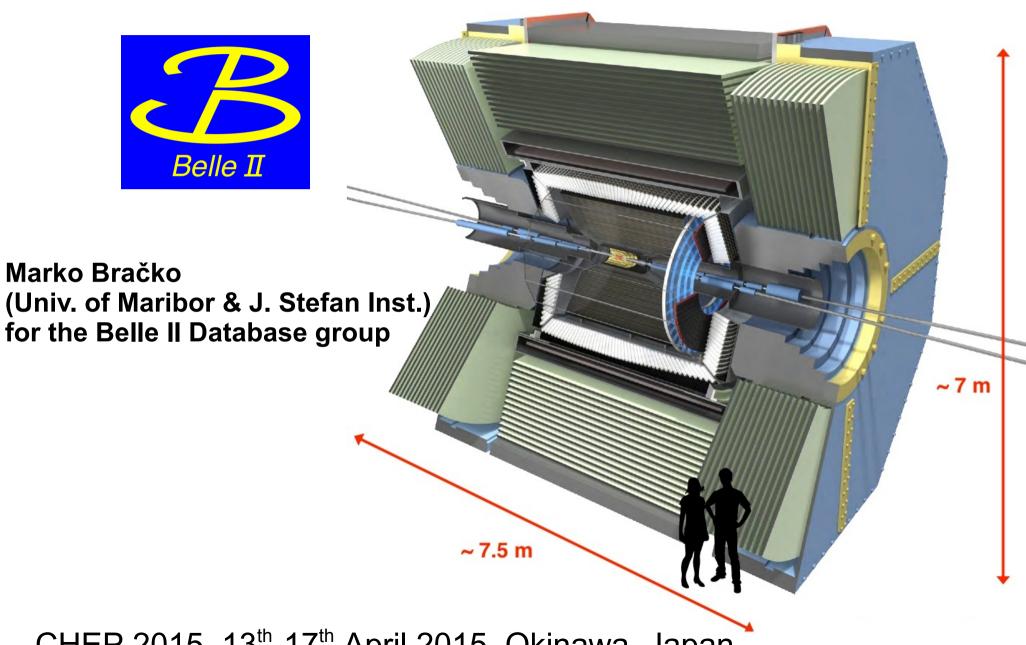
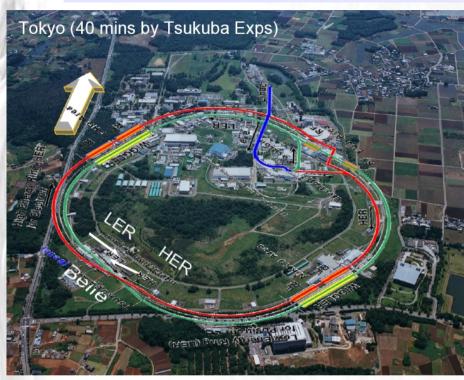
The Belle II Conditions Database



CHEP 2015, 13th-17th April 2015, Okinawa, Japan

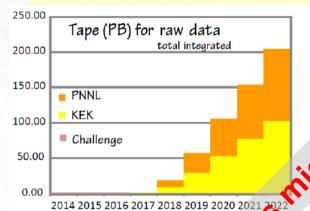
The Belle II experiment

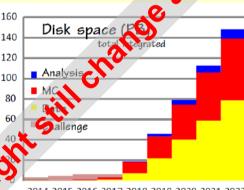




The Belle II experiment is a Super B Factory experiment, the successor of B Factorie(s):

- → Physics motivation is complementary to the LHC experiments: Belle II will search for physics beyond the SM, exploring quark flavour physics, tau physics and CP violation at the Intensity Frontier (precision measurements with high-intensity beams).
- → Total integrated luminosity: 50 ab⁻¹ (50-times Belle data).
- → Operation starts in October 2018.
- → Similar data rate as LHC experiments (and comparable in total requirements)!





2014 2015 2016 2017 2018 2019 2020 2021 2022

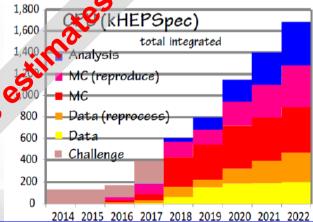
KEK laboratory, Tsukuba, Japan



Belle II collaboration:

~ 600 members;

~ 100 institutions from 23 countries/region



Data flow, analysis scheme



Focus of DAQ group

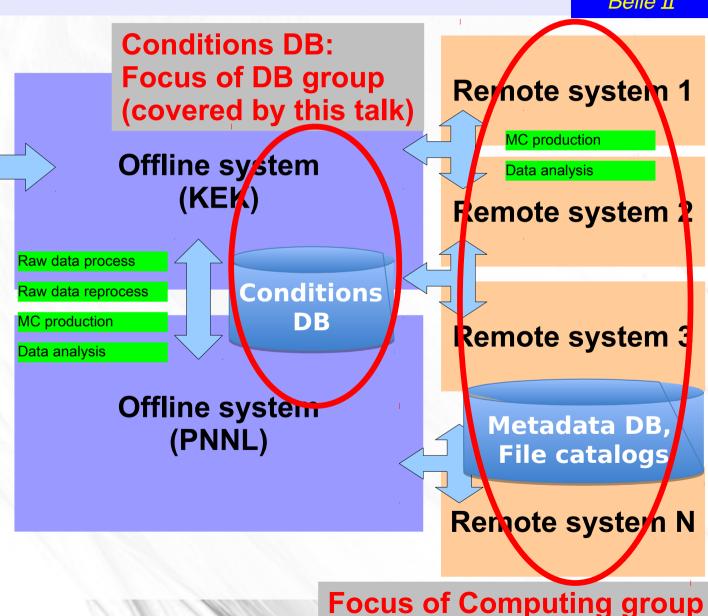
Online system (KEK)

Online database (Configuration DB, Logger DB)

Main DB types:

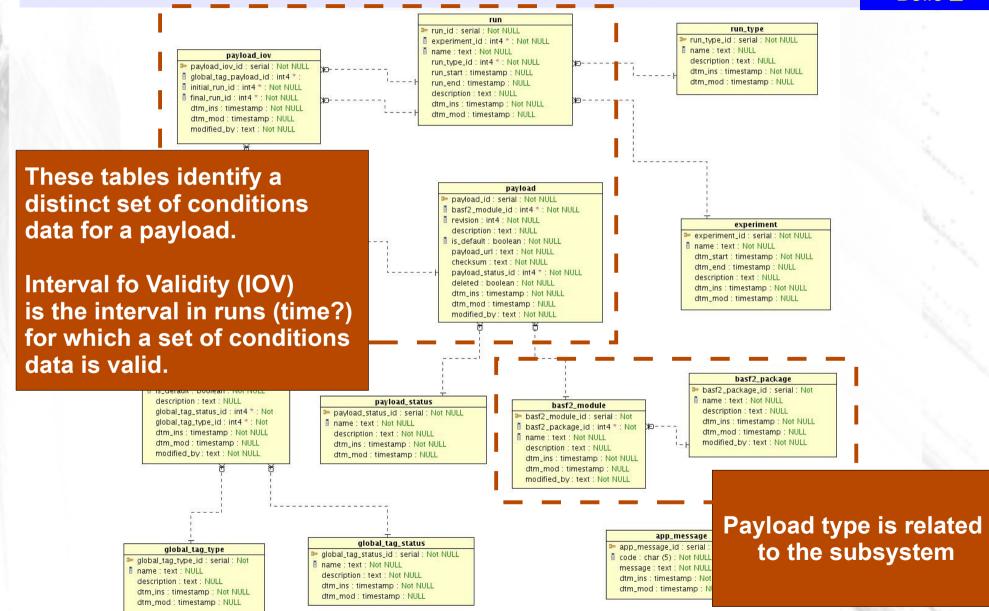
- Configuration DB
- Logger DB
- Conditions DB
- Metadata DB

• ...



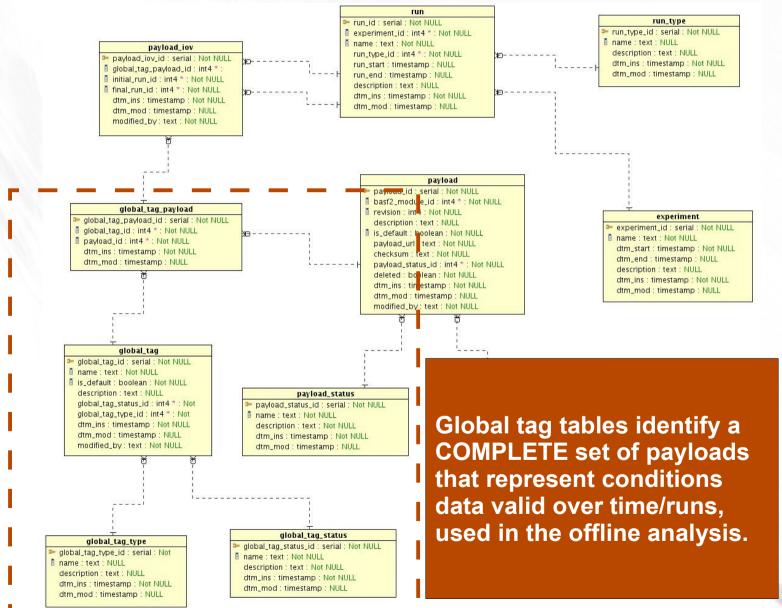
Conditions DB Schema





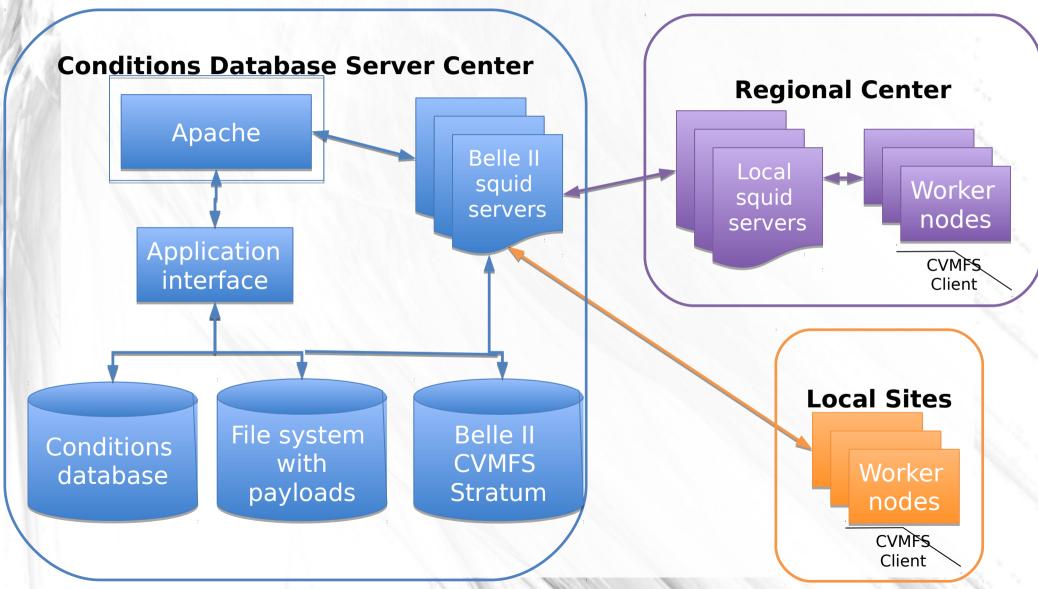
Conditions DB Schema





Distributed Conditions DB design





Distributed Conditions DB design



Conditions Database Server Center

Apache

Relle II

PostgreSQL database

Interface through Apache application interface on server. Generalized schema makes the interface simple (few required methods).

DB payloads contain **reference to file location on file system** (not the data itself).

Conditions database

File system with payloads

Belle II CVMFS Stratum

Regional Center

Local

Worker

File system for payloadsPayloads saved as files on file system.

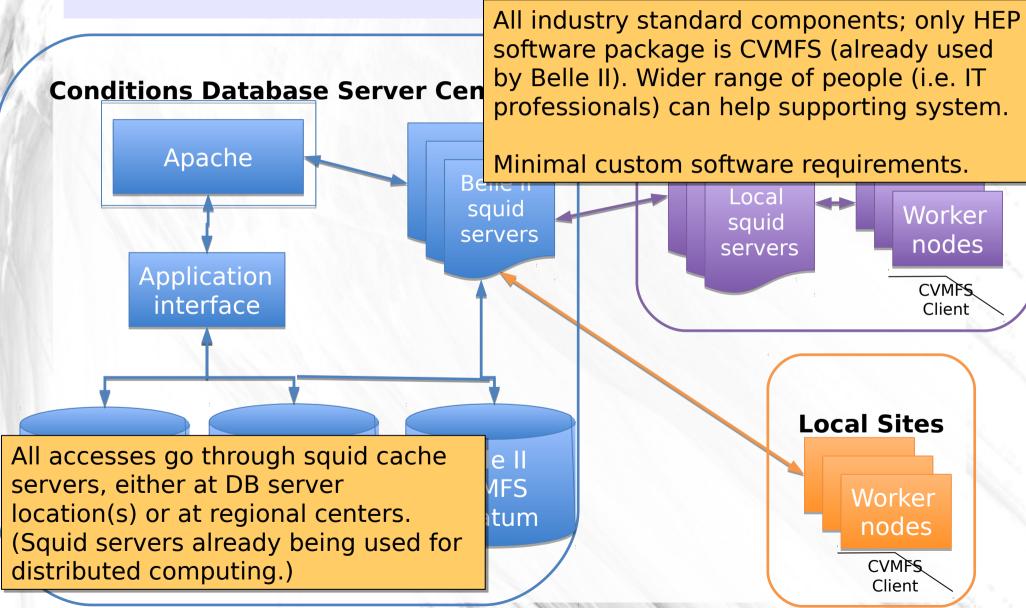
CVMFS used with squid servers to serve files when requested from nodes. (CVMFS client at worker nodes is required!)

Worker nodes

CVMFS.

Distributed Conditions DB design





Rest API (Application interface)



REST communicates via HTTP verbs (GET, POST, ...)

Global Tag

- GET /globalTag/{globalTagld}
- GET /globalTag/{globalTagName}
- ▶ GET /globalTags
- POST /globalTag/{globalTagType}
- GET /globalTagType
- GET /globalTagStatus

Experiment / Run

GET /experiments

With GET /globalTag one gets all the needed references

Payload

GET /payload/{payloadId}

GET /globalTag/{GlobalTagId}/payloads

GET /globalTag/{GlobalTagName}/payloads

GET /payloads

GET /payloads?gtName=<name>&expName=<name> &runName=<name>

POST /module/{moduleId}/payload

GET payload (name, experiment, run)



GET /payloads?gtName=<name>& expName=<name>&runName=<name>

Method GET

Description Returns all payloads if no query parameters specified.

Returns all payloads for a global tag, experiment, and run (all specified by name), if the query parameters are provided. All values are required. Only payloads associated with the global tag and having an IOV for the provided experiment and run will be returned.

Role Access AllDependencies

Parameter(s) gtName - the global tag name

expName - the experiment name

runName - the run name

Example Request

GET http://<domain>:<port>/b2s/rest/v1/payloads? gtName=TEST_GT1&expName=EXP_1&runName=RUN_1

Example Response

```
"payloadId": 1,
"checksum": "checksum".
"deleted": false.
"dtmlns": "2015-02-16T10:34:13.591717-08:00",
"dtmMod": "2015-02-16T10:34:13.591717-08:00",
"isDefault": true.
"modifiedBy": "adm cond",
"payloadUrl": "http://cvfms.org/pl1",
"revision": 1.
"payloadStatus": {
 "payloadStatusId": 2,
 "description": "VALID description",
 "dtmlns": "2015-02-16T10:34:13.574965-08:00".
 "dtmMod": "2015-02-16T10:34:13.574965-08:00".
 "name": "VALID"
"payloadlov": {
 "payloadlovId": 1,
 "dtmlns": "2015-02-16T10:34:13.633415-08:00",
 "dtmMod": "2015-02-16T10:34:13.633415-08:00".
 "modifiedBy": "adm cond",
 "payloadId": 1,
 "initialRunId": 1,
 "finalRunId": 1
```

POST payload



POST /module/{moduleId}/payload

Method POST

Description Creates a new payload

Role Access All

Dependencies payloadld and revision field values in the request will be ignored, both are generated by the service and their values will be returned in the response.

Parameter(s) moduleId - Integer value, identifying the basf2 module

Example Request

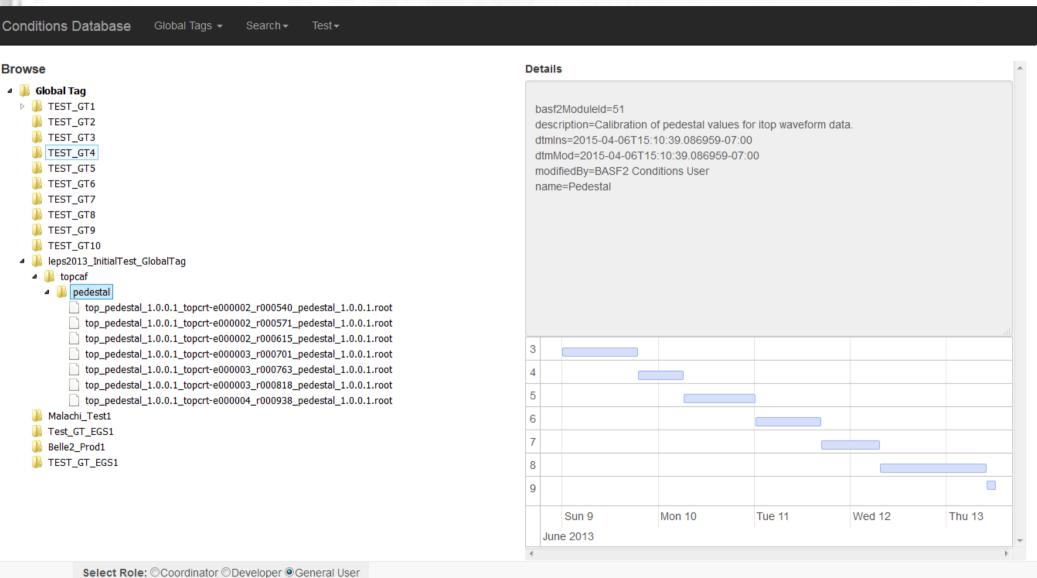
```
{ "payloadId": 1,
    "checksum": "checksum",
    "deleted": false,
    "dtmIns": "2015-02-11T15:59:43.953156-08:00",
    "dtmMod": "2015-02-11T15:59:43.953156-08:00",
    "isDefault": true,
    "modifiedBy": "adm_cond",
    "payloadUrl": "http://cvfms.org/pl1/NEW",
    "revision": 23 }
```

Example Response

```
"payloadId": 52,
"checksum": "checksum",
"deleted": false.
"dtmlns": "2015-02-16T12:17:34.934-08:00",
"dtmMod": "2015-02-16T12:17:34.934-08:00",
"isDefault": true.
"modifiedBy": "adm cond",
"payloadUrl": "http://cvfms.org/pl1/NEW",
"revision": 3,
"payloadStatus": {
 "payloadStatusId": 1,
 "description": "NEW description",
 "dtmlns": "2015-02-16T11:01:27.509082-08:00",
 "dtmMod": "2015-02-16T11:01:27.509082-08:00",
 "name": "NEW"
"basf2ModuleId": 1
```

GUI snapshot





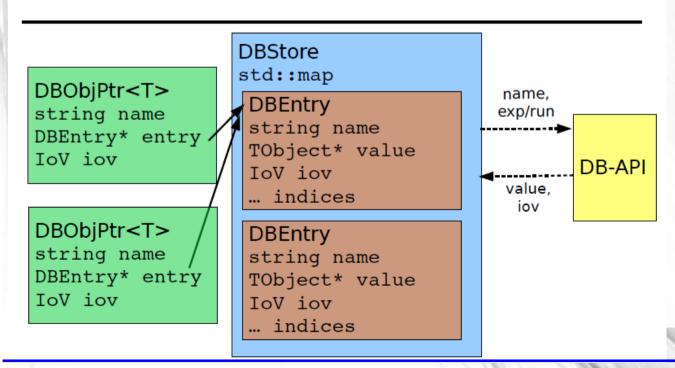
Database/BASF2 Interface



High-level Database interface:

- DB Access functions are integrated in the BASF2 (Belle II Analysis Framework);
 provide transparent and relatively easy usage of database for all users;
- Database payloads (associated to user-specified names) are Tobjects;
- Users should be able to change the specific payload using BASF2 steering files by changing DB mapping;
- Database Store object is used (similar as Datastore is used for the event data)

Database Store



Test system set-up (at PNNL)



- Goal: end-to-end test of components
 - Local cloud instances of various servers
- PostgreSQL
 - PostgreSQL database in place
- Apache server
 - Installed (no issues)
- DB server Application Interface
 - Finalized this month
- Squid server
 - In place this month
- Prototype system
 - Ready at PNNL in April 2015 (with GUI and CVMFS)
 - Release candidate in late April 2015
 - Similar systems then set up at additional sites (KEK, JSI-Ljubljana, ...)
 - Release for early adopters at May 2015 Belle II Computing/Software Workshop
 - Further progress by the June 2015 Belle II General Meeting

Plans



• <u>2015</u>:

REST API:

- Developed in 2014/early 2015
- Further design modifications & applications implemented

BASF2 interface:

- Recently implemented; back-end still missing; Tests, further development during 2015
- Ensure that good communication between online & offline DB is achieved (toghether with DAQ and computing group)

Distributed database system:

- Get the prototype system ready at PNNL
- Perform tests and optimizations using replicated systems at KEK and JSI
- Release candidate should be ready in late April
- · Release with the documentation ready for the May WS

Software&Computing Workshop in May:

- Evaluate the release and prepare it for early adopters
- Prepare/update the documentation
- Training session planned (another also at the June B2GM)

• Distributed computing usage:

- Test the DB system with distributed computing system before/during summer
- Modify/optimize the system based on performance tests
- Prepare CondDB v1.0 release candidate to be ready in late 2015 for production running (not optimized vet)