

Hall-D Online Skim System

(to be named later)

Oct. 5, 2019

David Lawrence

Online Skims

- “Skim” files contain a subset of events from the raw data stream
- Events formed from specialized triggers for calibration or normalization
- Produced by dedicated pass over entire data set on scicomp farm

<code>hd_rawdata_061580_002.evio</code>	20 GB
---	-------

<code>hd_rawdata_061580_002.BCAL-LED.evio</code>	45 MB
<code>hd_rawdata_061580_002.CCAL-LED.evio</code>	132 MB
<code>hd_rawdata_061580_002.FCAL-LED.evio</code>	68 MB
<code>hd_rawdata_061580_002.DIRC-LED.evio</code>	0.2 MB
<code>hd_rawdata_061580_002.ps.evio</code>	2,574 MB
<code>hd_rawdata_061580_002.random.evio</code>	48 MB
<code>hd_rawdata_061580_002.sync.evio</code>	0.4 MB

GOAL:

Generate these in counting house when data is taken

- reduce tape drive usage
- reduce Lustre activity
- reduce time waiting for skims

The Challenge of Online Skims:


- Start High Intensity running in the Fall
 - Larger data rates than ever seen in production
 - Single RAID partition cannot handle full rate (at least not stable)
- Requires scanning entirety of every file
 - Never done even in low intensity era
- Cannot be done with only RAID server compute capacity
 - Must distribute to farm nodes
- High data volumes+hardware limits necessarily couples data flow with skim system

hdskims + hdmk_skims.py









- high intensity will produce **4-5 files/min** (20GB)
- **hdskims**: skim through EVIO file and write blocks (40 events) containing at least one FP trigger to separate file (~10 sec)
- **hdmk_skims.py**: Run hdskims to create reduced EVIO file then run hd_ana with trigger_skims and ps_skim plugin on that to produce standard skim files (~40 sec)
- 20GB file processing time: classic method=4 min -- new method=1 min

Branch: davidl_hdskims ▾ [halld_recon](#) / [src](#) / [programs](#) / [Utilities](#) / [hdskims](#) / Create new file Upload files Find file History

This branch is 12 commits ahead, 8 commits behind master. Pull request Compare

 **faustus123** Built in support for generating SQL and committing it to skiminfo DB... ... Latest commit ec6b556 yesterday

..

 HDEVIOWriter.cc	Adding HDEVIOWriter and hdbyte_swapout files.	10 days ago
 HDEVIOWriter.h	Adding HDEVIOWriter and hdbyte_swapout files.	10 days ago
 SConscript	Significant changes to make hdskims work.	14 days ago
 hdbyte_swapout.cc	Adding HDEVIOWriter and hdbyte_swapout files.	10 days ago
 hdbyte_swapout.h	Adding HDEVIOWriter and hdbyte_swapout files.	10 days ago
 hdmk_skims.py	Built in support for generating SQL and committing it to skiminfo DB...	yesterday
 hdskims.cc	Built in support for generating SQL and committing it to skiminfo DB...	yesterday
 skiminfo.sql	Schema for skinfo DB along with entries for known trigger types.	yesterday

Skiminfo DB

- Complete trigger counts are accumulated during initial scan of raw data file

```
1
2 CREATE TABLE IF NOT EXISTS skiminfo (
3
4     run INT,
5     file INT,
6     UNIQUE KEY (run, file),
7     num_physics_events INT,
8     num_bor_events INT,
9     num_epics_events INT,
10    num_control_events INT,
11    first_event INT,
12    last_event INT,
13
14    NGTP0 INT DEFAULT 0,
15    NGTP1 INT DEFAULT 0,
16    NGTP2 INT DEFAULT 0,
17    NGTP3 INT DEFAULT 0
```

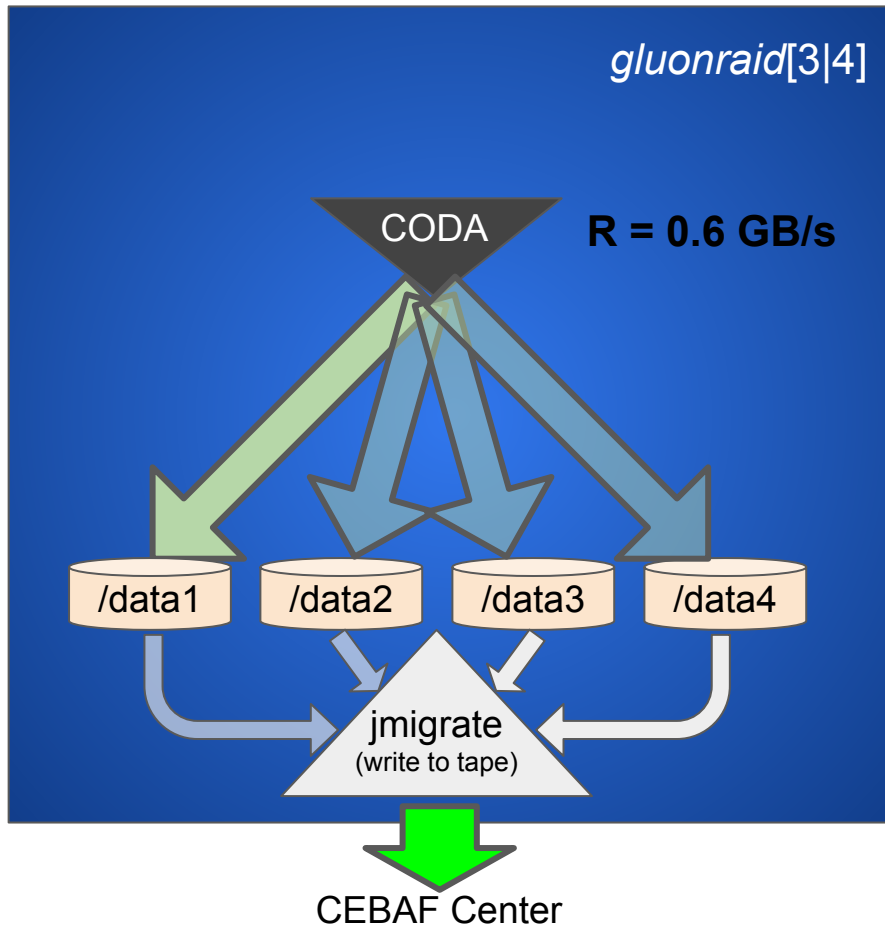
- First and last event number found for each file
- System writes these to DB so complete trigger statistics are recorded for each file
- RCDB?

```
38    NFP7 INT DEFAULT 0,
39    NFP8 INT DEFAULT 0,
40    NFP9 INT DEFAULT 0,
41    NFP10 INT DEFAULT 0,
42    NFP11 INT DEFAULT 0,
43    NFP12 INT DEFAULT 0,
44    NFP13 INT DEFAULT 0,
45    NFP14 INT DEFAULT 0,
46    NFP15 INT DEFAULT 0,
47
48    skim_host VARCHAR(256),
49    created TIMESTAMP
50 );
```

Hall-D Data recording

(up to now)

- Transport into RAID server via 40Gbps ethernet
- Event builder and Recorder run on directly on RAID server
- All files from one run written to single partition
- Files read from non-active partitions for writing to tape
- CODA configuration must be changed to switch to another RAID server



Fast Networks in Counting House

- 40 Gbps ethernet for DAQ system
 - EMUs
 - RAID servers
- 40(56) Gbps infiniband (IB) for everything
 - EMUs
 - RAID servers
 - farm nodes
 - **capable of RDMA**

hdrdmacp - **H**all-**D** Remote **D**irect **M**emory **A**ccess **C**o**P**y

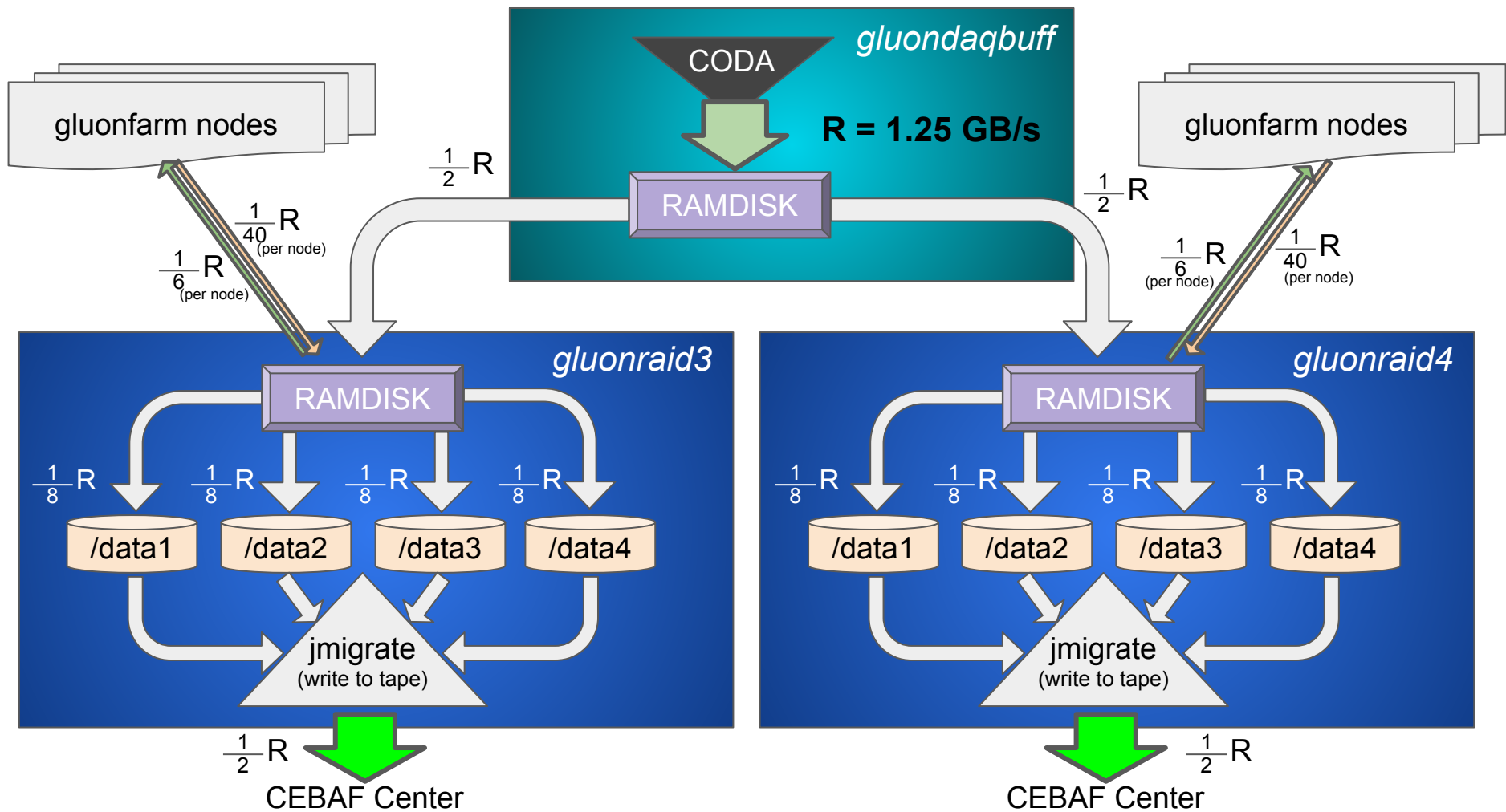
- Program runs as either server or client to copy file(s) over IB with minimal CPU (uses a feature of IB network card)
- Configured as systemd service on all gluons with IB connection
- Single stream transfers up to 1.5GB/s
- Multiple streams can transfer 3.5GB/s sustained
- Publishes statistics periodically as JSON formatted message using zeroMQ

subversion: (for us)

<https://halldsvn.jlab.org/repos/trunk/online/packages/miscUtils/src/hdrdmacp/>

github: (for the rest of the world)

<https://github.com/JeffersonLab/hdrdmacp>



```
sync_test6.config
#-----
# DAQ
stage: gluondaqbuff
source /media/ramdisk/@TESTDIR/active/*.evio
destination /media/ramdisk/@TESTDIR/rawdata_in

distribute: gluondaqbuff
source /media/ramdisk/@TESTDIR/rawdata_in/*.evio
destination gluonraid3:/media/ramdisk/@TESTDIR/active
destination gluonraid4:/media/ramdisk/@TESTDIR/active

stage: gluonraid3, gluonraid4
source /media/ramdisk/@TESTDIR/active/*.evio
destination /media/ramdisk/@TESTDIR/rawdata_staged_for_disk
destination /media/ramdisk/@TESTDIR/rawdata_staged_for_skim

#-----
# RAWDATA
#
# First copy from ramdisk to one of the RAID partitions and then
# make links in staged_for_tape and volatile directories.

distribute: gluonraid3, gluonraid4
source /media/ramdisk/@TESTDIR/rawdata_staged_for_disk/*.evio
destination /data1/@TESTDIR/rawdata_staged_for_disk
destination /data2/@TESTDIR/rawdata_staged_for_disk
destination /data3/@TESTDIR/rawdata_staged_for_disk
destination /data4/@TESTDIR/rawdata_staged_for_disk

stage: gluonraid3, gluonraid4
source /data1/@TESTDIR/rawdata_staged_for_disk/*.evio
destination /data1/@TESTDIR/rawdata/staged_for_tape/@RUNPERIOD/rawdata/Run@RUNNUMBER
destination /data1/@TESTDIR/rawdata/volatile/@RUNPERIOD/rawdata/Run@RUNNUMBER
stage: gluonraid3, gluonraid4
source /data2/@TESTDIR/rawdata_staged_for_disk/*.evio
destination /data2/@TESTDIR/rawdata/staged_for_tape/@RUNPERIOD/rawdata/Run@RUNNUMBER
destination /data2/@TESTDIR/rawdata/volatile/@RUNPERIOD/rawdata/Run@RUNNUMBER
stage: gluonraid3, gluonraid4
source /data3/@TESTDIR/rawdata_staged_for_disk/*.evio
destination /data3/@TESTDIR/rawdata/staged_for_tape/@RUNPERIOD/rawdata/Run@RUNNUMBER
destination /data3/@TESTDIR/rawdata/volatile/@RUNPERIOD/rawdata/Run@RUNNUMBER
stage: gluonraid3, gluonraid4
source /data4/@TESTDIR/rawdata_staged_for_disk/*.evio
destination /data4/@TESTDIR/rawdata/staged_for_tape/@RUNPERIOD/rawdata/Run@RUNNUMBER
destination /data4/@TESTDIR/rawdata/volatile/@RUNPERIOD/rawdata/Run@RUNNUMBER

#-----
# SKIMS
#
# - Copy to farm node
# - Run hdmy_skims.py on farm node to generate skim files
# - Copy skims back to raid server
# - Move to RAID disk
# -

distribute: gluonraid3
source /media/ramdisk/@TESTDIR/rawdata_staged_for_skim/*.evio
destination gluon100:/media/ramdisk/@TESTDIR/active
destination gluon101:/media/ramdisk/@TESTDIR/active
destination gluon102:/media/ramdisk/@TESTDIR/active
```

System driven by two types of operations:

stage: move and link files within a filesystem

distribute: transfer file to one of a number of other filesystems

Raw data files on RAID partition hard linked in **rawdata_staged_for_tape** and **volatile**

Still to be done

- No real back-pressure mechanism yet
 - System needs to recognize if it can't keep up with rate and set alarm (via CODA?)
- Integration with RCM
 - Processes must be started on many computers when DAQ starts
- System monitoring tool needs more development
- Thorough testing
- Unclear how much this will impact monitoring system (will certainly diminish capacity for full-recon monitoring)

Summary

- New paradigm for raw data flow in counting house
 - Splitting data stream at file level significantly reduces demand on individual hardware components
 - System handles distribution of files over nodes and partitions as well as shallow copies via hard links (*large configuration file*)
- RDMA
 - hrdmactp program written, tested, deployed as systemd service on gluons
- hdskims
 - Breaks skimming into 2 phases resulting x4 speedup
 - Automatically fills DB with trigger statistics for each file