Database

Tomoyuki Konno

TRG/DAQ workshop

2017/08/25, NTU, Taipei

DAQ database

- Relational database server is used to store configuration and logs
  - PostgreSQL 9.3 on 10 TB RAID6 storage
  - Connected to daqnet and b2nsm/b2epics networks
- **Configuration DB** gives configuration parameters
  - Text parser to objects / DB parser to objects
  - An instruction for DAQ DB is available for Belle II collaborators: https://confluence.desy.de/display/BI/DAQ+database
  - **Run record** is a kind of configuration DB table updated at run start/end
- **Logger DB** shows histories of detector status and DAQ activities
  - Text messages via NSM2 are stored into database
- Archiver for monitored values => covered by S. Park
  - Conversion of NSM2 into EPICS Archiver Appliance

Configuration DB

- Configuration DB is available using the command line tools below:
  - Text -> DB : **daqdbcreate** <filepath> <tablename>
  - DB -> text (stdout) : **daqdbget** <tablename> <configname>
- Running in global/local run control, HV control, some detector operations
  - Run record is also stored as same db table format

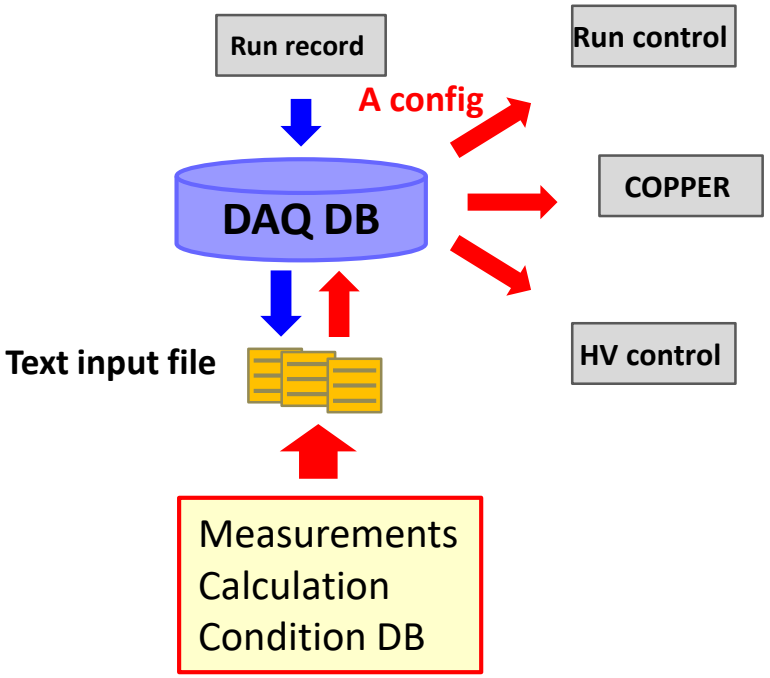
```
#

config      : cdc:fee:cpr2075:a:suppress:045

firm        : recbe_v52_170213.bit
mode        : suppress
delay.val   : 139
window.val  : 21
tdcth.val   : 3750
adcth.val   : 2
ped[0].val  : 220
ped[1].val  : 219
ped[2].val  : 218
ped[3].val  : 221
...
ped[47].val : 230

#
```

configuration
for a CDC FEE
(suppressed mode)



Relation DB and slow control (ECL)

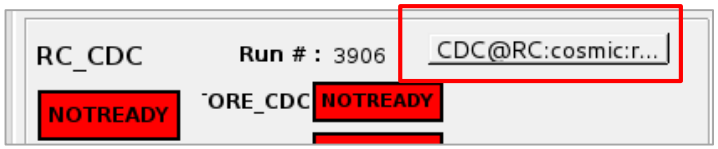
```
config : ecl:col:cpr5018:b:2017:08:18:04

sh_mask                : 4095
reg_num                : 544
reg_wdata              : 48
shaper_mask_low        : 255
shaper_mask_high       : 15
ttd_trg_rare_factor    : 97
ttd_trg_type           : 49
calib_ampl0_low        : 0
calib_ampl0_high       : 8
calib_ampl_step_low    : 0
calib_ampl_step_high   : 0
calib_delay0_low       : 9
calib_delay0_high      : 0
calib_delay_step_low   : 0
calib_delay_step_high  : 0
calib_events_per_step  : 0
sh_data[0].sh_num      : 4095
sh_data[0].reg_num     : 547
sh_data[0].reg_wdata   : 31
sh_data[1].sh_num      : 4095
sh_data[1].reg_num     : 544
sh_data[1].reg_wdata   : 48
sh_data[2].sh_num      : 4095
sh_data[2].reg_num     : 64
sh_data[2].reg_wdata   : 0
sh_data[3].sh_num      : 4095
sh_data[3].reg_num     : 545
sh_data[3].reg_wdata   : 0
reg30[0].val           : 61
reg30[1].val           : 57
```

```
void ECLFEE::load(RCCallback& callback, HSLB& hslb, const DBObject& obj)
{
  callback.log(LogFile::INFO, "Load ECL config");
  const DBObjectList o_sh_datas(obj.getObjects("sh_data"));
  for (size_t i = 0; i < o_sh_datas.size(); i++) {
    const DBObject& o_sh_data(o_sh_datas[i]);
    unsigned int sh_num  = o_sh_data.getInt("sh_num");
    unsigned int reg_num = o_sh_data.getInt("reg_num");
    unsigned int reg_wdata = o_sh_data.getInt("reg_wdata");
    rio_sh_wreg(callback, hslb, sh_num, reg_num, reg_wdata);
  }
  hslb.writefee8(0x20, obj.getInt("shaper_mask_low"));
  hslb.writefee8(0x21, obj.getInt("shaper_mask_high"));
  hslb.writefee8(0x38, obj.getInt("ttd_trg_rare_factor"));
  hslb.writefee8(0x39, obj.getInt("ttd_trg_type"));
  hslb.writefee8(0x40, obj.getInt("calib_ampl0_low"));
  hslb.writefee8(0x41, obj.getInt("calib_ampl0_high"));
  hslb.writefee8(0x42, obj.getInt("calib_ampl_step_low"));
  hslb.writefee8(0x43, obj.getInt("calib_ampl_step_high"));
  hslb.writefee8(0x45, obj.getInt("calib_delay0_low"));
  hslb.writefee8(0x45, obj.getInt("calib_delay0_high"));
  hslb.writefee8(0x46, obj.getInt("calib_delay_step_low"));
  hslb.writefee8(0x47, obj.getInt("calib_delay_step_high"));
  hslb.writefee8(0x48, obj.getInt("calib_events_per_step"));
  const DBObjectList o_reg30s(obj.getObjects("reg30"));
  for (size_t i = 0; i < o_reg30s.size(); i++) {
    const DBObject& o_reg30(o_reg30s[i]);
    unsigned int val = o_reg30.getInt("val");
    hslb.writefee8(0x30, val);
  }
}
```

Switching DAQ configuration

- DAQ configuration can be reloaded from DB by CONFIGURE
  - Run control has a nested scheme to replace configuration
  - nsm://set:<RC>:config in CSS
    or
    $ rcrequest <RC> CONFIGURE in command line
- CDC case:
  - suppress:... for usual run
  - raw:... for calibration




```
nodename              : CDC01
config                : RC:suppress:2017:06:14:02

node[0].name          : ROPC201
node[0].rcconfig      : suppress
node[1].name          : CPR2001
node[1].rcconfig      : suppress
node[2].name          : CPR2002
node[2].rcconfig      : suppress
node[3].name          : CPR2003
node[3].rcconfig      : suppress
...
```

Load latest config. with "RC:suppress:..."

```
nodename              : CDC01
config                : RC:raw:2017:06:14:02

node[0].name          : ROPC201
node[0].rcconfig      : raw
node[1].name          : CPR2001
node[1].rcconfig      : raw
node[2].name          : CPR2002
node[2].rcconfig      : raw
node[3].name          : CPR2003
node[3].rcconfig      : raw
...
```

Load latest config. with "RC:raw:..."

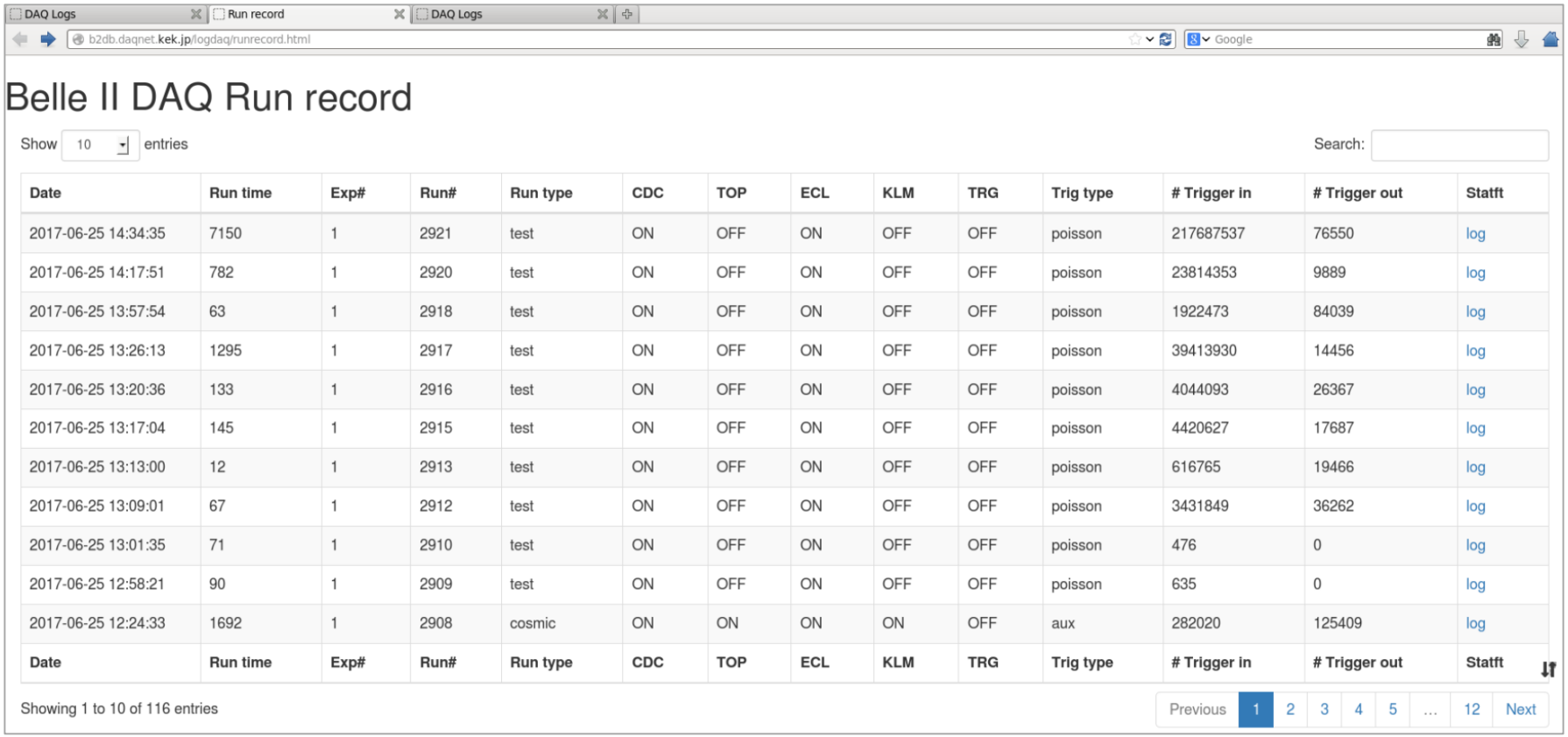
Run record

- Run record stores NSM vars at run start and end into database
  - Editable variable list
    - var : <nsm node>@<var name>
    - name : name in database
- Currently global run record are stored
  - Detector enabled / disabled
  - Run type (manually edited)
  - Run start time
  - Trigger count (in/out)
  - Statft (FTSW status summary)
- Easily extends to local run control
  - Detector run control can be added new run records

configuration of run record input

```
#
vars[0].var   : RUNCONTROL@cdc.used
vars[0].name  : RC.CDC.used.I
vars[1].var   : RUNCONTROL@top.used
vars[1].name  : RC.TOP.used.I
vars[2].var   : RUNCONTROL@ecl.used
vars[2].name  : RC.ECL.used.I
vars[3].var   : RUNCONTROL@klm.used
vars[3].name  : RC.KLM.used.I
vars[4].var   : RUNCONTROL@trg.used
vars[4].name  : RC.TRG.used.I
vars[5].var   : STORAGE@record.runtype
vars[5].name  : STORAGE.runtype.S
vars[6].var   : TTD@ftstate
vars[6].name  : TTD.state
vars[7].var   : TTD@statft
vars[7].name  : TTD.statft
vars[8].var   : TTD@tstart
vars[8].name  : TTD.run.tstart
vars[9].var   : TTD@trun
vars[9].name  : TTD.run.length
vars[10].var  : TTD@tincnt
vars[10].name : TTD.trig.count.in
vars[11].var  : TTD@toutcnt
vars[11].name : TTD.trig.count.out
vars[12].var  : TTD@trigger_type
vars[12].name : TTD.trig.type
#
```

Run record page



Belle II DAQ Run record

Show 10 entries Search:

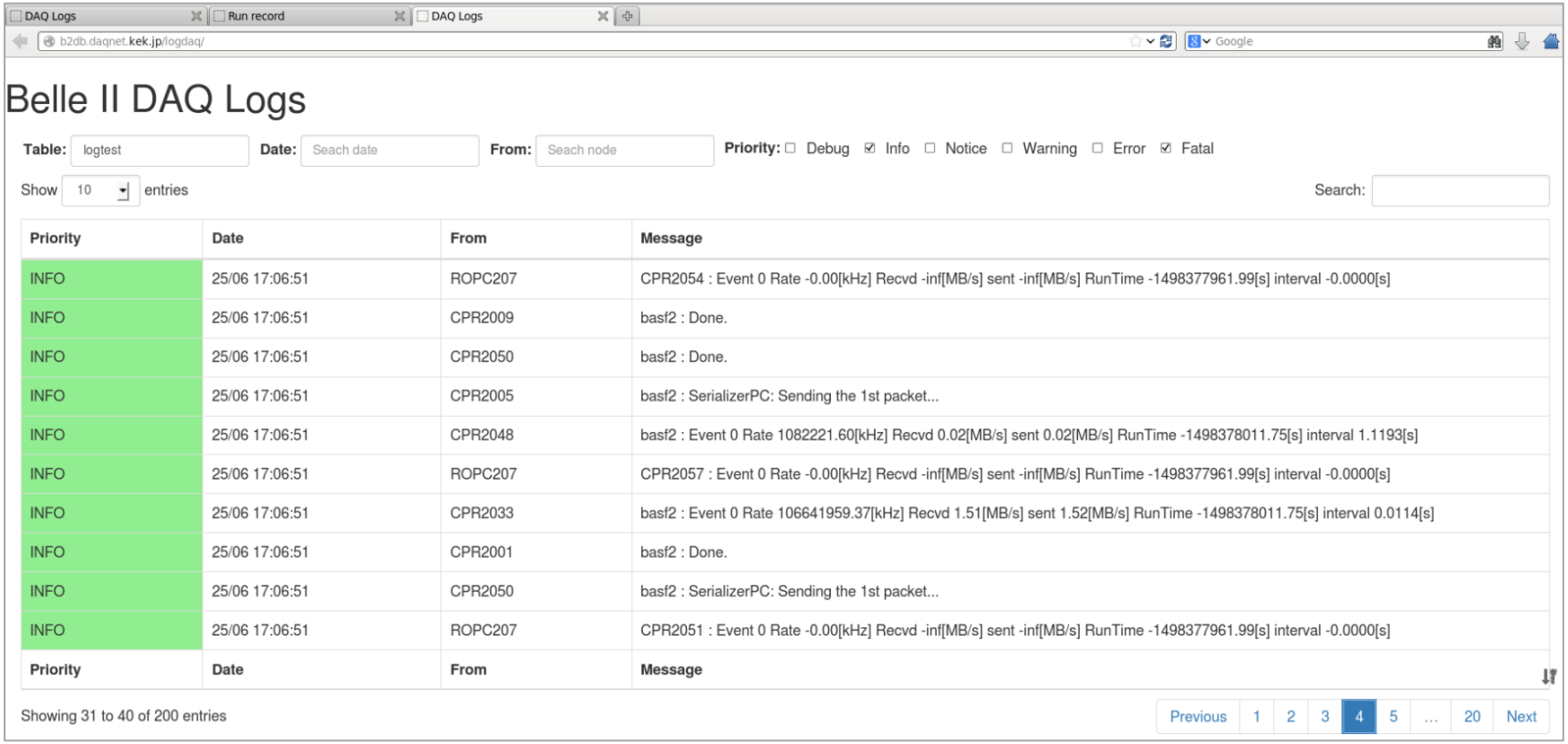
Date	Run time	Exp#	Run#	Run type	CDC	TOP	ECL	KLM	TRG	Trig type	# Trigger in	# Trigger out	Statft
2017-06-25 14:34:35	7150	1	2921	test	ON	OFF	ON	OFF	OFF	poisson	217687537	76550	log
2017-06-25 14:17:51	782	1	2920	test	ON	OFF	ON	OFF	OFF	poisson	23814353	9889	log
2017-06-25 13:57:54	63	1	2918	test	ON	OFF	ON	OFF	OFF	poisson	1922473	84039	log
2017-06-25 13:26:13	1295	1	2917	test	ON	OFF	ON	OFF	OFF	poisson	39413930	14456	log
2017-06-25 13:20:36	133	1	2916	test	ON	OFF	ON	OFF	OFF	poisson	4044093	26367	log
2017-06-25 13:17:04	145	1	2915	test	ON	OFF	ON	OFF	OFF	poisson	4420627	17687	log
2017-06-25 13:13:00	12	1	2913	test	ON	OFF	ON	OFF	OFF	poisson	616765	19466	log
2017-06-25 13:09:01	67	1	2912	test	ON	OFF	ON	OFF	OFF	poisson	3431849	36262	log
2017-06-25 13:01:35	71	1	2910	test	ON	OFF	ON	OFF	OFF	poisson	476	0	log
2017-06-25 12:58:21	90	1	2909	test	ON	OFF	ON	OFF	OFF	poisson	635	0	log
2017-06-25 12:24:33	1692	1	2908	cosmic	ON	ON	ON	ON	OFF	aux	282020	125409	log
Date	Run time	Exp#	Run#	Run type	CDC	TOP	ECL	KLM	TRG	Trig type	# Trigger in	# Trigger out	Statft

Showing 1 to 10 of 116 entries Previous 1 2 3 4 5 … 12 Next

http://b2db.daqnet.kek.jp/logdaq/runrecord.html

- Information in slow control network is recorded at run start/stop
- Selected records at run end are summarized into the web page

DAQ log page



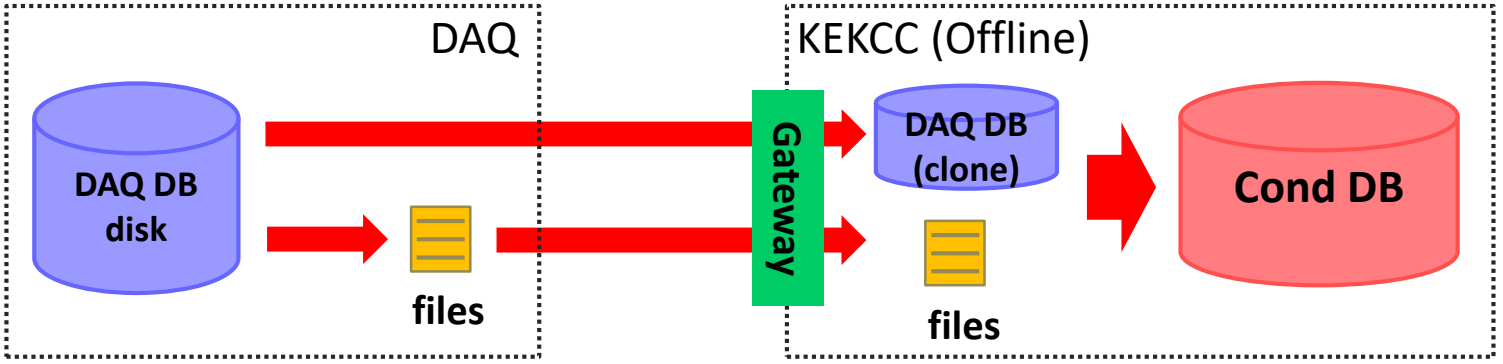
- Text logs are collected from slow control processes via NSM2
- Web interface with PHP + Bootstrap
  - Search using Node name, Priority, (Date), (Messages)
- Masking messages according to log level

DAQ Log level

- There are 6 log levels:
  - DEBUG : debugging info (not shown by default)
  - INFO : information showing system running well
  - NOTICE : also good information but need to confirm (e.g. run start)
  - WARNING : error message but run can continue
  - ERROR : error message to stop run
  - FATAL : critical error message to be sent to expert
- I'm now reducing # of messages to improve shifter's convenience
  - Too many event counters misses sometime important messages
  - HLT software should be more clear for reasons of trouble
- Some detector (TOP) starts sending log messages but a bit too many
  - Called FEE interface class (TOPFEE) in COPPER slow control
  - Maeda-san will tune threshold of warning messages

Porting DAQ DB to offline

- Replication of DAQ DB to Offline
  - DB contents are pulled from offline database
  - DAQ accepts postresql (5432) port from KEKCC
  - Config DB is converted into Cond DB by database group
- Transport **dumped files** for monitoring
  - http (80) access to get files by wget or curl
  - KEKCC side downloads files periodically (one per day?)
- **A dedicated IP in KEK network is assigned to DAQ DB server**
  - Thanks to Yamagata-san
  - Transfer scheme is not established yet, need contribution from DB group



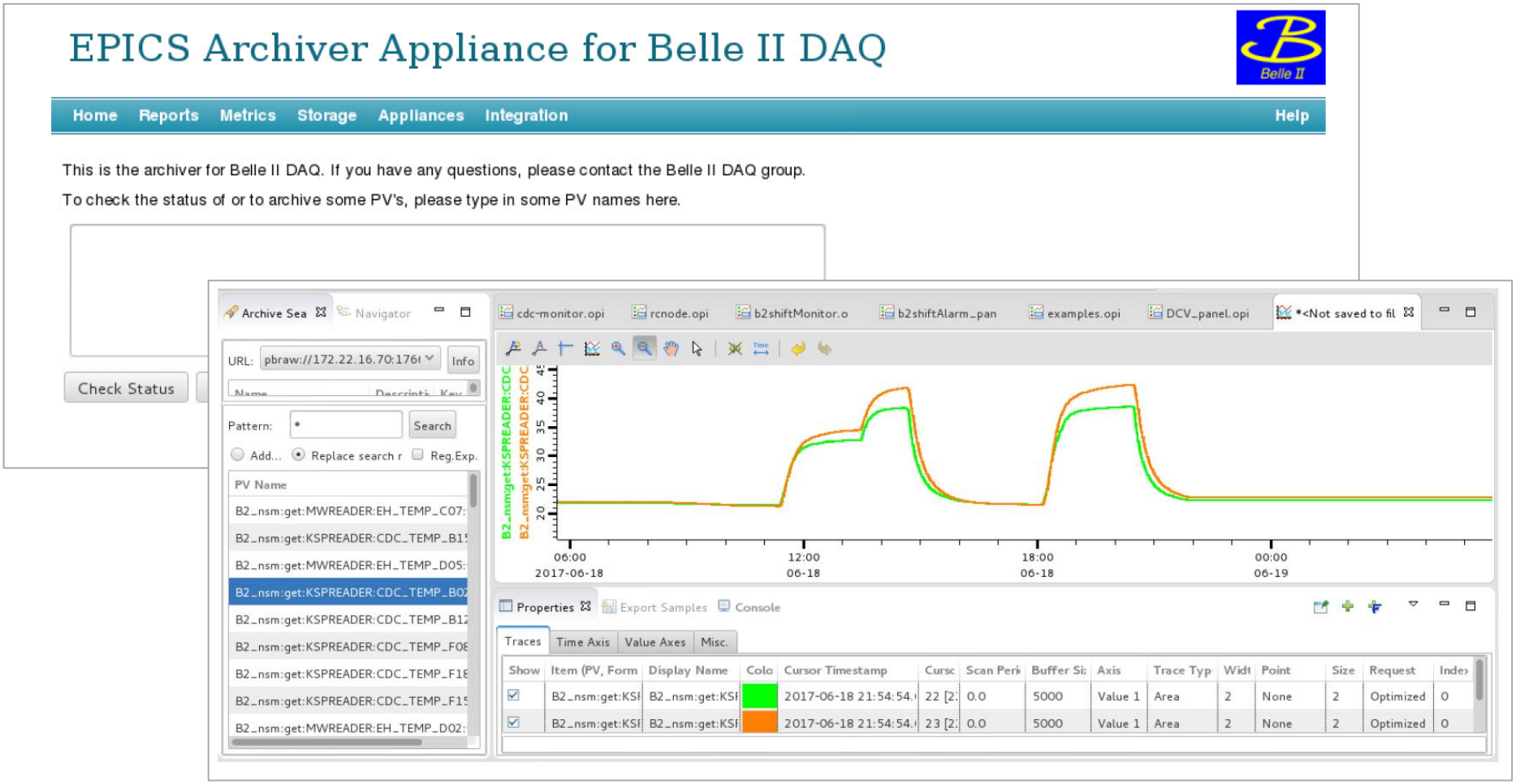
Summary

- Several functions in the DAQ database are implemented and running
  - Configuration DB provides configuration parameters
  - Run record is stored into database every run start/end
    => variables are collected from NSM variables
  - Logger DB shows message histories about detector/DAQ status
- A little but important progress for porting DAQ DB to offline system
  - Now DAQ DB server is open to KEKCC
  - Database is commonly available for Belle II members
  - Need feedback from Database group

Archiving System

From S. Park
(Yonsei University)

- EPICS Archiver Appliance : https://slacmshankar.github.io/epicsarchiver_docs/
  - Archiver of EPICS records => NSM2-to-EPICS converter is introduced
  - Available from web and Control System Studio



Archiving System

From S. Park
(Yonsei University)

- File based recording scheme is adapted with data size reduction in time
- Json format for global use is possible

○ Features of the archiver

http access is available to copy records

- Retrieving data type: txt, raw (pbraw://), json (web browser)
  - Developer says that csv, svg, mat(matlab) also available.
- Data is stored of text file on the below directory structure
  - "$sts/B2_nsm/get/KSPREADER/CDC_TEMP_B01/####.pb"
- Three kinds of storage depends on the time

STS (short term)	MTS (medium term)	LTS (long term)
– ~ day – Fast but small storage is efficient	– ~ week ~ month	– ~ month ~ year – Large but slow storage is efficient – Compression option