# Belle II use case

Hideki Miyake (KEK)


2017 May 29th, DIRAC users' workshop@Warsaw

# Belle2 DIRAC system: overview

KEK
- Production
  - 6 DIRAC node
  - 4 DB node
    - 2 SSD + 2 HDD
  - 1 Web portal (DIRAC slave)

Slaves (mainly for SiteDirector)

PNNL, Nagoya, Napoli, UVic

DDM
Condor
HPC

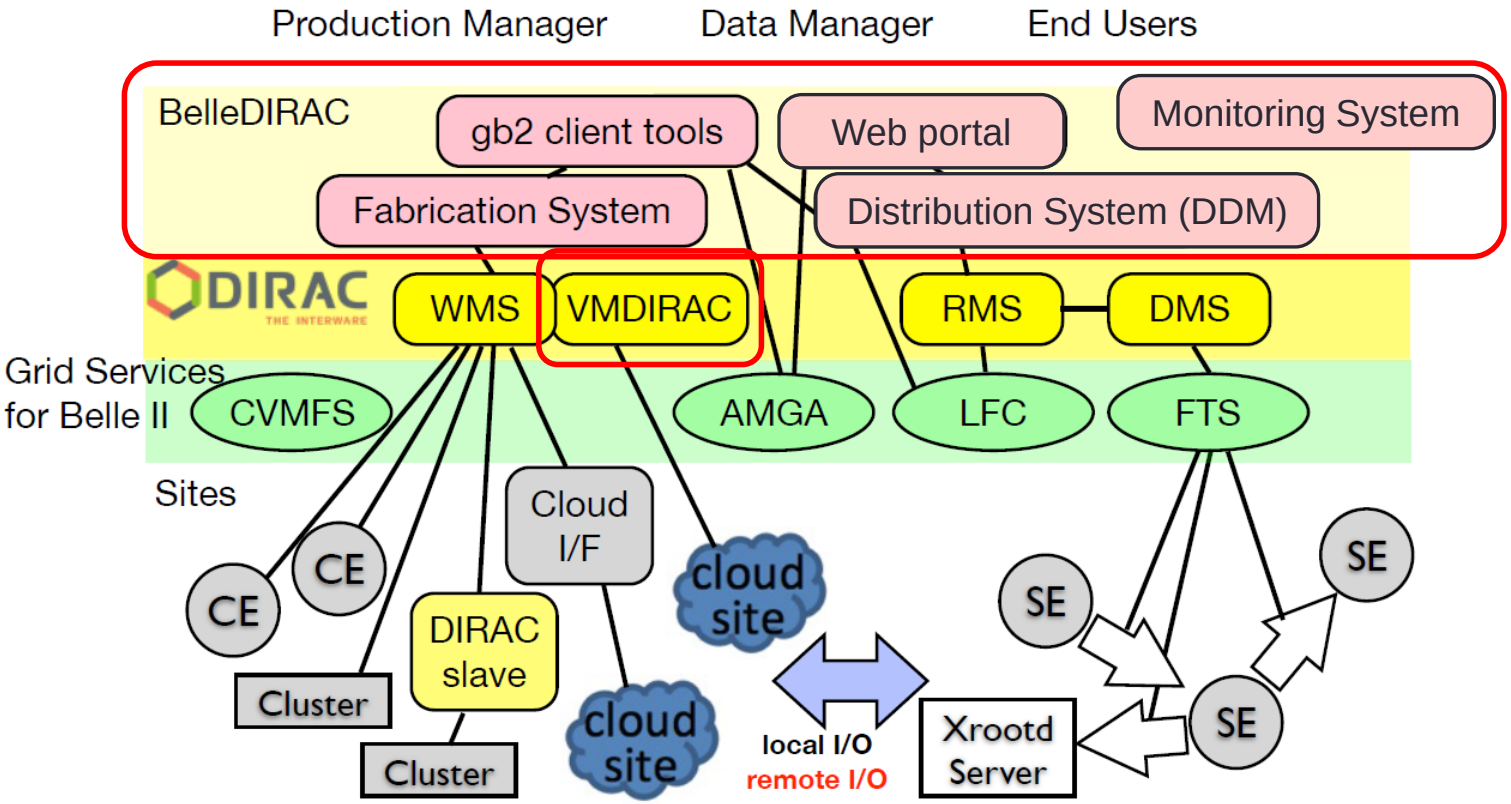
SSH

Condor (Cloud)

- Development (some nodes have independent configuration)
  - KEK, Nagoya, PNNL, Krakow, Melbourne, CINVESTAV
- Certification (used to test release candidate)
  - PNNL

# DIRAC extension for Belle2

- Belle II is developing own extension to fulfill our computing model, especially for “data management block”
- Own + inherited systems



# Data Management Block

- A unit of Belle II data handling
  - All files stored on same SE
  - Dataset can consist of multiple DMBs (= different SEs)
- A DMB contains fixed number of files (say 1000 files)
  - If one file is unavailable by any reason, replaced by alternative
  - Job failure, SE down before transfer…

Fabrication System

Job goes to data location
No input data relocation for now

  - Each file is stored on temporary “local” SE → assembled by DDM

Distribution System (DDM)

Dataset: /xx/yy/BdecayA

/xx/yy/BdecayA/sub1

XXX_120_YYY_task120.root
XXX_121_YYY_task121.root
XXX_122_YYY_task122.root
XXX_123_YYY_task123.root
XXX_121_YYY_task128.root

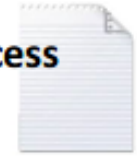
/xx/yy/BdecayA/sub2

XXX_124_YYY_task124.root
XXX_125_YYY_task125.root
XXX_126_YYY_task126.root
XXX_127_YYY_task127.root

Convention: Serial ID_Task ID

# Belle II Production System

**Definition**
- MC prod / data process
- Type (BB, ττ, ccbar..)
- # of events
- software version
- etc..



**PS**

-Production
-Distribution
-Merge

**Production manager (human)**
- Define "Production"

**Belle DIRAC**

**Distributed data management system**
- Check status of storages
- Define "Transfers"
- Gather outputs to major storage

DDM

output info

**Monitor**

**Fabrication system**
- Define jobs
- Re-define failed job
- Verify output files

**DIRAC**

DIRAC Transfer management

DMS

DIRAC Job management

TS, WMS

Submit job on site

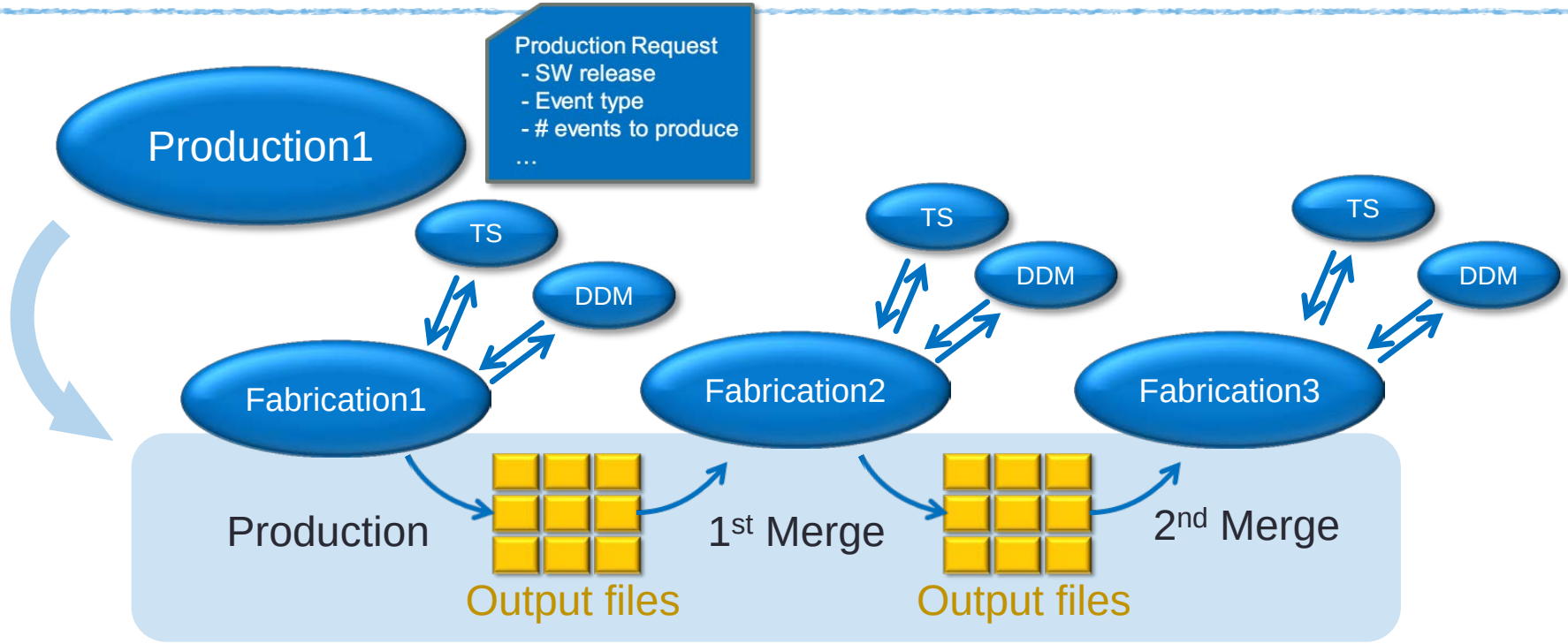
gbasf2


**Resource**

Destination storage ← FTS3 ← Temporally Storage ← Computing site

Courtesy by Yuji Kato

# Belle II Production workflow




- In Belle II production, job is handled by TS while data transfer is handled by DDM
- Fabrication communicates with them and test output file
  - Treatment of failure job, bad file (for both TS and DDM)
  - File validation (LFC, AMGA, SE metadata including checksum)
- ProductionManagement handles communication among Fabrications (e.g. hand output files to next Fabrication)

# Fabrication System: status and plan

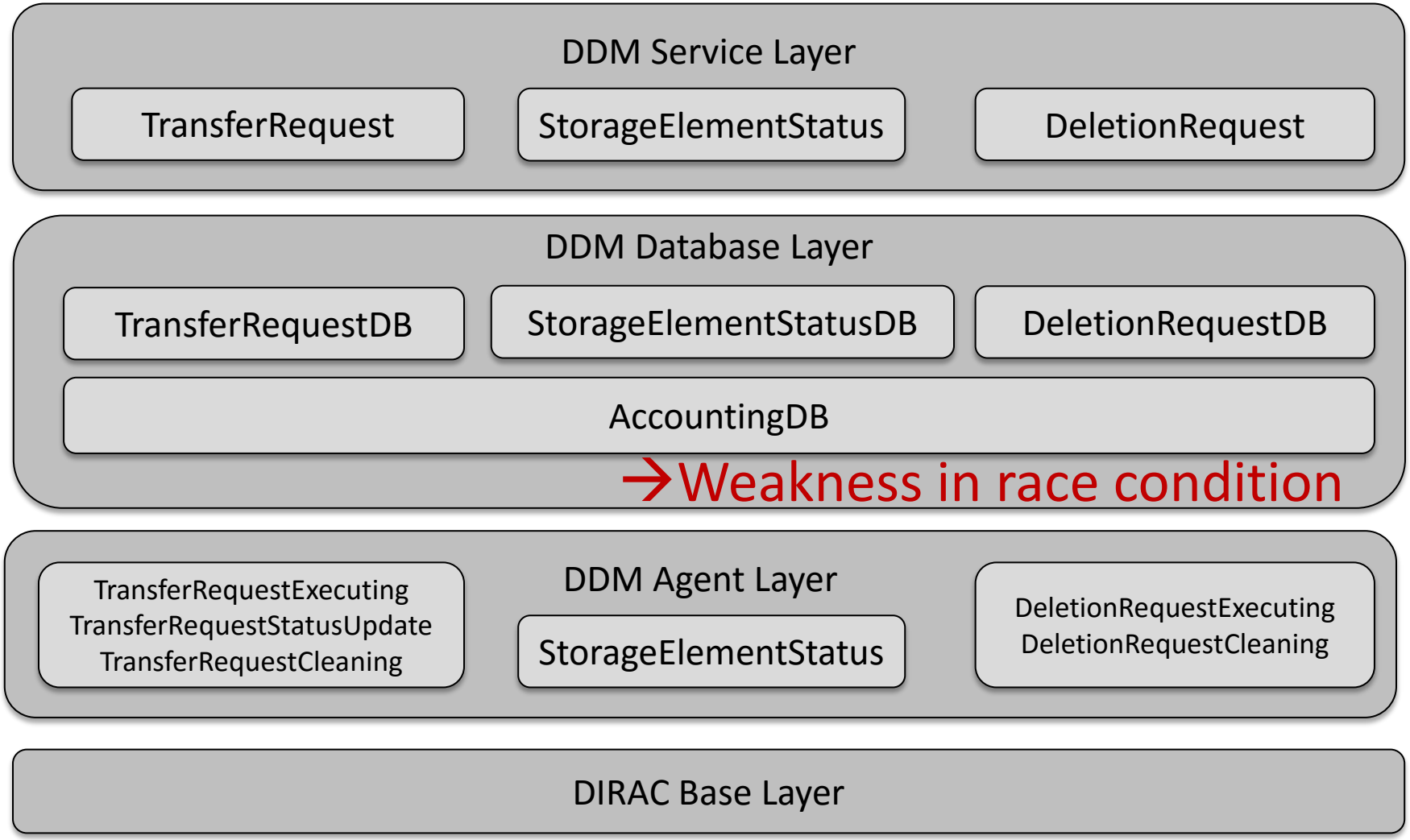
- Prototype has been utilized and proven the concept
  - Single output (one job creates one file)
  - Single thread Agent + single service handler
  - Simple DB schema (not well optimized)
  - Non-sequential file name (by failure job or dropped files)
  - Many human interventions

Improved performance and automatization
More practical for use case

- "FabricationSystem"
  - Multiple output per job
  - Multi thread Agent + multiple service handler
  - Optimized DB schema (index, split table, cache)
  - Sequential file name
  - More automatized verification, self diagnosis

# Working DDM Layout (version 0.1)

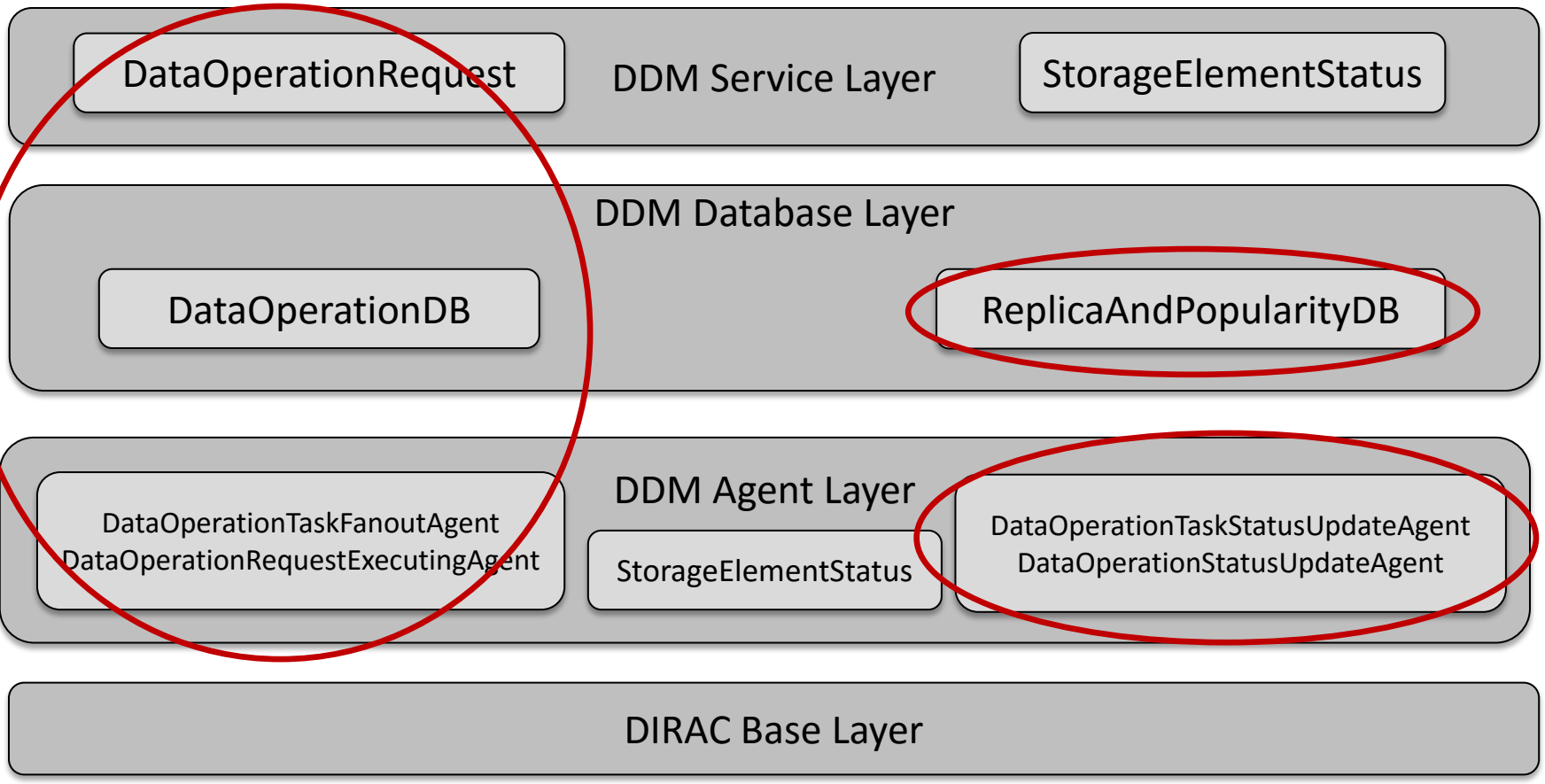




# New DDM layout (version 0.2)



- **Validating before MC9 in July**




→Unified data operation, data block level replica DB

# DDM Development & Operation Overview



- **Short Term (Alpha version)**
  - Provide patches to the exiting DDM components
  - Develop and test any blockers within the existing DDM framework
- **Medium Term (Beta version)**
  - Migrate to the *DataOperation* model
  - Develop the *Replica* components
  - Develop the *Popularity* components
  - Migrate *StorageHealth* to RSS
  - Integrate *NetworkingHealth* to RSS
- **Operation**
  - Continue to participate in the MC production
  - Prepare for the full dress rehearsal

# Latest Scalability Tests



- Tests:
  - *Replicate*AndRegister 1K LFNs from PNNL-TMP-SE to KEK-DISK-TMP-SE
  - *D*
  - *R* -DISK-TMP-SE
  - *D*
  - *R* -DISK-TMP-SE
  - *D*
  - *R* K-DISK-TMP-SE
  - *Delete* 10K LFNs from KEK-DISK-TMP-SE

1. ReplicateAndRegister: PNNL→KEK
2. Remove from KEK

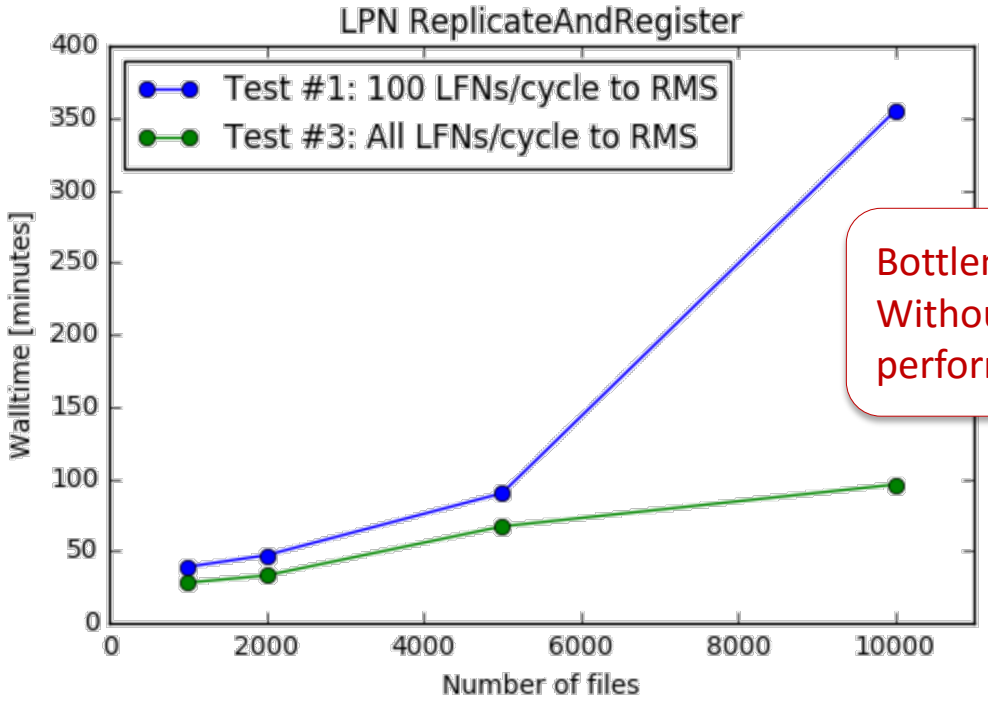
Tested by 1K, 2K, 5K, 10K LFNs

- We removed the LFC metadata assignment (checksum and size) in DataOperationExecutingAgent to see if RMS was more efficient at retrieving this information
  - Results were effectively the same

# ReplicateAndRegister Scaling



- We significantly improved the scaling of the DDM
  - Scaling appears to be linear



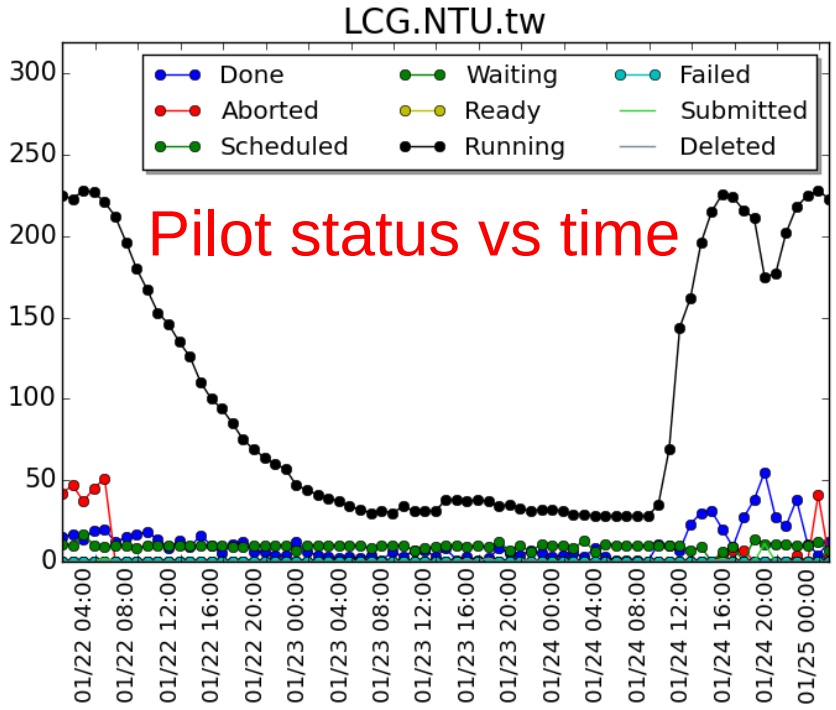

Bottleneck came from throttling by DDM
Without the throttling, linear performance are seen up to 10K files

Practical system tuning/redesign is ongoing

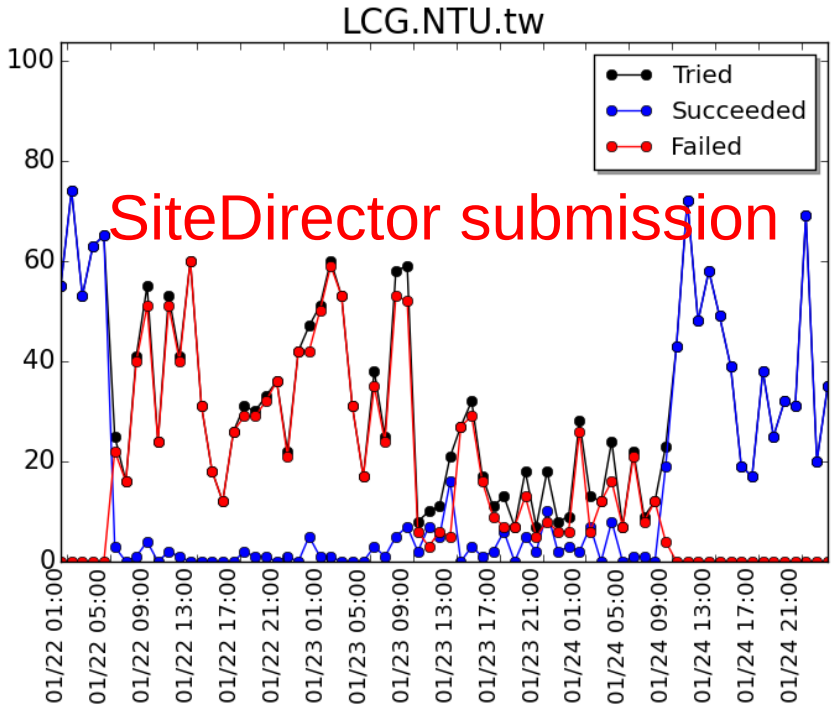
- LFC get metadata is now the bottleneck
- We need to update code to be multithreaded (LFC call, etc.)

# Belle II Monitoring System

- Basic concept: multi dimensional monitoring based on time-trend → diagnosed by automatic issue detector




PilotTrend shows decreasing
Of number of running jobs

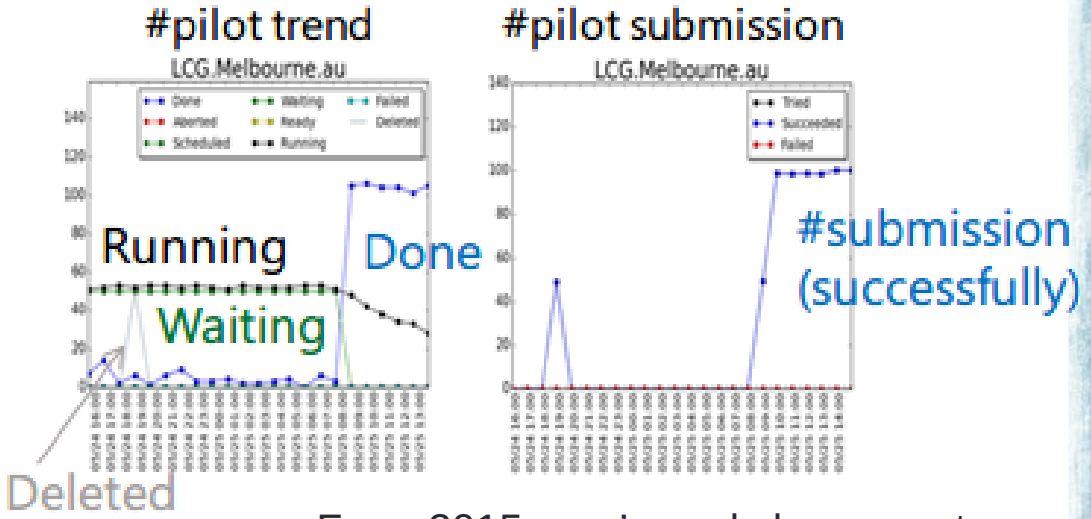



PilotSubmission shows
failure of the pilot submission

# Monitoring integration to DIRAC

- Various monitoring items have been collected, so far
  - WMS, DMS, RMS, downtime(GOCDB), SE free space, production…
  - Many of them are common among various VOs

- However, the system was implemented on non-DIRAC system




From 2015 user's workshop report

- This year, integrated to DIRAC as B2MonitoringSystem
  - Various types of plots are produced and collected in single platform (B2PlotDisplay, next page)
  - An Agent checks stuck components and automatically restart.

# B2Plot Display

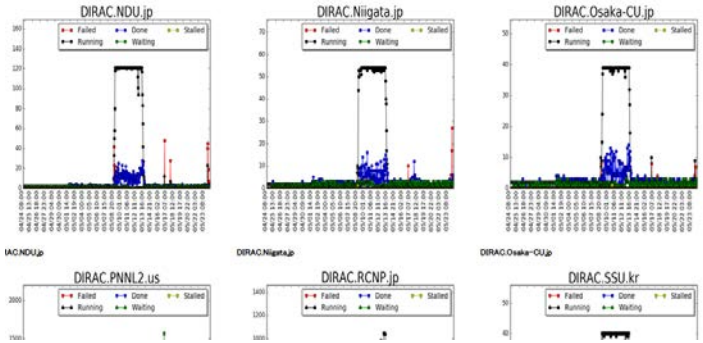
Pilot Trend | Pilot Submission | Pilot Processing | Pilot Waiting | Job Trend | JobStatus | Job Summary | Replication Trend | DDM Trend | Storage Accounting | Production Progress | DownTime

Collect various type of plots in single place.

## Example of tools useful also for base DIRAC

- **PilotSubmission**
  -Analyze log file of SiteDirectors and check if pilot submission is succeeded.
- **PilotTrends**
  -“Snapshot” of pilot statuses are recorded and plotted.
- **PilotProcessing**
  -Plot life time is plotted to find pilots finished before receiving payload with due to trouble.
- **JobSummary**
  -Plot total number of Running/Waiting jobs to check enough jobs are submitted
- **JobStatus**
  - Accounting plots “Running jobs” grouped by “FinalMinorStatus” for each site is shown.
- **Replication Status**
  - Plot Request status with type “ReplicateAndRegister”,
- **DownTime**
  -Retrieve information from GOCDB and format it in DIRAC style (Affected Site, SE)

Plot for PilotSubmission
Red colors means submission failure

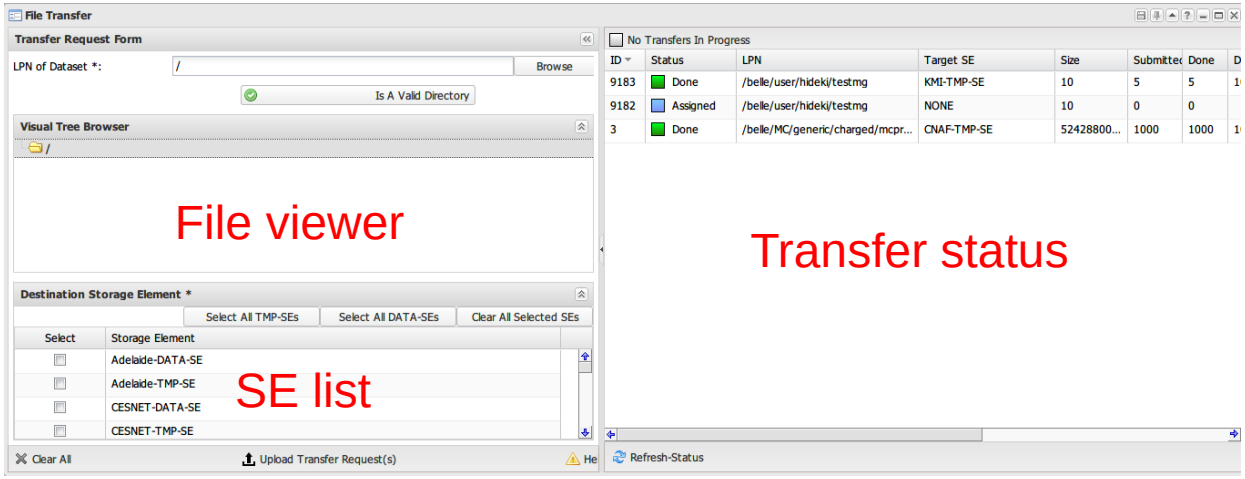



Grasp all site activities at once (cached plots)

# WebApp development

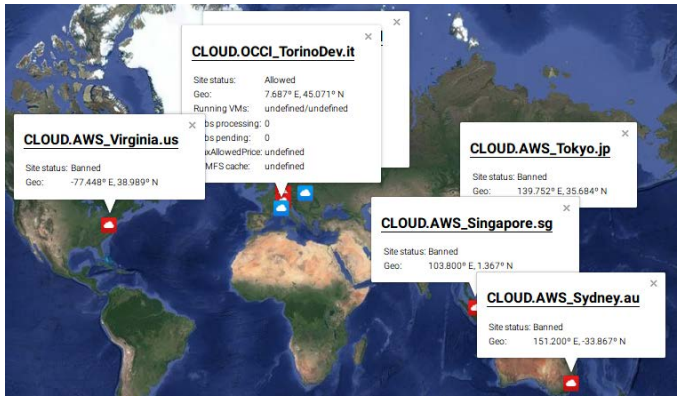
- “Transfer Map”
  - CE (job stats)
  - SE (free space)
  - Transfer volume


- “File Transfer”… GUI for DDM



# VMDIRAC in BelleDIRAC

- Tools migration from BelleDIRAC to VMDIRAC:
  - Cloud Resources Map (Params: CVMFSproxy, runningVMs, maxVMs)
  - EC2 Spot Price Agent (Params: zones, instances. Write MaxPrice to DB every few seconds)



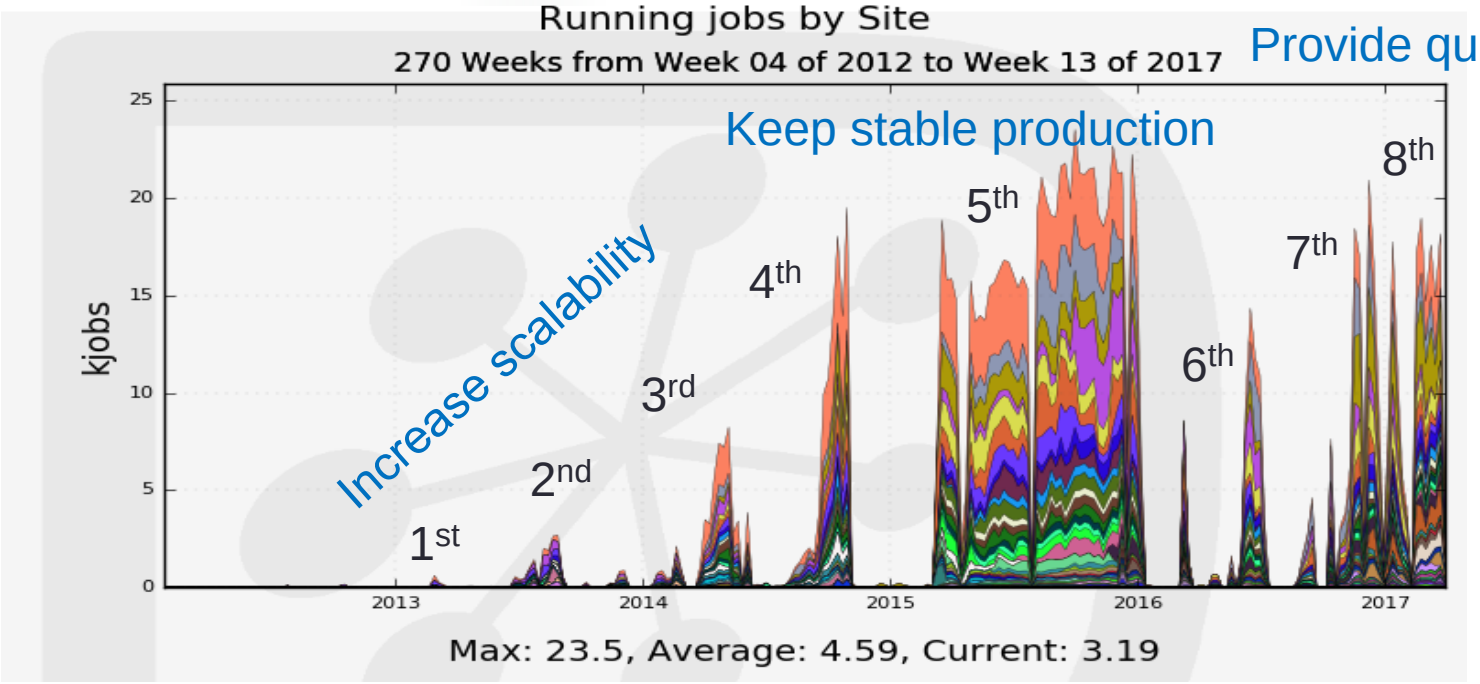

- Two use cases to be tested on Amazon:
- Tests with HPC cluster (cfnCluster) on EC2, maintained by sshSiteDirector
- Build and test new tools to work with commercial clouds:
  - New SiteDirector for REST API with VOMS (SAML in future?) authentication. Let's talk about generic API:
    - API (job/{submit,kill,status,output}, ce/{status}) and VOMS validation.
    - Proxy cert as text file in HTTPS request for VOMS authentication.
      - Delegate and store or pass directly to batch system?
    - JDL as text file in HTTPS request
      - Delegate and store or pass directly to batch system?
    - ISB tarball as binary data.
    - Example: curl -X POST -F 'isbfile=@123.tar' -F 'proxycert=@x509up' -F 'jobfile=@123.jdl' https://host/job/submit

# Belle II DC history at a glance

- “History of automatization”

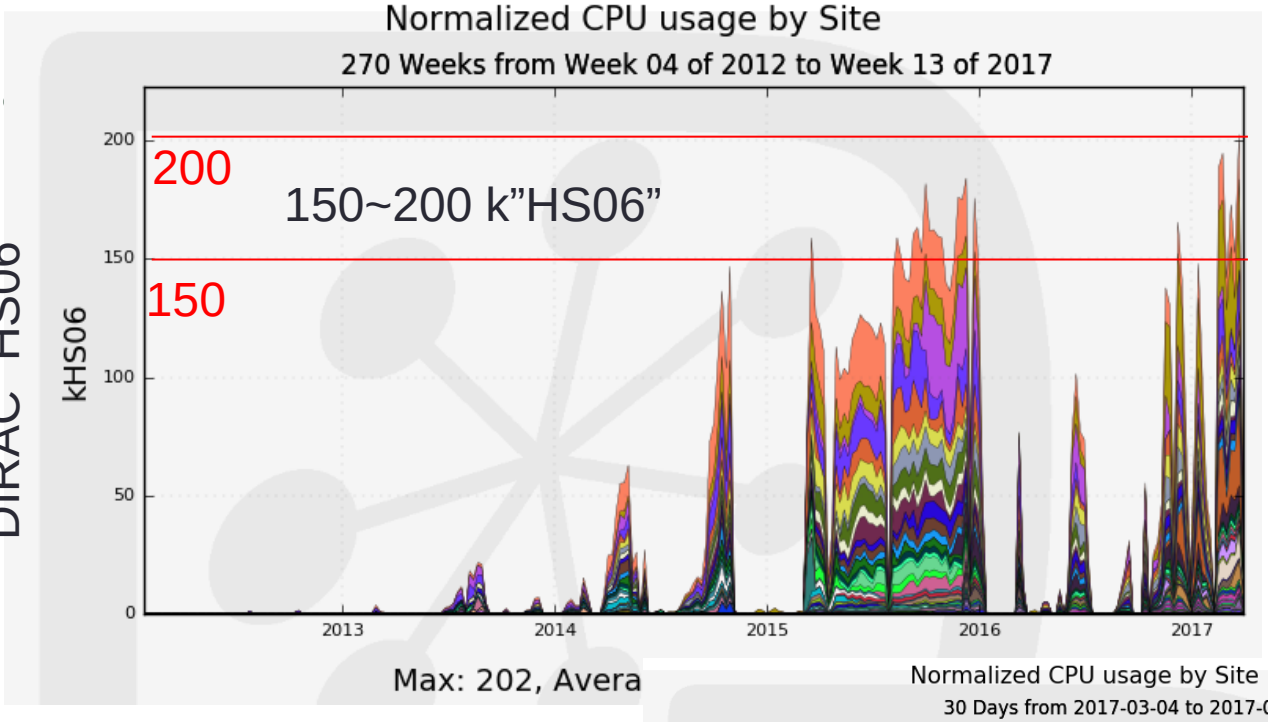
| | | | |
|---|---|---|---|
| Job workflow | Manual | Prototype | Fabrication |
| Data transfer | Job based | TS (FTS) | DDM (FTS) |
| Production workflow | Manual | | ProdSys |
| Monitoring | Manual | | Automatic issue detector |

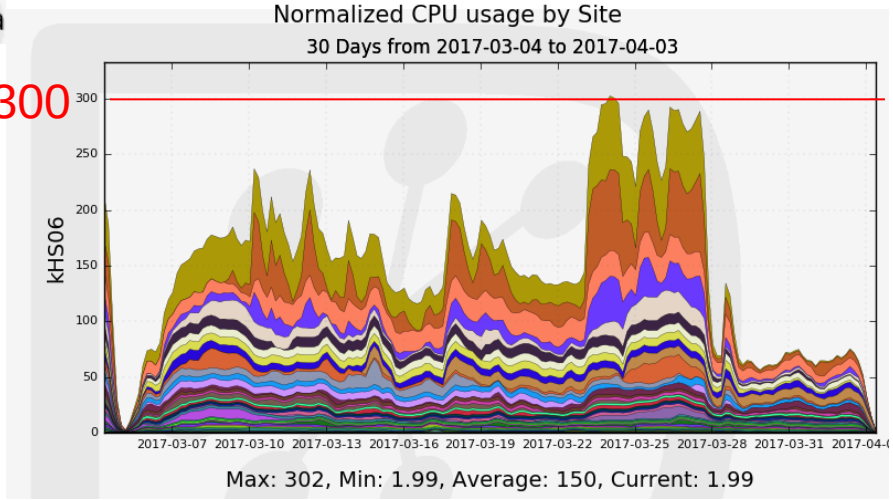
Human
DIRAC
Belle II development



# Performance: job management

DIRAC "HS06"

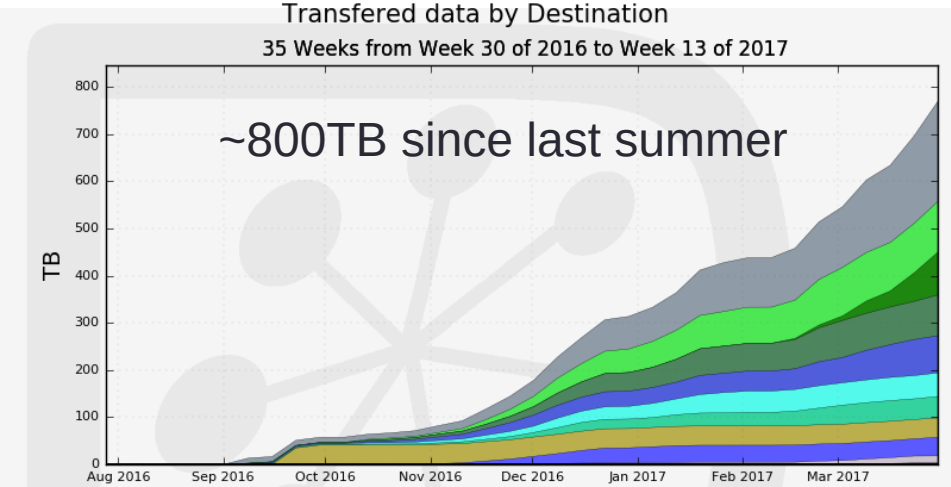



Instantly recorded ~300

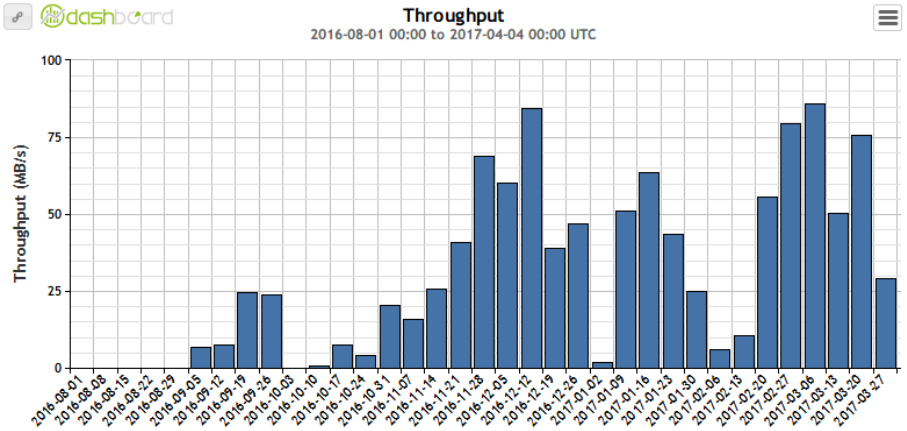
# Performance: data management

- FTS3 data transfer




~20 “local” SE

~8 “major” SE




Typically ~70MB/s

File size: O(10MB)~O(1GB)

# Prospects and plans

- More automatization (reduce human intervention)
  - Especially for data transfer and dynamic system control by monitoring info
- System tuning under realistic usage (e.g. analysis jobs)
- Scalability/throughput improvement
- Continuous development/upgrade
  - Planning Jenkins based automatic unit test
  - Seamless system update

- Cosmic ray data processing (2017)
  - First real use case to try raw data processing workflow

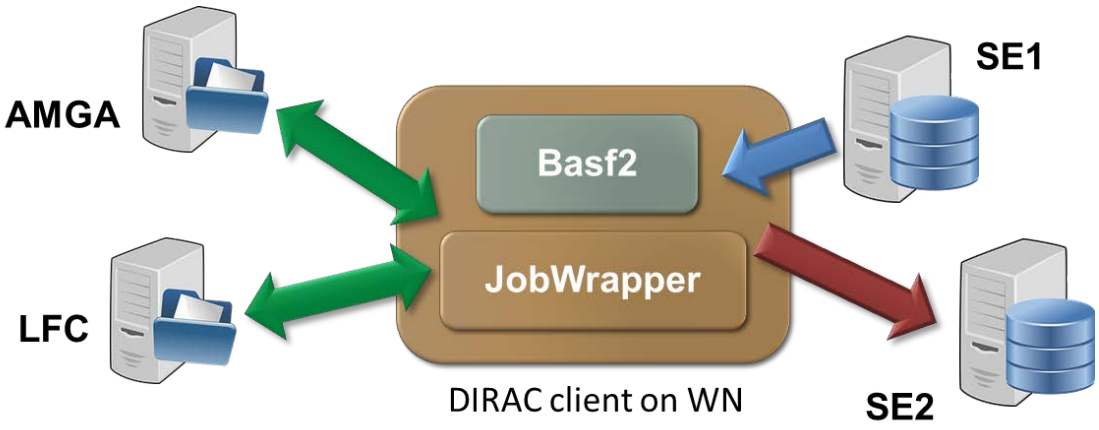
- System dress rehearsal (2017): before Phase-2 runs
  - To try the full chain workflow from raw data to skim

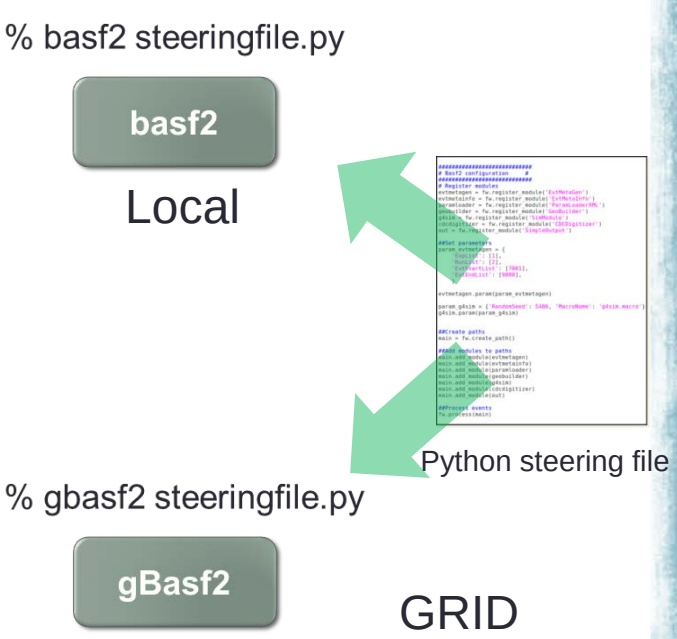
- Start of the phase 2 run in 2018

# Backup

# gBasf2

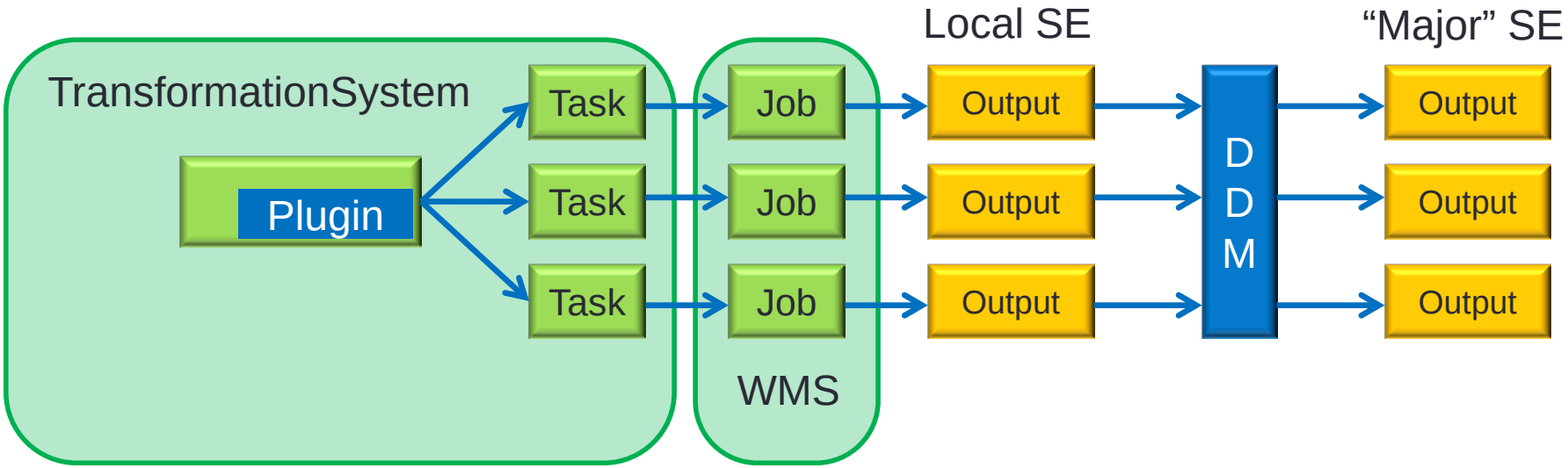
- Basf2 is our analysis software framework
  - Modular basis job executing platform
- Interface to distributed computing is given by gBasf2 (grid Basf2)
  - Provide transparent job execution on DC environment
    - Data input/output, file catalog/metadata registration…
  - Provide also collection of tools to handle job and data through DIRAC API (gb2 tools)

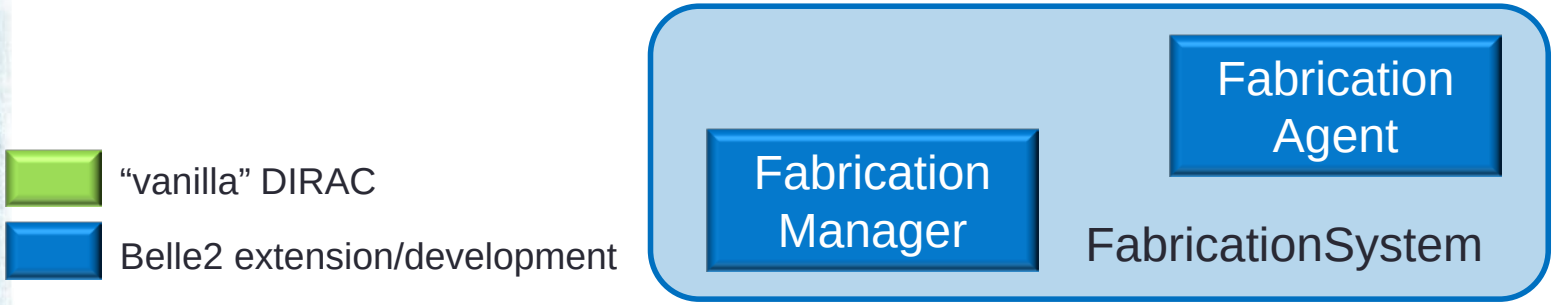


# Workflow: overview

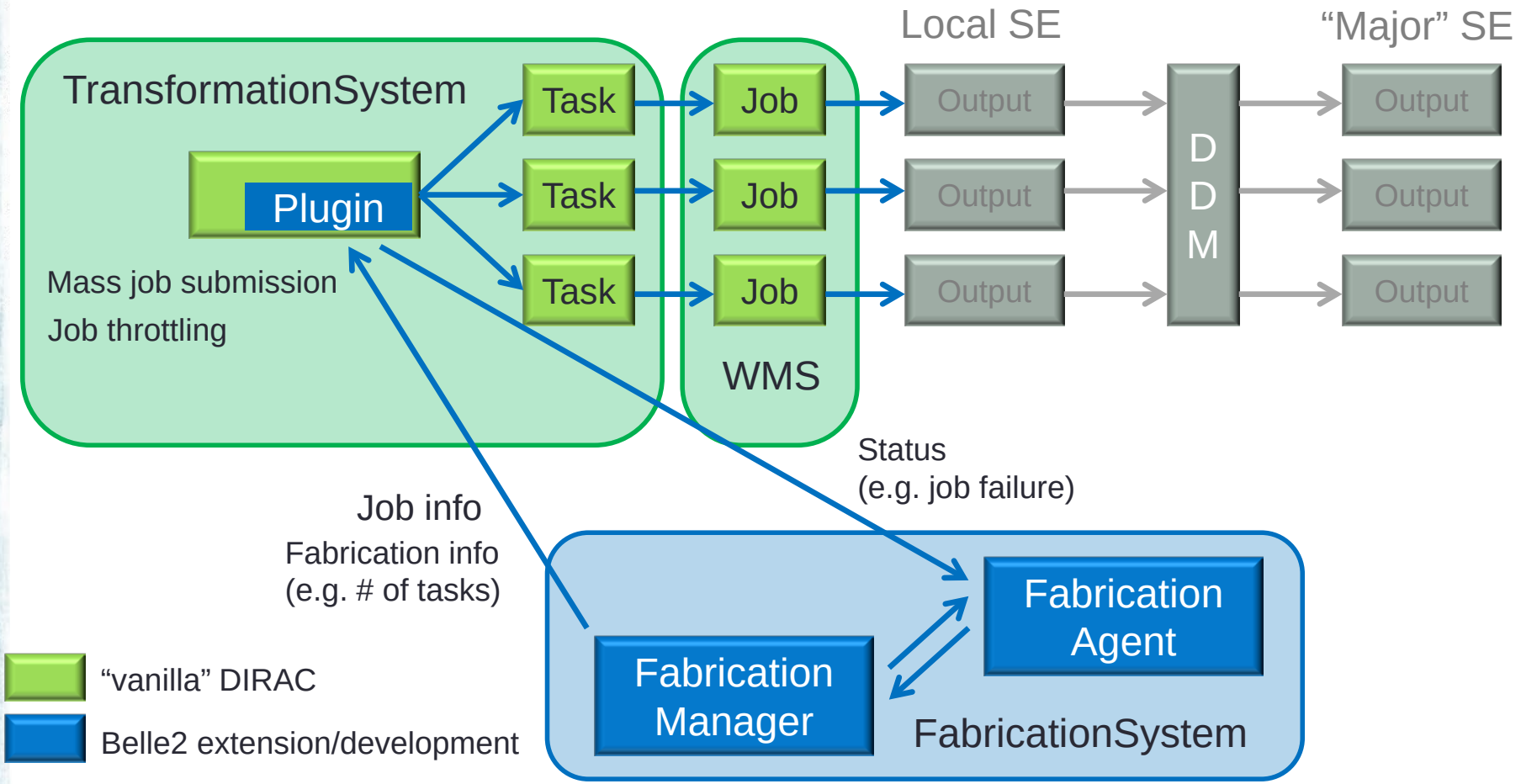
- Fabrication System exploits existing DIRAC components; TransformationSystem (TS) and WorkloadManagementSystem (WMS)
- TS is controlled by our plugin extension



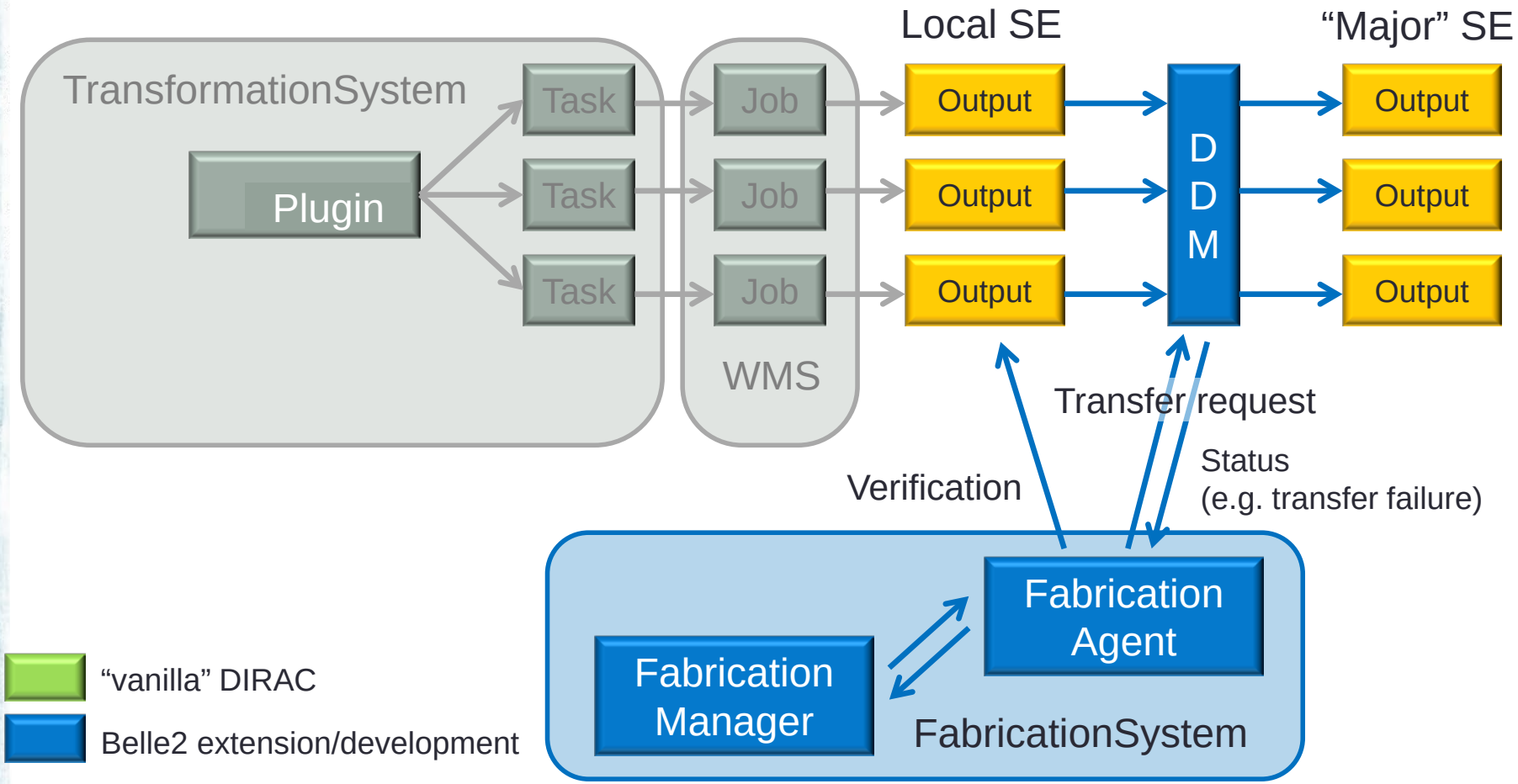
# Workflow: job management

- FS controls both job submission and failure job resubmission
- Each job status is monitored through TS



# Workflow: file management

- Once output file is created by GRID job, verifies each
- If file status is good, ask for DDM to transfer the file



# DDMS dress rehearsal

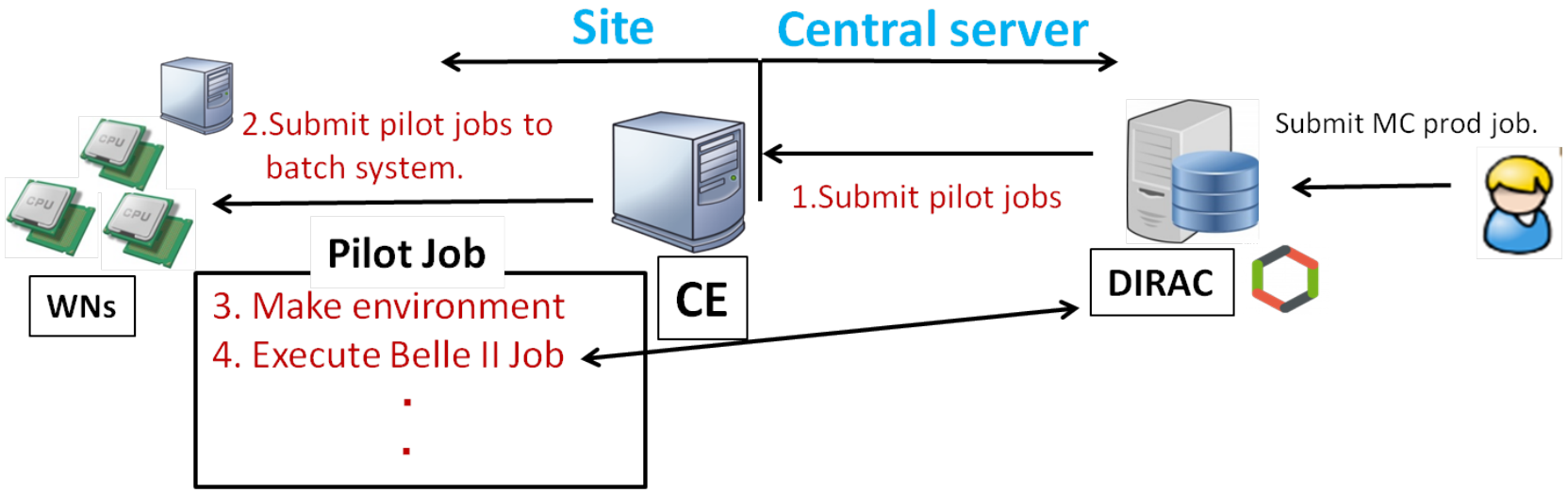
Pacific Northwest NATIONAL LABORATORY
*Proudly Operated by Battelle Since 1965*

- Testing data transfer rates using the DDMS between KEK and PNNL
- Transferring 5GB files we see consistent throughput rate around 12 Gbps
- Validates the current setup for RAW data transfer throughout the duration of the experiment
- Tests below were take between March 8th to 22nd 2017

# Monitoring system (passive way)



- Many interfaces → Need to identify "where the trouble happens"




- Store and process information of each step in database.
- Analyze log file to identify the origin of problem further.
- If problem is detected, show it in the web.

**Automatic issue detector**

Sites

- DIRAC.TIFR.in
  - Health checker info. : "Short pilot jobs" has been found since 20:20:00 UTC on 2016/12/25.(details)
- LCG.NTU.tw
  - GGUS ticket : "[TW-NTU-HEP] Job aborted with BLAH error"(125175) has been submited at 02:57:16 UTC on 2016/11/25.
  - Health checker info. : "CRL has expired" has been found since 21:20:00 UTC on 2016/12/17.
- LCG.Napoli.it
  - Job submission check : Pilot submission failure has been found since 06:25:00 UTC on 2016/12/26. (details)

# Monitoring system (active way) 

**Actively collect site status by submitting diagnosis job.**

## SiteCrawler:

Check the site environment to execute Belle II job

Site status summary

| site | worker node | CPU | #core | memory | OS | Kernel | rpm | cvmfs | releases | CPU Norm. | last updated |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ARC.DESY.de | batch0905.desy.de | Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz | x32 | 3015MB/cores | Scientific Linux release 6.8 (Carbon) | 2.6.32-642.6.2.el6.x86_64 | 2 problems found | Rev. 132 | OK (release-00-07-02) | 8.5 HS06 | 2016/12/26 15:25:10 |
| ARC.LMU2.de | vm-141-40-254-95 | QEMU Virtual CPU version 2.3.1 | x9 | 3567MB/cores | Scientific Linux release 6.8 (Carbon) | 2.6.32-642.6.2.el6.x86_64 | 4 problems found | Rev. 132 | OK (release-00-07-02) | 7.5 HS06 | 2016/12/26 15:23:29 |

## Job Submission check:

As DIRAC does not record failure reason, job submission is tried and record the result.

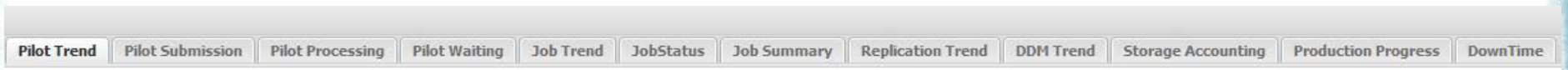
CE Job Submission test result

FaultDetail=[SSL authentication failed in tcp_connect(): check password, key file, and ca file.]

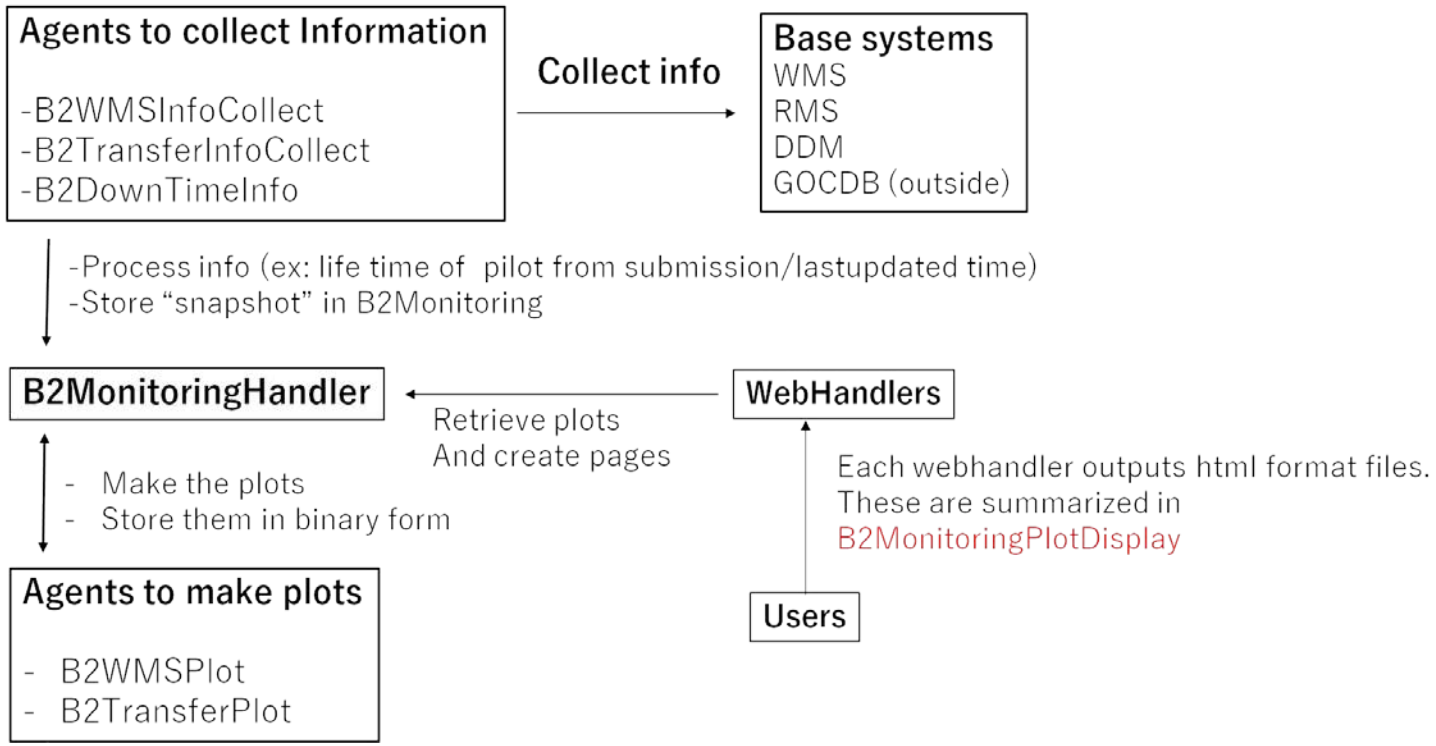
| sitename | CE | queue | status | last updated time |
|---|---|---|---|---|
| LCG.Cosenza.it | recas-ce-01.cs.infn.it | cream-pbs-belle | submission_failed | 2016/12/26 10:20:13 UTC |
| LCG.KEK.jp | kek2-ce02.cc.kek.jp | cream-lsf-gridbelle_heavy | ABORTED | 2016/12/26 10:00:18 UTC |

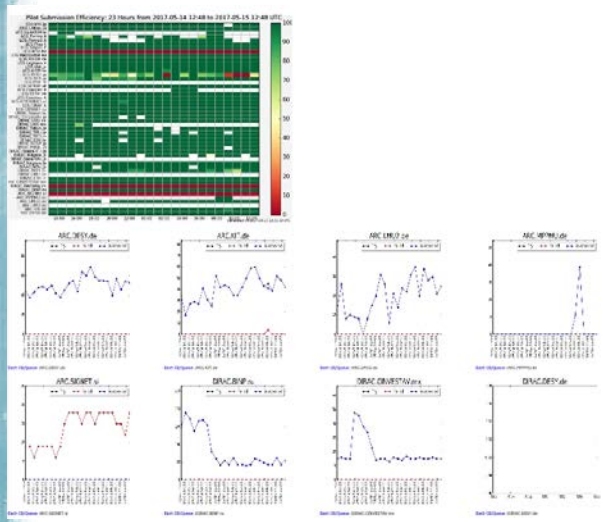
**Our activity maximize the availability of the resource!**

# B2Plot Display



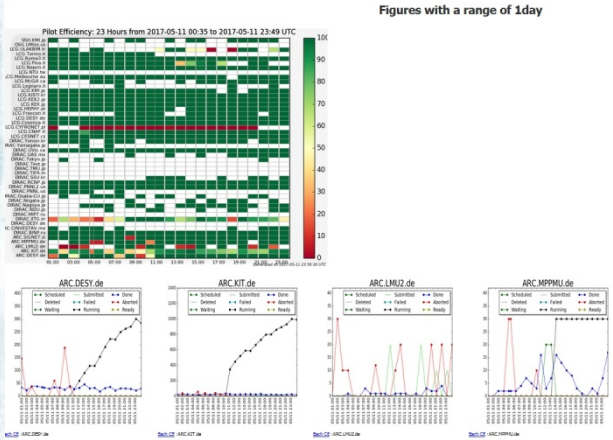

- Collect various monitoring tools in single place.
- Various agents collect information from base system
  - Process
  - Make snapshots
- Everything is written inside the DIRAC





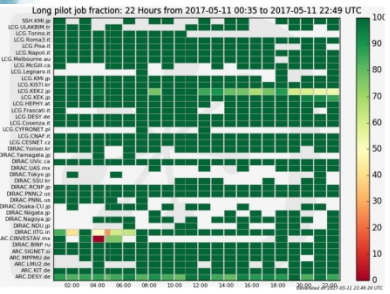
## PilotSubmission

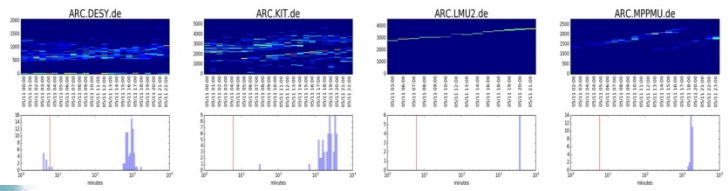
- A agent which analyze SiteDirector log is developed.
- ”Tried”, “Succeeded”, “Failed” submissions are plotted
- 2D plot shows Succeeded/Tried for all the Site.
- Possible to show each CE/Queue by clicking link in each plot.
- 13 GGUS submitted from this monitor in ~half year operation




## PilotTrends

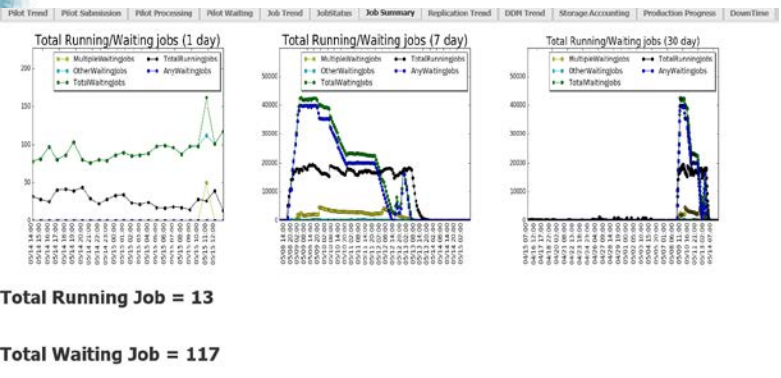
- Agent collect “snapshot” of pilot status.
- Current running/waiting plots are plotted
- In addition, # of terminal status in 1hour is also plotted.
- 2D plot shows (Done)/(Done+Aborted+Failed) for each site.



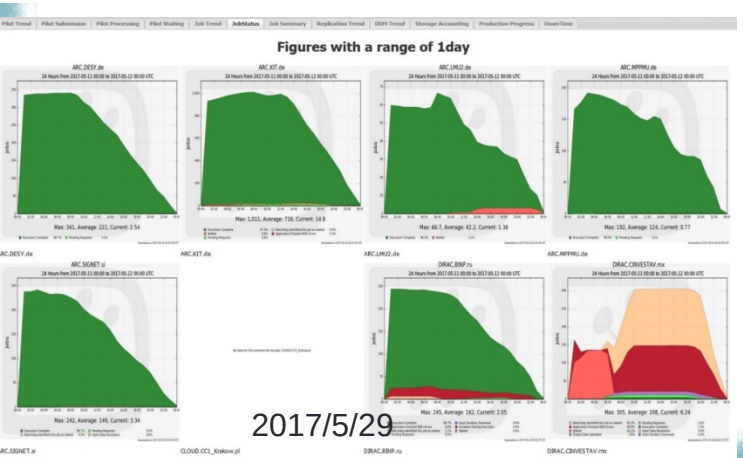

## PilotProcessing

- Life time (Submission-LastUpdatedTime) for pilot jobs with “Done” are plotted in minutes.
- Red line in each color shows possible shortest life time (JobAgent Cycle*Polling time)
- DIRAC installation failure, small disk space etc can be detected with this monitor
- 6 GGUS tickets submitted in ~half year operation.




## JobSummary

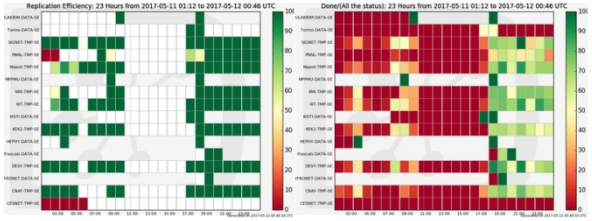
- Monitor trend **of total running/waiting jobs** to check enough jobs are submitted for shifters

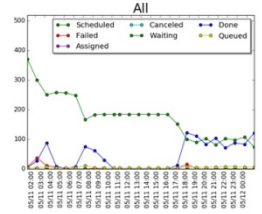


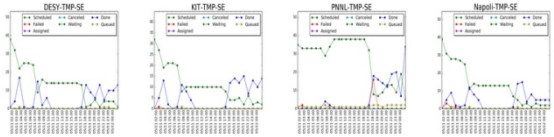

## JobStatus

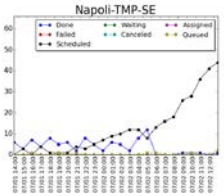
- Put Accounting plots “Running Jobs” grouped by “FinalMinorStatus” for each site

- **By combining monitoring tools shown above, we can clearly identify where the source of problem.**




## Replication Status

- Plot the request status with type "ReplicateAndRegister"
- Piling up of the "Scheduled" is sign of problem. (like stuck of FTSAgent).
- 2D plot (left) shows Done/(Done+Failed)
- 2D plot (right) shows Done/(All other status)

### Affected Sites/SE

| Site/SE | Name | Down/Total CE (only for sites) |
|---|---|---|
| Site | LCG.CNAF.it | 1/5 |
| SE | CNAF-TMP-SE | - |
| SE | CNAF-DATA-SE | - |

### Overview (Link for shift log)

| Start time (UTC) | End time (UTC) | Description | Link |
|---|---|---|---|
| 2017-05-14 06:00 | 2017-05-26 06:00 | ce01-lcg dismission | GOCDB page |

### Hosts

| Service | Host name | Severity |
|---|---|---|
| gLExec/CREAM-CE | ce01-lcg.cr.cnaf.infn.it | OUTAGE |

## DownTime

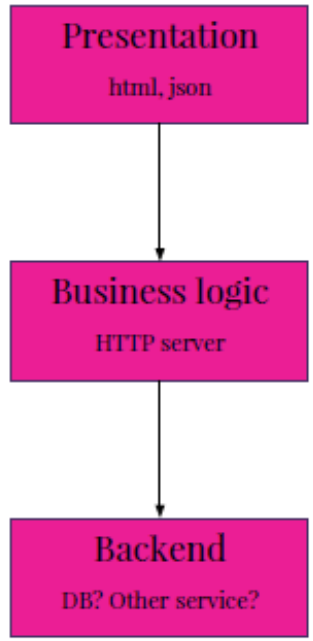
- Retrieve information from GOCDB and Format in DIRAC style (Site, SE)

# Serverless architecture with Lambda, API Gateway, Batch

**Serverless logic tier:**

Amazon API Gateway is a service for creating and managing HTTP APIs

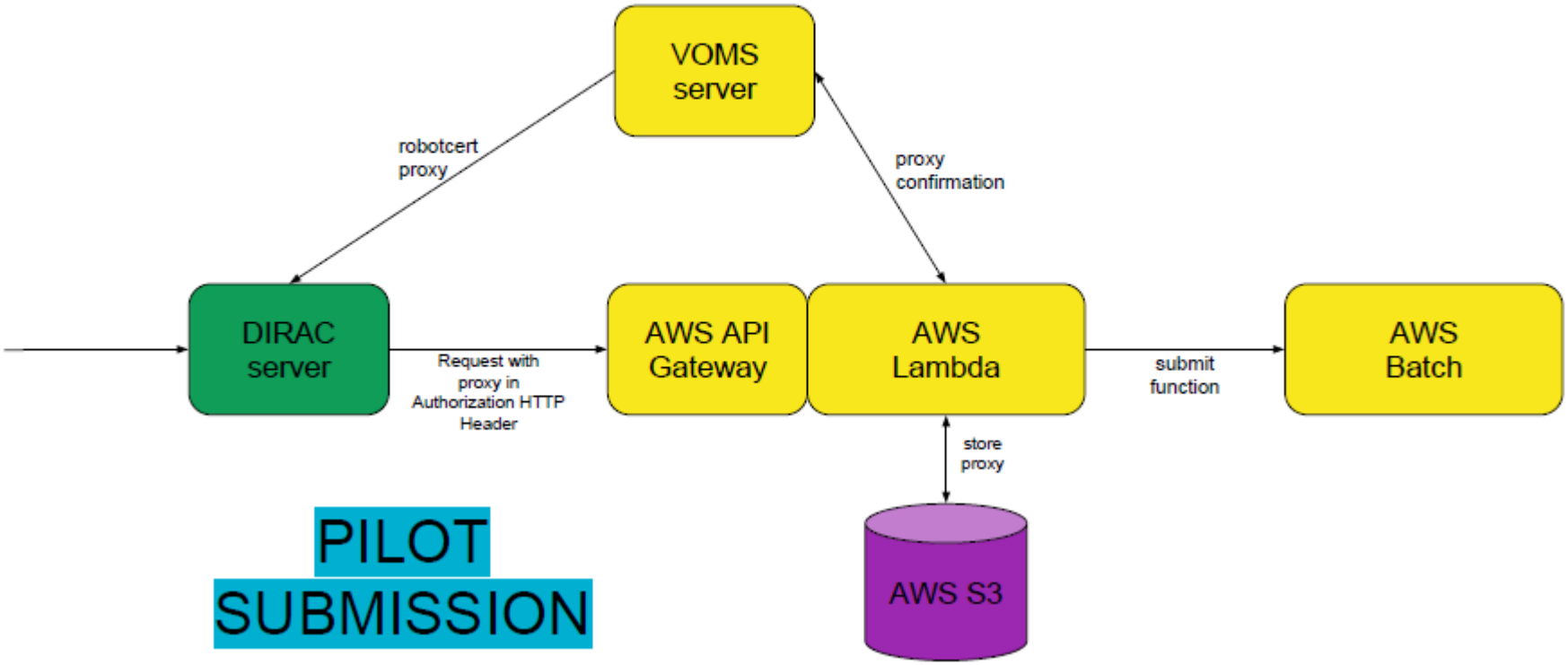
AWS Lambda is a compute service that lets you run code functions without provisioning or managing servers. AWS Lambda executes your code only when needed and scales automatically.




AWS Batch is an experimental batch computation service

- announced two months ago
- based on containers clusters
- we can run Ruby/Python/Node/Bash jobs
- No classical endpoint. The idea is to use AWS API Gateway and AWS Lambda to provide HTTP interface with X509 authentication.

# AWS Lambda with VOMS authentication



# VOMS and X509 proxy

**How can we validate a VOMS proxy cert in "cloud function" ?**

- Proxy chain validation:
  - check number of issuers
  - check each issuer (verify public key, verify DN)
- Valid DN ? Lifetime (hasExpired?) Has VOMSExtensions?
- Is this DN registered in VO's VOMS? Is a member of VO's group ? days to membership expiration ?
- Presence of CSRF prevention HTTP header ("X-VOMS-CSRF-GUARD")

# AWS cfnCluster

AWS cfnCluster tool is setting up whole HPC cluster.

Available schedulers: **sge, openlava, torque, slurm.**

Head node could be operated by **SSHSiteDirector.**

cfnCluster is using CloudFormation (InfrastructureAsCode) to setup autoscaling environment.