COPPER and Readout PC

1. COPPER/ROPC preparation status
2. SL5 -> SL6 OS upgrade
3. Event size
4. readout status of each sub-detector

S. Yamada (KEK)

1, COPPER/ROPC preparation status

Status of COPPERs

† https://belle2.cc.kek.jp/~twiki/bin/view/Detector/DAQ/EventSizeOfEachSubDetector

	Required # of COPPERs†	**Assembled**	**Installed in E-hut**	**Used in other places**	**Comments**
SVD	***48***	**0**	**0**	**0**	- Should be prepared before phase II
CDC	***75***	**75**	**51**	**4**	- All boards are needed after CDC installation
TOP	*16*	16	16	0	- 1 in Univ. of Hawaii
ARICH	*18*	18	14	4	- 4 COPPERs in used in the ARICH clean room on B4
ECL	*26*	26	26	0	- 18 COPPERs for Barrell ECL are used for CRT
bKLM	*4*	6	6	0	- 1 COPPER is connected with the detector
eKLM	*4*	9	9	0	
TRG	***10?***	**7**	**7**	**0**	- # of required COPPERs is not finalized. (around 10)

-> COPPERs except for SVD has been assembled

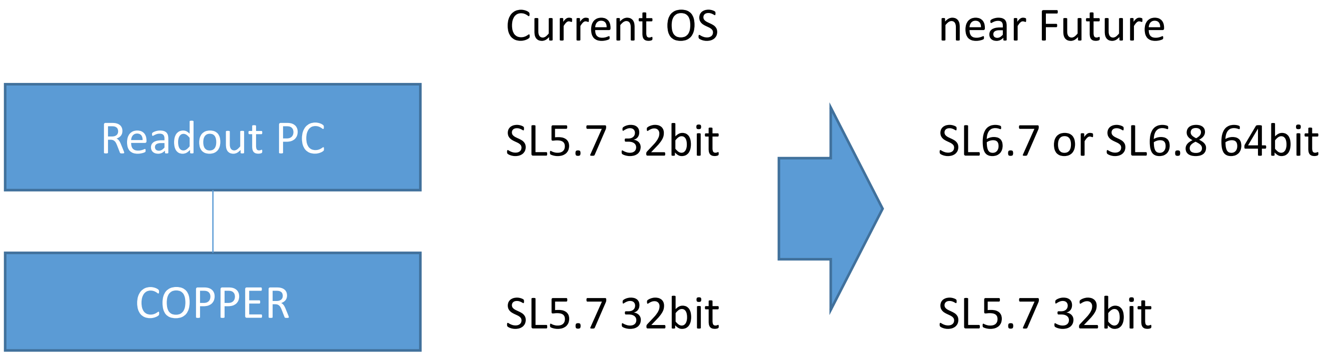
-> The rest of HSLBs for SVD and TRG have already been delivered at KEK.

Status of readout PCs

- 37 PCs were purchased.
    - 3 PCs will be purchased this JFY.
- 34 of them were alrady installed in E-hut.

2, Upgrade of OS in readout PCs (1)

Reason of the upgrade : security issues and discontinuing SL5 support by software group.



	Current OS	near Future
Readout PC	SL5.7 32bit	SL6.7 or SL6.8 64bit
COPPER	SL5.7 32bit	SL5.7 32bit

- Basf2 on SL5 will not be supported by software group after KEKCC upgrade.

-> It is not recommended to rely on basf2 program running on COPPERs.

- For global DAQ program on Readout PC and COPPERs, they still use daq and rawdata package but not using basf2 modules. (But they are still using daq and rawdata package).
  - Don’t need external/tools library and performance of DAQ program is a bit improved.
  - For pocket DAQ, basf2 modules(SeqROOToutput) is still needed to record data on readout PCs. Other parts can be replaced by the new program w/o basf2 modules.

2, Upgrade of OS in readout PCs (2)

Already updated to SL6

- svd01-08
- cdc01-04,06
- arich01-03

Following PCs are currently being used by detector groups.

- ecl01, ecl04
- top01-03
- cdc05,cdc07
- klm02

I will ask each group’s experts when upgrade should be done.

When upgrading OS, I will do clean install of SL6 because software RAID will be also Installed then.

I will backup /home directory before the upgrade work, but readout PCs aren’t supposed to be used to store important data or files. So please move or take backup of those important files elsewhere by yourselves. (I’m not taking regular backup.)

3, Event size

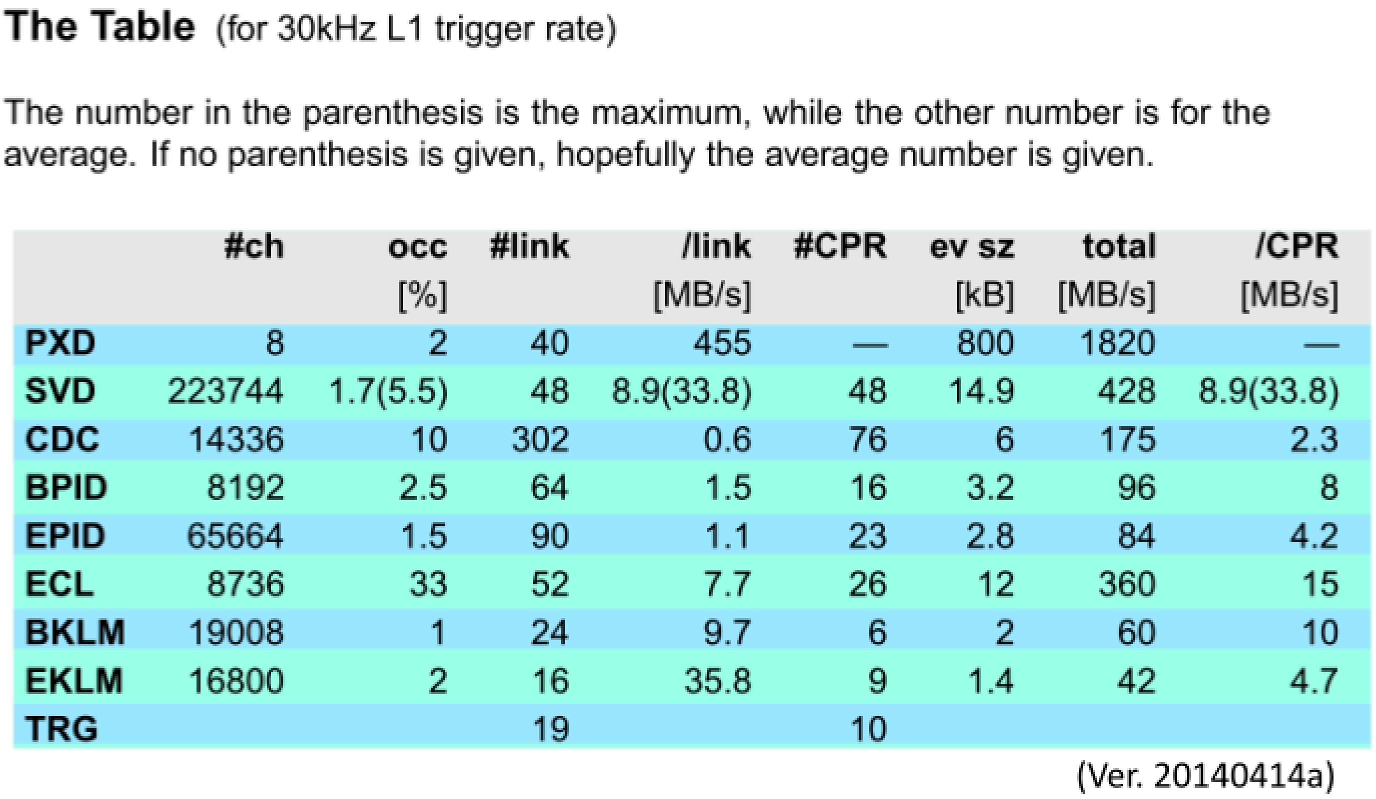
Event size estimation

2015 Feb. TRG/DAQ workshop

Baseline :

Summary table made by Nakao-san :

(https://belle2.cc.kek.jp/~twiki/pub/Detector/DAQ/WebHome/belle2-daq-spec.20140414a.pdf)



The Table (for 30kHz L1 trigger rate)

The number in the parenthesis is the maximum, while the other number is for the average. If no parenthesis is given, hopefully the average number is given.

	#ch	occ [%]	#link	/link [MB/s]	#CPR	ev sz [kB]	total [MB/s]	/CPR [MB/s]
PXD	8	2	40	455	—	800	1820	—
SVD	223744	1.7(5.5)	48	8.9(33.8)	48	14.9	428	8.9(33.8)
CDC	14336	10	302	0.6	76	6	175	2.3
BPID	8192	2.5	64	1.5	16	3.2	96	8
EPID	65664	1.5	90	1.1	23	2.8	84	4.2
ECL	8736	33	52	7.7	26	12	360	15
BKLM	19008	1	24	9.7	6	2	60	10
EKLM	16800	2	16	35.8	9	1.4	42	4.7
TRG			19		10			

(Ver. 20140414a)

Please tell DAQ group when the above value becomes obsolete.
(Due to the change of data-format, hit-rate and occupancy etc.)

-> But not so many feedbacks so far…

Request from computing gr.

BPAC focused review for software and computing in June.
-> The computing group is requested to estimate the event size.
According to Browder-san's e-mail,
"we do need a more realistic number by the end of September or earlier."

-> Only each sub-detector group can estimate the number, obviously.

But using packer modules and MC data, T. Kuhr-san write a script to obtain
Raw data object size and we can estimate the number.

$ basf2 raw.py (-> the script is in appendix of these slides)

But packer modules are not guaranteed to produce realistic number in physics run.
As far as my brief check this month, ARICH(fixed), TOP and PXD(fixed) packers produce data of different mode or format from that of physics run.

Each detector group needs to commit those numbers more to avoid future conflicts with computing and DAQ groups.

I tried running Thomas-san’s script with packer modules).

SVD packer

	Prev. estimation (DAQ Twiki)	Thomas-san’s script (EventGen)	Thomas-san’s script (BHWide)
FEE data size [kB/ev]	14.9	17.6	17.0
FEE overhead [kB/ev]	N.A.	N.A	N.A
Hit size[byte/hit]	3.75		
Occupancy[%]	1.7	**0.42** ☺	
DAQ overhead[kB/ev]	3.3	3.3	3.3
Total data flow[MB/s]	545	626	607

Occupancy is decreased but FEE data size is increased.
Total data flow : moderate increase

Current assignment of ROPC : 12 -> ?

CDC packer

	Prev. estimation (DAQ Twiki)	Thomas-san's script (EventGen)	Thomas-san's script (BHWide)
FEE data size [kB/ev]	6.0 +3.6	15.3	9.3
FEE overhead [kB/ev]		3.6	3.6
Hit size[byte/hit]	4	8 ☹	8
Occupancy[%]	10	10.2	5.0
DAQ overhead[kB/ev]	7.8 ☹	7.8	7.8
Total data flow[MB/s]	522	693	512

- Occupancy estimation is not changed.
- Hit size is increased.
- Total data flow : a bit increase

Current assignment of ROPC : 11 -> ?

TOP packer

	Prev. estimation (DAQ Twiki)	Thomas-san's script (EventGen)	Thomas-san's script (BHWide)
FEE data size [kB/ev]	3.2	7.5	4.0
FEE overhead [kB/ev]		0.26	0.26
Hit size[byte/hit]	15.6	20	20
Occupancy[%]	2.5	4.5	2.3
DAQ overhead[kB/ev]	1.9	1.9	1.9
Total data flow[MB/s]	152	275	181

- Occupancy estimation -> Matt-san's talk?
- Hit size is increased.
- Total data flow : increase

Current assignment of ROPC : 3 -> ?

ARICH packer

	Prev. estimation (DAQ Twiki)	Thomas-san's script (EventGen)	Thomas-san's script (BHWide)
FEE data size [kB/ev]	2.8	4.9	4.9
FEE overhead [kB/ev]		3.7	3.7
Hit size[byte/hit]	3.5	5	5
Occupancy[%]	1.5	0.32 ☺	0.33 ☺
DAQ overhead[kB/ev]	1.9	1.9	1.9
Total data flow[MB/s]	140	201	202

- Occupancy is decreased.
- Hit size + ovhead are increased
- Total data flow : increased

Current assignment of ROPC : 4 -> ?

The output of raw.py : ROOT(not .sroot) size = RawARICHs 6.62 kB/ev

bKLM packer

	Prev. estimation (DAQ Twiki)	Thomas-san's script (EventGen)	Thomas-san's script (BHWide)
FEE data size [kB/ev]	2	4.1	4.0
FEE overhead [kB/ev]	0.5		
Hit size[byte/hit]	8	8	8
Occupancy[%]	1	2.7 ☹	2.6 ☹
DAQ overhead[kB/ev]	0.4	0.4	0.4
Total data flow[MB/s]	72	137	131

- Occupancy is incerased.
- Total data flow : increased

Current assignment of ROPC : 2 -> ?

PXD

- Size of ROOT(not sroot) object : 19.3kB/ev (getzipbytes=15.7kB) ☺
- If HLT trigger rate is 10kHz, ROOT object size of raw PXD data will be 19.3kB*10k = 200MB/s.

-> Lange-san will discuss his estimation in his talk.

ECL, eKLM

- Packer modules are under construction.

RawTRG, RawFTSW

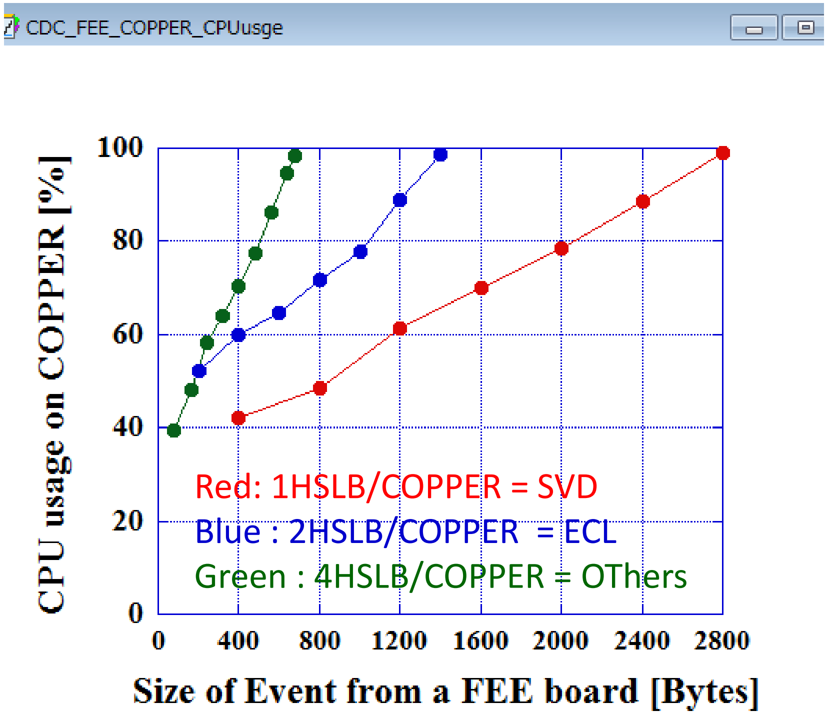
- Format has not been finalized ?
- Negligible small ?

Let’s compare the outputs by MC+packer with the estimation by each expert at the tomorrow’s session.

Effect of event-size on COPPER

Bottleneck : COPPER CPU usage to process data.

(Other possible bottleneck : output bandwidth of COPPER = 1GbE = 125MB/s)



= size of so called ‘detector buffer’

SVD Layer3 (around 1kB/event ?) may become the bottleneck among all links.

- It is difficult to increase # of COPPER boards for SVD because it has already only one HSLB/COPPER.

- For other sub-detectors, doubling COPPERs is not easy due to the shortage of spare COPPER boards.

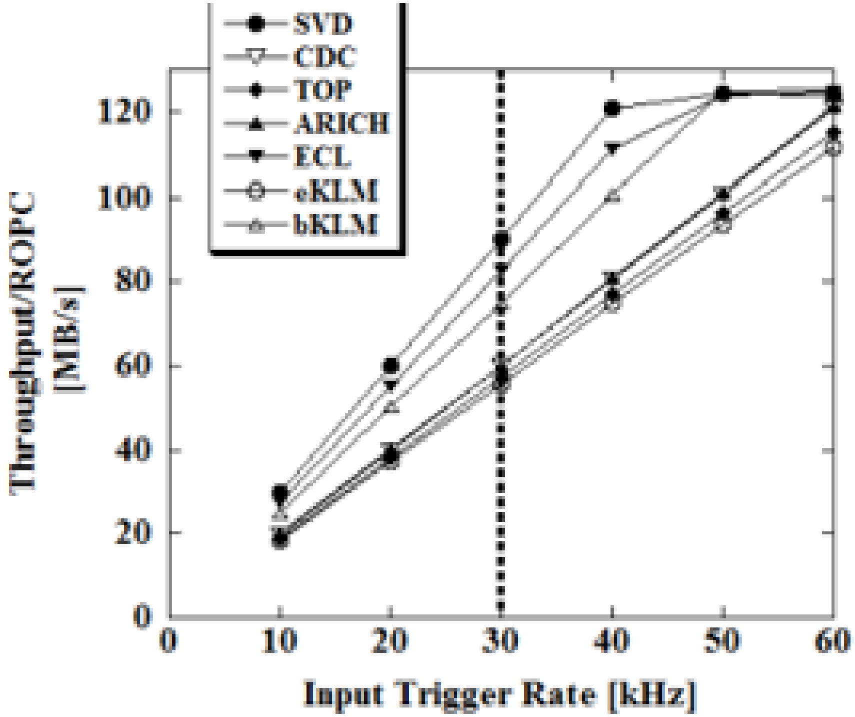
Effect of event size on readout PC (1)

Bottleneck : Output bandwidth of ROPC to HLT = 1GbE = 125MB/s

(Other possible bottleneck : CPU usage in ROPC)

Parameter : Event size and # of COPPERs/ROPC

Not so much space for the increase, according to the previous size estimation.



Hopefully, no more increase of the size estimation…

# of COPPERs/ROPC
SVD : 4 -> total ROPC 12
CDC : 7 -> total ROPC 11
TOP : 6 -> total ROPC 3
ARICH: 6 -> total ROPC 4
ECL : 5 -> total ROPC 4
bKLM : 4 -> total ROPC 2
eKLM : 4 -> total ROPC 2
TRG -> 1?
Sum of ROPC = 39

Size of detector buffer / HSLB used to make the left figure. [bytes]

SVD: 672 bytes
CDC : 44 bytes
TOP : 52 bytes
ARICH : 56 bytes
ECL : 232 bytes
bKLM : 128 bytes
eKLM : 88 bytes

Effect on readout PC (2)

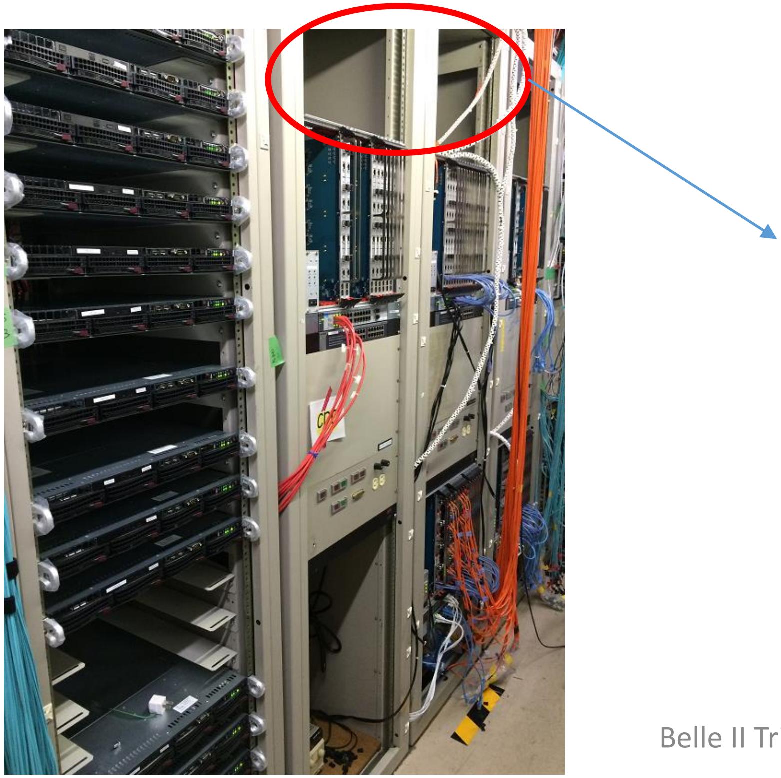
If readout PC system should be reinforced,

1) Increase # of Readout PC
2) Increase # of NIC/PC

can be considered.

Procurement status :
40 readout PCs have been purchased before phase II run.

After Phase II, we can again consider whether more ROPC should be purchased or not.



This space above COPER crates will be used for locating additional PCs, if necessary.

4, Readout status

My understating of readout status of each sub-detector group

	Achievement	Future
SVD ☺	Readout data via COPPER in two DESY beam tests	The last DESY beam test. Phase II
CDC ☺	Continuous cosmic ray test.	Read out from CDCTG. Global CRT.
TOP	Cosmic ray test and FW development	CRT with feature extraction. Global CRT
ARICH	Reading out from 6 FEE boards (1module).	Increase the # of FEEs and stable operation.
ECL ☺	Continuous cosmic ray test.	Global CRT
KLM	Cosmic ray test and FW development	Cosmic ray test with global DAQ and Global CRT.

I can only report the TOP readout test here, since I joined Maeda-san and Gaz-san’s test.

Stability issue of (installed) TOP DAQ during the readout test

(last B2GM)

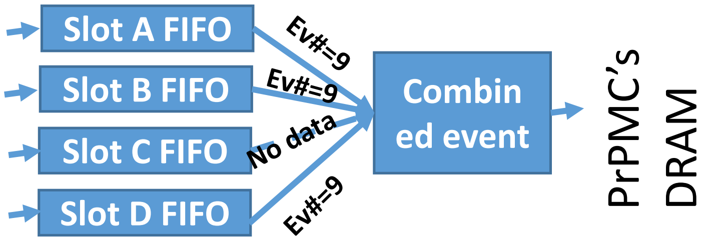
Dummy trigger rate :

- 1 or 2Hz -> It sometimes goes to BUSY state(= COPPER FIFO almost full) just after the run-start. After retrying a few times, it started running.
- >= 3Hz : It stopped just after the run-start. Restarting for a few times did not work.

It is not as stable as other sub-detector's readout.

From the COPPER side, it seems that some HSLBs do not receive the same number of events as that from other HSLBs, which can cause deadlock.

```
cpr3005 $ staths | grep event
(a)   event=16 total=132kB (avg=8268B last=16492B max=16492B)
(b)   event=56 total=611kB (avg=10912B last=16492B max=16496B)
(c)   event=8 total=66kB (avg=8268B last=16492B max=16492B)
(d)   event=59 total=644kB (avg=10916B last=16492B max=16496B)
```



- If the information from FEE side is available, like received trigger # from FTSW by FEE or sent event # to COPPER, the situation will be clearer.
- The implementation of feature extraction may change the situation. (Currently, one event is divided to 16 fragments by FEE to reduce event size.)

Stability issue of (CDC+)TOP DAQ during the readout test

- Ale-san pointed out that (CDC+)TOP DAQ crashed after 4hours data-taking last Sunday.

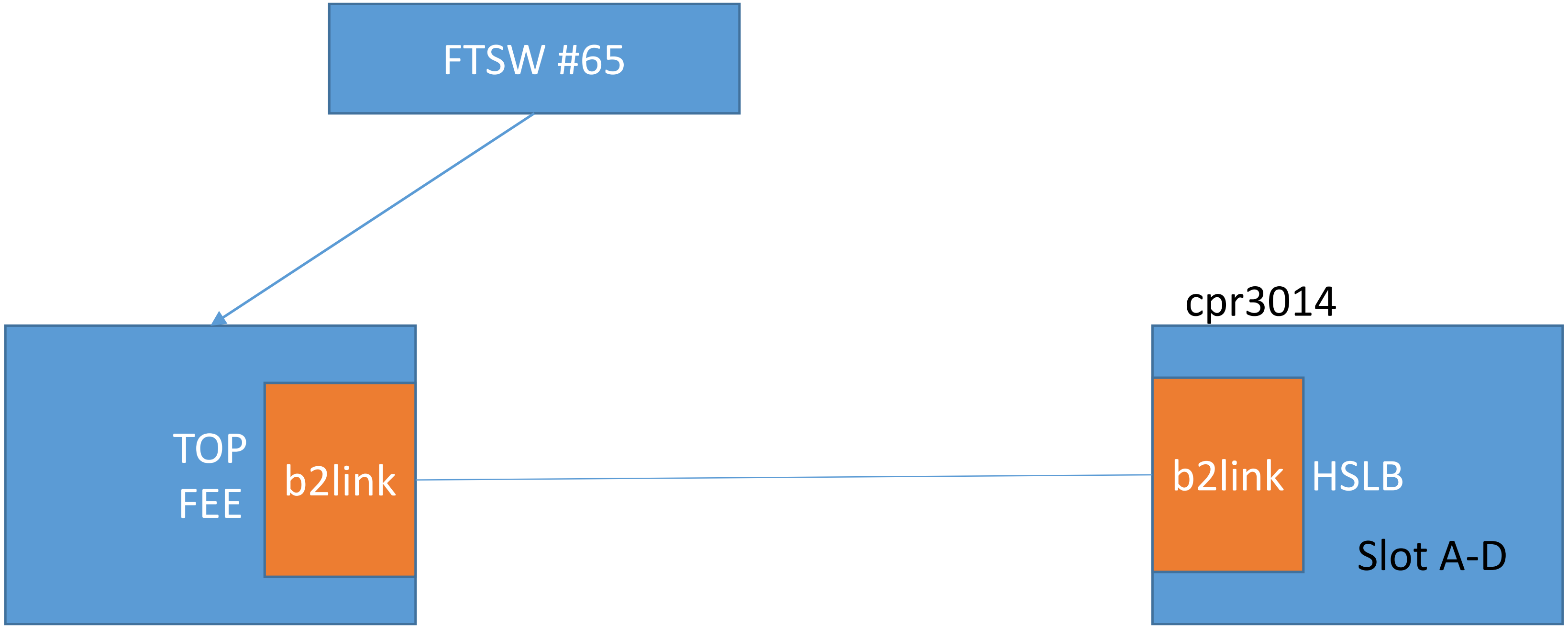
Memory leak of ROPC DAQ software for large events
-> Fixed. (svn. **29424**)
-> 13hours data-taking @ 0.22Hz

Consistency check of # of events for installed TOP modules

Setup :

Firmware : feature extraction version (No event division)

(A) # of triggers from FTSW



Nakao-san implemented
(B) # of evens sent from FEE via b2link.

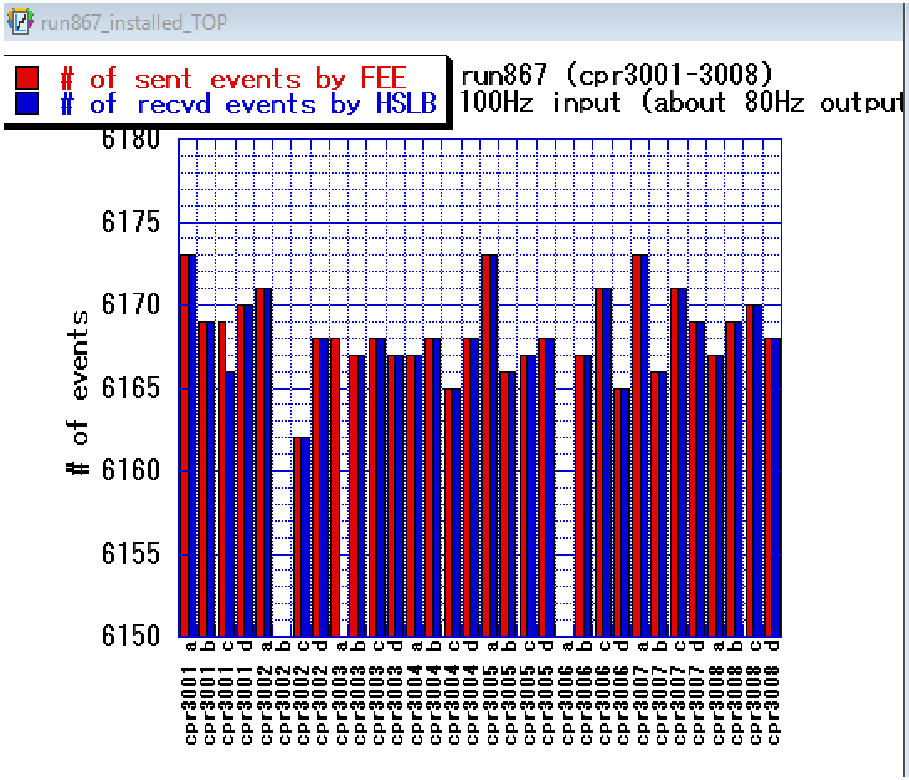
(C) HSLB received events

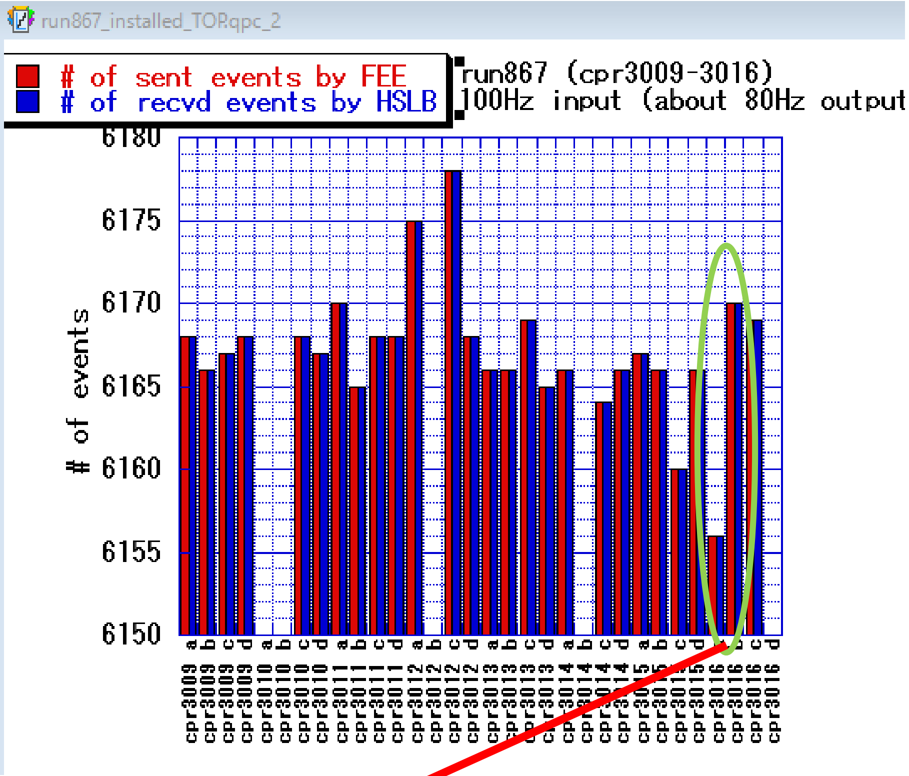
Commands to obtain above information:

```
(A) ttd1 $ statft -65
(B) cpr3014 $ reghs -a(bcd) fee32 5
(C) cpr3014 $ staths | grep event
```

2, Trigger rate 100Hz w/ all COPPERs except for some HSLB slots, which did not send any events in run855

-> DAQ got stuck after 7255 triggers were generated, because COPPER FIFO became almost full.





- ➢ There was difference in # of sent events between HSLBs on a COPPER. When the difference becomes 14(=14*30kB=412kB), COPPER FIFO becomes almost full.
- ➢ Statft said 7255 triggers were distributed to FEEs, but at most 6178 events were sent from FEE.

Summary

- Event size estimation will affect the resourse of DAQ and computing.
    - Need to closely watch the simulation
    - Each sub-Detector gr. is responsible to the event size estimation except for the DAQ overhead.
- All sub-detector groups have started to take data w/ b2link+COPPER.
    - Stable operation is a goal before the global CRT.

APPENDIX

raw.py for event size estimation

You can copy and run the script by yourself.

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

################################################################
# This steering file generates, simulates, and reconstructs
# a sample of 10 BBbar events.
#
# Usage: basf2 example.py
#
# Input: None
# Output: output.root
#
# Example steering file - 2011 Belle II Collaboration
################################################################

from basf2 import *
from simulation import add_simulation
from rawdata import add_packers
from glob import glob
import ROOT

# create path
main = create_path()

# specify number of events to be generated
eventinfosetter = register_module('EventInfoSetter')
eventinfosetter.param('evtNumList', [100])
main.add_module(eventinfosetter)

# generate BBbar events
main.add_module('EvtGenInput')

# detector simulation
bg = glob('/sw/belle2/bkg/*.root')
add_simulation(main, bkgfiles=bg)

# do raw data packing and unpacking
add_packers(main)
))

# full output
main.add_module('RootOutput', outputFileName='raw.root')

# process events and print call statistics
process(main)
print(statistics)


# print raw data event sizes
root_file = ROOT.TFile('raw.root')
tree = root_file.Get('tree')
events = tree.GetEntriesFast()
print('Events: %d' % events)
for branch in tree.GetListOfBranches():
    name = branch.GetName()
    if name.startswith('Raw'):
        size = branch.GetTotBytes('*') * 1.0
        zipsize = branch.GetZipBytes('*') * 1.0
        print("%s %.2f (%.2f)" % (name, size/1024./events, zipsize/1024./events
```