Belle II with DIRAC and BelleDIRAC

Ueda I.

2024.Jun.19 - DUW10

Belle II Computing System

Belle II has been using DIRAC for many years

- (since prehistoric era to me, DIRAC CS tells 2012)

and Rucio since 2021

- **DIRAC** as the main framework
  - Configuration, RSS, WMS, DMS, RMS, Transformation, Accounting/ Monitoring, ...
  - With extension BelleDIRAC
    - User interface, Production System, Monitoring, Interfaces for Rucio (incl. RucioFileCatalogClient, RucioClient, ...), BellePilotCommands
- **Rucio** used for data management
  - File Catalog (file names, attributes, replica locations)
  - Automated distribution, deletion, control of number of replicas
  - Configuration (SEs, accounts, etc.) sync'ed from DIRAC DS with a DIRAC Agent (RucioSynchronizer)
  - Transfer/Deletion monitor and Data volume accounting monitor

DIRAC DMS and Rucio?

Data distribution / deletion

- Done by Rucio
  - Pre-defined subscriptions => automatic distribution
  - Manual management of rules => to trigger replication / deletion
  - Production system + Rucio interface in DIRAC to trigger deletion of intermediate files

FileCatalog

- DIRAC FileCatalog plug-in 'RucioFileCatalogClient' lets DIRAC components/APIs to access Rucio File Catalog transparently
  - We are now also making metadata APIs in BelleDIRAC RucioFileCatalogClient (see later slide)

DIRAC jobs

- Uses DIRAC DMS APIs for downloading/uploading input/output files

End-user client tools

- Downloads/uploads files with DIRAC APIs
  - Download can be done with Rucio APIs with the option
- Sets Rucio rules to trigger replication
  - Replication asynchronously done by Rucio
- Sets a lifetime to Rucio rules to trigger deletion
  - Deletion asynchronously done by Rucio

Belle II Computing System

BelleDIRAC KEK, BNL + a few extra nodes to run special SiteDirectors

- For productions and Analyses

BelleRawDIRAC KEK

- At KEK to register raw data to Grid

Rucio BNL

- For data management

FTS KEK, BNL

- For actual file transfers

AMGA KEK

- The metadata catalog => **to be replaced with Rucio**

VOMS, IAM KEK

Belle II will keep using VOMS at KEK

CVMFS Repository, Stratum0 KEK

- Pilot files, pre-installed client

The servers at KEK are all to be replaced this summer... (KEKCC renewal)

VOMS and IAM

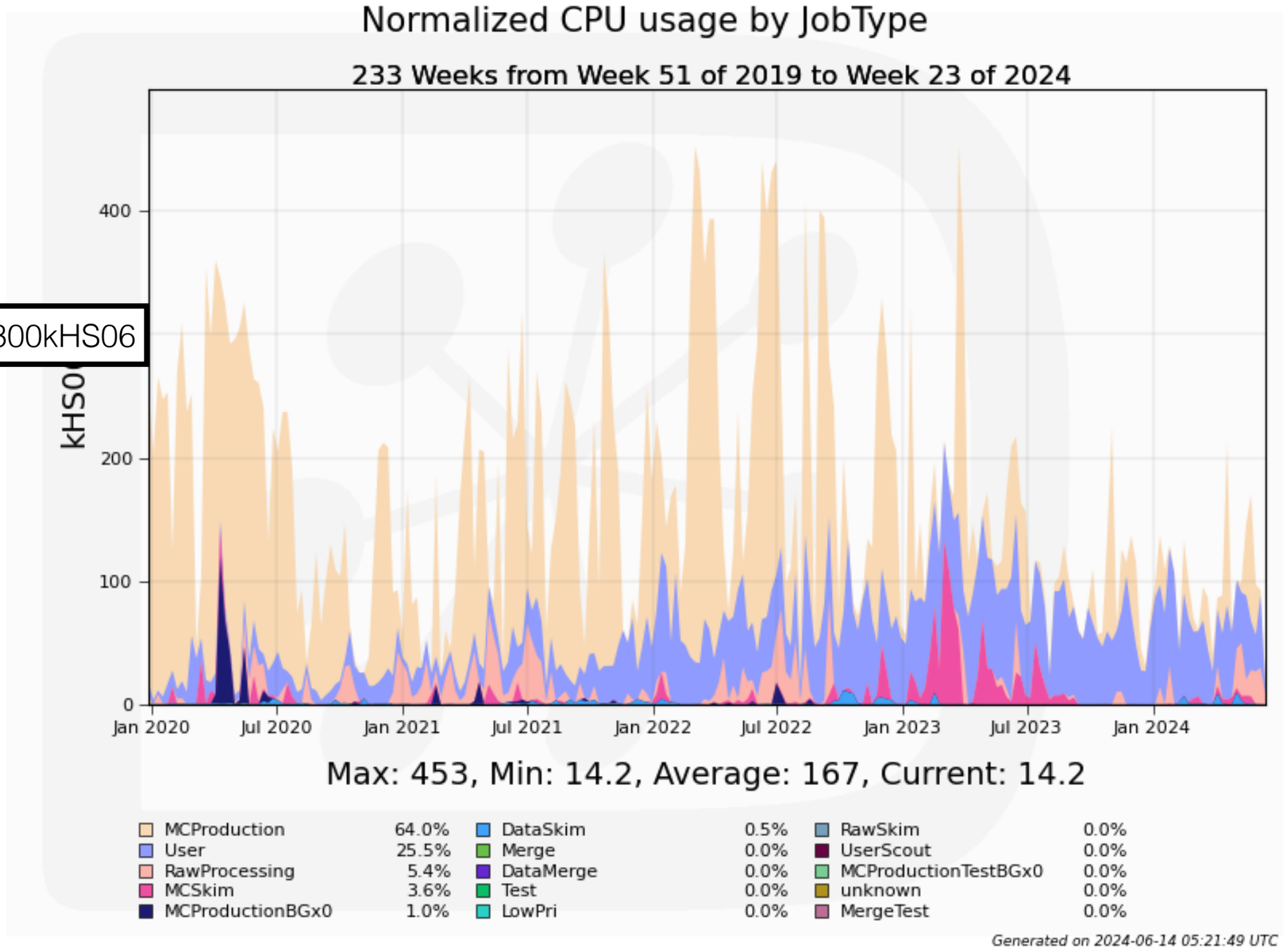
Belle II will keep using VOMS at KEK

- KEKCC keeps running VOMS on RHEL7

IAM at KEK

- **was successfully used in submitting pilot jobs with tokens**
  - In the validation setup against a single site
  - Not in production yet
- will **not** provide VOMS proxies (as far as VOMS is usable)
  - IAM VOMS attribute authority not deployed
- Belle II may not start using IAM heavily until we see some clearer future
  - IAM versions
  - Token versions
- i.e. currently not sync'ing VOMS registration to IAM

Belle II Computing Activities



300kHS06

MCProduction

Currently dominant but not constant

Analysis

Increasing on Grid

RawProcessing

Not large while the accumulated detector data are small

MCSkim DataSkim

Evolving to produce more effective data for analysis

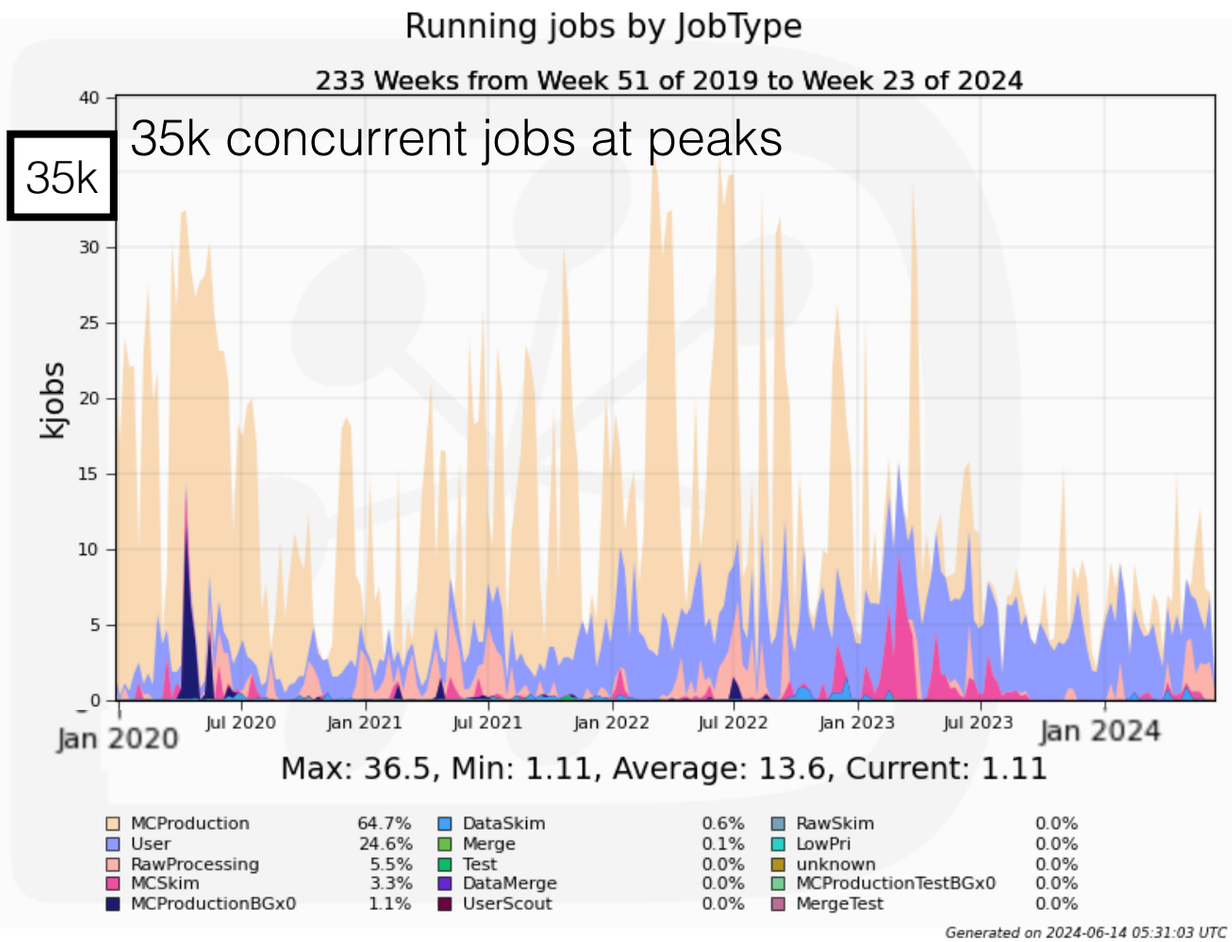
Belle II Computing: Jobs

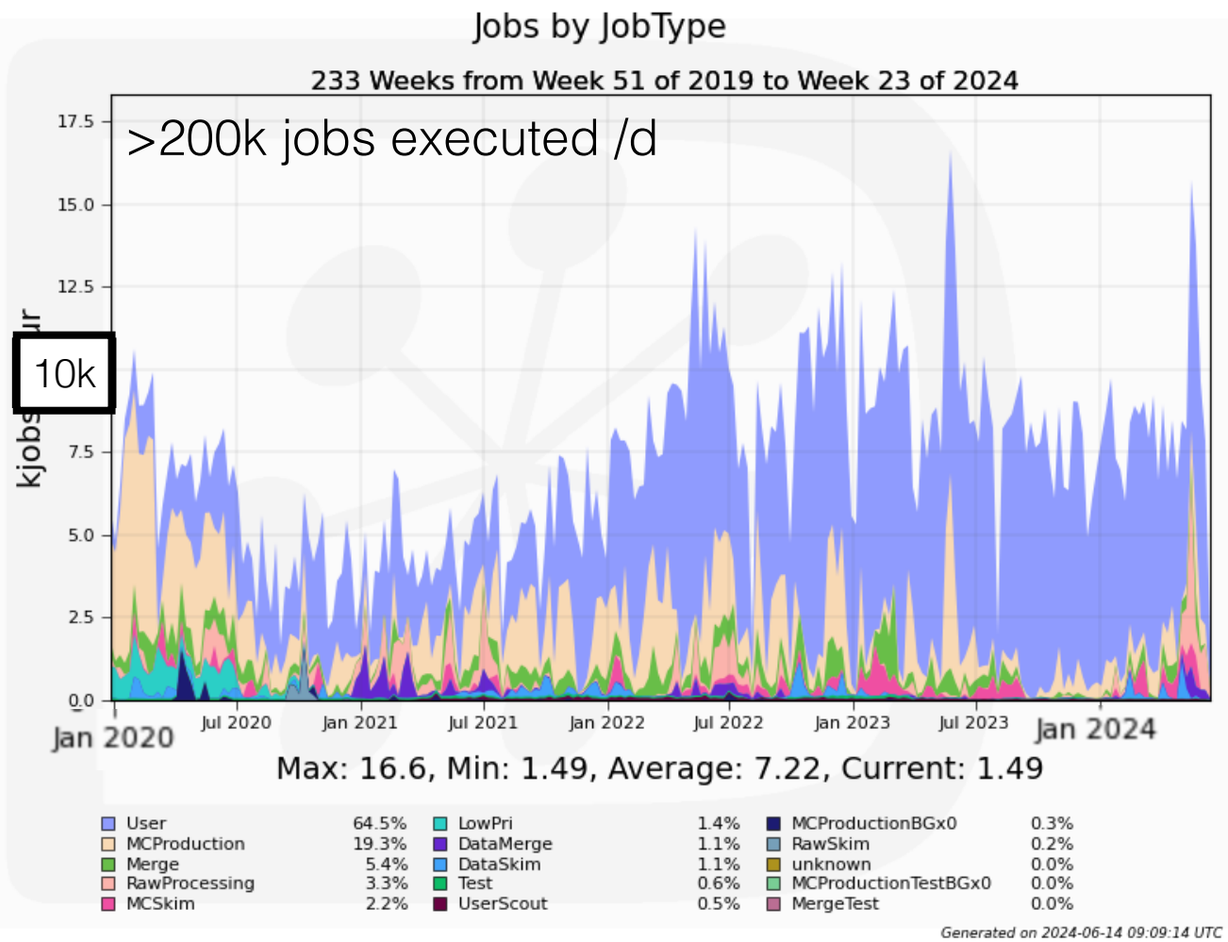
Production jobs = Intermittent

- Usually long jobs with good behaviour, though not constant

Analysis jobs = constant pressure with higher execution rate

- Large number of short jobs <= Many short "runs"
- Counter measures include reducing the size of **Input Sandbox files**

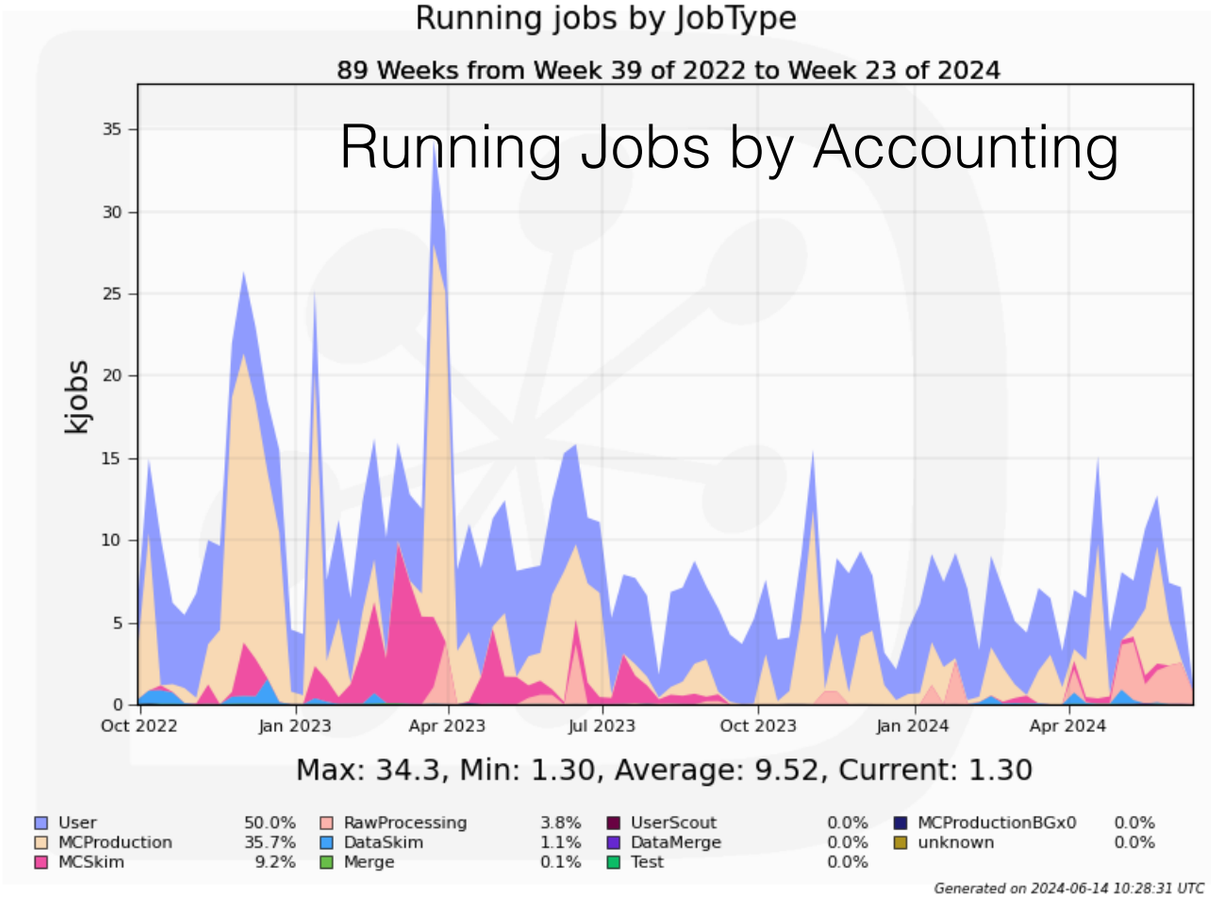




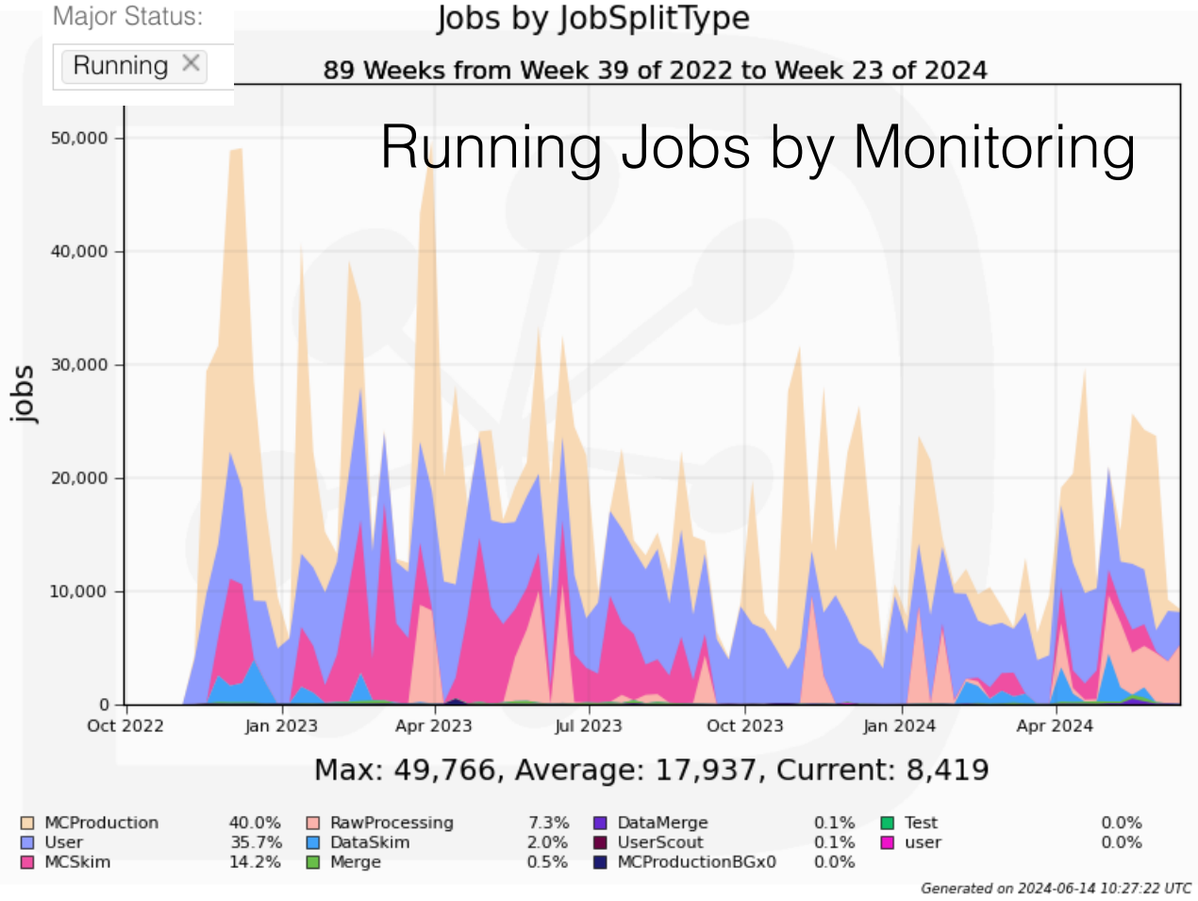
WMS Monitoring

We have also started using WMS Monitoring

Reports: Accounting
Category: Job
Plot To Generate: Running jobs
Group By: JobType



Reports: Monitoring
Category: WMS Monitoring
Plot To Generate: NumberOfJobs
Group By: JobSplitType
Major Status: Running

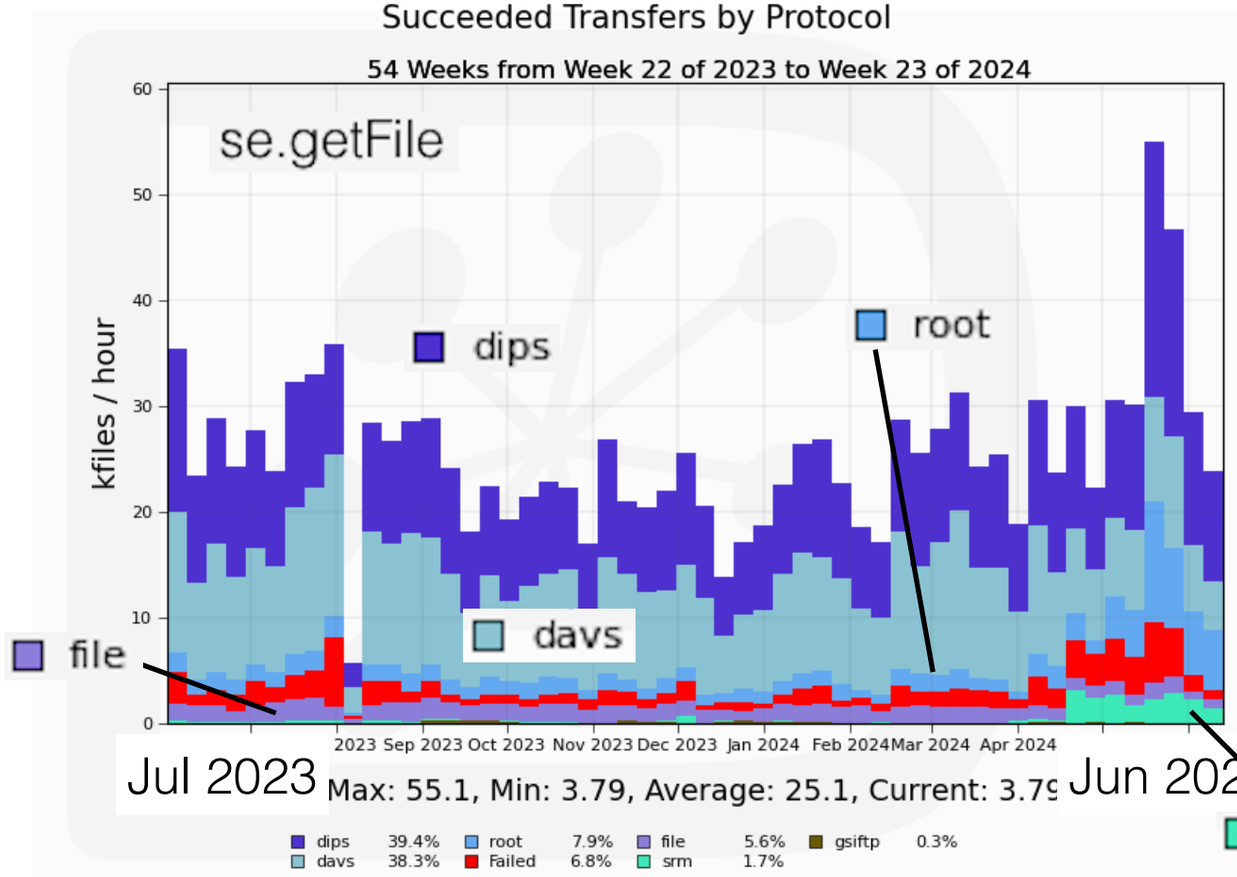


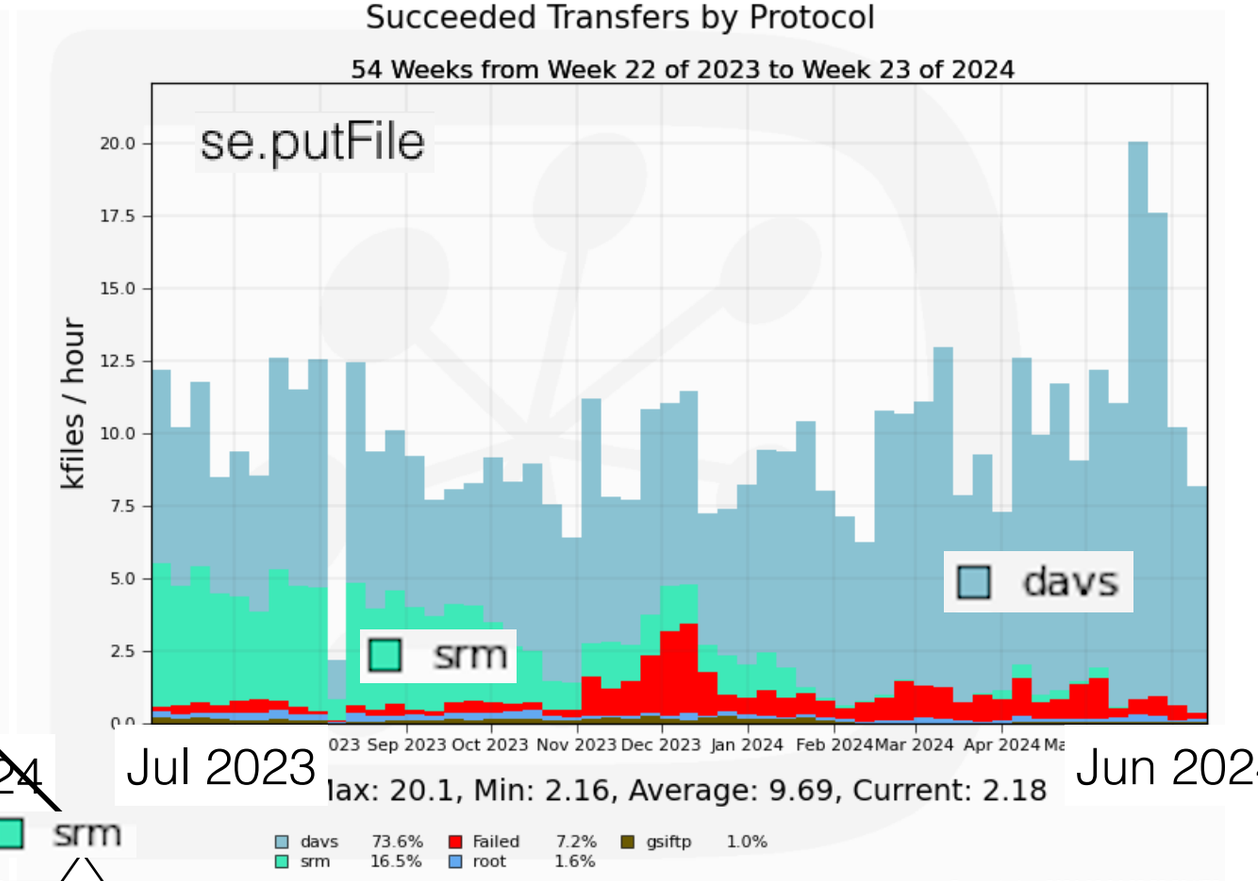
Data Operation

File transfer between SEs with Rucio

Download/upload files with DIRAC APIs

- Mostly 'davs' replacing 'srm' (with recent regression => later slide)
- Read with 'dips' from SandboxStore
- Read with 'root' where preferred (eg. to utilize XCache)





recent regression => later slide

TAPE with REST

Staging files from TAPE with REST

- In Belle II, it is done by Rucio, not by DIRAC
- Started manually testing SE-by-SE, one successful, another failing, ...
- Protocol section https/davs added, or to be added, to DIRAC CS
  - to be sync'ed to Rucio SE attributes

DIRAC v8.0

Upgrade to DIRAC v8.0 took quite some time

- Adapting the BelleDIRAC codes
  - Python 2 => Python 3
  - DIRAC v7.3 => v8.0
- Switching pilots Python 2 => Python 3 step-by-step while running v7.3 (April)
  - setup DIRAC with our own client installation on cvmfs, instead of installing DIRAC
  - still with BellePilotCommands, not to have too much change while running v7.3
    - Now with v8.0 we plan to use vanilla PilotCommands (after finishing other priority works)

Currently using DIRAC v8.0.44 in production

- Deployed in production BelleDIRAC on May 14
- Still v7.3.38 for BelleRawDIRAC until the end of the current data taking period
  - v8.0.44 already tested with a test instace

Aiming for the latest (v8.0.48 or later)

- FIX: (#7631) Avoid printing out ... in SQLAlchemy
- NEW: (#7630) HTCondorCE: UseSSLSubmission option (*)
- ...

DIRACOS2

DIRACOS 2.31

- Currently in use in production
- We tried 2.38, but found issues
  - The blocker was the proxy key length
- File downloads with davs suddenly started showing timeout since around Mar. 26
  - No such issue observed when tried with diracos2.38
  - Tried to update gfal2/davix/curl in diracos2.31, while keeping python3.9 and openssl as is, but not easy and gave up...

Proxies with 2048-bit key

- dirac-proxy-init patched as suggested to force upload of proxy
- ProxyDB also needed a patch
- ProxyDB getting populated with ones with 2048-bit key
- Plan to set a deadline for the remaining with 1024-bit key

... and start using the latest(?) diracos2

DIRACOS2 on cvmfs

Imaginary "standard" client installation

- bash DIRACOS-Linux-$(uname -m).sh
- source diracos/diracosrc
- pip install VODIRAC==x.y.z (with VODIRAC dependency to DIRAC)
- then, after multiple installation of VODIRAC x.y.z1, x.y.z2, we may have the same diracos installation under multiple paths
  - /cvmfs/repo/.../VODIRAC/x.y.z1/$SYSTEM/diracos
  - /cvmfs/repo/.../VODIRAC/x.y.z2/$SYSTEM/diracos
- This may result in a possible inefficient use of cvmfs
  - Installing and publishing the same thing multiple times
  - Different jobs (running different VODIRAC versions) read the same libraries from different paths...

Non-all-in-one Installation

- DIRACOS2
  - cd /cvmfs/belle.kek.jp/grid/diracos2/2.31 && bash DIRACOS-Linux-$(uname -m).sh
  - source /cvmfs/belle.kek.jp/grid/diracos2/2.31/Linux-x86_64/diracos/diracosrc
- DIRAC
  - cd /cvmfs/belle.kek.jp/grid/DIRAC/8.0.44 && export PYTHONUSERBASE=$(pwd)/$system && pip install --prefix $PYTHONUSERBASE DIRAC==8.0.44
  - export PYTHONPATH=${DIRACROOT}/lib/python${PythonVersion}/site-packages:${PYTHONPATH}

Monitoring

WMS Monitoring / RMS Monitoring

- have been in production since last year
- We need to tune the parameters -- polling time, expiration

New Monitoring Types

- After v8 migration, we have started to store all Elasticsearch monitoring items listed in
- https://github.com/DIRACGrid/DIRAC/wiki/DIRAC-8.0#new-monitoring-types
- Recently we switched Matcher's share correction source from MySQL to Elasticsearch

Kibana/Grafana

- Validation instances setup
  - Some plots from Elasticsearch and others from MySQL
  - based on what is done in Dev LHCb
- A test Kibana setup for production instance , not open to everyone
- An official Kibana/Grafana to be setup for wider accesses

Soon we will drop MySQL WMSHistory, if no issue is observed.

Centralized logging

- No centralized logging has been utilized. It may come after all the work along with the KEKCC renewal

WMS

HTCondor CE

- We have started submitting jobs with SSL in production
  - to the CEs which has dropped GSI support
  - with a BelleDIRAC extension of '`HTCondorCE`' => '`HTCondorSSLCE`'
    - without hearing any needs in the other VOs
    - Now seeing DIRAC v8.0.47 has UseSSLSubmission option...
- We have successfully submitted pilot jobs with tokens
  - In our validation environment
    - First pilot submission within a half-day.
  - Tornado & TokenManager installed and pilot submission with tokens worked
    - TokenDB installed later for the Service ERRORs in the log...
    - TokenManager should not need TokenDB just for submitting pilots

ARC CE

- We have started switching CEType ARC => AREX
- Still need investigation (later slide)

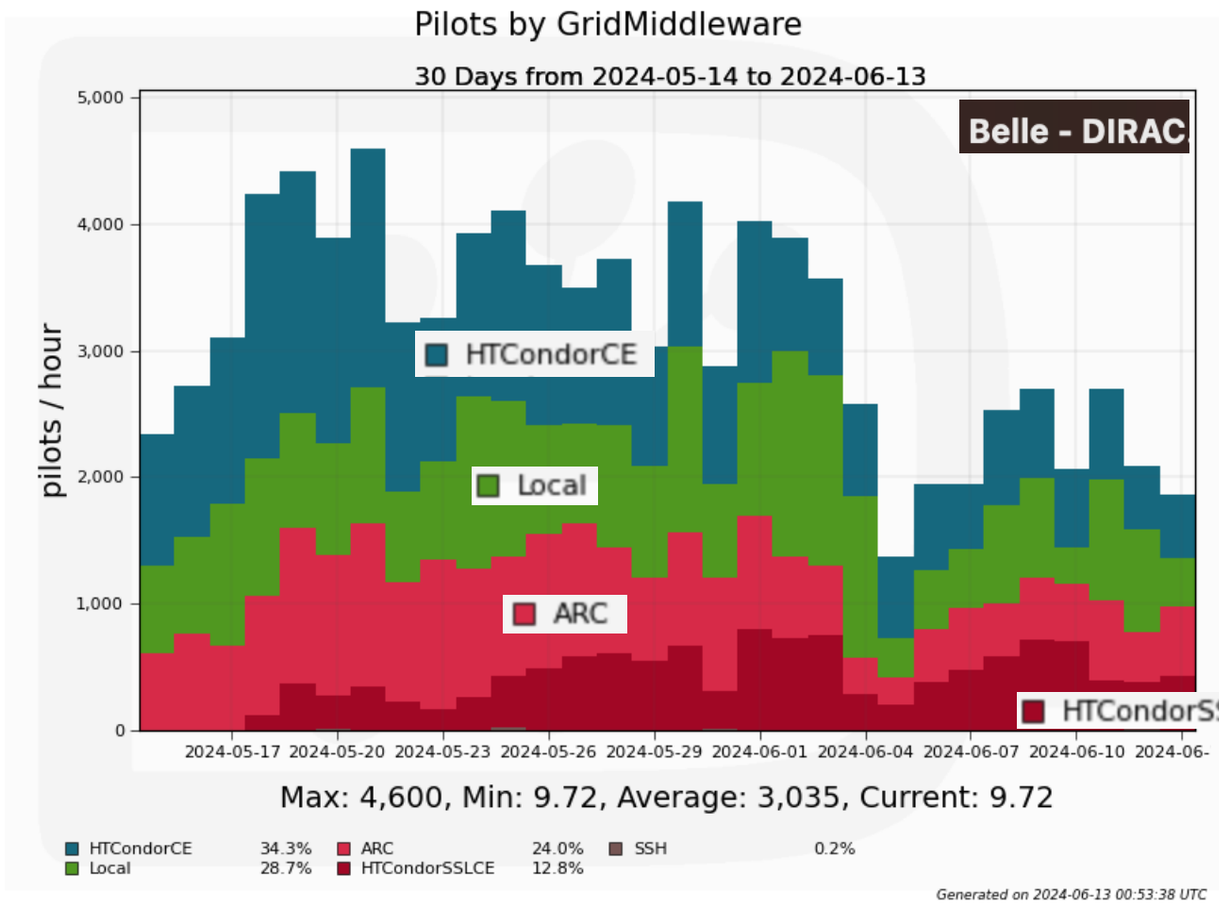
CE Types

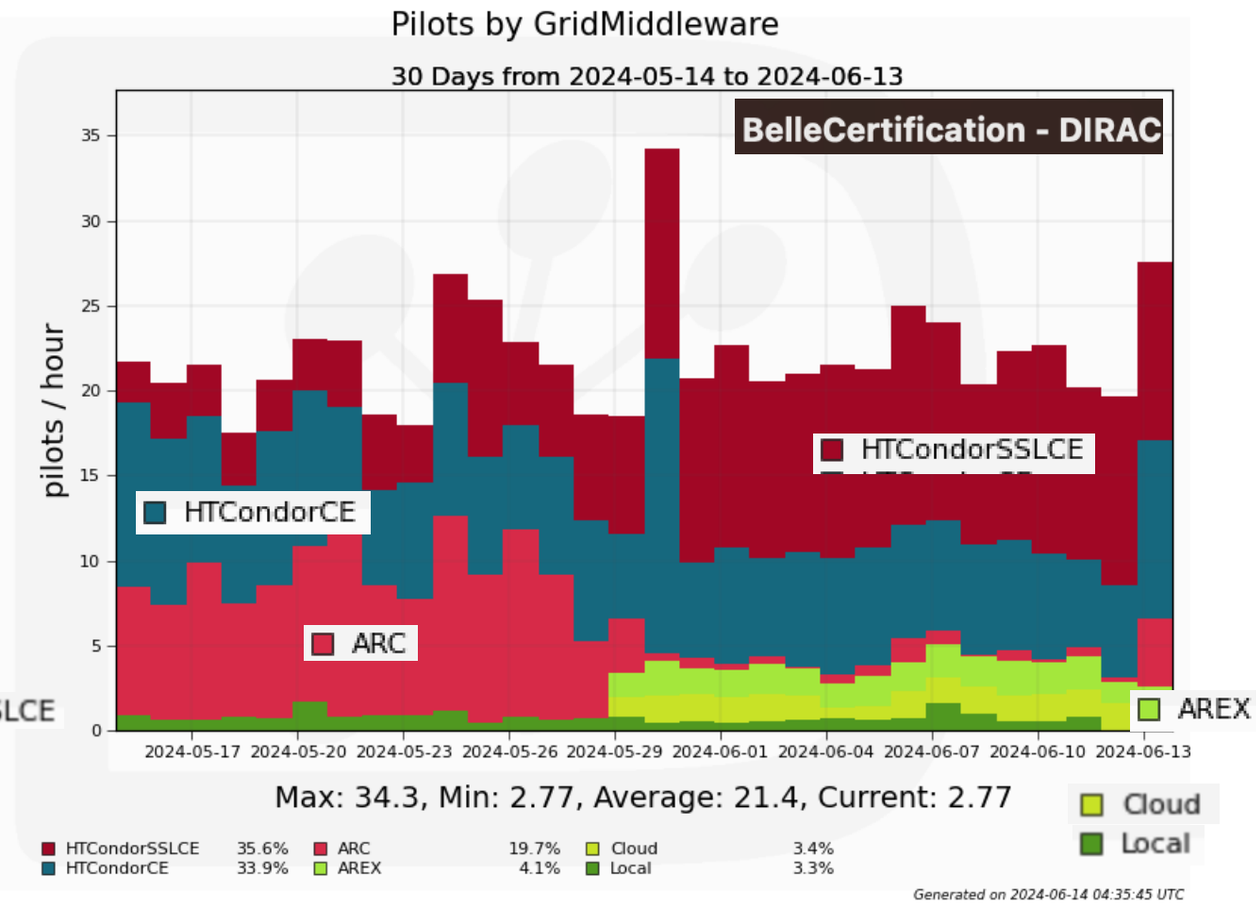
HTCondor-CE via SSL

- We have been testing SSL against some HTCondor-CEs with vanilla DIRAC HTCondorCE since last year (DIRAC v7.3)
  - impossible with v8.0.44 when we upgraded in May, though now possible with v8.0.47
- => Custom HTCondorSSLCE to enable SSL (see previous slide)

ARC => AREX

- Started validating AREX in the validation setup

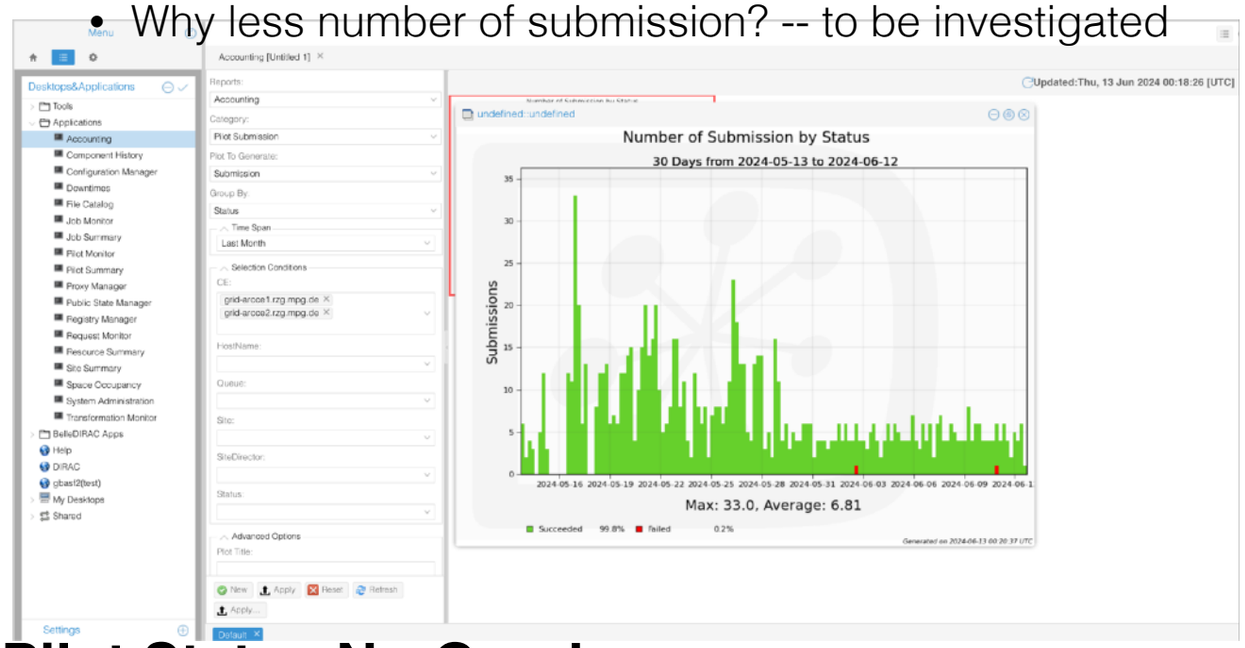




AREX

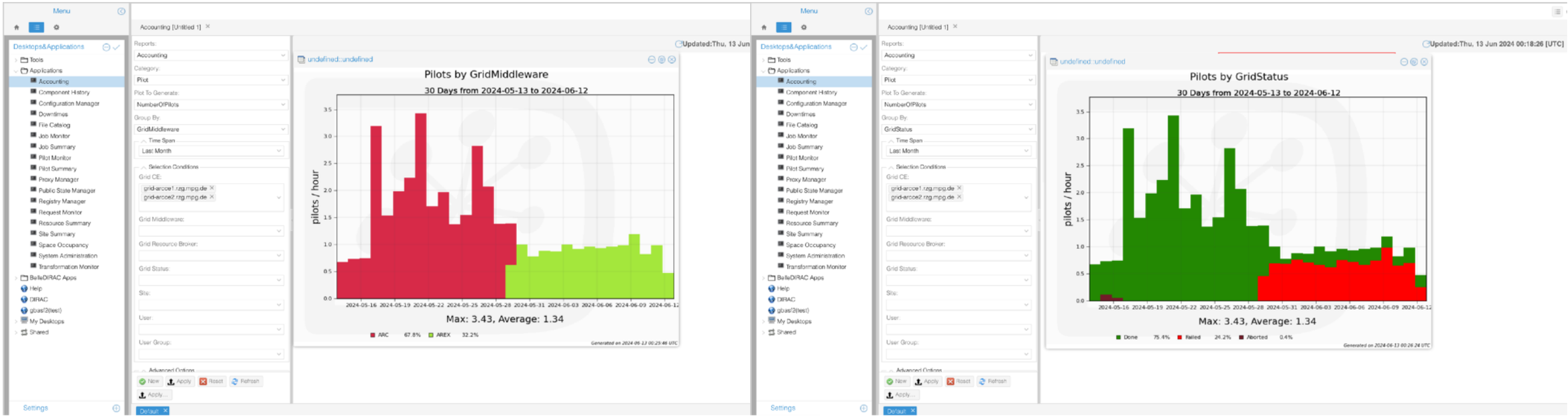
Pilot Submission OK-ish

- Why less number of submission? -- to be investigated



Pilot Status No-Good

- Why failing?



AREX?

Some sites may not be supporting AREX?

- OK

```
$ arcinfo -c grid-arcce2.rzg.mpg.de
Computing service: MPPMU (production)
  Information endpoint: ldap://grid-arcce2.rzg.mpg.de:2135/Mds-Vo-Name=local,o=grid (org.nordugrid.ldapng)
  Information endpoint: ldap://grid-arcce2.rzg.mpg.de:2135/o=glue (org.nordugrid.ldapglue2)
  Information endpoint: https://grid-arcce2.rzg.mpg.de:443/arex (org.nordugrid.arcrest)
  Information endpoint: https://grid-arcce2.rzg.mpg.de:443/arex (org.ogf.glue.emies.resourceinfo)
  Submission endpoint: https://grid-arcce2.rzg.mpg.de:443/arex (status: ok, interface: org.nordugrid.arcrest)
  Submission endpoint: https://grid-arcce2.rzg.mpg.de:443/arex (status: ok, interface: org.ogf.glue.emies.activitycreation)
  Submission endpoint: gsiftp://grid-arcce2.rzg.mpg.de:2811/jobs (status: ok, interface: org.nordugrid.gridftpjob)
```

- NG

```
$ arcinfo -c hpc.arnes.si
ERROR: LDAP query timed out: hpc.arnes.si
Computing service: hpc.arnes.si
  Information endpoint: ldap://hpc.arnes.si:2135/Mds-Vo-name=local,o=Grid (org.nordugrid.ldapng)
  Submission endpoint: gsiftp://hpc.arnes.si:2811/jobs (status: ok, interface: org.nordugrid.gridftpjob)

$ arcinfo -c pikolit.ijs.si
ERROR: Can't contact LDAP server (pikolit.ijs.si)
Computing service: pikolit.ijs.si
  Information endpoint: ldap://pikolit.ijs.si:2135/Mds-Vo-name=local,o=Grid (org.nordugrid.ldapng)
  Submission endpoint: gsiftp://pikolit.ijs.si:2811/jobs (status: ok, interface: org.nordugrid.gridftpjob)

$ arcinfo -c arcce03.esc.qmul.ac.uk
ERROR: LDAP query timed out: arcce03.esc.qmul.ac.uk
Computing service: arcce03.esc.qmul.ac.uk
  Information endpoint: ldap://arcce03.esc.qmul.ac.uk:2135/Mds-Vo-name=local,o=Grid (org.nordugrid.ldapng)
  Submission endpoint: gsiftp://arcce03.esc.qmul.ac.uk:2811/jobs (status: ok, interface: org.nordugrid.gridftpjob)
```

Cloud Resources

Cloud Scheduler

- The main computing resources on clouds are provided from U.Victoria (Canada) as pledged+opportunistic resources
- UVic runs their own "Cloud Scheduler"
  - for ATLAS and Belle II
- CEType = Local

CloudComputingElement

- CEType = Cloud
- We started trying with it after the DR23

Multi-Core Jobs

We plan to run multi-core jobs

- Some raw data processing jobs take >24h for larger input files
  - multi-core jobs helps
- Current (single-core) MCProduction jobs produce very small files
  - => multiple merge steps
  - 8-core jobs will produce/process 8x more events within the same duration
  - => less merge factor

Multi-core pilots to run multi-core jobs

- Existing mechanism in DIRAC seems very flexible
  - To run n-core jobs in m-core pilots (1 <= n <= m)
  - Probably with principal aim running single-core jobs in 8-core queues?
- We just want to run multi-core jobs in multi-core queues
  - Using Tag = MultiCore and RequiredTag = MultiCore
  - With a fixed and the same number of cores for pilot and payload jobs
    - currently 8, following the "standard" -- the standard in WLCG may change

What can be included in vanilla DIRAC

RucioFileCatalogClient metadata APIs

- The current vanilla RucioFileCatalogClient has LFC-like APIs for file names, attributes, and replica locations
- BelleDIRAC RucioFileCatalogClient has now **metadata APIs**
  - **Pull request #7383** for vanilla yet to be updated

Scout Job

- Working on https://github.com/DIRACGrid/DIRAC/pull/7251
  - **Executor/Scouting.py** to change the Job Status, so that they would be held until "scouting" finishes
  - **Agent/ScoutingJobStatusAgent.py** to change the Job Status to Waiting or Failed depending on the result of "scouting"
- Client tool to define scout jobs not included
  - One needs to define what to do in "scouting"
  - what we have is too Belle II specific
    - We duplicate the job script and reduce the number of events to process

RucioClient

- A set of interfacing methods to utilize Rucio client APIs
  - in extension of DMS

Outlook

In the coming weeks

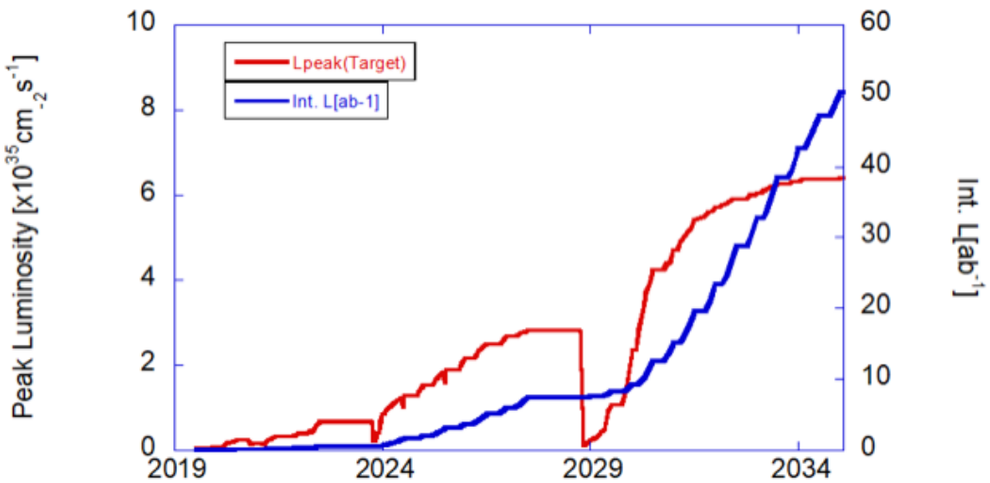
- Get rid of 1024-bit proxies and move to a newer diracos2
- Start using IAM with DIRAC in production
- More HTCondor-CEs with SSL/token (at least in validation setup)
- More ARC-CEs with AREX (at least in validation setup)
  - Start using AREX against ARC-CEs in production
- More tape SEs with REST API (at least in validation setup)
  - Start using REST against tape SEs in production

In the coming months

- Complete the switch of metadata catalog: AMGA => Rucio
- Switch-over to the new nodes in the new KEKCC (and CentOS7 => EL9)
- Elasticsearch => OpenSearch (hopefully)

In the further future

- More unit tests, CI, ...
- Analysis jobs with Transformation?
- Much more data to be managed
- Much more jobs to run



https://www.belle2.org/research/luminosity/

BelleDIRAC

To run productions and analysis

DIRAC version

- Production: **diracos2.31** + **v8.0.44**
- Validation: diracos2.31 + v8.0.44 => to be updated to the latest for the next release

Servers

- 5 main DIRAC + 4 MySQL + 3 Elasticsearch + 2 WebApp servers at KEK
  - can keep running during the annual power maintenance
- 1 DIRAC server with MySQL at BNL
  - to run Rucio-related DIRAC components
- 6 additional DIRAC servers at KEK
  - to run components that can be down during the annual power maintenance
- 2 additional DIRAC servers at the universities (Canada, Japan)
  - to run special SiteDirectors
- Test servers
  - Certification: validation of new BelleDIRAC releases
  - Migration: to test BelleDIRAC against upgrades of base DIRAC
  - Developments: multiple instances at KEK, BNL, universities

Clients

- Installed on **cvmfs**

BelleRawDIRAC

To upload and register raw data files to Grid

- A separate and independent system from BelleDIRAC (Yet another VO extension)
  - Independent from deployment plans and possible troubles of BelleDIRAC
  - Independent choice of base DIRAC possible
- Now also to transfer raw data from online storage to offline storage
  - Online storage as XRootD SE
- Replication is done by Rucio
  - BelleRawDIRAC to monitor the replication and verify the replicas

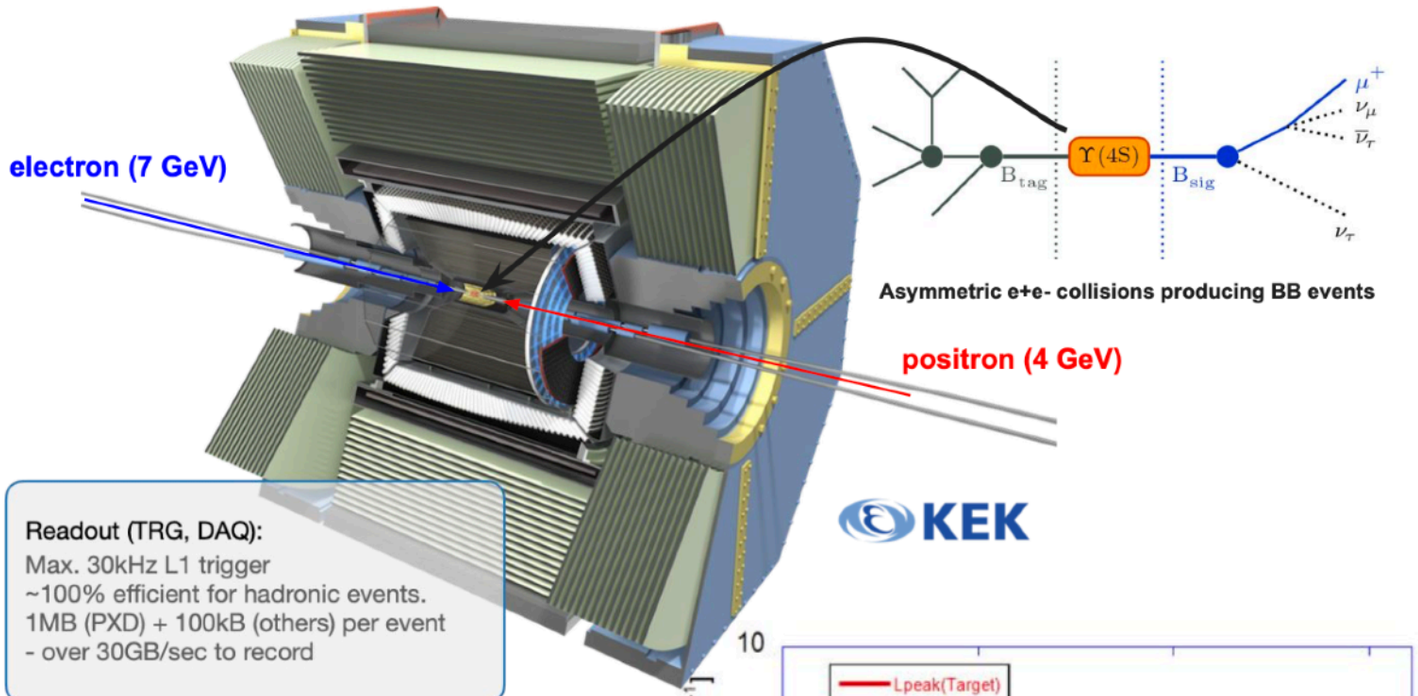
DIRAC version

- Production: diracos**2.31**+**v7.3.38** => to be upgraded during the summer shutdown
- Validation: diracos**2.38**+**v8.0.44**

Servers

- 2 main DIRAC servers for redundancy + 1 test server at KEK
  - To register raw data files and verify their replication to the Raw Data Centers
- 21 servers sub-divided into 3 groups to be assigned to the above
  - To run a set of Agents to transfer raw data from online storage to offline storage
  - And another set of Agents to upload the files to the Grid SE

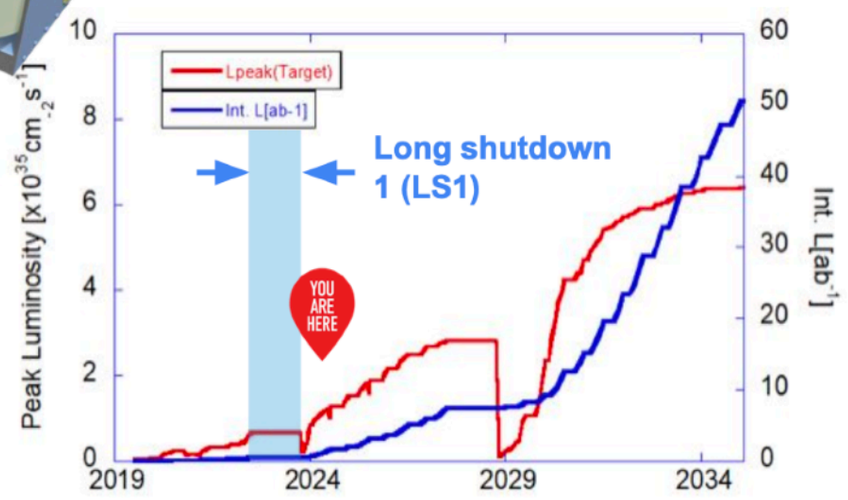
Belle II in a Nutshell



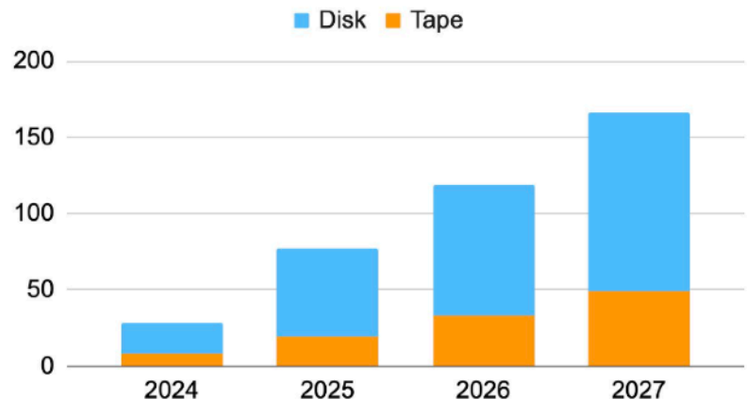
Readout (TRG, DAQ):
Max. 30kHz L1 trigger
~100% efficient for hadronic events.
1MB (PXD) + 100kB (others) per event
- over 30GB/sec to record

KEK

- Int L = 50 ab^{-1} at the end of the experiment **(x50 than the previous B factories)**



- In the high-luminosity scenario, the size of the dataset is **~ O(10) PB/year**



Offline computing:
Distributed over the world via grid.