Update on HPS Computing Operations and EVIO/LCIO Conversion

Menu:

- Extension of current disk space
- Projected current year needs for HPS computing
- Update on EVIO/LCIO conversion

Disk Space at SLAC for the HPS group

- Current 1.5TByte space becoming full and was already filled once and then cleaned

```
$ df /nfs/slac/g/hps
Filesystem           1K-blocks      Used Available Use% Mounted on surrey04a:/vol/vol1/g.hps
                     1572864000 1304617728 268246272  83% /a/surrey04a/vol/vol1/g.hps
```

Users consuming the most space: Matt (~700GB), Sho (~400GB)

- SLAC CD has accepted to allocate 10TB on an NFS server for us. The proposed path is simply /nfs/slac/g/hps2

Access to SLAC CPU

- SLAC uses the LSF batch system with balancing done through shares

SHARE_INFO_FOR: long/

USER/GROUP	SHARES	PRIORITY	STARTED	RESERVED	CPU_TIME	RUN_TIME
lcdprodgrp	836	278.667	0	0	0.0	0
lcd	418	139.333	0	0	0.0	0
rpgrp	418	137.393	0	0	217.9	0
rdgrp	342	113.982	0	0	2.4	0
glastgrp	8707	40.554	58	0	195628.8	263975
glastdata	4707	39.538	32	0	103115.2	140685
babarAll	26332	8.051	821	0	4149681.8	6637443
exoprodgrp	1000	2.535	76	0	841052.4	2094663
atlasgrp	34307	2.031	3700	0	29825262.0	90663412
glastusers	2000	1.520	310	0	1972853.6	15309514
AllUsers	2142	0.577	1053	0	2839731.5	9809620
cdmsdata	1000	0.574	485	0	1467043.5	256041
theorygrp	1000	0.571	486	0	1493049.4	1395119

Where do HPS members end up ...

- Those with no specified batch users group affiliation end up in All Users

```
[noric02] ~ $ bugroup | grep mgraham | awk '{print $1}'
babarAll
objyusers
[noric02] ~ $ bugroup | grep meeg | awk '{print $1}'
[noric02] ~ $ bugroup | grep omoreno | awk '{print $1}'
```

```
$ bugroup | grep homer | awk '{printf("%s ",$1)}'
babarSkim babarM2M babarTMSkim babarTMMerge babarOff babarBmixing babarBreco babarCharm babarCharmon
babarESemil babarIhbd babarIhps babarISemil babarLepbc babarQ2body babarRadpen babarSin2b babarTauqed babar2body
babar3body babarnorm babarAll genmpigrp objyusers pachyderms u-rhel4-32
```

Net SLAC Batch Resources

•

```
 $ source GeneralCPUs.sh
bal     #cores per CPU=4     #nodes =72      #cores =288     tot cores=288   CPUF=10         tot CPUF=2880
boe     #cores per CPU=4     #nodes =133     #cores =532     tot cores=532   CPUF=10         tot CPUF=5320
dol     #cores per CPU=8     #nodes =38      #cores =304     tot cores=304   CPUF=15.61      tot CPUF=4745.44
fel     #cores per CPU=8     #nodes =291     #cores =2328    tot cores=2328  CPUF=11         tot CPUF=25608
heq     #cores per CPU=8     #nodes =192     #cores =1536    tot cores=1536  CPUF=14.58      tot CPUF=22394.9
kis     #cores per CPU=4     #nodes =68      #cores =272     tot cores=272   CPUF=10         tot CPUF=2720
yil     #cores per CPU=4     #nodes =154     #cores =616     tot cores=616   CPUF=8.46       tot CPUF=5211.36
NET cores = 5876        NET CPUF = 68879.7
```

HPS Projected Computing needs urgently need to plan the FY12/13 budget

- First draft of document prepared and needs feedback from HPS

Computing Resource Needs for the HPS 2012 Test Run
24/01/12

by the HPS Software Group

Introduction:

In 2012 the HPS experiment will run around the end of the first quarter collecting data at Jefferson Labs and the analyzing it and making corresponding simulated data at SLAC and UNH. The needs of the experiment will be driven by:

- the data storage for the raw and reconstructed data
- simulated data
- n-tuples produced by analysts
- computing cycles for the simulation and then reconstruction of both the real and simulated data
- computing cycles for the analysis work

These needs are quantified below:

Production needs:

The raw data event sizes per subdetector are provided by the test run proposal at:
https://confluence.slac.stanford.edu/download/attachments/86676777/HPSTestRunProposal-February18.pdf?version=1&

	Event Size (bytes)
ECAL	504
SVT	2821
Total	3325

The trigger rate expected for the first week is 10KHz during commissioning and then upto 50KHz. 30KHz is used as the averaget trigger rate:

	frequency	pass thru rate	Hz
Trigger run1	30000		30000
L3 run1		1	30000
			Bytes/sec
Data Output rate run1			9.98E+07

The net number events for the requested two week run time is:

	runtime(days) per run	uptime fraction	Events
total events	14	50.00%	3.63E+10

Note that I have included a 50% inefficiency since it is very unlikely that we would achieve the maximum data rate early in this two week test run.

Thus the total raw data volume expected is:

Total Data Volume	
60.33	Tbytes

The estimated reconstructed data volume is:

	passing fraction				
Filter	100.00%			3.63E+10	events per pass
		X raw	Size	Net Size	
Recon Data		2	60.33	120.66	Tbytes
MC Data		0.1	90.49	9.05	Tbytes
and the net reconstructed data volume will be				129.71	Tbytes

and the estimated amount of CPU needed using typical current cores is:

	Lumi in multiples of raw data cnt	#Raw Events	CPU (s)/event	CPU seconds
ReconSpecInt2000	2	3.63E+10	0.1	7.26E+09
MC SpecInt2000	0.1	3.63E+09	5	1.81E+09

Assuming the raw data and reconstructed data are all stored at SLAC then this yields 320 TBytes with 100 TBytes being imported from JLab. By the way, note that we may not keep the raw EVIO data at SLAC but it may be stored with the LCIO files and so I've kept it in the net count. Immediate access to the reconstructed data will be needed at SLAC which means that an import rate of at least 5 Tbytes/day will need to be maintained. The reconstructed data will need to be exported to UNH and simulation and reconstruction output from UNH will need to be imported. This is roughly 100 Tbytes being transferred in each direction but spread out over two months.

The max. see by ATLAS between SLAC and other US sites

The CPU required is ~9 Giga CPU seconds. However, note that the data reconstruction includes two passes. To accomplish the reconstructions and simulation over a period of eight weeks 9 Giga CPU seconds/(86400 seconds/day * 56 days) = 1860 cores. Assuming that UNH can provide about half **?** the processing power then the request to SLAC from HPS would be for an allocation allowing ~1000 cores to be occupied by HPS production jobs for two months.

The net storage (raw + reconstructed needed) needed is ~200 Tbytes. Since at this time it is unlikely to be able to have an allocation for this much disk space, the HPSS tape storage system will be used. At any given time at least 20% of this should be accessible directly from disk to reduce the load on the tape drives and minimize the staging wait. Thus a 40 Tbytes disk cache is needed managed with XROOTD/SCALA.

Means for avoiding potential costs will be discussed at the PPA Scientific Computing meeting this Friday

Analysis computing needs:

The time to analyze one event will likely be roughly equivalent to the reconstruction time but the events will be filtered. Assuming 10% of the events pass the filters the amount of CPU time needed will be 20 GigaEvents * 10% * 0.1 CPU seconds = 200 M CPU seconds. If this is spread over two months then an allocation of ~100 cores/day would satisfy the needs. Storage need estimates from discussions within the group are 10 Tbytes. At least a 50% contingency on this would be wise.

LCSIM support

Conditions database:

The HPS group requires the help of the Tony Johnson and his Data Handling Group group to prepare the conditions database (CDB) needed for the HPS test run in spring 2012. From discussions with Tony, it sounds like this can be accomplished by Tony advising an assistant (most likely from his group) over a period of one month. This would be building upon work previously done for EXO and CDMS. The CDB will use a MySQL instance to store run and time dependent constants for the HPS subdetectors as well as relevant accelerator, ambient and data quality monitoring constants. The MySQL database will need to be public so as to be accessible offsite in particular for sites at Jefferson Labs and the University of New Hampshire. The deadline for having a product that HPS can use to test storing the initial constants is February 10th, 2012. The FTEs needed for this one month project is 0.10.

Computing Expertise

- Finally there is a projection for the fraction of time needed from Tony, Jeremy and Norman

Feeding EVIO to LCSIM

- Input raw data in EVIO format
- Apply map from channel address to geometric representation of detector element

  see Mauri's slide
- Store mapped raw data in hash tables with keys

  class **accessrawhpsdata** extends SIOLCParameters

  with hash tables like

  Map<String,int[]> intMap

  also for floats and strings and can easily be extended to other types

Next

- Two possible paths are allowed
  - LCSIM drivers can access the header information and hash tables in memory (i.e. « get » functions are provided in the **accessrawhpsdata** class)
  - Functions exist in **accessrawhpsdata** for persisting the tables in LCIO format and retrieving them