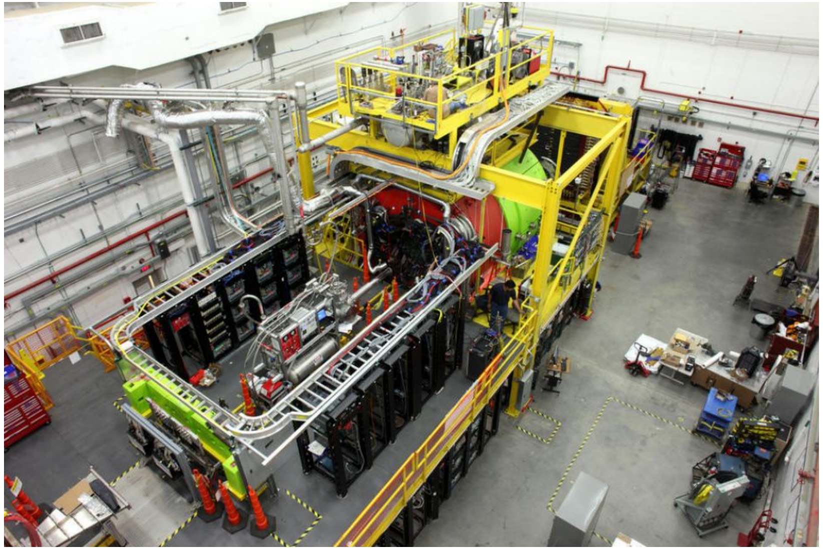
HOSS

(Hall-D Online Skim System)

May 19, 2020

David Lawrence

Motivation



The GlueX detector in Hall-D at Jefferson Lab

GlueX is a fixed target experiment that uses a high intensity linearly polarized 12GeV photon beam incident on a proton target

DAQ Rate:	1.25GB/s
Raw data:	~6PB/yr
CPU(incl. sim):	~200Mcore-hr/yr

Problem:
Single primary physics trigger with several calibration triggers all recorded in a single output stream.

This leads to inefficient use of resources offline to filter the (rare) calibration events into separate files.

Skims

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

hd_rawdata_071783_337.evio	20 GB
hd_rawdata_071783_337.BCAL-LED.evio	6.8 MB
hd_rawdata_071783_337.CCAL-LED.evio	0.3 MB
hd_rawdata_071783_337.FCAL-LED.evio	7.1 MB
hd_rawdata_071783_337.DIRC-LED.evio*	1.7 MB
hd_rawdata_071783_337.ps.evio*	69.7 MB
hd_rawdata_071783_337.random.evio	20.1 MB
hd_rawdata_071783_337.sync.evio	0.4 MB
hd_root_tofcalib_071783_337.root	12.3 MB
TOTAL	118.4 MB

(<0.7% of total data volume)

GOAL:

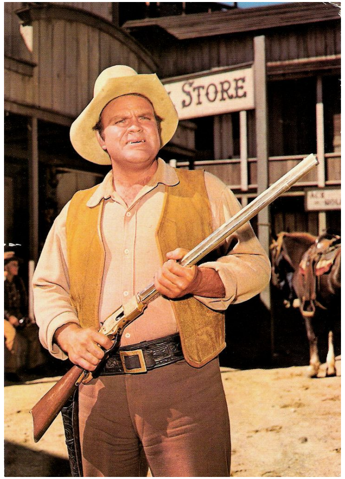
Generate these in counting house when data is taken

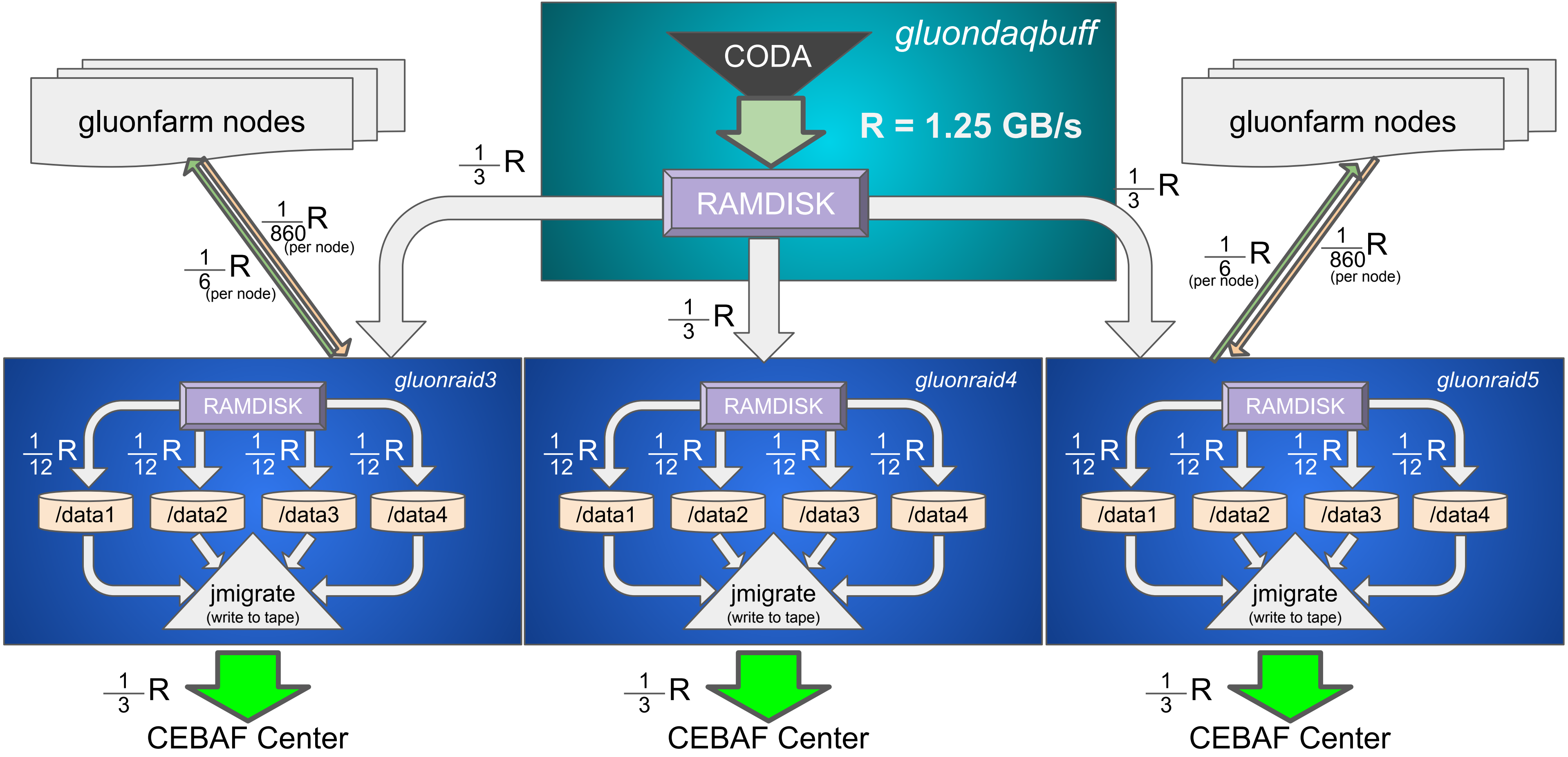
- reduce tape drive usage
- reduce Lustre activity
- reduce time waiting for skims

** not all ps or DIRC triggers saved*

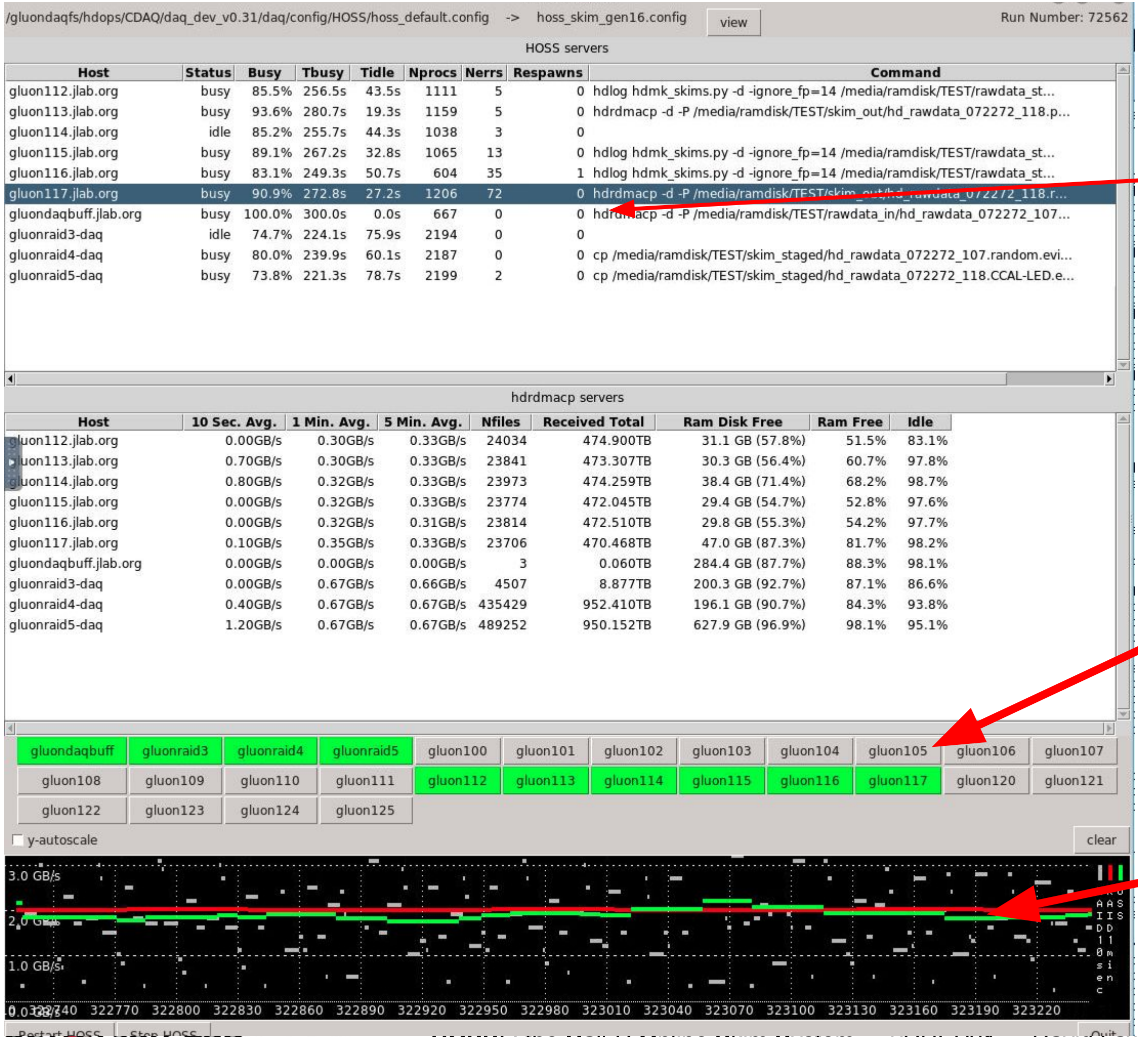
HOSS uses ...

- RAM disks for temporary, fast file staging
- RDMA (Remote Direct Memory Access) over IB
  - fast file transfer
  - implemented as Linux systemd service
- Python for process management
  - single central config. file defines rules for files transfers and processing
- ZeroMQ for monitoring
  - pub-sub system for both RDMA processes python script status
- MySQL DB for storing file path through system
  - Also stores full trigger counts as byproduct




gluondaqbuff
CODA
R = 1.25 GB/s
RAMDISK
gluonfarm nodes
gluonfarm nodes
$\frac{1}{3}$R
$\frac{1}{3}$R
$\frac{1}{3}$R
$\frac{1}{860}$R (per node)
$\frac{1}{6}$R (per node)
$\frac{1}{6}$R (per node)
$\frac{1}{860}$R (per node)
gluonraid3
gluonraid4
gluonraid5
RAMDISK
$\frac{1}{12}$R
/data1
/data2
/data3
/data4
jmigrate
(write to tape)
$\frac{1}{3}$R
CEBAF Center
CEBAF Center
CEBAF Center

HOSS Status GUI



/gluondaqfs/hdops/CDAQ/daq_dev_v0.31/daq/config/HOSS/hoss_default.config -> hoss_skim_gen16.config view Run Number: 72562

HOSS servers

Host	Status	Busy	Tbusy	Tidle	Nprocs	Nerrs	Respawns	Command
gluon112.jlab.org	busy	85.5%	256.5s	43.5s	1111	5	0	hdlog hdmk_skims.py -d -ignore_fp=14 /media/ramdisk/TEST/rawdata_st...
gluon113.jlab.org	busy	93.6%	280.7s	19.3s	1159	5	0	hdrdmacp -d -P /media/ramdisk/TEST/skim_out/hd_rawdata_072272_118.p...
gluon114.jlab.org	idle	85.2%	255.7s	44.3s	1038	3	0	
gluon115.jlab.org	busy	89.1%	267.2s	32.8s	1065	13	0	hdlog hdmk_skims.py -d -ignore_fp=14 /media/ramdisk/TEST/rawdata_st...
gluon116.jlab.org	busy	83.1%	249.3s	50.7s	604	35	1	hdlog hdmk_skims.py -d -ignore_fp=14 /media/ramdisk/TEST/rawdata_st...
gluon117.jlab.org	busy	90.9%	272.8s	27.2s	1206	72	0	hdrdmacp -d -P /media/ramdisk/TEST/skim_out/hd_rawdata_072272_118.r...
gluondaqbuff.jlab.org	busy	100.0%	300.0s	0.0s	667	0	0	hdrdmacp -d -P /media/ramdisk/TEST/rawdata_in/hd_rawdata_072272_107...
gluonraid3-daq	idle	74.7%	224.1s	75.9s	2194	0	0	
gluonraid4-daq	busy	80.0%	239.9s	60.1s	2187	0	0	cp /media/ramdisk/TEST/skim_staged/hd_rawdata_072272_107.random.evi...
gluonraid5-daq	busy	73.8%	221.3s	78.7s	2199	2	0	cp /media/ramdisk/TEST/skim_staged/hd_rawdata_072272_118.CCAL-LED.e...

hdrdmacp servers

Host	10 Sec. Avg.	1 Min. Avg.	5 Min. Avg.	Nfiles	Received Total	Ram Disk Free	Ram Free	Idle
gluon112.jlab.org	0.00GB/s	0.30GB/s	0.33GB/s	24034	474.900TB	31.1 GB (57.8%)	51.5%	83.1%
gluon113.jlab.org	0.70GB/s	0.30GB/s	0.33GB/s	23841	473.307TB	30.3 GB (56.4%)	60.7%	97.8%
gluon114.jlab.org	0.80GB/s	0.32GB/s	0.33GB/s	23973	474.259TB	38.4 GB (71.4%)	68.2%	98.7%
gluon115.jlab.org	0.00GB/s	0.32GB/s	0.33GB/s	23774	472.045TB	29.4 GB (54.7%)	52.8%	97.6%
gluon116.jlab.org	0.00GB/s	0.32GB/s	0.31GB/s	23814	472.510TB	29.8 GB (55.3%)	54.2%	97.7%
gluon117.jlab.org	0.10GB/s	0.35GB/s	0.33GB/s	23706	470.468TB	47.0 GB (87.3%)	81.7%	98.2%
gluondaqbuff.jlab.org	0.00GB/s	0.00GB/s	0.00GB/s	3	0.060TB	284.4 GB (87.7%)	88.3%	98.1%
gluonraid3-daq	0.00GB/s	0.67GB/s	0.66GB/s	4507	8.877TB	200.3 GB (92.7%)	87.1%	86.6%
gluonraid4-daq	0.40GB/s	0.67GB/s	0.67GB/s	435429	952.410TB	196.1 GB (90.7%)	84.3%	93.8%
gluonraid5-daq	1.20GB/s	0.67GB/s	0.67GB/s	489252	950.152TB	627.9 GB (96.9%)	98.1%	95.1%

🟩 gluondaqbuff	🟩 gluonraid3	🟩 gluonraid4	🟩 gluonraid5	gluon100	gluon101	gluon102	gluon103	gluon104	gluon105	gluon106	gluon107
gluon108	gluon109	gluon110	gluon111	🟩 gluon112	🟩 gluon113	🟩 gluon114	🟩 gluon115	🟩 gluon116	🟩 gluon117	gluon120	gluon121
gluon122	gluon123	gluon124	gluon125								

Errors and Respawns are more common on some servers.

System is fault tolerant.

HOSS Processes

hdrdmcp server Processes

Transfer rates recorded by hdrdmacp servers

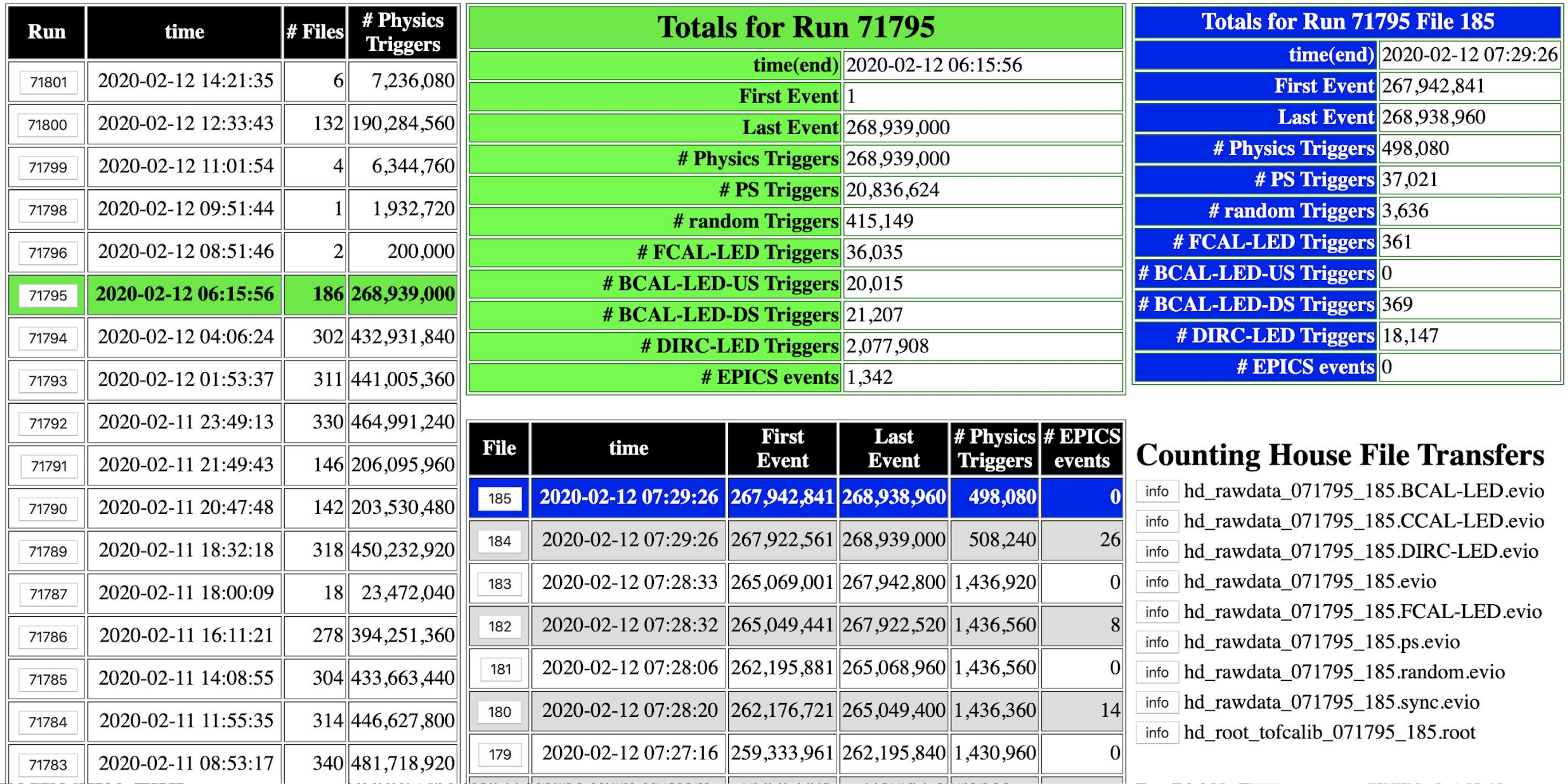
- green = good
- red = bad
- grey = not used

n.b. right click to restart
left click for log file

Transfer rates recorded by hdrdmacp servers

- green = transferred to skim nodes
- red = transferred to RAID (1min avg)
- grey = transferred to RAID (10 sec avg)
-

HOSS Run/File Info. DB



Run	time	# Files	# Physics Triggers
71801	2020-02-12 14:21:35	6	7,236,080
71800	2020-02-12 12:33:43	132	190,284,560
71799	2020-02-12 11:01:54	4	6,344,760
71798	2020-02-12 09:51:44	1	1,932,720
71796	2020-02-12 08:51:46	2	200,000
71795	**2020-02-12 06:15:56**	**186**	**268,939,000**
71794	2020-02-12 04:06:24	302	432,931,840
71793	2020-02-12 01:53:37	311	441,005,360
71792	2020-02-11 23:49:13	330	464,991,240
71791	2020-02-11 21:49:43	146	206,095,960
71790	2020-02-11 20:47:48	142	203,530,480
71789	2020-02-11 18:32:18	318	450,232,920
71787	2020-02-11 18:00:09	18	23,472,040
71786	2020-02-11 16:11:21	278	394,251,360
71785	2020-02-11 14:08:55	304	433,663,440
71784	2020-02-11 11:55:35	314	446,627,800
71783	2020-02-11 08:53:17	340	481,718,920

Totals for Run 71795	
time(end)	2020-02-12 06:15:56
First Event	1
Last Event	268,939,000
# Physics Triggers	268,939,000
# PS Triggers	20,836,624
# random Triggers	415,149
# FCAL-LED Triggers	36,035
# BCAL-LED-US Triggers	20,015
# BCAL-LED-DS Triggers	21,207
# DIRC-LED Triggers	2,077,908
# EPICS events	1,342

Totals for Run 71795 File 185	
time(end)	2020-02-12 07:29:26
First Event	267,942,841
Last Event	268,938,960
# Physics Triggers	498,080
# PS Triggers	37,021
# random Triggers	3,636
# FCAL-LED Triggers	361
# BCAL-LED-US Triggers	0
# BCAL-LED-DS Triggers	369
# DIRC-LED Triggers	18,147
# EPICS events	0

File	time	First Event	Last Event	# Physics Triggers	# EPICS events
185	**2020-02-12 07:29:26**	**267,942,841**	**268,938,960**	**498,080**	**0**
184	2020-02-12 07:29:26	267,922,561	268,939,000	508,240	26
183	2020-02-12 07:28:33	265,069,001	267,942,800	1,436,920	0
182	2020-02-12 07:28:32	265,049,441	267,922,520	1,436,560	8
181	2020-02-12 07:28:06	262,195,881	265,068,960	1,436,560	0
180	2020-02-12 07:28:20	262,176,721	265,049,400	1,436,360	14
179	2020-02-12 07:27:16	259,333,961	262,195,840	1,430,960	0

Counting House File Transfers

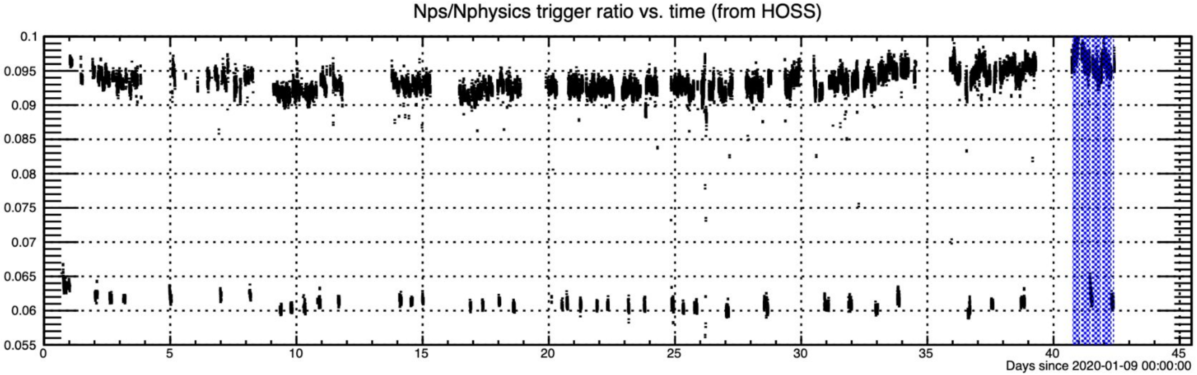
info	hd_rawdata_071795_185.BCAL-LED.evio
info	hd_rawdata_071795_185.CCAL-LED.evio
info	hd_rawdata_071795_185.DIRC-LED.evio
info	hd_rawdata_071795_185.evio
info	hd_rawdata_071795_185.FCAL-LED.evio
info	hd_rawdata_071795_185.ps.evio
info	hd_rawdata_071795_185.random.evio
info	hd_rawdata_071795_185.sync.evio
info	hd_root_tofcalib_071795_185.root

DB of Trigger Counts is Useful Beam Diagnostic

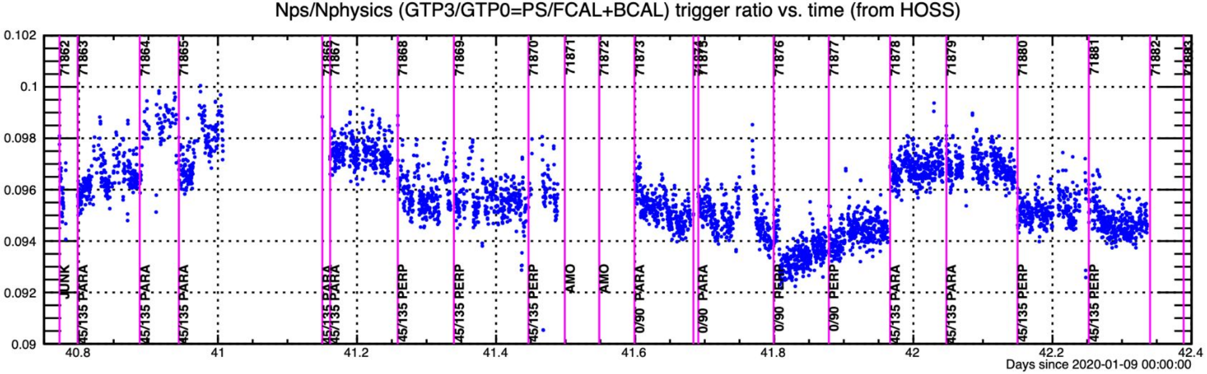
- A labor intensive analysis identified possible beam dependence on polarization direction

- Byproduct of HOSS system is exact count of various trigger types

- Data from HOSS DB showed effect clearly and even showed drift within a single ~4hr run



Nps/Nphysics trigger ratio vs. time (from HOSS)



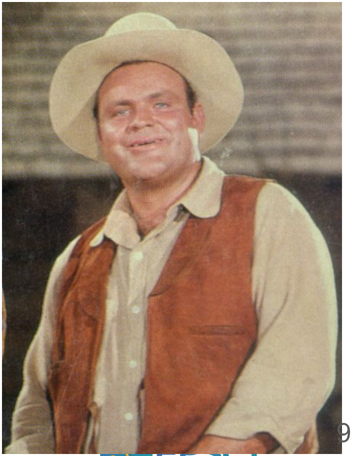
Nps/Nphysics (GTP3/GTP0=PS/FCAL+BCAL) trigger ratio vs. time (from HOSS)

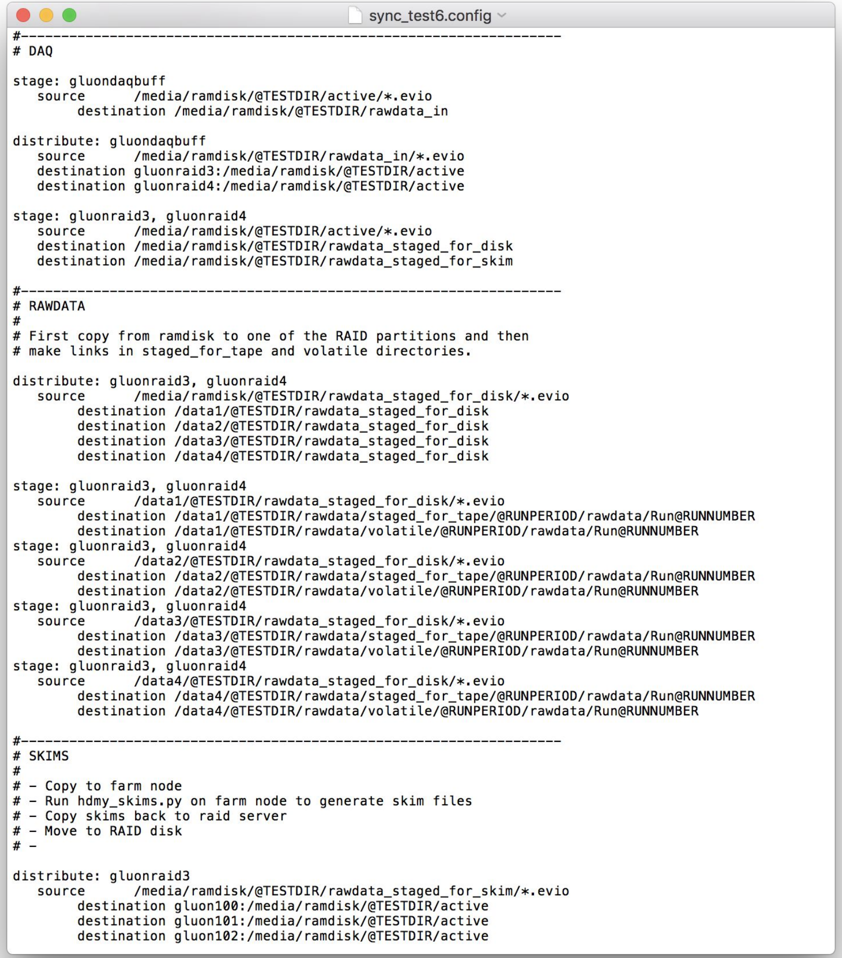
HOSS Summary

- New paradigm for raw data flow in Hall-D counting house
  - Splitting data stream at file level significantly reduces demand on individual hardware components *(e.g. RAID disks)*
  - System handles distribution of files over nodes and partitions as well as shallow copies via hard links *(large configuration file)*

- RDMA
  - hdrdmacp program written, tested, deployed as systemd service on gluons
  - >2PB cumulatively transferred through servers without crashing

- hdskims
  - Breaks skimming into 2 phases resulting x4 speedup
  - Automatically fills DB with trigger statistics for each file





```
#-------------------------------------------------------------------
# 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
```

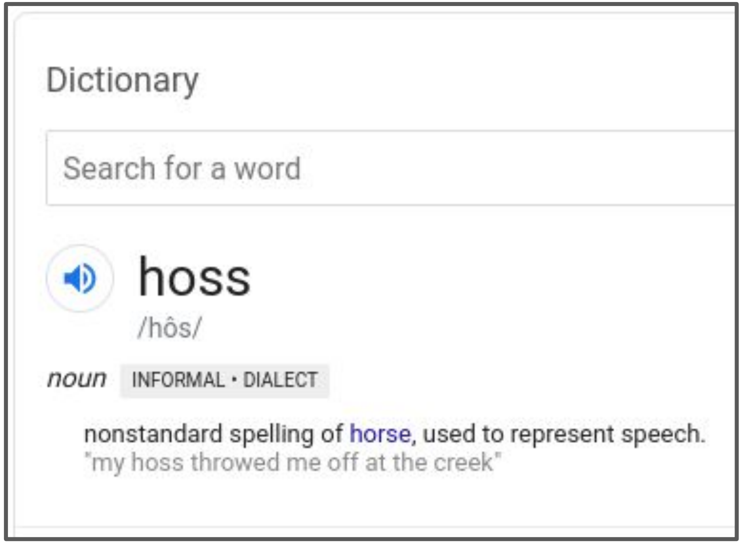
System driven by two types of operations:

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**

In this talk, “HOSS” will not refer to these ….



Dictionary

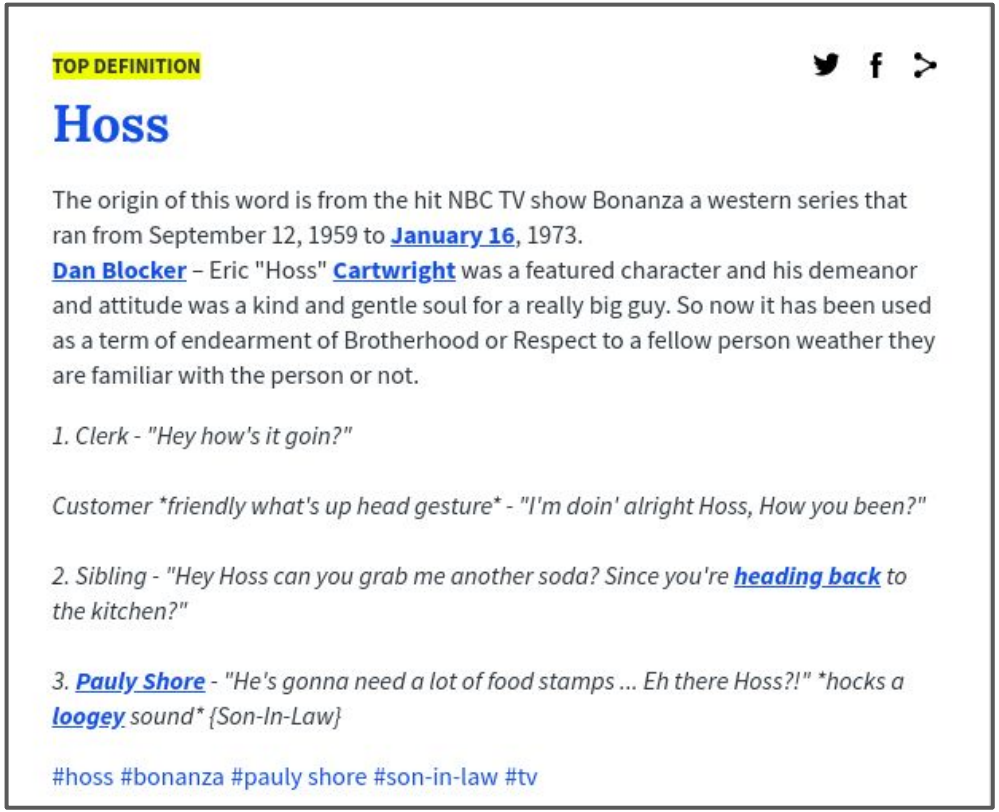
Search for a word

hoss

/hôs/

noun INFORMAL • DIALECT

nonstandard spelling of horse, used to represent speech.
"my hoss throwed me off at the creek"



TOP DEFINITION

Hoss

The origin of this word is from the hit NBC TV show Bonanza a western series that ran from September 12, 1959 to January 16, 1973.
Dan Blocker – Eric "Hoss" Cartwright was a featured character and his demeanor and attitude was a kind and gentle soul for a really big guy. So now it has been used as a term of endearment of Brotherhood or Respect to a fellow person weather they are familiar with the person or not.

1. Clerk - "Hey how's it goin?"

*Customer *friendly what's up head gesture* - "I'm doin' alright Hoss, How you been?"*

2. Sibling - "Hey Hoss can you grab me another soda? Since you're heading back to the kitchen?"

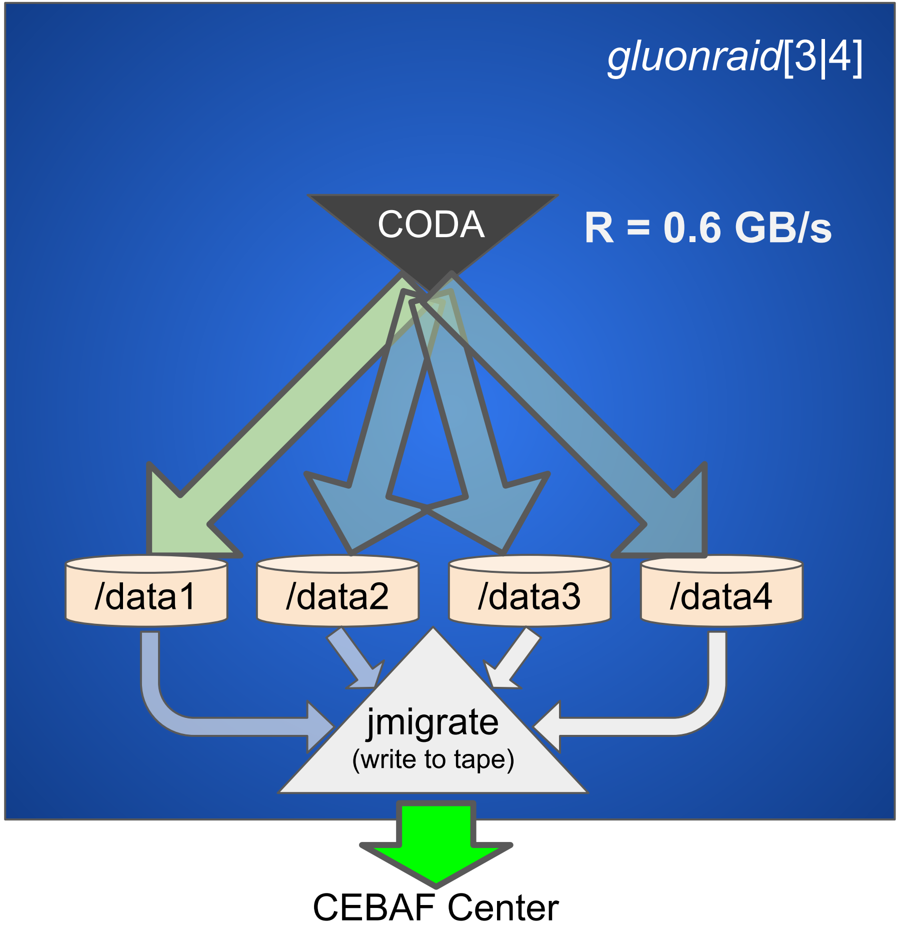
*3. Pauly Shore - "He's gonna need a lot of food stamps ... Eh there Hoss?!" *hocks a loogey sound* {Son-In-Law}*

#hoss #bonanza #pauly shore #son-in-law #tv

Hall-D Data recording

(for low intensity running)

- Transport into RAID server via 40Gbps ethernet
- Event builder and Recorder run 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



The Challenge of Online Skims:

- High Intensity Running
  - Larger data rates than ever seen in production
  - Single RAID partition cannot handle full rate (at least not stable)

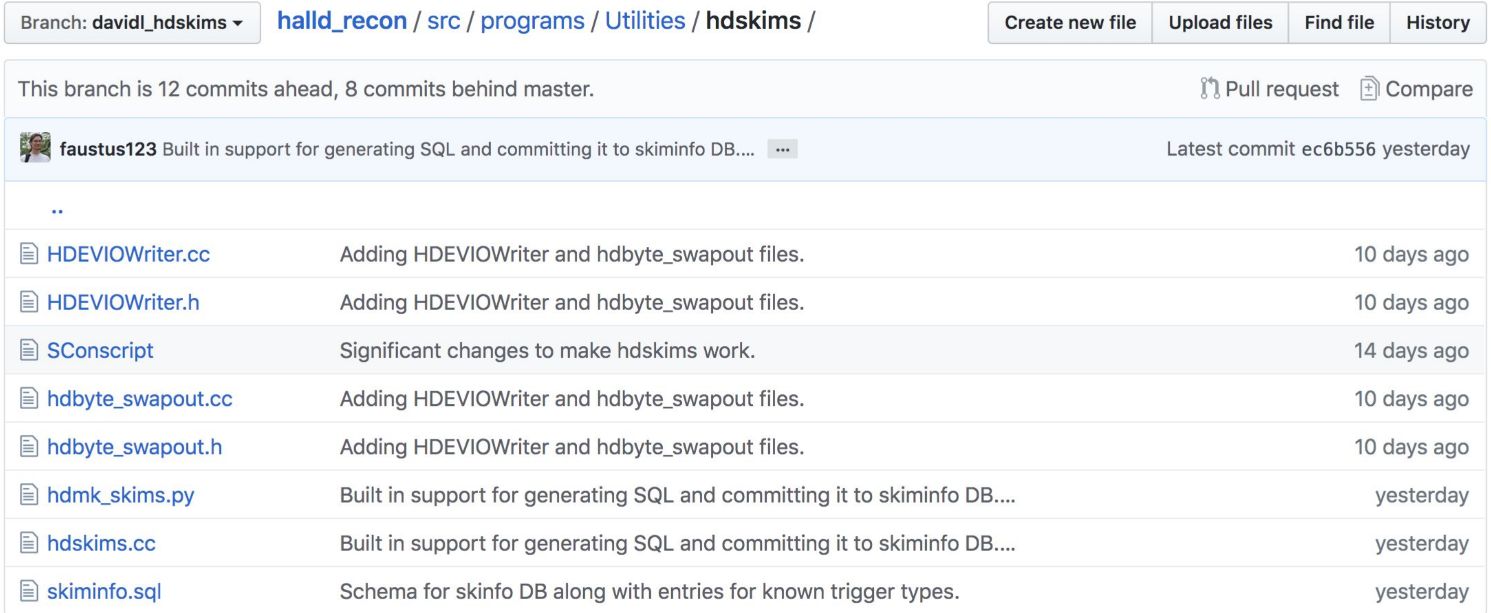
- Requires scanning entirety of every file
  - Never done even in low intensity era

- 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 produces **~3 files/min** (20GB)
- **hdskims**: skim through EVIO file and write blocks (40 events) containing at least one FP trigger to separate file (~10 sec using RAM disk)
- **hdmk_skims.py**: Run hdskims to create reduced EVIO file then run hd_ana with trigger_skims and ps_skim plugins to produce standard skim files (~40 sec)
- 20GB file processing time: <u>classic method=**4 min**</u> -- <u>new method=**1 min**</u>



halld_recon / src / programs / Utilities / **hdskims** /

This branch is 12 commits ahead, 8 commits behind master.

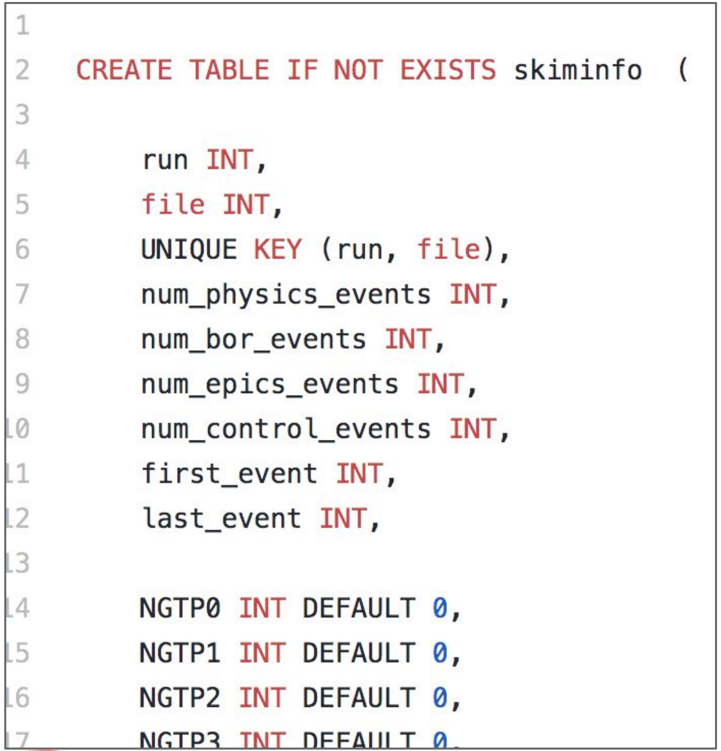
faustus123 Built in support for generating SQL and committing it to skiminfo DB.... Latest commit ec6b556 yesterday

..

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

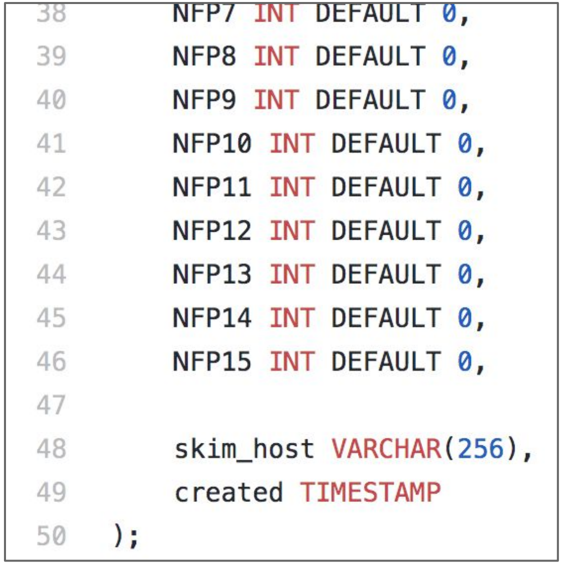


```
CREATE TABLE IF NOT EXISTS skiminfo  (

    run INT,
    file INT,
    UNIQUE KEY (run, file),
    num_physics_events INT,
    num_bor_events INT,
    num_epics_events INT,
    num_control_events INT,
    first_event INT,
    last_event INT,

    NGTP0 INT DEFAULT 0,
    NGTP1 INT DEFAULT 0,
    NGTP2 INT DEFAULT 0,
    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
- Counts for each trigger recorded



```
        NFP7 INT DEFAULT 0,
        NFP8 INT DEFAULT 0,
        NFP9 INT DEFAULT 0,
        NFP10 INT DEFAULT 0,
        NFP11 INT DEFAULT 0,
        NFP12 INT DEFAULT 0,
        NFP13 INT DEFAULT 0,
        NFP14 INT DEFAULT 0,
        NFP15 INT DEFAULT 0,

        skim_host VARCHAR(256),
        created TIMESTAMP
);
```

hdrdmacp - **Hall-D Remote Direct Memory Access CoPy**

- 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