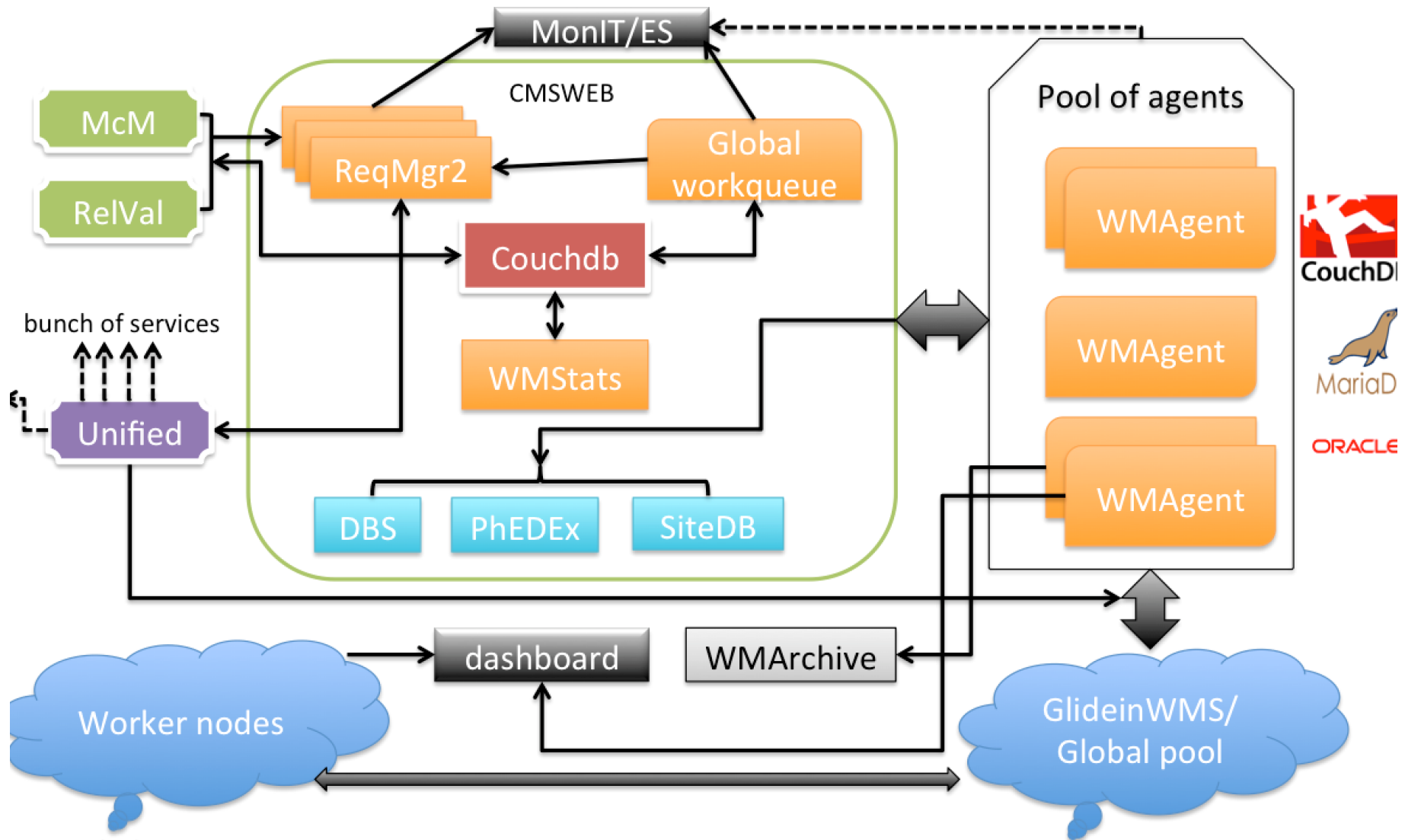
Data and Work Unit Relation



WM System Overview

- What does WM do?
 - WM System takes workflow spec as input and produce files, blocks, and datasets as output. Then it posts the data information to external services.
 - Different workflow can produce the same dataset (add files to the same dataset)
- Many functions and aspects are omitted in this talk.

Schematic of the Production System



Request

Creating Request (Web GUI)



ReqMgr

User: sryu

Home

Create

Approve

Assign

Batches

Create request interface

Workflow template: MonteCarloFromGEN

Table

JSON

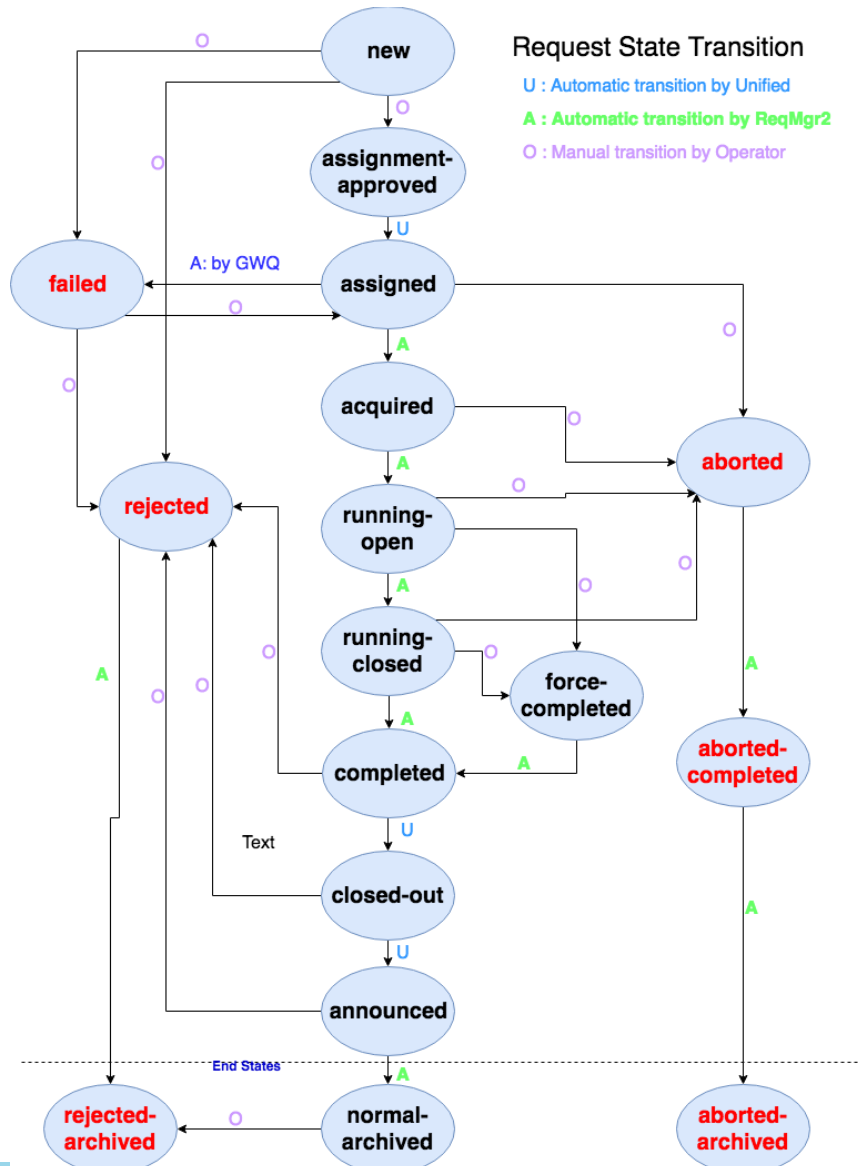
Please use the following table to fill out template form. If you need to extend it fill free to switch to JSON format and edit JSON directly.

Field	Value
InputDataset	REPLACE-InputDataset
Group	DATAOPS
DeleteFromSource	False
EventStreams	None
TimePerEvent (seconds)	12.0
Memory (MBytes)	2300.0
Comments	
RobustMerge	True
Requestor	REPLACE-Requestor
ProcessingString	REPLACE-ProcessingString
Multicore	1
SizePerEvent (KBytes)	512.0
ConfigCacheID	REPLACE-ConfigCacheID
DQMHarvestUnit	hvRun

Request

```
{  
  "AcquisitionEra": "RunIISummer15GS",  
  "CMSSWVersion": "CMSSW_7_1_25_patch2",  
  "Campaign": "Campaign-OVERRIDE-ME",  
  "Comments": "Automatic EventAware setting 507 EpJob, around 2LpJob",  
  "ConfigCacheID": "e86c9397d893f309853b83f87f674f6b",  
  "ConfigCacheUrl": "https://cmsweb.cern.ch/couchdb",  
  "CouchDBName": "reqmgr_config_cache",  
  "DQMUploadUrl": "https://cmsweb-testbed.cern.ch/dqm/dev",  
  "DbsUrl": "https://cmsweb-testbed.cern.ch/dbs/int/global/DBSReader/",  
  "GlobalTag": "MCRUN2_71_V1::All",  
  "Group": "DATAOPS",  
  "IncludeParents": false,  
  "InputDataset": "/H1H2Jet_M80_400To800_13TeV-Calchep/RunIWinter15pLHE-  
MCRUN2_71_V1-v1/LHE",  
  "Memory": 1200,  
  "PrepID": "TEST-EXO-RunIISummer15GS-08440",  
  "PrimaryDataset": "H1H2Jet_M80_400To800_13TeV-Calchep",  
  "ProcessingString": "MCRUN2_71_V1",  
  "RequestType": "MonteCarloFromGEN",  
}
```


Request State Transition

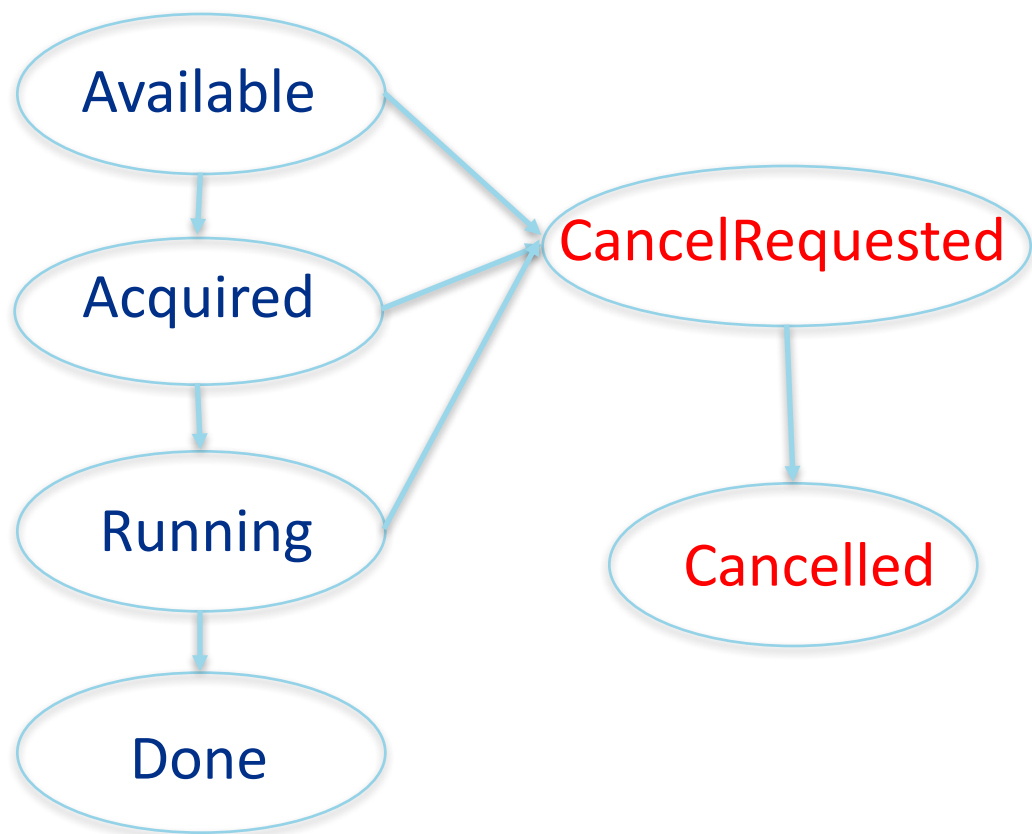


ReqMgr2 Handles most of the state transition through collecting Global WorkQueue (WQE status), and WMStats (Collective Job states).

Unified handles automation of some transitions (assigned, close-out, announced) which were manual CompOps task.

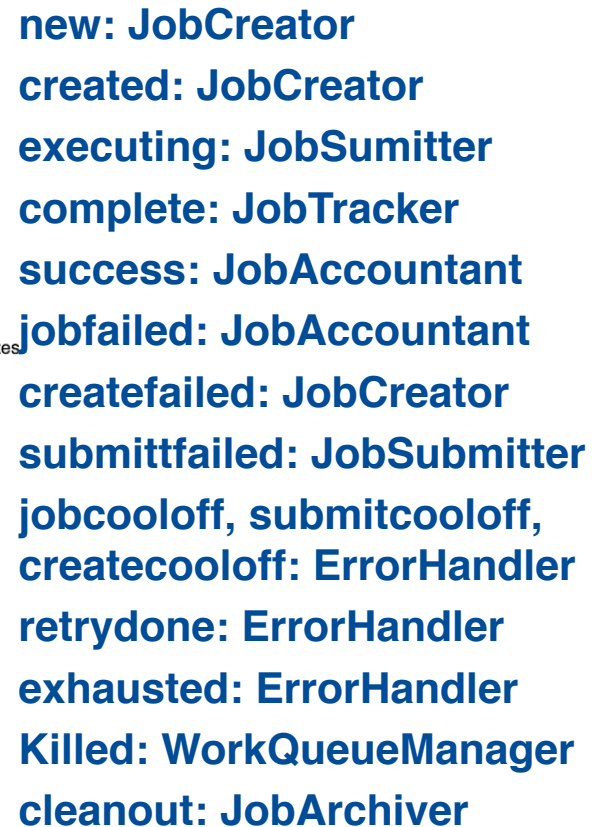
CompOps can also make the state transition for rejecting, aborting, and force-completing requests.

WorkQueue Element State Transition



Job state information is propagated to WQE. Collective WQEs' states are used for determining the request states.

Job State Transition (WMBS)



Multiple Request Handling

- Resource Control
 - Retrieve site threshold information from SSB;
 - Set the site threshold for each agent by job type;
 - Track current running and pending jobs;
 - Submit jobs if there are available sites.
- Priority Queue:
 - Organize works by priority per site;
 - Some higher priority jobs can go through even if site limit is reached.

Other Functions

- Monitoring
- Book keeping (Accounting)
- Update to external services (PhEDEx, DBS, Dashboard, WMArchive, MonIT)
- Cleaning up

ReqMgr2

- Provide the tools for request creation and management(REST API, Web GUI);
- Validate request parameters;
- Manage a request life cycle (tools to update status, priority and other parameters);
- Store the request data in DB (couch db backend);
- Provide search tools on request data (REST API, Web GUI).
- Misc:
 - ACDC - stores input file information for failed jobs;
 - Workload Summary – stores the request summary information with performance matrix.

- Home
- Create
- Approve
- Assign
- Batches

Assign interface

Settings

Requests

[Search](#)

[Search](#)

Sort by: [Date](#) [Status](#) [User](#) [Group](#)

Sort by: [Pattern](#) [Campaign](#)

☐ Select all requests

Assign

Reject

<input type="checkbox"/> Request: pdmvserv task HIG-RunIIIFall17DRPremix-00840_v1 T 180307 174741 1815 Status: assignment-approved Date: <input type="text" value="2018-03-07 15:47:41 GMT"/> User: pdmvserv Group: ppd	Splitting
<input type="checkbox"/> Request: pdmvserv task HIG-RunIIIFall17MiniAODv2-01241_v1 T 180414 220601 1394 Status: assignment-approved Date: <input type="text" value="2018-04-14 19:06:01 GMT"/> User: pdmvserv Group: ppd	Splitting
<input type="checkbox"/> Request: pdmvserv task TOP-RunIIIFall17DRPremix-00119_v1 T 180307 174947 9370 Status: assignment-approved Date: <input type="text" value="2018-03-07 15:49:47 GMT"/> User: pdmvserv Group: ppd	Splitting
<input type="checkbox"/> Request: pdmvserv task TOP-RunIIIFall17DRPremix-00121_v1 T 180316 192545 7574 Status: assignment-approved Date: <input type="text" value="2018-03-16 17:25:45 GMT"/> User: pdmvserv Group: ppd	Splitting
<input type="checkbox"/> Request: pdmvserv task HIG-RunIIIFall17DRPremix-00854_v1 T 180307 183104 8241 Status: assignment-approved Date: <input type="text" value="2018-03-07 16:31:04 GMT"/> User: pdmvserv Group: ppd	Splitting

Home

Create

Approve

Assign

Batches

Request pdmvserv_task_TOP-PhaseIISummer17wmLHEGENOnly-00092__v1_T_180510_063921_6474

Table

JSON

Splitting

Config

Comment

Status: **running-open**

✓ running-open
running-closed
force-complete
aborted

Submit

Field	Value
RequestPriority	<div>110000</div>

Config Cache List

- [Task1: TOP-PhaseIISummer17wmLHEGENOnly-00092_0: ConfigCacheID: 935d8b1700b4ffd0928f55c60c2926a6](#)

State Transition

- new:** 2018-05-10 04:39:21 UTC
- assignment-approved:** 2018-05-10 04:40:31 UTC
- assigned:** 2018-05-10 04:59:57 UTC
- acquired:** 2018-05-10 05:16:29 UTC

Unified

- Originally CompOps' task;
- Distribute data (Primary and Secondary input);
- Assign requests based on the data distribution;
- Validate the processing completion. (DBS, PhEDEx) – closed-out;
- Trigger the recovery (automatically create recovery workflow);
- Announce requests;
- Release the lock for DDM when data is transferred to tape storage;
- Distribute data output dataset.

WorkQueue

- Retrieve work (assigned request) from ReqMgr2;
- Split them in small pieces (WorkQueue Elements);
 - Dataset
 - Block
 - MC
 - Resubmit (ACDC)
- Calculate estimated jobs for each WQE used for matching the resources;
- Update locations of the WQEs periodically;
- Delete the WQEs when request is completed;
- Communicate to Local WorkQueue using couch replication.

WMStats

- Monitor current status of requests and collective job summary;
- Work as a debugging tool by providing a hierarchical view on requests;
- Provide search tool for finding specific requests (REST API, Web GUI);
- Display error, warning information about requests and infrastructure (WM Applications/WMAgent components).

filter

campaign:

workflow:

type:

status:

input dataset:

output dataset:

site whitelist:

team:

agent:

filtered stats

requests: 3443 total events: 1084.7 G processed events: 6.8 G created: 6.7 M cooloff: 6593 success: 5.6 M failure: 176803 queued: 384487 running: 99789 pending: 437823

Category

» Requests

» Jobs

all requests

progress

number of jobs

Category: all

Show 25 entries

Show 25 entries							Previous		1	2	3	4	5	...	138	Next
D	L	workflow	status	type	priority	queue injection	job progress	event progress	lumi progress	failure rate	Estimated Completion	cool off				
		pdmvserv_task_EXO-RunIIFall17MiniAODv2-00308_v1_T_180510_084153_40	new	TaskChain	85000	0.0%	0.0%	0.0%	0.0%	0.0%	N/A	0				
		pdmvserv_task_TOP-PhaseIISummer17wmLHEGENOnly-00092_v1_T_180510_063921_6474	running-open	TaskChain	110000	46.5%	0.0%	0.0%	0.0%	100.0%	72228 h 7 m	0				
		pdmvserv_task_TOP-PhaseIISummer17wmLHEGENOnly-00093_v1_T_180510_063910_5648	running-open	TaskChain	110000	16.5%	1.9%	0.0%	0.0%	75.9%	1748 h 47 m	0				
		pdmvserv_task_SMP-PhaseIISummer17wmLHEGENOnly-00049_v1_T_180510_063808_5724	running-open	TaskChain	110000	65.8%	0.2%	0.0%	0.0%	43.8%	4628 h 6 m	0				
		pdmvserv_task_B2G-PhaseIITDRFall17wmLHEGS-00056_v1_T_180510_043148_3566	new	TaskChain	90000	0.0%	0.0%	0.0%	0.0%	0.0%	N/A	0				
		pdmvserv_task_B2G-PhaseIITDRFall17wmLHEGS-00064_v1_T_180510_043141_4018	new	TaskChain	90000	0.0%	0.0%	0.0%	0.0%	0.0%	N/A	0				
		pdmvserv_task_B2G-PhaseIITDRFall17wmLHEGS-00061_v1_T_180510_043135_4529	new	TaskChain	90000	0.0%	0.0%	0.0%	0.0%	0.0%	N/A	0				
		pdmvserv_task_B2G-														

	TOP-PhaseIISummer17wmLHEGENOnly-00093_0	jobfailed	T2_US_Vanderbilt	8001	22	CMSEException,FatalException,ReportManipulatingError,WMAgentStepExecutionError	
--	---	-----------	------------------	------	----	--	---

Showing 1 to 9 of 9 entries

Previous

1

Next

1 2 3 4 5 6 7 8 9 10

- **Job Name:** fa140913-67a7-49a9-97ca-e69ccd2bc37e-816
- **WMBS job id:** 256817
- **Workflow:** pdmvserv_task_TOP-PhaseIISummer17wmLHEGENOnly-00093__v1_T_180510_063910_5648
- **Task:** /pdmvserv_task_TOP-PhaseIISummer17wmLHEGENOnly-00093__v1_T_180510_063910_5648/TOP-PhaseIISummer17wmLHEGENOnly-00093_0
- **Status:** jobfailed
- **Input dataset:**
- **Site:** T2_CH_CSCS_HPC
- **Agent:** vocms0253.cern.ch
- **ACDC URL:** https://cmsweb.cern.ch/couchdb/acdcserver
- **► State Transition**
- **Exit code:** 8001
- **Retry count:** 0
- **Errors:**
 - logArch1
 - cmsRun1
 - **Fatal Exception (Exit Code: 8001)**
 - An exception of category 'ExternalLHEProducer' occurred while


```
[0] Processing global begin Run run: 1
[1] Calling method for module ExternalLHEProducer/'externalLHEProducer'
```

 Exception Message:
Child failed with exit code 1.
 - **CMSEException (Exit Code: 8001)**
 - Exit 8001: CMSEException Exception from cmsRun


```
Adding last 25 lines of CMSSW stderr:
WARNING: In non-interactive mode release checks e.g. deprecated releases, production architectures are disabled.
WARNING: There already exists /srv/job/WMTaskSpace/cmsRun1/CMSSW_9_3_9_patch1 area for SCRAM_ARCH slc6_amd64_gcc630.

Adding last 25 lines of CMSSW stdout:
1 GenXSecAnalyzer      PostGlobalBeginR PostGlobalBeginR
```

WMAgent (Overview)

- Contains many components;
- Components can be classified (roughly) as follows:
 - Make job state transition
 - Do other work (Cleaning, Monitoring, Archiving, Reporting to external service)
- Communication among components is indirectly made through job state;.
- Durable: Recovers itself from fault (not always).

WorkQueueManager

- Pull work from the [Global WorkQueue](#) into the local workqueue;
- Perform the workflow bootstrap inside the agent (sandbox and pileup list creation);
- Inject elements of work from the local queue into database;
- Update input data locations (in case data location has changed after work is pulled);
- Clean up the WQEs in the local queue when workflow is finished (In case there is leftover);
- Kill requests if necessary.

JobCreator

- Create jobs:
 - The heart and brain of JobCreator is job splitting. Different job splitting algorithms are applied to create jobs;
 - The goal is creating optimal job size (~8h job).
- Job splitting:
 - Determine input files, lumis, events;
 - FileBased, EventBased, LumiBased, EventAwareLumibased,
- Created jobs are pickled and stored in designated location for the submission.

JobSubmitter

- Load job objects from database;
- Retrieve pickled job files from disk;
- Cache this information in main memory (Cache refresh condition is configurable);
- Retrieve resource thresholds from the database and calculate available site by counting current running jobs for each site;
- Create the batch job spec file;
- Submit jobs by priorities to a batch system (condor) until the resources are fully utilized or the available jobs are exhausted.

JobStatusLite

- Track jobs from condor (or other batch system);
- Update condor job state to database (sub state of executing):
 - Pending, Running, Completed, Error
- Kill stuck and “Error” state jobs.

JobTracker

- Update database with job report file location;
- Check BossAir db (condor job status DB) and update the jobs to “completed” state;
- Jobs failed by JobStatusLight move to “jobfailed” status directly.

JobUpdater

- Handle priority requests in WMAgent;
- Periodically compare priorities of active requests;
- Change the priority of jobs in local workqueue, database and condor. This will make priority changes effective for already created jobs.

JobAccountant

- Fetch all jobs in "complete" state from database;
- Parse the job report file then put output modules, output files and metadata information into database;
- Move the job state to success or jobfailed;
- Set the file parentage in database.

ErrorHandler

- Check the retry counter and max retry setting;
- Move the job status to cooloff or exhausted status;
- Gather information (i.e. input files) for exhausted jobs and upload to ACDC database for retrying.

ACDC is the database in cmsweb. It contains input file information on failed jobs (equivalent to DBS).

RetryManager

- Apply configurable retry logic for “cooloff” jobs to determine retry condition;
- Use plugin architecture for different retry logic (default is SquaredAlgo - each retry increases the time by squares of retry times);
- Move job state to "created" when condition is met.

JobArchiver

- Clean job cache area (job report and log files);
- Archive files in local archive area (never deleted until Agent redeployed);
- Move job state to "cleanout" state.

TaskArchiver

- Track finished state for tasks, requests;
- Report completed state to Global WorkQueue (request manager to update request status);
- Archive request summary to workload summary database;
- Clean up agent database for already archived requests;
- Clean up disk space for already archived requests.

AnalyticsDataCollector

- Collect job summary information by task and workflow;
- Report to WMStats:
 - How many jobs in each state per sites per task/requests;
 - Aggregate performance matrix;
 - Data from different WMAgents is aggregated in WMStats.

AgentStatusWatcher

- Report Agent status to WMStats:
 - Failed components/threads;
 - Proxy expiration report;
 - Disk usage warning;
- Control agent draining (clean up procedure for agent redeployment).

DBS3Upload

- Resolve file parentage;
- Create blocks;
- Close blocks (using configurable conditions);
- Upload blocks with files (attaching additional information – dataset, file detail information).

PhEDExInjector

- Inject files to PhEDEx;
- Close blocks in PhEDEx;
- Make PhEDEx subscriptions;
- Delete blocks (for copy+delete subscriptions – T0);
- Mark files in the db when injected.

ArchiveDataReporter

- Extract information from a job report (performance, error info, site info, campaign, etc);
- Upload collected information to WMArchive.

WMAgent (DB backend)

- Relational database (Oracle, MariaDB)
 - WMBS (file, job, site, ect)
 - Resource Control (site threshod)
 - BossAir (batch system (condor) job status)
 - DBSBuffer (file, block, dataset record for DBS and PhEDEx)
- NoSQL database (CouchDB)
 - wmagent_jobdump/fwjrs - frame work job report information
 - wmagent_jobdump/jobs - job status information inserted/updated by agent
 - stat_summary - summary snap shot of wmagent_jobdump/fwjrs
 - wmagent_summary - summary information from wmagent_jobdump/jobs and stat_summary inserted/updated by agent (replicated to wmstats)

Tier0 Workload Management

Differences: Only one agent

- No ReqMgr, WorkQueue, Unified: Only WMAgent;
- Tier0 specific component: *Tier0Feeder*:
 - Detects new CMS data from StorageManager bookkeeping database;
 - Based on meta-data from StorageManager/HLT and a Tier0 configuration (read from file) injects data into Tier0 WMAgent instance and creates workflows to process it;
- Different request life cycle (AnalyticsDataCollector);

Common functionality:

- Actual workflow processing happens like in a WMAgent;
- Data publication in DBS/PhEDEx like in a WMAgent.

Tier0 Agent (WMAgent variation)

Certain WMAgent features are only used in the Tier0

- Difference due to interaction to ReqMgr2, WorkQueue;
- Difference due to data processing:
 - Workflow definitions (Repack, Express, PromptReco);
 - Job types used exclusively in Tier0 workflows;
 - Job splitting algorithms used exclusively in Tier0 workflows.
- Difference due to in the Tier0 policy reasons:
 - Paused job retry logic (jobs aren't allowed to auto-fail);
 - Automatic deletion of output data after all transfers are done.

Tier0 WMStats

- Request Monitoring tool for Tier0;
- Same code base (Less features);
- No information from ReqMgr2 and WorkQueue;
- Run number information;
- Different request life cycle.

Technology

- Python 2.7 → Python 3 transition plan;
- CherryPy → flask, go;
- CouchDB 1.6.1 – latest 2.1.1;
- Oracle/MariaDB;
- JavaScript (jQuery, Yui, Protovis) – not using modern framework;
- Scalability and limitations (Alan's section).

References

- Request state definition:
<https://github.com/dmwm/WMCore/wiki/Request-Status>
- ReqMgr2: <https://github.com/dmwm/WMCore/wiki/Request-Manager>
- WorkQueue:
<https://github.com/dmwm/WMCore/wiki/WMAgent>
- WMStats: <https://github.com/dmwm/WMCore/wiki/WMStats>
- WMAgent: <https://github.com/dmwm/WMCore/wiki/WMAgent>
- Unifiled:
<https://indico.cern.ch/event/503133/contributions/2012968/>
<https://indico.cern.ch/event/505613/contributions/2230726/>