

New STORE Status

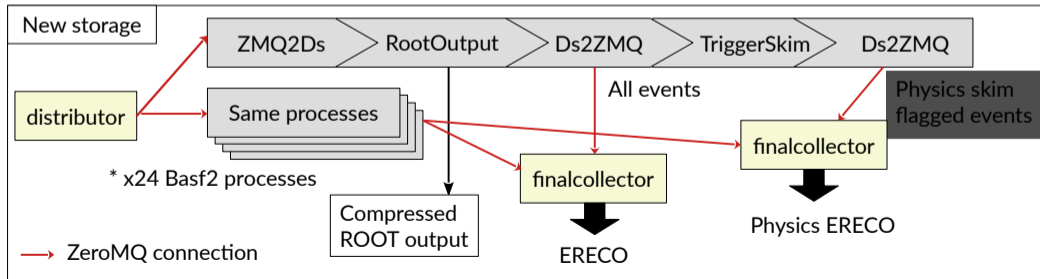
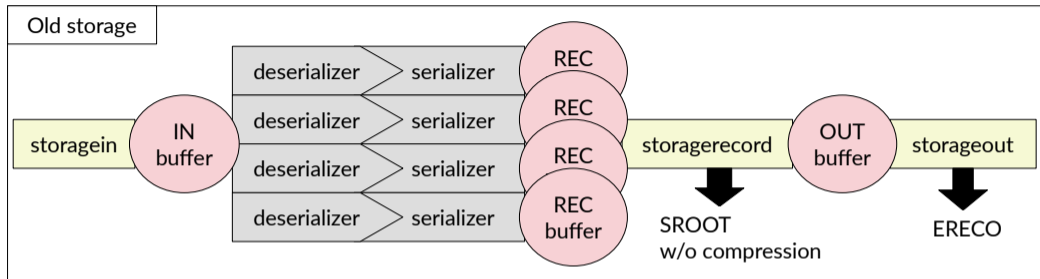
Seokhee Park

seokhee.park@kek.jp

KEK IPNS

2022, Dec. 1st

Introduction



■ Storage server

- ▶ CPU: 32-48 threads (depending on the unit #)
- ▶ Three RAID cards and disk enclosures which consists of 12×4 TB HDD
 - We will use RAID-6 with a hot spare → 36 TB per partition
 - Total 24 BASF2 processes → 8 files are written in a partition
- ▶ Currently, the storage is using diskless PXE boot via NFS
 - If h1tct1 is down, online to offline file transfer will be also down
 - During the LS1, I will install OS in the local disk and replace the HLT control network to daqnet

Slow control

RC Command

LOAD

ABORT

HLT13

36754

50.8 Hz

6.8 MB/s

RC_HLTB_HLTDAQ Run # : 20

NOTREADY DISTRIBUTOR NOTREADY
COLLECTOR NOTREADY
Bridge : JQMSEVER NOTREADY
BHLTB_HLTDAQ EB1 OFF
H NOTREADY

H NOTREADY

RC_ERECOTB_ERECORun # : 20

NOTREADY UTADAPTER NOTREADY
COLLECTOR NOTREADY
Bridge : COLLECTOR NOTREADY
BERECOTB_ER UTADAPTER NOTREADY
JQMSEVER NOTREADY

NOTREADY

RC_STORETB_HLTDAQRun # : 20

NOTREADY DISTRIBUTOR NOTREADY
LECTOR_ALL NOTREADY
Bridge : ECTOR_PHY NOTREADY
BSTORETB_HLT EB2RX NOTREADY
WORKER NOTREADY

RC_ERECOTB_ERECORun # : 20

NOTREADY ADAPTER_PHY NOTREADY
ECTOR_PHY NOTREADY
Bridge : ECTOR_PHY NOTREADY
BERECOTB_ER ADAPTER_PHY NOTREADY
SERVER_PHY NOTREADY

ERWK NOTREADY

Run setting

Run type : hlttest

HLT script :

beam_reco_monitor

HTB_HLTDAQ NOTREADY ETB_HLTDAQ NOTREADY

_ERECODAQ NOTREADY

CODAQ_PHY NOTREADY

RUNRECORD NOTREADY

Run status

xp # : 1

un # : 20

Run control

NOTREADY

Data flow

#####

Global DAQ test

- **Global DAQ test is being continued with HLT05 and RECL**

- **HLT05**

- ▶ CPU: Two Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz (48-thread in total)
- ▶ MEM: 32 GB (Not much memory consumption for the BASF2 processes)

- **Test environment**

- ▶ Including RECL only
- ▶ hlttest / 1000 input rate / passthrough

Global DAQ test

■ System resource usage of the test environment

- ▶ CPU usage per BASF2 process: ~80% (top result)
- ▶ Total (all 3 RAID) disk writing: 150 MB/s → 50 MB/s per RAID

■ 3 kHz physics run estimation using exp26 run1968

- ▶ $22,089,949 \text{ evt} / 1976 \text{ s} = 11,230 \text{ Hz}$
- ▶ Total output file size after ROOT conversion: 303 GB
- ▶ 3 kHz physics run writes output file with 41 MB/s → 14 MB/s per RAID

■ Our test environment is more tough than the final goal

■ Results

- ▶ 17 hours run creates 9 TB root output files into the storage without any issue
- ▶ Simultaneously, b2file-merge and file copying was tested
 - No performance degradation found = No output rate drop and event loss
- ▶ However, sometimes (or frequently), run goes to ERROR right after the events are reaching to the storage
 - SALS makes next run work
 - This was not happened in the test bench, so I don't have any idea for now

■ Concerns

- ▶ Sometimes, beam_reco_monitor is requested and the 3 kHz beam_reco_monitor cannot be handled by current setup
 - The bottleneck comes from CPU
 - If we do not use compression, disk size might be ~300 MB/s per RAID → 1 TB/h
- ▶ I don't know how large input rate to be allowed for local run
 - At least, detnull will be introduced during LS1

Post-processing tool and DB

■ Post-processing tool

- ▶ will do file merging, renaming, datafiles related table update, etc.
- ▶ is written by python3 with psycopg3
 - I revised the script using proper placeholder, suggested by Yamagata-san
- ▶ For file recovery, I discussed with Thomas Kuhr
 - We conclude to wait a realistic case as an starting point

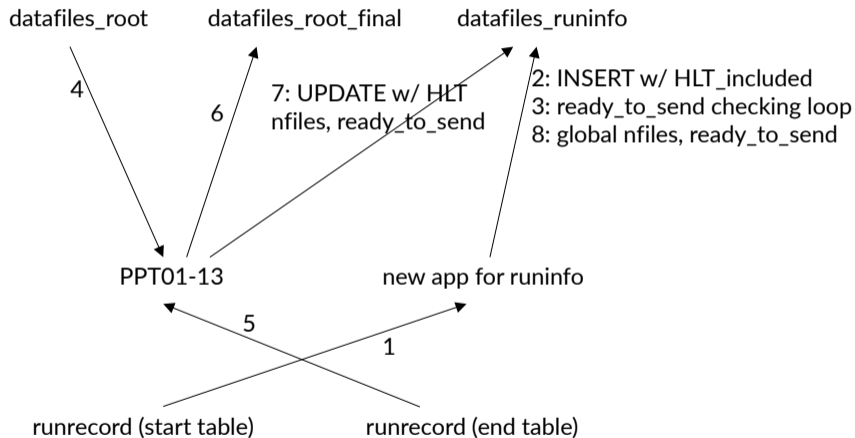
■ In the previous DAQ-CC meeting, we conclude to make additional database table to keep the run information related with output file

- ▶ To managing this additional table, I'm writing new python program
- ▶ The program can be run anywhere, but I think that it's good to put in rc01

Post-processing tool and DB

```
CREATE TABLE datafiles_runinfo (  
  id bigserial primary key NOT NULL,  
  expno bigserial NOT NULL,  
  runno bigserial NOT NULL,  
  runtype text NOT NULL,  
  nfiles serial NOT NULL DEFAULT 0,  
  ready_to_send boolean NOT NULL DEFAULT FALSE,  
  transfer_done boolean NOT NULL DEFAULT FALSE,  
  files_removed boolean NOT NULL DEFAULT FALSE,  
  
  HLT01_included boolean NOT NULL DEFAULT FALSE,  
  HLT01_ready_to_send boolean NOT NULL DEFAULT FALSE,  
  HLT01_transfer_done boolean NOT NULL DEFAULT FALSE,  
  HLT01_files_removed boolean NOT NULL DEFAULT FALSE,  
  HLT01_nfiles serial NOT NULL DEFAULT 0,
```

Post-processing tool and DB



Partial SALS

- EB2 managing 'keep connection'!
- Inside the storage, we should consider output file naming
 - ▶ Currently, the new file name is assigned while loading
 - ▶ After applying the partial SALS, there is no chance to LOAD with run number changing
 - ▶ I need to modify file naming mechanism

To do list during LS1

1. Debug the frequent ERROR after START
 2. Complete post-processing tool and new app for runinfo table
 3. Create `datafiles_root`, `datafiles_root_final`, and `datafiles_runinfo` table into the b2s1ow DB
 4. Take global run including post-processing tool and the runinfo app with ERECO connection
 5. Complete the daqnet setup for STORE and update `restart.sh`
 - ▶ Reinstallation is needed to add software array for local disk
 6. Apply new STORE into the store01-05 and include them into the global run
 7. Test local run slow control (pcie40 local run is available from hlt06-10)
 - ▶ `detnull` run, too
- In parallel, changing file naming mechanism for partial SALS