Hall-D Software

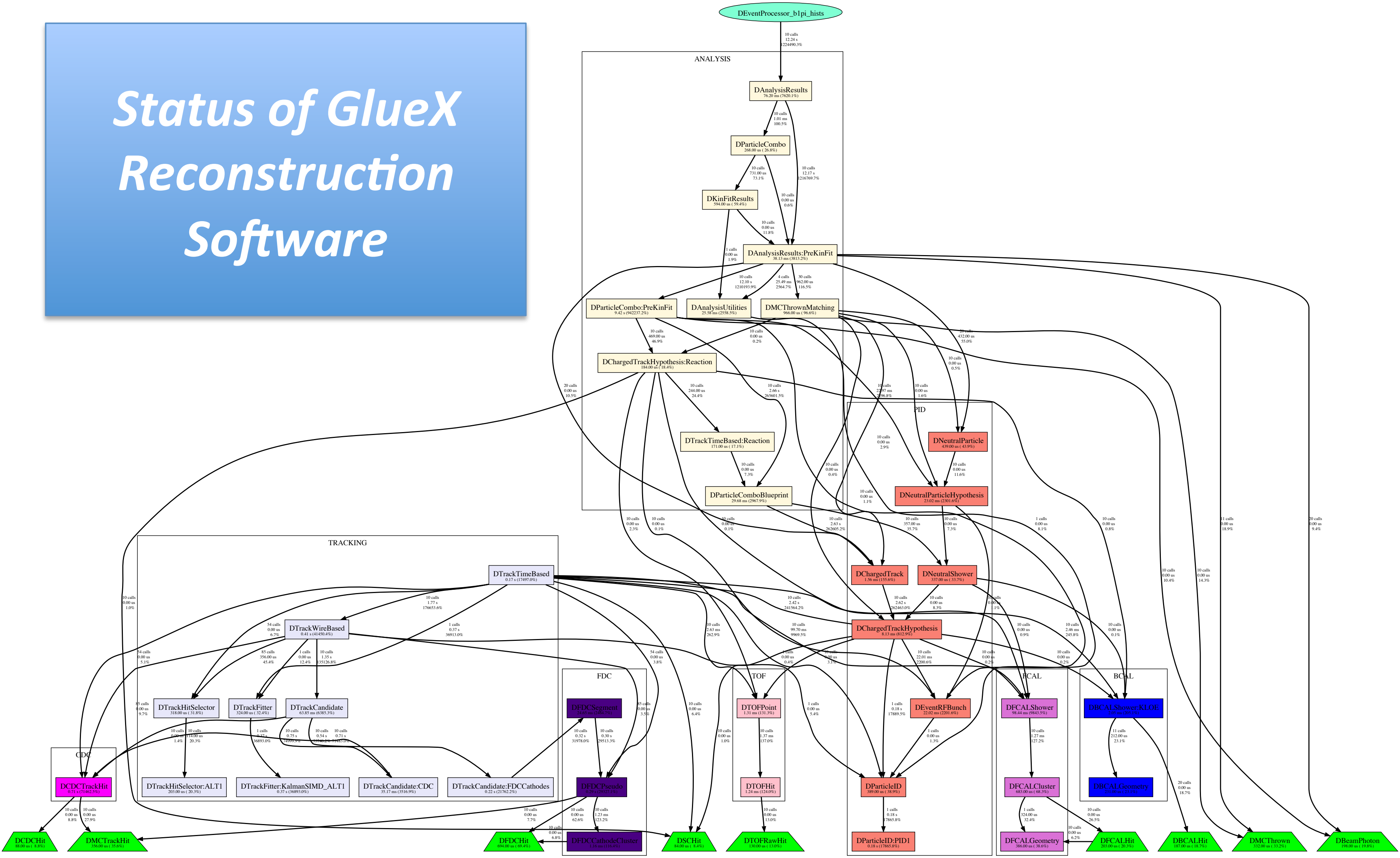
David Lawrence

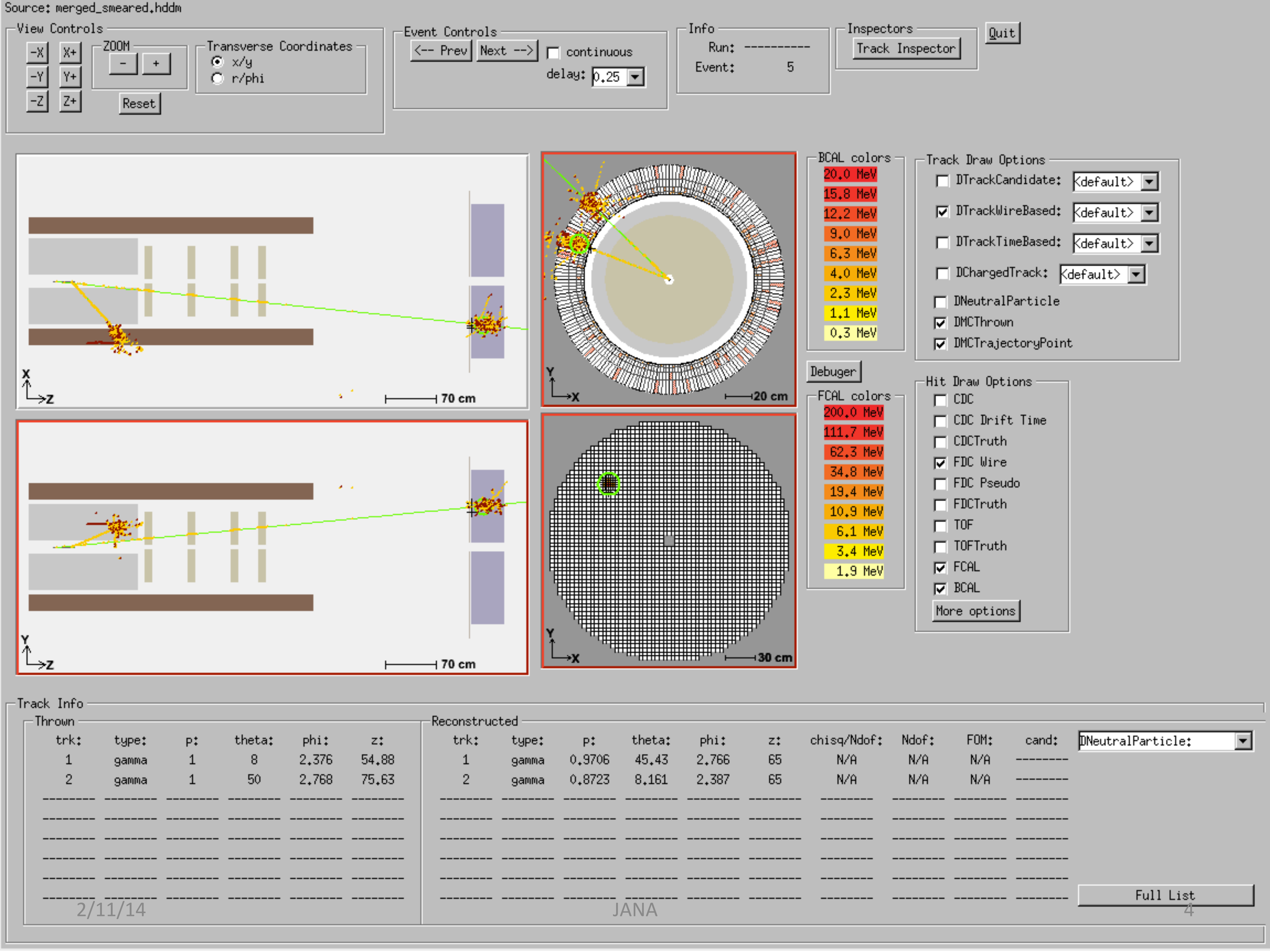
June 7, 2012

What is JANA?

- **J**Lab **ANA**lysis framework (more accurately, a *reconstruction)* framework
- C++ framework that formalizes the organization of algorithms and data transfer for event based processing
- Multi-threaded event processing
- Numerous additional features:
  - Configuration parameters
  - Web-based Resource retrieval
  - Plugins
  - Automatic ROOT tree creation
  - Calibration DB API

Status of GlueX Reconstruction Software





Source: merged_smeared.hddm

Transverse Coordinates
● x/y
○ r/phi

□ continuous
delay: 0.25

Run: ----------
Event: 5

BCAL colors

- 20.0 MeV
- 15.8 MeV
- 12.2 MeV
- 9.0 MeV
- 6.3 MeV
- 4.0 MeV
- 2.3 MeV
- 1.1 MeV
- 0.3 MeV

FCAL colors

- 200.0 MeV
- 111.7 MeV
- 62.3 MeV
- 34.8 MeV
- 19.4 MeV
- 10.9 MeV
- 6.1 MeV
- 3.4 MeV
- 1.9 MeV

Track Draw Options

- □ DTrackCandidate: <default>
- ☑ DTrackWireBased: <default>
- □ DTrackTimeBased: <default>
- □ DChargedTrack: <default>
- □ DNeutralParticle
- ☑ DMCThrown
- ☑ DMCTrajectoryPoint

Hit Draw Options

- □ CDC
- □ CDC Drift Time
- □ CDCTruth
- ☑ FDC Wire
- □ FDC Pseudo
- □ FDCTruth
- □ TOF
- □ TOFTruth
- ☑ FCAL
- ☑ BCAL

Track Info

Thrown

trk:	type:	p:	theta:	phi:	z:
1	gamma	1	8	2.376	54.88
2	gamma	1	50	2.768	75.63
--------	--------	--------	--------	--------	--------
--------	--------	--------	--------	--------	--------
--------	--------	--------	--------	--------	--------
--------	--------	--------	--------	--------	--------
--------	--------	--------	--------	--------	--------
--------	--------	--------	--------	--------	--------

Reconstructed

DNeutralParticle:

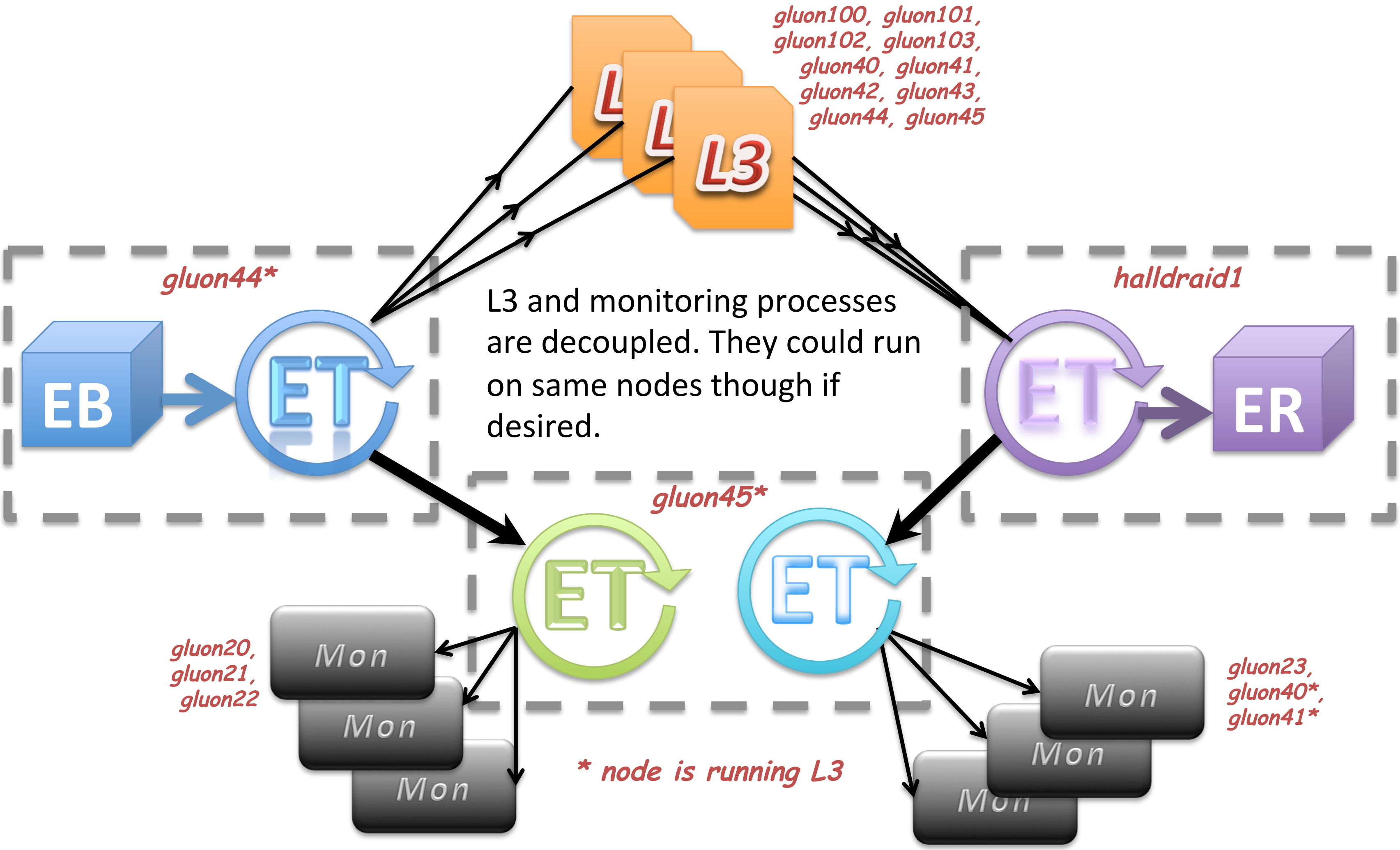
trk:	type:	p:	theta:	phi:	z:	chisq/Ndof:	Ndof:	FOM:	cand:
1	gamma	0.9706	45.43	2.766	65	N/A	N/A	N/A	--------
2	gamma	0.8723	8.161	2.387	65	N/A	N/A	N/A	--------
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------
--------	--------	--------	--------	--------	--------	--------	--------	--------	--------

Distributed Computing in GlueX

- Online systems
  - Monitoring farm *(ET)*
  - L3 trigger farm *(ET)*
- Offline systems
  - Raw data reconstruction/analysis *(Auger/PBS)*
  - Simulation *(Open Science Grid/Auger/PBS)*

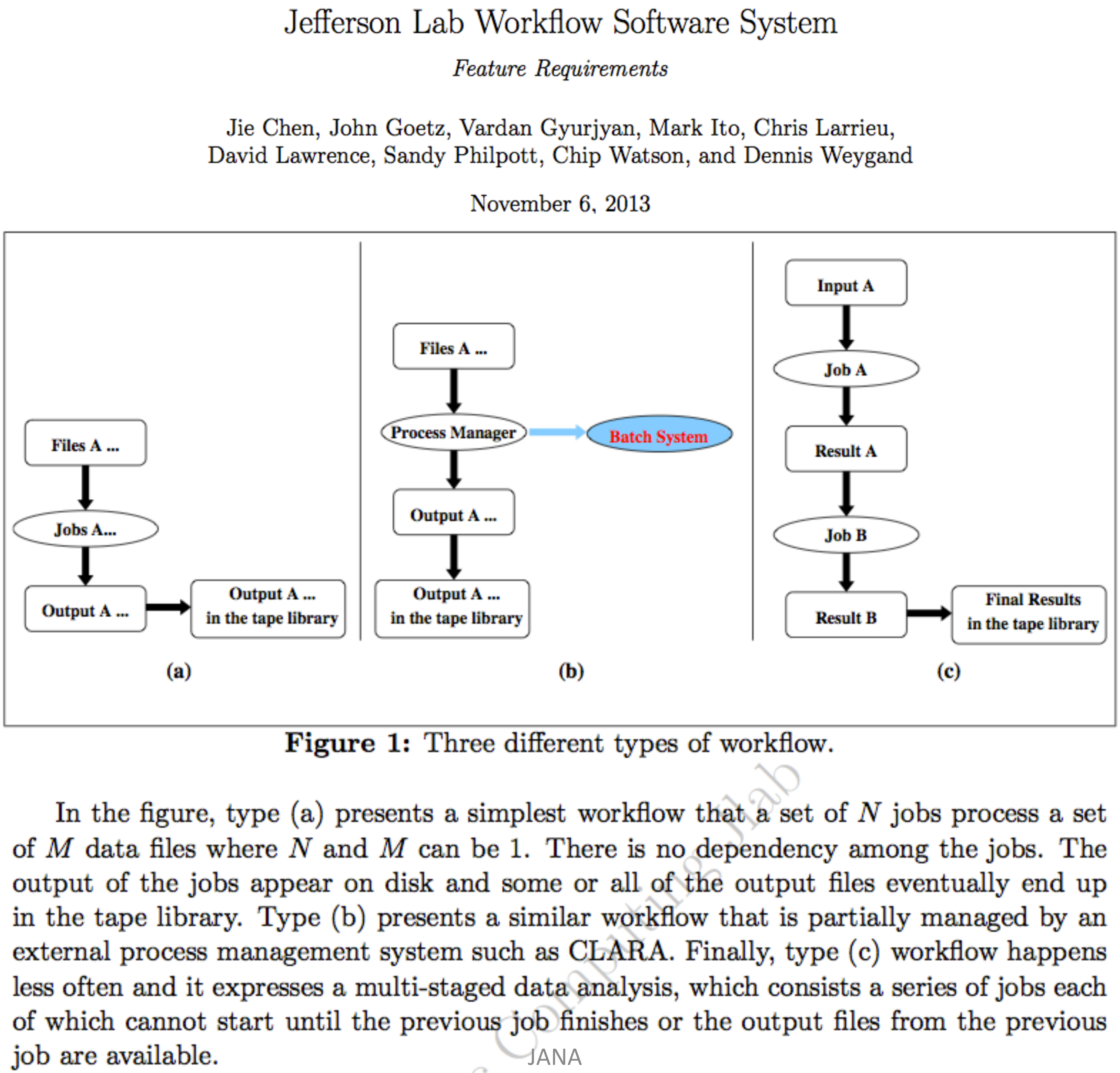
L3 and monitoring architecture

for 2013 Online Data Challenge



L3 and monitoring processes are decoupled. They could run on same nodes though if desired.

**n.b. all L3 machines connected via InfiniBand*



Jefferson Lab Workflow Software System

Feature Requirements

Jie Chen, John Goetz, Vardan Gyurjyan, Mark Ito, Chris Larrieu,
David Lawrence, Sandy Philpott, Chip Watson, and Dennis Weygand

November 6, 2013

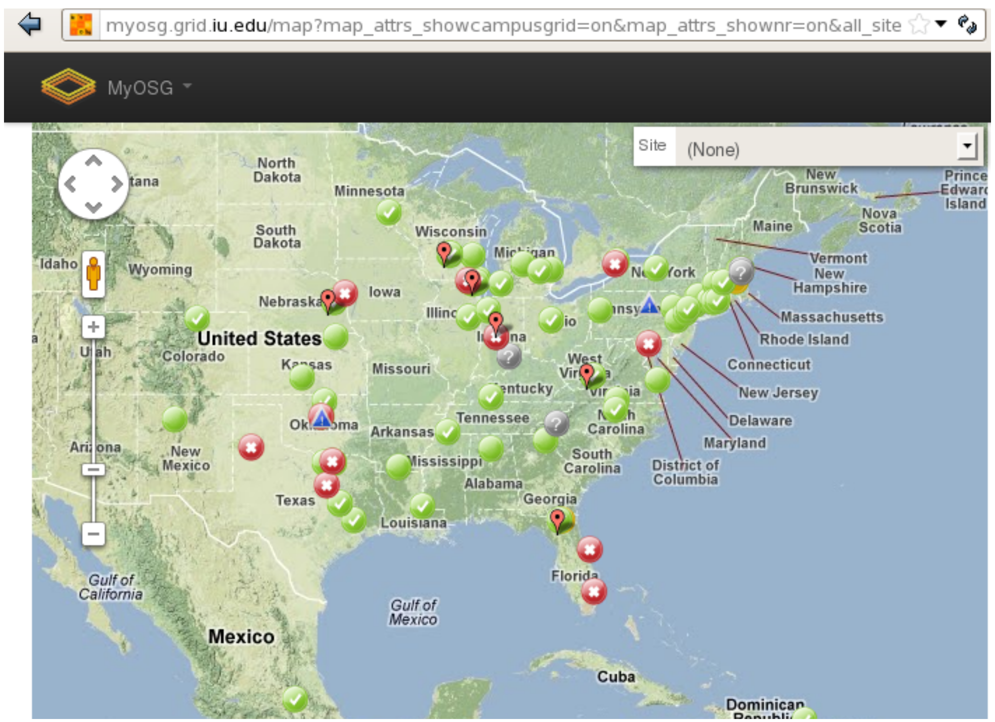
Files A ...
Jobs A...
Output A ...
Output A ... in the tape library
(a)
Files A ...
Process Manager
Batch System
Output A ...
Output A ... in the tape library
(b)
Input A
Job A
Result A
Job B
Result B
Final Results in the tape library
(c)

Figure 1: Three different types of workflow.

In the figure, type (a) presents a simplest workflow that a set of N jobs process a set of M data files where N and M can be 1. There is no dependency among the jobs. The output of the jobs appear on disk and some or all of the output files eventually end up in the tape library. Type (b) presents a similar workflow that is partially managed by an external process management system such as CLARA. Finally, type (c) workflow happens less often and it expresses a multi-staged data analysis, which consists a series of jobs each of which cannot start until the previous job finishes or the output files from the previous job are available.

OSG Context

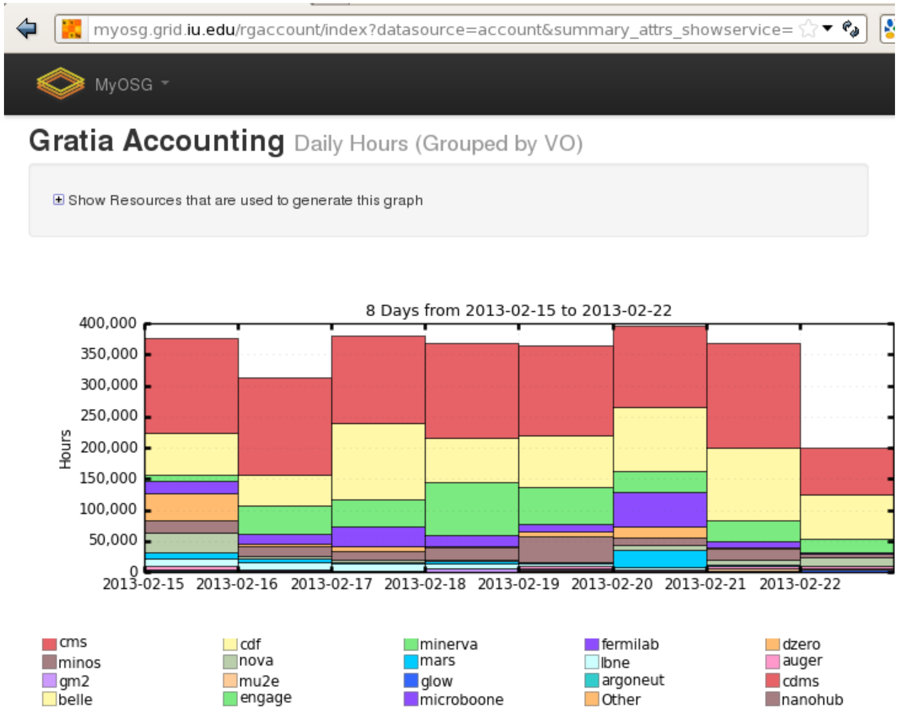
- Open Science Grid – founded 2004
- primary driver supporting LHC experiments in N/S America
- over 75,000 cores, running a distribution of Linux
- sites at 72 institutions including 42 universities, 90 sites (US, Brazil)
- centrally managed and operated by full-time staff (GOC @ I.U.)



myosg.grid.iu.edu/map?map_attrs_showcampusgrid=on&map_attrs_shownr=on&all_site

MyOSG

Site (None)



myosg.grid.iu.edu/rgaccount/index?datasource=account&summary_attrs_showservice=

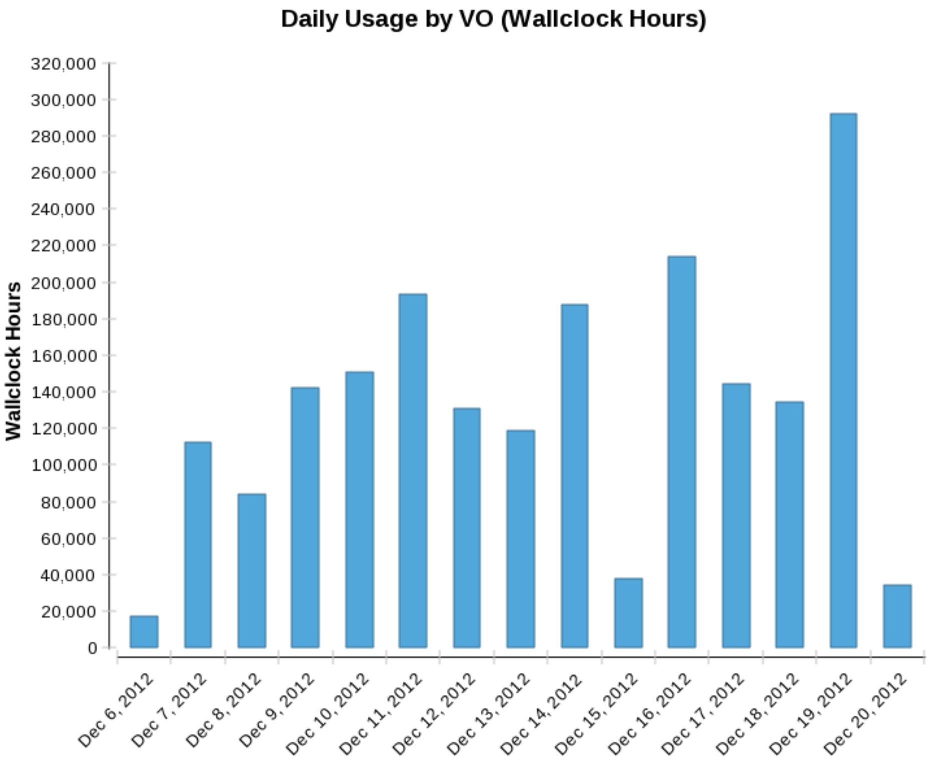
MyOSG

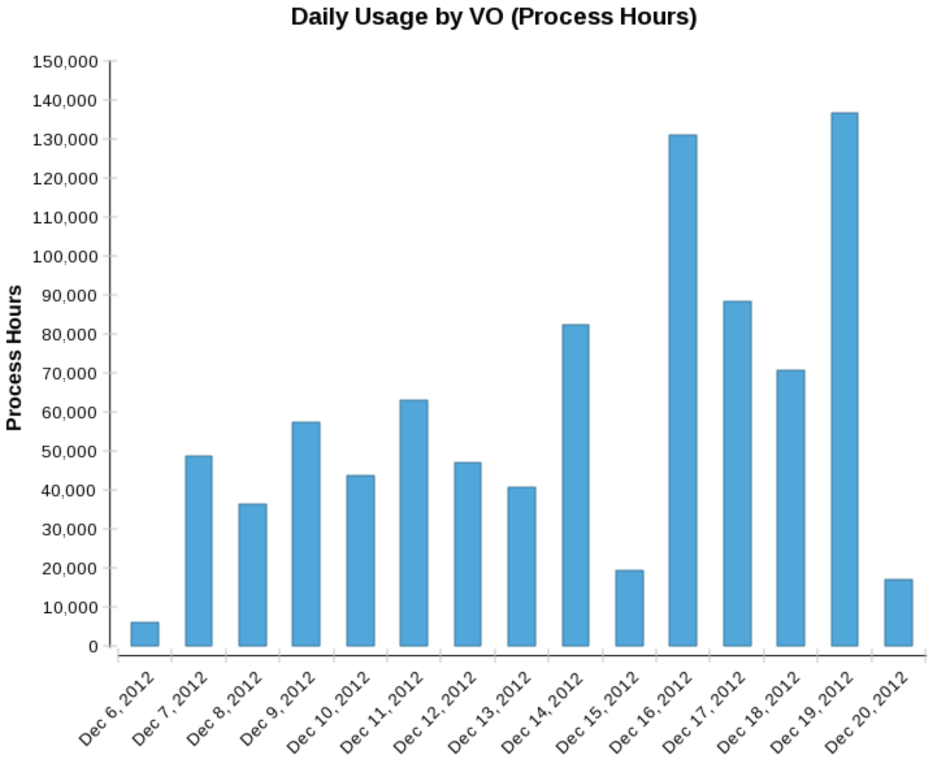
Gratia Accounting Daily Hours (Grouped by VO)

⊞ Show Resources that are used to generate this graph

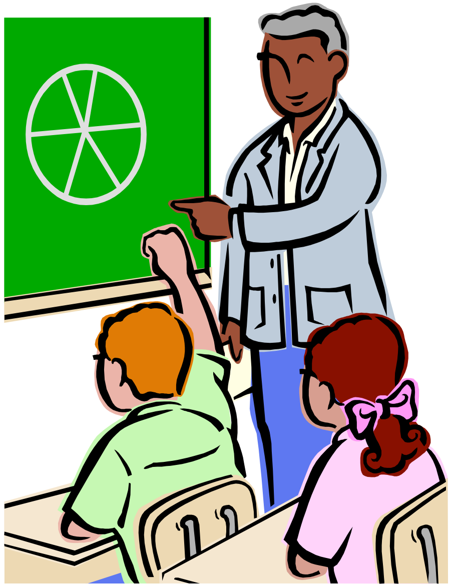
Achievements

- cpu availability was very high (>10,000 cores peak)
- production efficiency was not great (40 – 60%)
- part of inefficiency is due to pre-emption (opportunistic)
- understanding sources of inefficiency is reason why we stopped @5B events

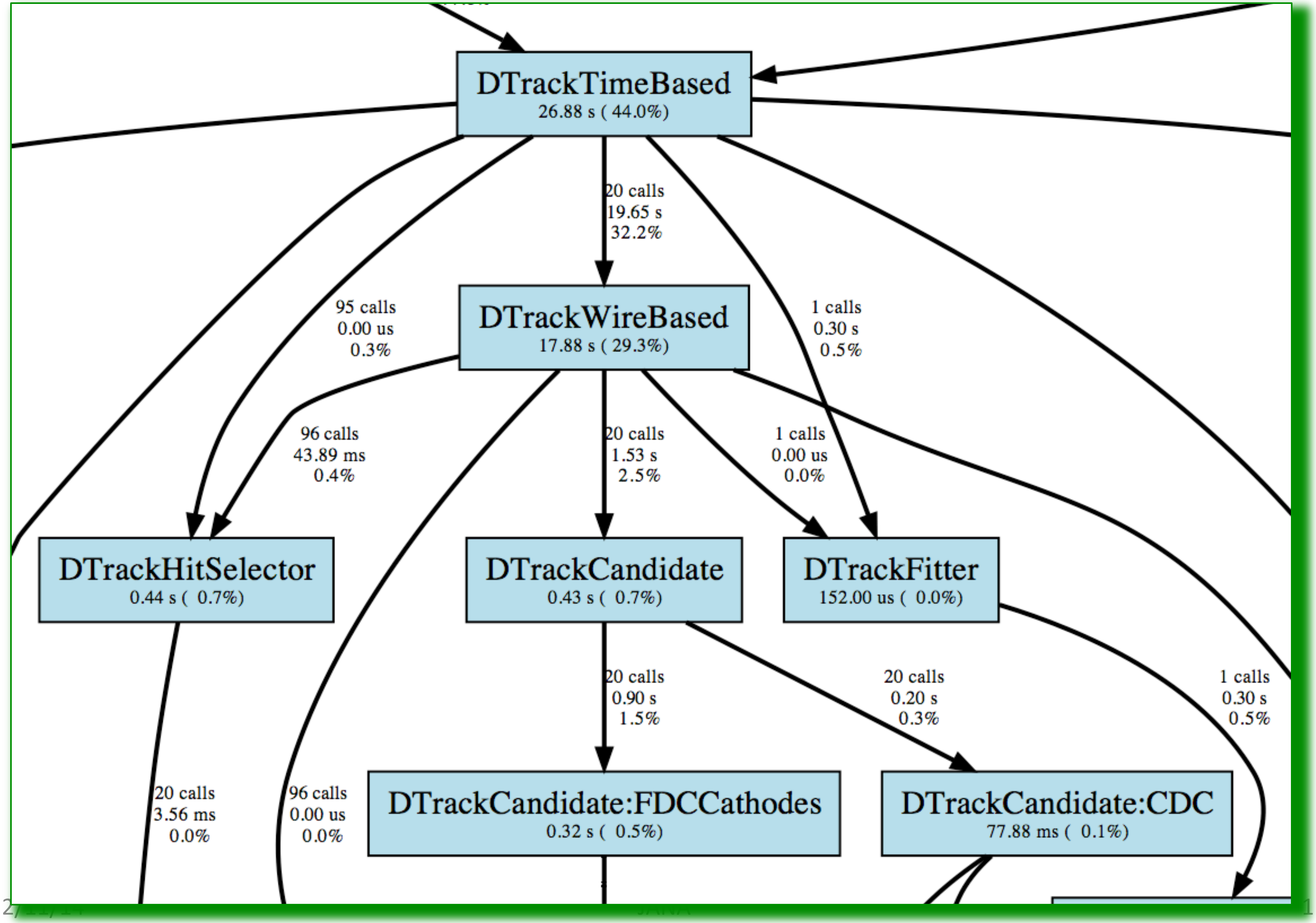




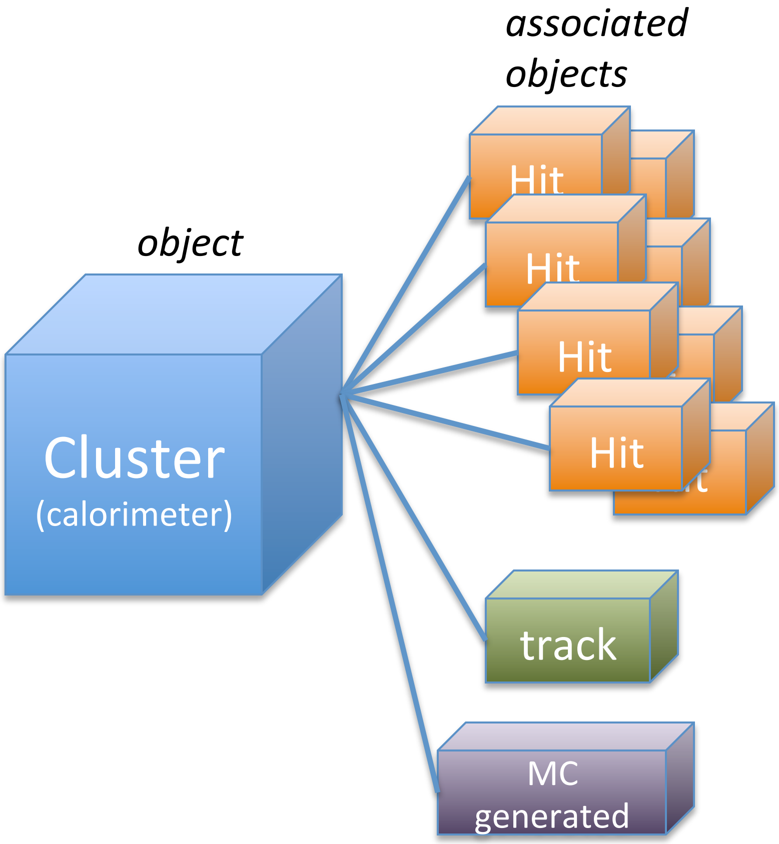
Backup Slides



A closer look at *janadot*



Associated Objects



- A data object may be associated with any number of other data objects having a mixture of types
  - Each data object has a list of “associated objects” that can be probed using a similar access mechanism as for event-level object requests

```
vector<const DCluster*> clusters;
loop->Get(clusters);
for(uint i=0; i<clusters.size(); i++)
{
     vector<const DHit*> hits;
     clusters[i]->Get(hits);
     // Do something with hits …
}
```

Configuration Parameters

in a factory's init method one might write …

Variables are data members of factory class

```
MIN_SEED_HITS = 4;
MAX_STEP_SIZE = 3.0; // cm
```

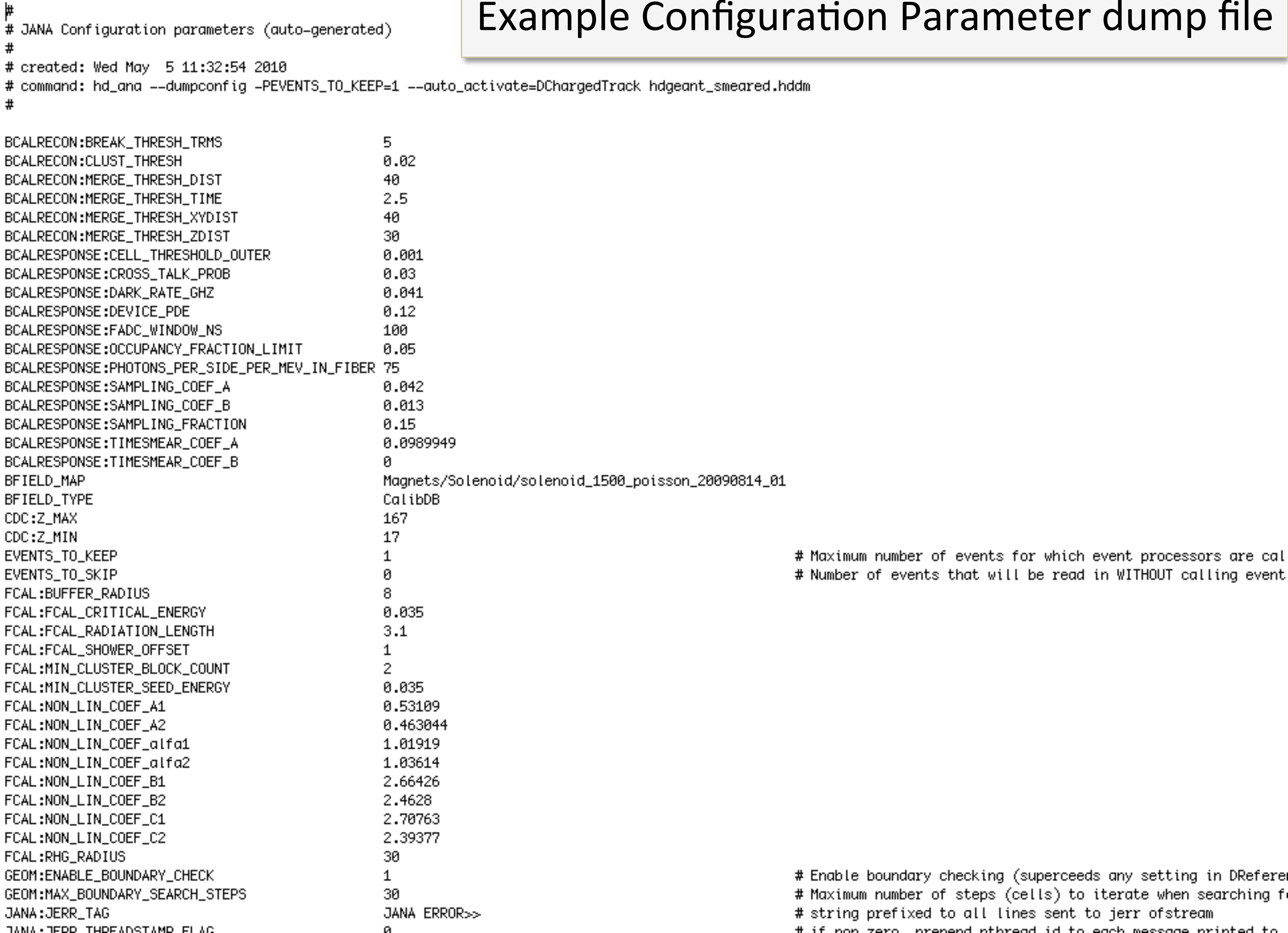
Value may be overwritten if user specifies a value at run time

```
gPARMS->SetDefaultParameter("TRKFIND:MIN_SEED_HITS",MIN_SEED_HITS);
gPARMS->SetDefaultParameter("TRK:MAX_STEP_SIZE" , MAX_STEP_SIZE
     , "Maximum step size in cm to take when swimming a track with adaptive step sizes");
```

***NEW:** Optional 3rd argument allows description to be stored with parameter*

- Parameters can be set via command line or configuration file
  - Complete list of parameters can be dumped using option `--dumpconfig`
    - Parameters can read in using option `--config=`*`filename`*

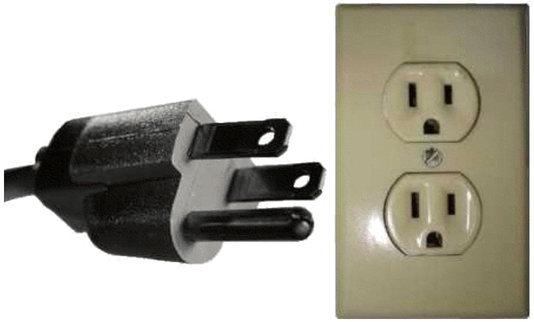
Example Configuration Parameter dump file



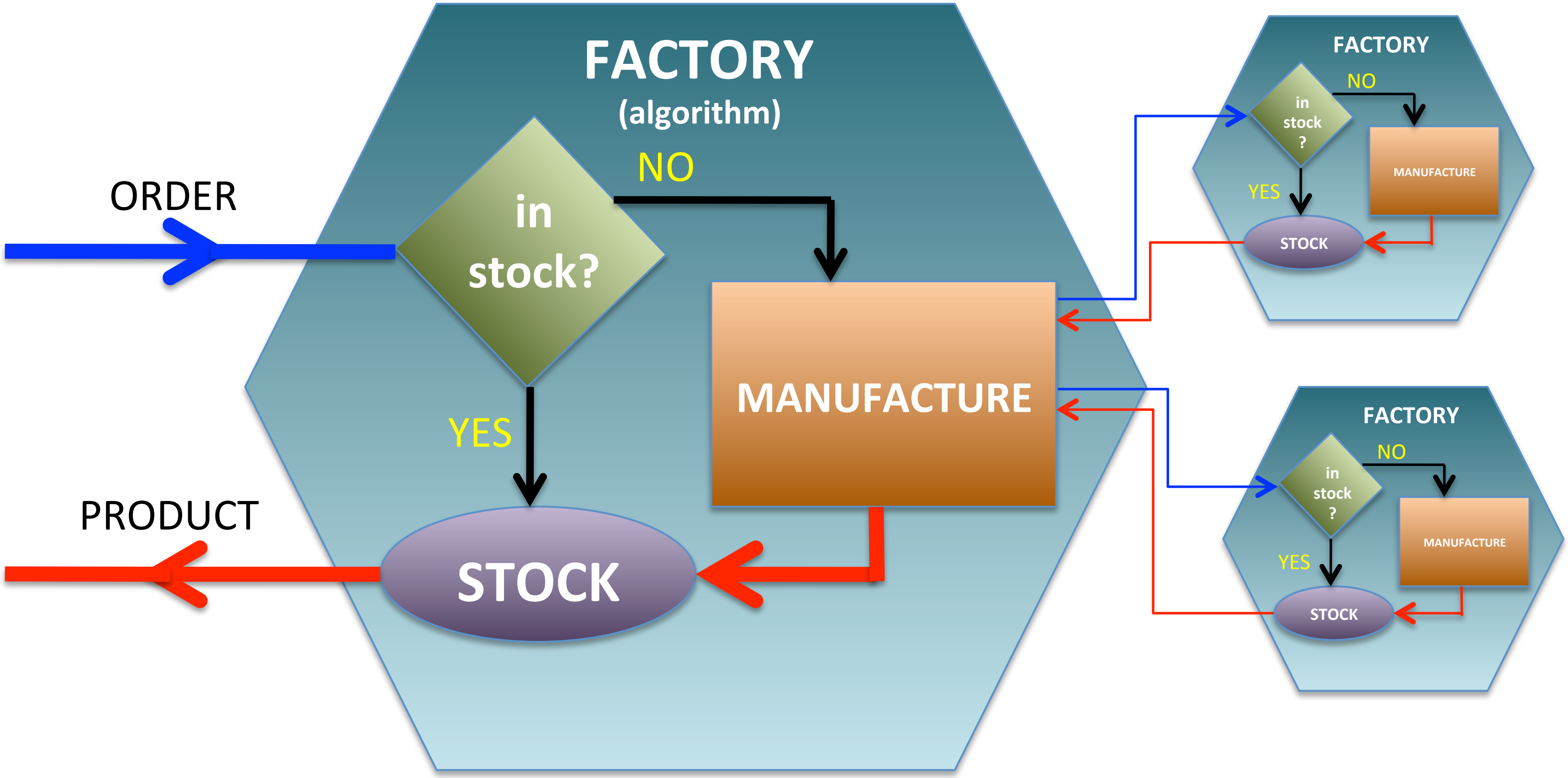
```
#
# JANA Configuration parameters (auto-generated)
#
# created: Wed May  5 11:32:54 2010
# command: hd_ana --dumpconfig -PEVENTS_TO_KEEP=1 --auto_activate=DChargedTrack hdgeant_smeared.hddm
#

BCALRECON:BREAK_THRESH_TRMS                    5
BCALRECON:CLUST_THRESH                         0.02
BCALRECON:MERGE_THRESH_DIST                    40
BCALRECON:MERGE_THRESH_TIME                    2.5
BCALRECON:MERGE_THRESH_XYDIST                  40
BCALRECON:MERGE_THRESH_ZDIST                   30
BCALRESPONSE:CELL_THRESHOLD_OUTER              0.001
BCALRESPONSE:CROSS_TALK_PROB                   0.03
BCALRESPONSE:DARK_RATE_GHZ                     0.041
BCALRESPONSE:DEVICE_PDE                        0.12
BCALRESPONSE:FADC_WINDOW_NS                    100
BCALRESPONSE:OCCUPANCY_FRACTION_LIMIT          0.05
BCALRESPONSE:PHOTONS_PER_SIDE_PER_MEV_IN_FIBER 75
BCALRESPONSE:SAMPLING_COEF_A                   0.042
BCALRESPONSE:SAMPLING_COEF_B                   0.013
BCALRESPONSE:SAMPLING_FRACTION                 0.15
BCALRESPONSE:TIMESMEAR_COEF_A                  0.0989949
BCALRESPONSE:TIMESMEAR_COEF_B                  0
BFIELD_MAP                                     Magnets/Solenoid/solenoid_1500_poisson_20090814_01
BFIELD_TYPE                                    CalibDB
CDC:Z_MAX                                      167
CDC:Z_MIN                                      17
EVENTS_TO_KEEP                                 1                                                  # Maximum number of events for which event processors are cal
EVENTS_TO_SKIP                                 0                                                  # Number of events that will be read in WITHOUT calling event
FCAL:BUFFER_RADIUS                             8
FCAL:FCAL_CRITICAL_ENERGY                      0.035
FCAL:FCAL_RADIATION_LENGTH                     3.1
FCAL:FCAL_SHOWER_OFFSET                        1
FCAL:MIN_CLUSTER_BLOCK_COUNT                   2
FCAL:MIN_CLUSTER_SEED_ENERGY                   0.035
FCAL:NON_LIN_COEF_A1                           0.53109
FCAL:NON_LIN_COEF_A2                           0.463044
FCAL:NON_LIN_COEF_alfa1                        1.01919
FCAL:NON_LIN_COEF_alfa2                        1.03614
FCAL:NON_LIN_COEF_B1                           2.66426
FCAL:NON_LIN_COEF_B2                           2.4628
FCAL:NON_LIN_COEF_C1                           2.70763
FCAL:NON_LIN_COEF_C2                           2.39377
FCAL:RHG_RADIUS                                30
GEOM:ENABLE_BOUNDARY_CHECK                     1                                                  # Enable boundary checking (superceeds any setting in DReferen
GEOM:MAX_BOUNDARY_SEARCH_STEPS                 30                                                 # Maximum number of steps (cells) to iterate when searching fo
JANA:JERR_TAG                                  JANA ERROR>>                                       # string prefixed to all lines sent to jerr ofstream
JANA:JERR_THREADSTAMP_FLAG                     0                                                  # if non zero, prepend pthread id to each message printed to
```

Plugins

- JANA supports plugins: pieces of code that can be attached to existing executables to extend or modify its behavior
- Plugins can be used to add:
  - Event Processors
  - Event sources
  - Factories (additional or replacements)
- Examples:
  - Plugins for creating DST skim files
    - Reconstruction is done once with output to multiple files
    - `hd_ana --PPLUGINS=kaon_skim,ppi+pi-_skim run012345.evio`
  - Plugins for producing subsystem histograms
    - Single ROOT file has histograms from several pieces of code
    - `hd_root --PPLUGINS=bcal_hists,cdc_hists,tof_hists ET:GlueX`



Factory Model

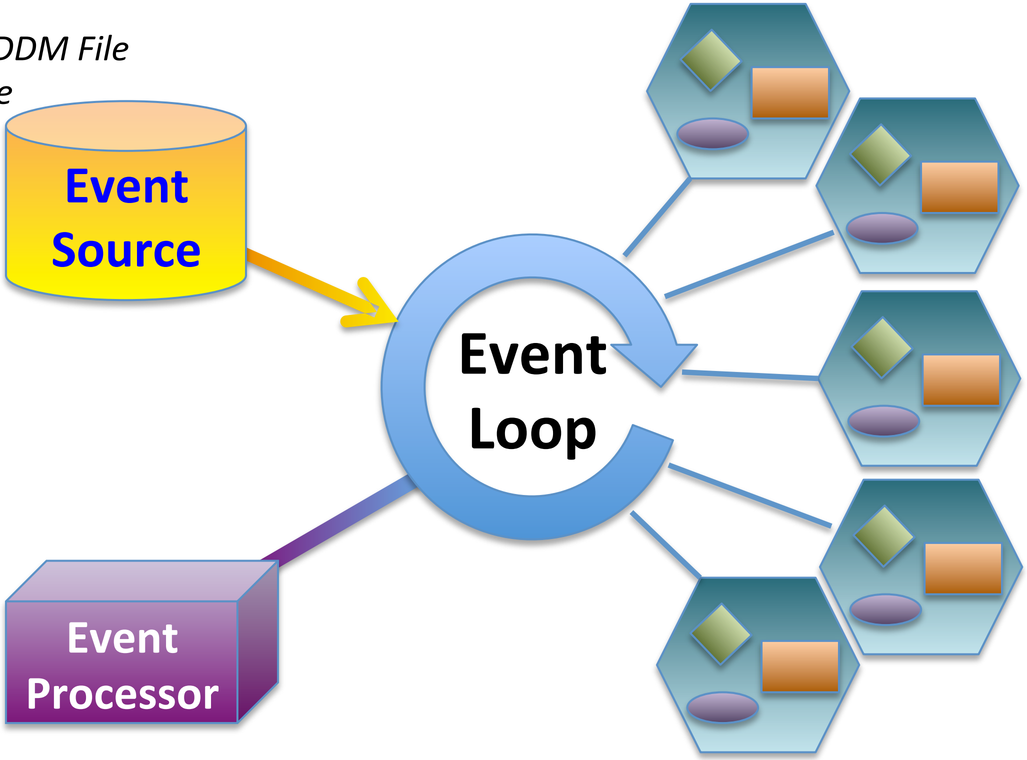


Data on demand = Don't do it unless you need it

Stock = Don't do it twice

Conservation of CPU cycles!

Complete Event Reconstruction



HDDM File

EVIO File

ET system

Web Service

Event Source

Event Loop

Event Processor

User supplied code

Fill histograms
Write DST
L3 trigger

Framework has a layer that directs object requests to the factory that completes it

Multiple algorithms (factories) may exist in the same program that produce the same type of data objects

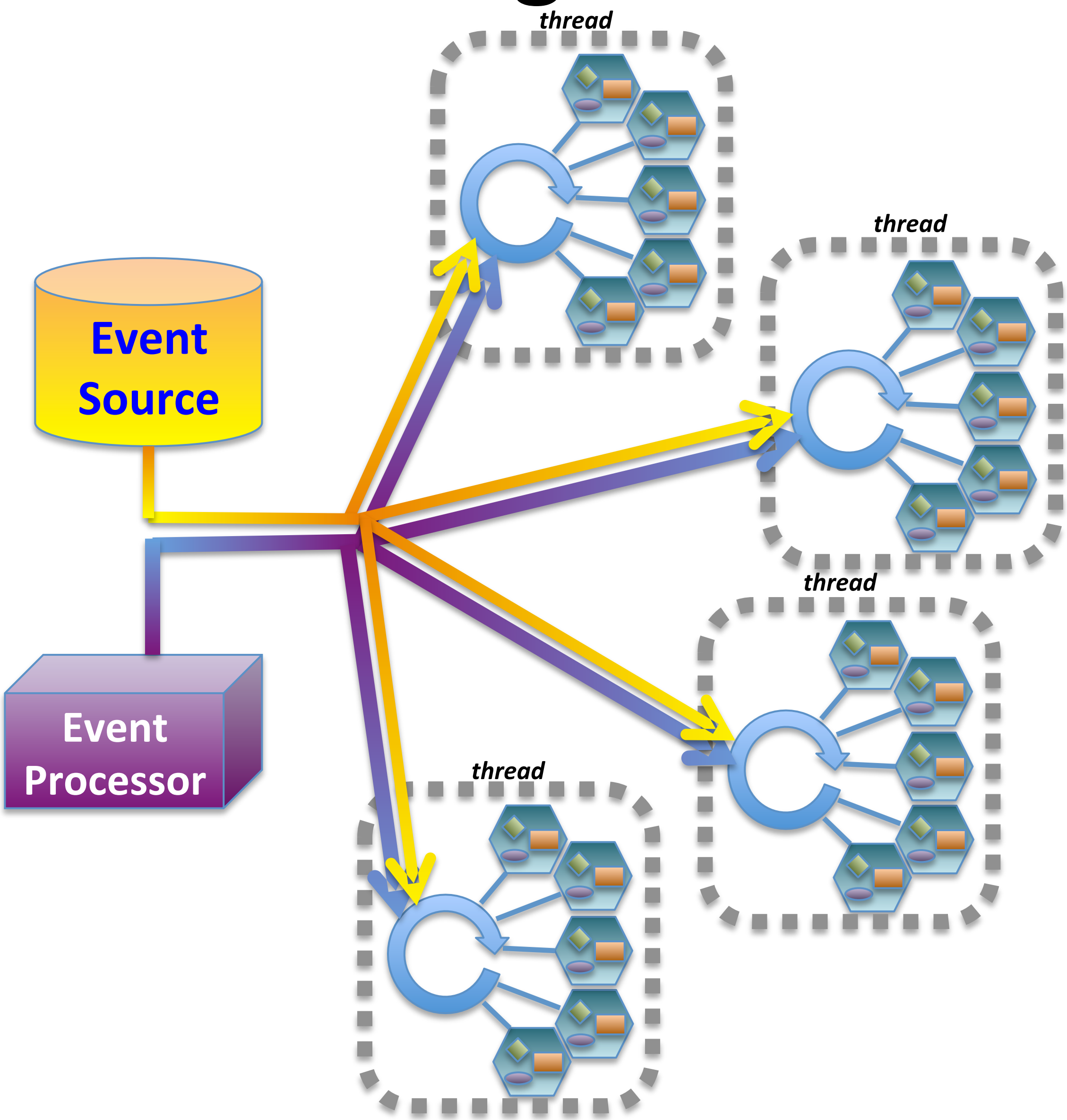
This allows the framework to easily redirect requests to alternate algorithms specified by the user at run time

Multi-threading

- *Each thread has a complete set of factories making it capable of completely reconstructing a single event*

- *Factories only work with other factories in the same thread eliminating the need for expensive mutex locking within the factories*

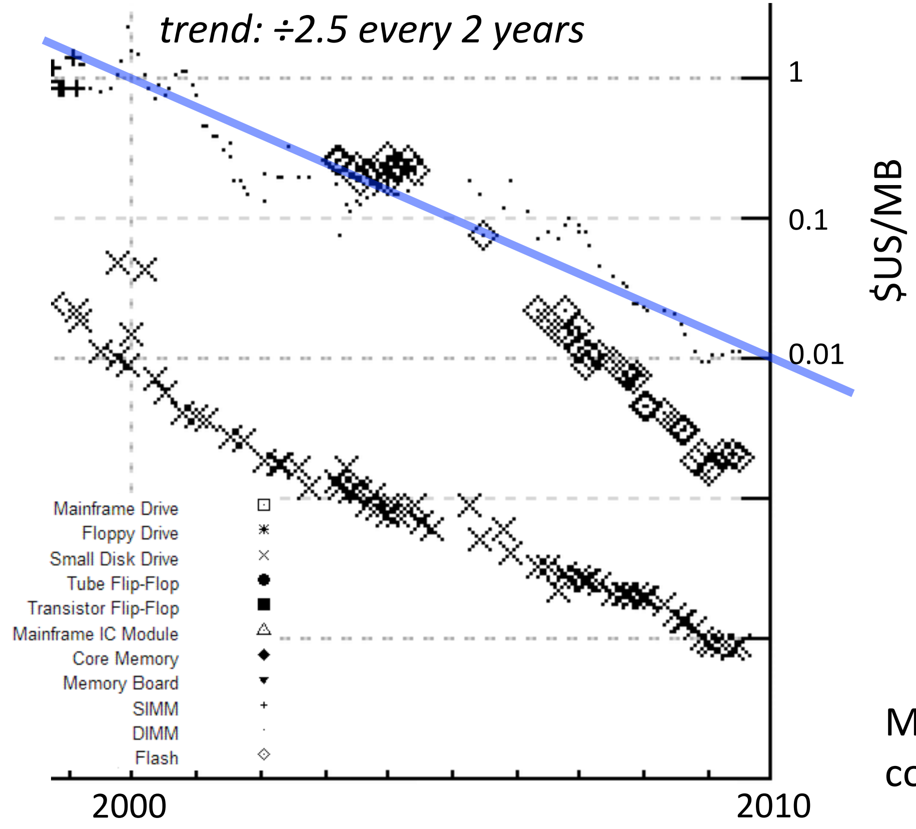
- *All events are seen by all Event Processors (multiple processors can exist in a program)*



Memory prices for last 2 decades

Memory Prices

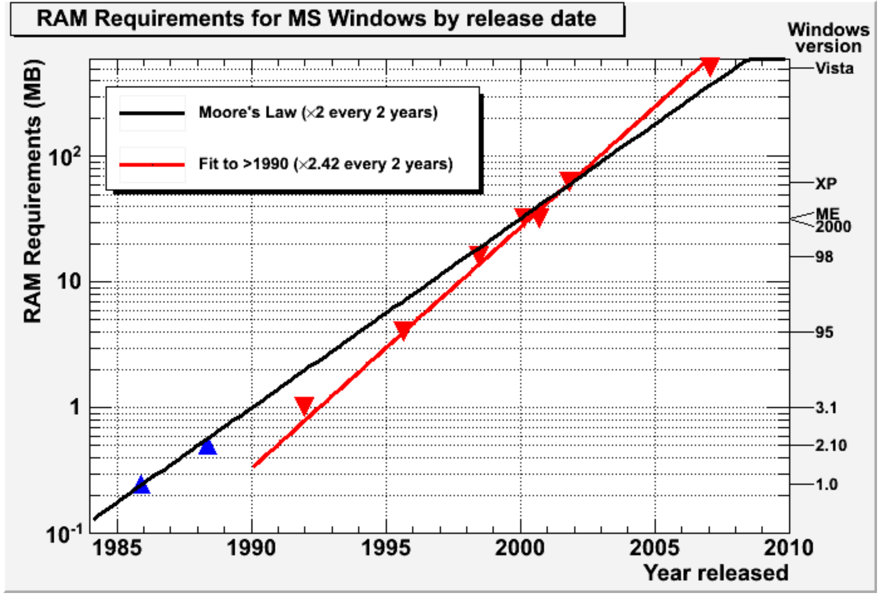
trend: ÷2.5 every 2 years



©2009 John C. McCallum
thanks to gnuplot

Memory Requirements

trend: ×2.4 every 2 years



David Lawrence 2010

Memory cost for a desktop computer has been roughly constant over the last 2 decades.

The cost of a desktop system has been roughly constant so this keeps memory at a constant fraction of the total cost.

Multiple cores + memory

Multi-core processors are already here and commonly used. Industry has signaled that this will be the trend for the next several years. Consequence: Parallelism is ***required***

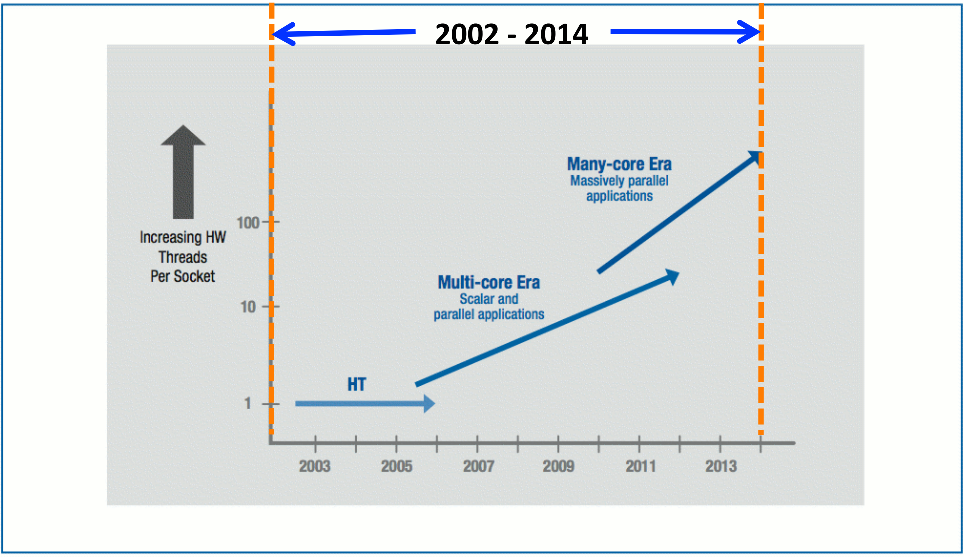
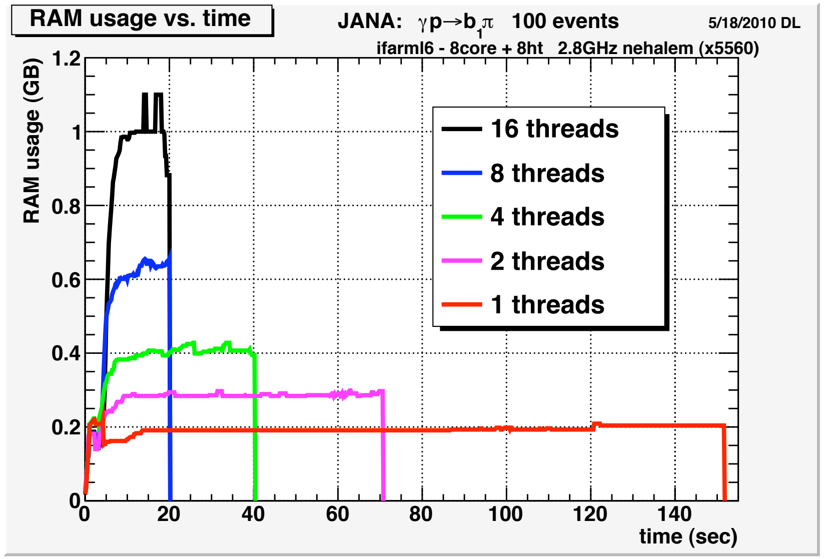
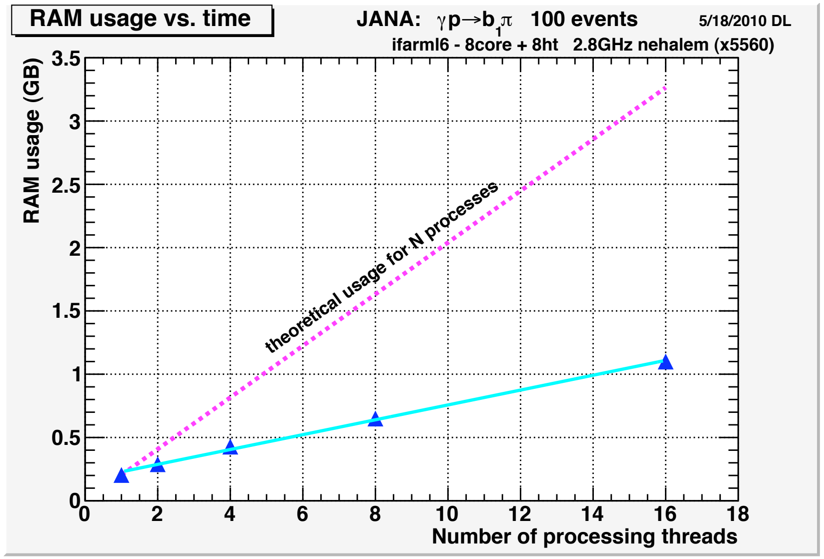


Figure 1: Current and expected eras of Intel® processor architectures

Maintaining a fixed memory capacity per core will become increasingly expensive due to limitations on the number of controllers that can be placed on a single die (#pins).

Prediction is that number of cores in the "Many-core Era" will increase faster than Moore's law adding to the difficulty in maintaining a fixed memory capacity per core.



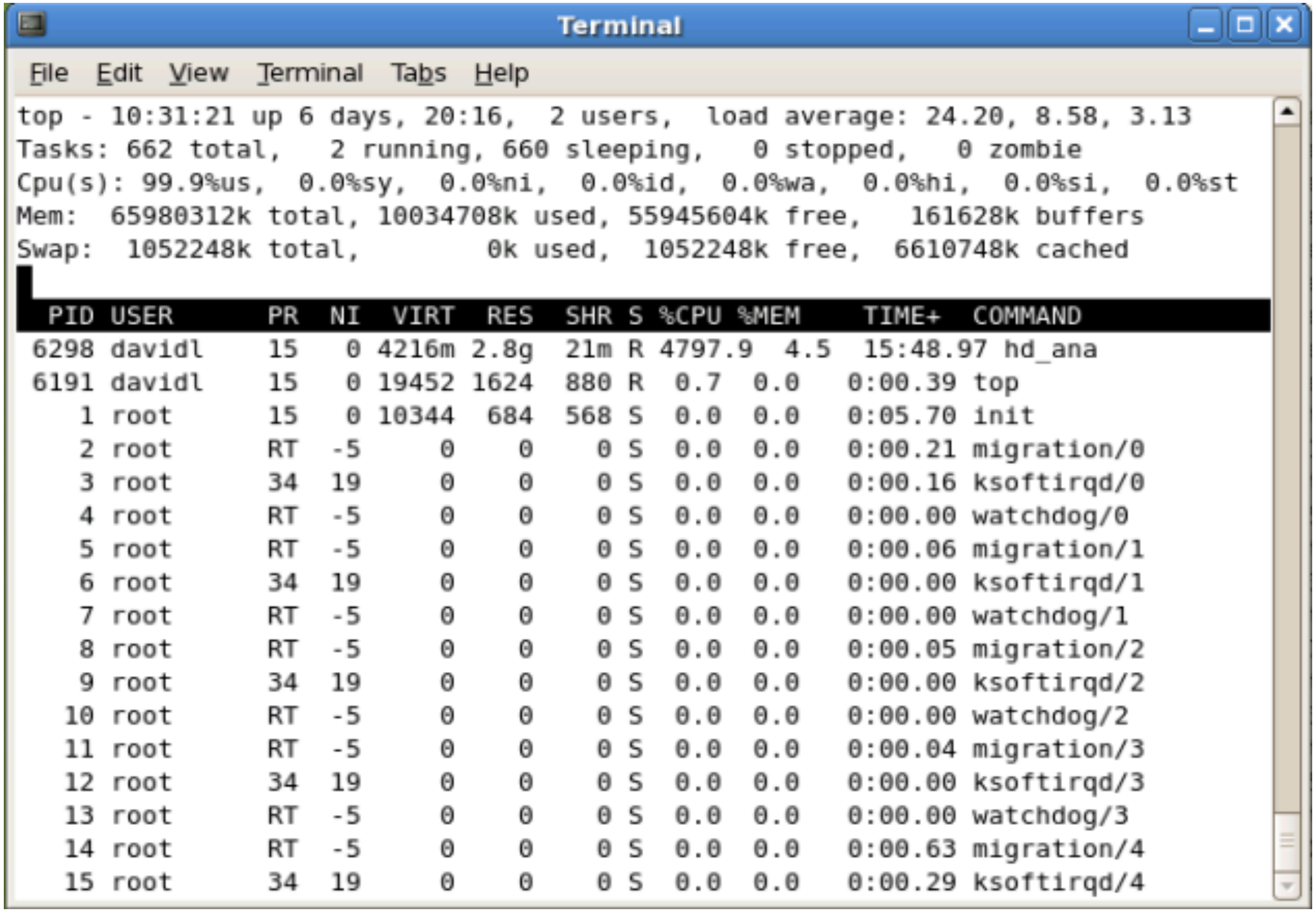


top

Memory usage is between 3 GB and 4 GB for single process running with 48 processing threads

CPU completely utilized in user space (like we want it!)

Negligible time spent sleeping in mutex locks or system calls

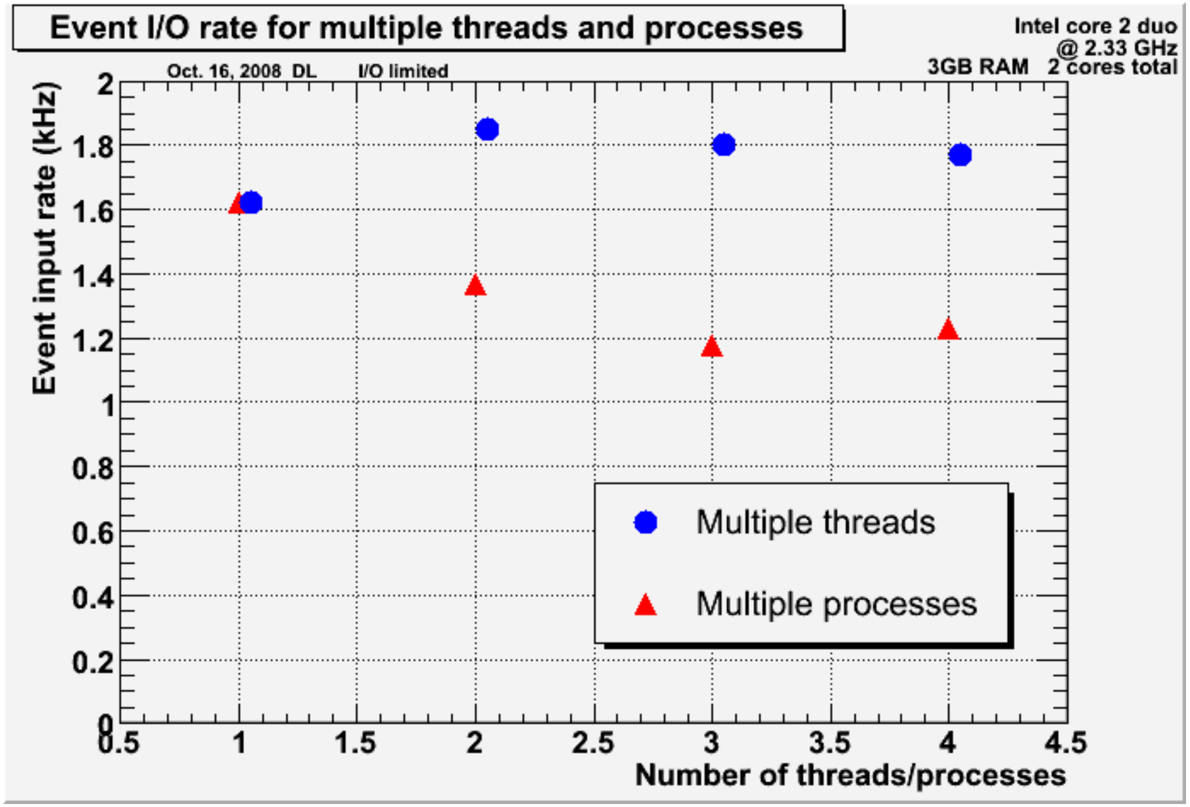


```
top - 10:31:21 up 6 days, 20:16,  2 users,  load average: 24.20, 8.58, 3.13
Tasks: 662 total,   2 running, 660 sleeping,   0 stopped,   0 zombie
Cpu(s): 99.9%us,  0.0%sy,  0.0%ni,  0.0%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:  65980312k total, 10034708k used, 55945604k free,   161628k buffers
Swap:  1052248k total,        0k used,  1052248k free,  6610748k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 6298 davidl    15   0 4216m 2.8g  21m R 4797.9  4.5  15:48.97 hd_ana
 6191 davidl    15   0 19452 1624  880 R  0.7  0.0   0:00.39 top
    1 root      15   0 10344  684  568 S  0.0  0.0   0:05.70 init
    2 root      RT  -5     0    0    0 S  0.0  0.0   0:00.21 migration/0
    3 root      34  19     0    0    0 S  0.0  0.0   0:00.16 ksoftirqd/0
    4 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
    5 root      RT  -5     0    0    0 S  0.0  0.0   0:00.06 migration/1
    6 root      34  19     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/1
    7 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/1
    8 root      RT  -5     0    0    0 S  0.0  0.0   0:00.05 migration/2
    9 root      34  19     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/2
   10 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/2
   11 root      RT  -5     0    0    0 S  0.0  0.0   0:00.04 migration/3
   12 root      34  19     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/3
   13 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/3
   14 root      RT  -5     0    0    0 S  0.0  0.0   0:00.63 migration/4
   15 root      34  19     0    0    0 S  0.0  0.0   0:00.29 ksoftirqd/4
```

I/O Scaling

Multiple processes simultaneously reading and writing to the same local disk will cause the disk head to thrash, ultimately leading to an I/O bottleneck

Multiple threads will stream events from a single file leading to much less competition for the head position


Event I/O rate for multiple threads and processes
Intel core 2 duo
@ 2.33 GHz
3GB RAM 2 cores total
Oct. 16, 2008 DL
I/O limited
Event input rate (kHz)
Number of threads/processes
● Multiple threads
▲ Multiple processes