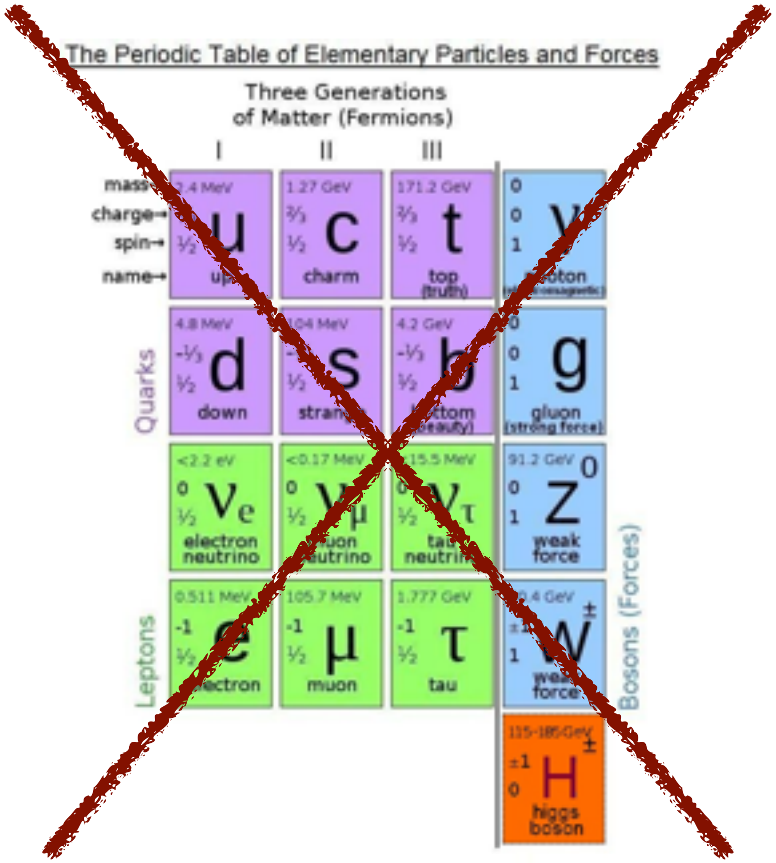
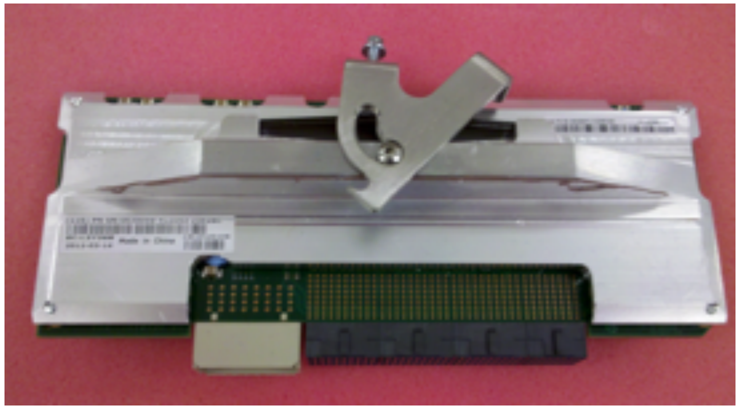
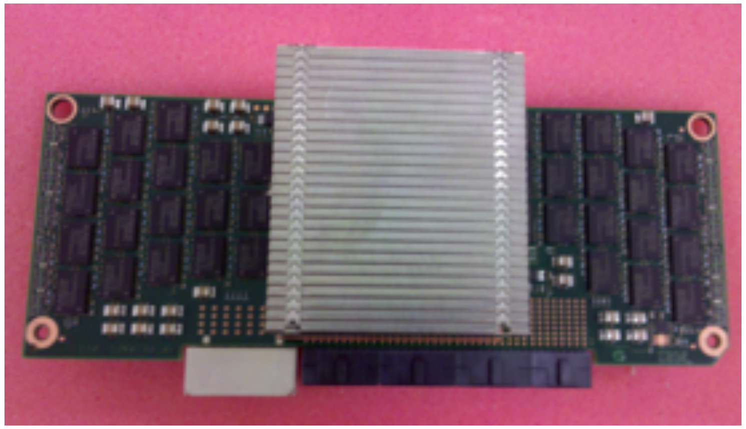
Using ALCF Systems: An Overview of the Blue Gene/Q and DA Systems, Storage, Software, and Other Notes

William Scullin
ALCF Catalyst Group

Building Blocks of the Universe

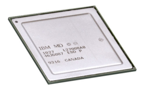

The Periodic Table of Elementary Particles and Forces
Three Generations of Matter (Fermions)
I
II
III
mass→
charge→
spin→
name→
2.4 MeV ⅔ ½ u up
1.27 GeV ⅔ ½ c charm
171.2 GeV ⅔ ½ t top (truth)
0 0 1 γ photon (electromagnetic)
4.8 MeV -⅓ ½ d down
104 MeV -⅓ ½ s strange
4.2 GeV -⅓ ½ b bottom (beauty)
0 0 1 g gluon (strong force)
<2.2 eV 0 ½ νe electron neutrino
<0.17 MeV 0 ½ νμ muon neutrino
<15.5 MeV 0 ½ ντ tau neutrino
91.2 GeV 0 1 Z0 weak force
0.511 MeV -1 ½ e electron
105.7 MeV -1 ½ μ muon
1.777 GeV -1 ½ τ tau
80.4 GeV ±1 1 W± weak force
115-185 GeV ±1 0 H± higgs boson
Quarks
Leptons
Bosons (Forces)



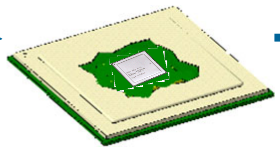


Anatomy of a Blue Gene/Q (Not Ours)

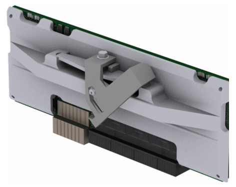
1. Chip
16 cores



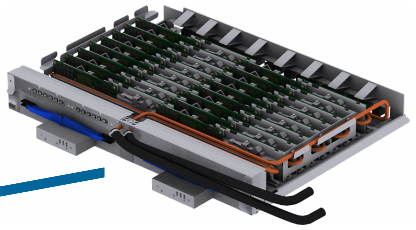
2. Module
Single chip



3. Compute card
One single chip module,
16 GB DDR3 memory



4. Node board
32 compute cards,
optical modules,
link chips, torus

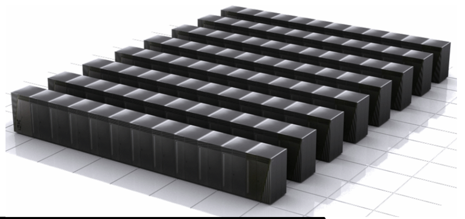


5b. I/O drawer
8 I/O cards with 16 GB
8 PCIe Gen2 slots

5a. Midplane
16 node boards

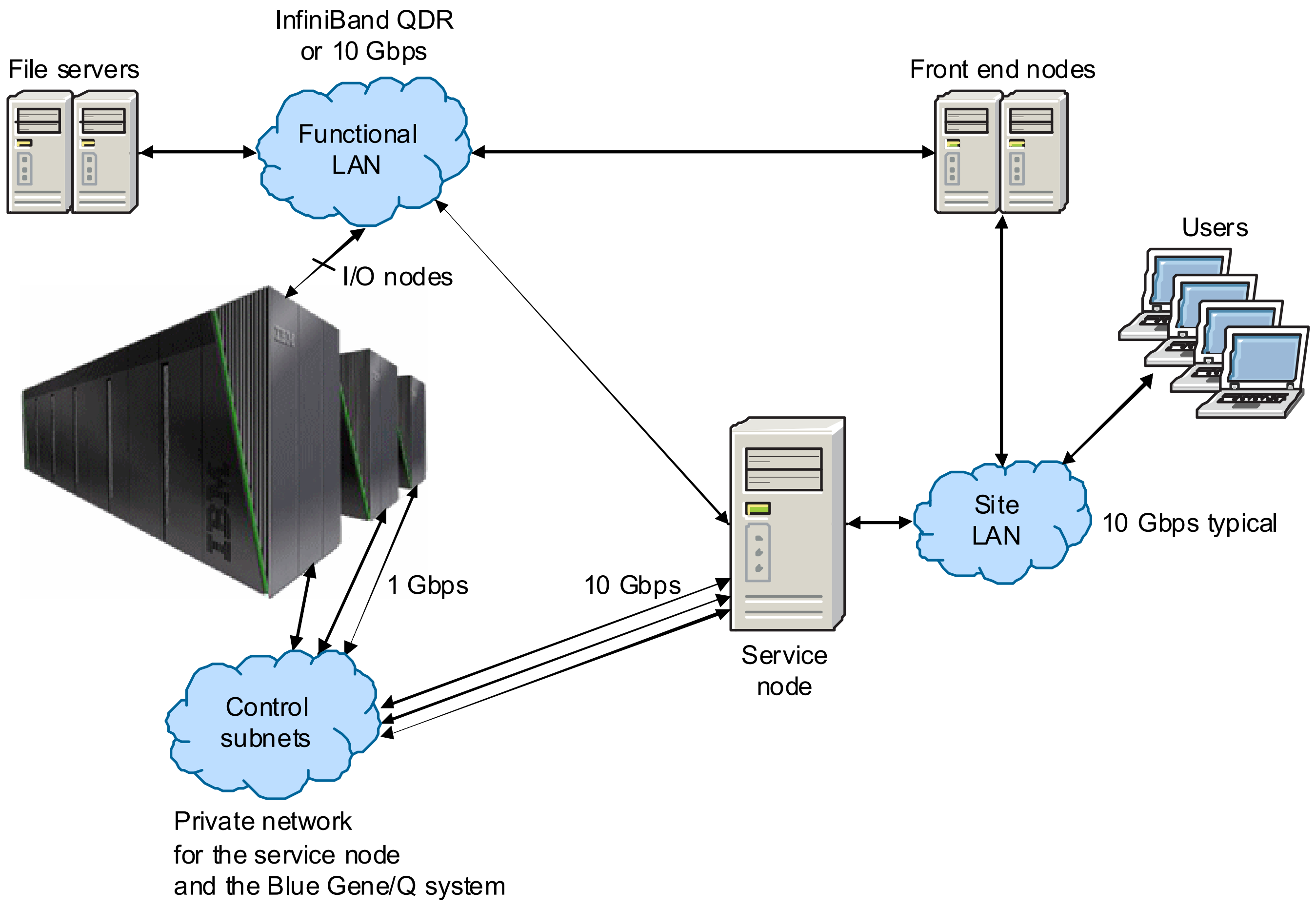
6. Rack
1 or 2 midplanes
0, 1, 2, or 4 I/O drawers

7. Multi-rack system



Per rack	
Peak performance	**209 TF**
Sustained *(Linpack)*	**~170+ TF**
Power efficiency*	**~2.1 GF/W**
***Without optical interconnect**	

No Blue Gene/Q is an island



System Details

Blue Gene /Q System	Vesta	Mira	Cetus
Function	Test & Development	INCITE Production	INCITE Test & Development
Login address	vesta.alcf.anl.gov	mira.alcf.anl.gov	cetus.alcf.anl.gov
Login OS	RHEL 6.5	RHEL 6.5	RHEL 6.5
Login CPUs	2 Power 740: - Power 7 - 48 cores @ 3.7 GHz	4 Power 740(s): - Power 7 - 48 cores @ 3.7 GHz	1 Power 740: - Power 7 - 48 cores @ 3.7 GHz
Login memory	64GB	64GB	64GB
BG /Q CPU (16 core)	1.6 GHz PowerPC A2	1.6 GHz PowerPC A2	1.6 GHz PowerPC A2
# Nodes / # Cores	2048 / 32,768	49,152 / 786,432	4096 / 65,536
BG/Q Memory	32TB (16GB per node)	640TB (16GB per node)	64TB (16 GB per node)
# I/O nodes (Ratio)	64 @ IB (32:1)	384 @ IB (128:1)	8 @ IB (128:1)
Blue Gene/Q Driver	V1R2M2 + efixes to 15	V1R2M2 + efixes to 15	V1R2M2 + efixes to 15

- Login to ALCF Resources is via ssh with cryptocard authentication
- Round robin DNS will place you on an available login node named:
  <machine>lac{number}.alcf.anl.gov
- Logins are for compilation and job submission only -
- You may use parallel make, but be responsible

System Details

Analytics System	Tukey
Function	INCITE Production
Login address	tukey.alcf.anl.gov
Node OS	RHEL 6.5
Node CPUs	Two 2GHz AMD Opteron 6128 CPUs per node (8 cores per CPU, 16 cores total)
Node memory	64GB
GPUs	Two NVIDIA Tesla M2070 GPUs per node
GPU memory	6GB
Nodes	100
Interconnect	QDR Infiniband interconnect
Local scratch	450 GB

- Login to ALCF Resources is via ssh with cryptocard authentication
- Round robin DNS will place you on an available login node named:
  login{number}.tukey
- Logins are for compilation and job submission only
- Four tukey nodes are currently in use for GPFS token management

Driver and OS status

- We're currently running Red Hat Enterprise Linux 6.5
  - this covers logins, the control system and infrastructure, and IO nodes
  - We are continuously patching where possible
    - holding back on anything that strongly impacts the toolchain, control system, or availability
    - 99% of packages are stock RHEL 6.5 ppc64 packages
  - everything is 64-bit
    - we will install 32-bit packages and applications if absolutely necessary
    - so far there haven't been any cross-compiling surprises
- Current driver is the V1R2M2 driver
  - Vesta (testing and development) receives all new efixes and drivers first. Generally, there are two weeks between an efix or new driver being installed on Vesta and installation in the production environment
  - V1R2M3 has just been released, but we haven't deployed it yet
  - Vesta should have it installed, but not active soon

Login node notes

- Be a good citizen!
  - Don’t run parallel make for more than a fourth of the cores (16) than a login has
  - Clean up your files when done
    - We purge /tmp, but it’s better if you clean up after compiles
  - We will terminate processes impacting stability and other’s use of the logins
    - We will contact you if we do
    - We can usually accommodate your workflows – just ask
- Cetus and Mira
  - All important filesystems are synced and shared
  - Cetus is for testing and development jobs - not production
  - You cannot submit Cetus (prod) jobs from Mira (prod-devel) and vice-versa
- Vesta file systems are not shared on Cetus and Mira for security reasons
- If you lack a .soft file file at login we can create one or basic content should look like:

  +mpiwrapper-gcc.legacy # note that legacy means corse locking, not outdated

  @default

Filesystems and Layout

- /soft
  - we'll get back to this
  - mounted everywhere
  - a GPFS bind mount
- /projects
  - mounted everywhere, not backed up
  - check quota with /usr/local/bin/myprojectquotas
  - GPFS
  - also referred as
    - /gpfs/mira-fs0/projects on Mira, Cetus, and Tukey
    - /gpfs/mira-fs1/projects on Mira, Cetus, and Tukey
    - /gpfs/vesta-fs0/projects on Vesta
- /home
  - mounted everywhere, backed up on Mira, Cetus, and Tukey only
  - check quota with /usr/local/bin/myquota
  - GPFS
  - also referred to as
    - /gpfs/mira-home on Mira, Cetus, and Tukey
    - /gpfs/vesta-home on Vesta

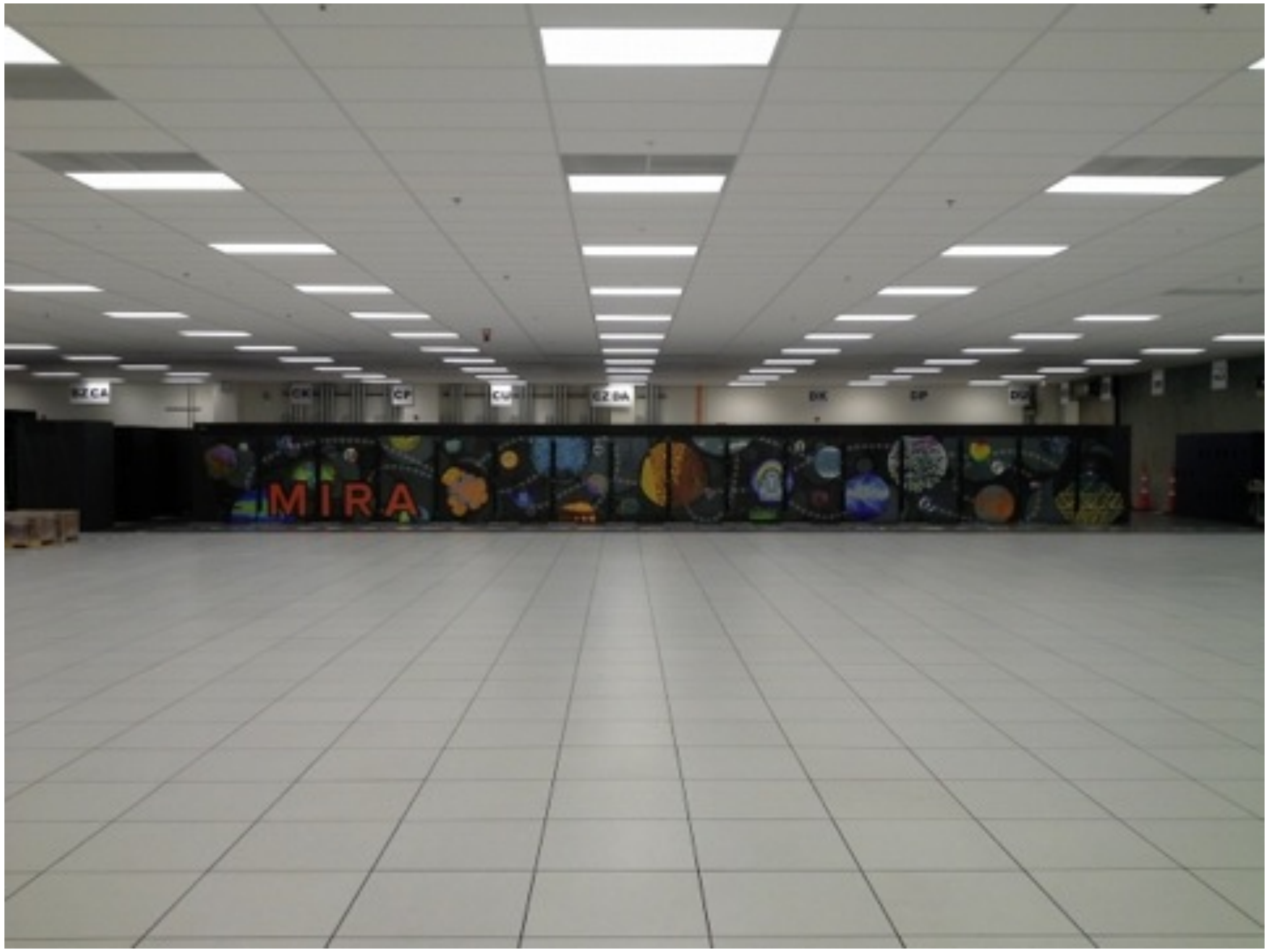
/soft layout and finding things

- We try to organize /soft to make things easy to find
  - Arrangements will be by function, ie: compilers in /soft/compilers, performance tools in /soft/perftools, softenv and modules in /soft/environment.
- softenv keys should be authoritative
- front end software (editors, X, games) is installed in RHEL's default locations

Cetus

Vesta

Mira



Tukey (or part of it)

A Random Mantis Shrimp

Cobalt

- Very similar to other queuing systems with caveats
  - general usage: qsub -t <time> -n <nodes> --mode <mode> -i <stdin> <binary>
  - modes set processes per node or specifies a script
    - -c{1,2,4,8,16,32,64}
    - memory is divided evenly with hardware partitioning
  - -n gives you nodes
  - --proccount gives you total processes
  - -i or --input provides stdin
- Script mode
  - allows for functionality similar to other batch systems
  - #COBALT directives must be at the top of your script
  - see wiki for details (use runjob in scripts and consult the man page - only in scripts)
- Examples can be found in /soft/cobalt/examples

Block naming and Cobalt

- Block names follow logical names, not hardware names
- Why?
  - 32 character limit on block names
  - Mira allows many more degrees of freedom in block configurations
  - Allows us to state which midplanes, and by extension which hardware is in use in a given location when the hardware locations make little sense
  - Makes sub-block setup easier
- One rack has the topology 4x4x4x8x2
- One midplane is 4x4x4x4x2

Decoding Block Names

- LOC-CCCCC-XXXXX-[T]-[PPPP]-SIZE
- LOC = location identifier, like ANL, CHR, VES, CET, MIR, EAS can be up to 7 characters.
- CCCCC = The bottom right front corner as described as a set of 5-dimensional coordinates ABCDE. This corresponds to the node location of the node of rank 0 in a ABCDE-type mapping scheme (node 0).
- XXXXX = The top left rear corner of the block in each dimension (node n-1).
- T = an optional identifier indicating which dimensions are Mesh and which are torus. This is a bitmasked value (0 = toruS, 1 = mesh). No value implies the maximum number of torus dimensions for that block
- PPPP = indicator of passthrough extents in each dimension. This will have a value of 0, 1, or 2.
- SIZE = The overall size in nodes of the block. This should correspond to the product of the extents.

Block name example

- A sample logical address could be: MIR-04C00-48FF2-7-2048
- Think LOC-CCCCC-XXXXX-[T]-[PPPP]-SIZE

- This corresponds to:
  - one midplane in the A dimension, first midplane in A
  - one midplane in the B dimension, starting at the second midplane (row 1, to be exact)
  - one midplane in the C dimension,
  - Four midplanes in the D dimension,
  - A,B,C dimensions are mesh, D is a torus
  - 2048 Nodes

- Old style it might be: MIR-R14-R15-2048

Rules of the Road

- Blocks must have a sufficient ratio of IO nodes to Compute Nodes
- Blue Gene/Q IO blocks are persistent and may be shared
  - writing to /tmp and not removing files is anti-social
  - we do not offer a way to retrieve files written to /tmp at this time
  - crashing an IO block crashes compute blocks reliant on that IO block
- All GPFS filesystems available on the login nodes are available on the IO nodes for access from the computes
- You are always the only user process on compute node cores and memory

Resource Isolation (Mira and Cetus)

- You have I/O node isolation (only user) when compute node counts are:
  - >= 512 (midplane or above): The IONs, computes and blocks are all yours within that block)
- Your I/O nodes are shared when node counts are:
  - >= 256 : you share IONs
- The I/O nodes share an IB switch complex with all the other I/O nodes
- Take away: measure and debug I/O performance issues running on at least a midplane and over several runs

Resource Isolation (Vesta)

- Vesta is configured for users doing development
- Resource sets where you are the only one on our resources.
  - >= 128 (4 node boards): The IONs, computes and blocks are all yours within that block)
- You are sharing when:
  - > 128: you share I/O blocks
  - > 32: you may be sharing an I/O node via a split I/O link
- You are always the only thing on the compute node's compute cores and memory
- Crashing an I/O block takes out jobs on any attached compute blocks

Allocation Management

- Every user must have at least one Project they are assigned to
- Projects are then given allocations
  - Allocations have an amount, start, and end date and are tracked separately; Charges will cross allocations automatically. The allocation with the earliest end date will be charged first, until it runs out, then the next, and so on
- Charges are based on the partition size, NOT the # of nodes or cores used!
- Reservations are charged for the full time they are active
- Will be managed with clusterbank
  - Use the 'cbank' command (see 'cbank --help')
- Examples:

  # list all charges against a particular project

  - cbank -l charge -p <projectname>

  # list all active allocations for a particular project

  - cbank -l allocation -p <projectname>

When things go wrong… running

- Cobalt jobs, by default, produce three files prefixed with either the job number or the name specified with qsub's **–O** option:
  - **$PREFIX.output** with output to standard out from you application
  - **$PREFIX.error** with output from the control system, scheduler, and your application directed to standard error
  - **$PREFIX.cobaltlog** with a record of the environment and the submission command
    - Generated at submit time
    - Very useful for the support team in debugging
- Only cobaltlog is generated at submit time, the others at runtime
- At boot, the .error file will have a non-zero size for non-script jobs
  - Most of the messages are related to booting, it's a decent way to follow startup progress
  - Script jobs leave the handling of standard error up to you, though most do use the .error file
  - We'll walk through a normal boot in a moment…
- If you think there is an issue, it's best to save all three files
- If you have a script job, save the script in an unedited state

Normal Execution Produces a lot of stderr

```
2015-05-18 14:48:20.101 (INFO ) [0xfffab08c330] ibm.runjob.AbstractOptions: using properties file /bgsys/local/etc/
bg.properties
2015-05-18 14:48:20.102 (INFO ) [0xfffab08c330] ibm.runjob.AbstractOptions: max open file descriptors: 65536
2015-05-18 14:48:20.102 (INFO ) [0xfffab08c330] ibm.runjob.AbstractOptions: core file limit: 0
2015-05-18 14:48:20.103 (INFO ) [0xfffab08c330] 29518:tatu.runjob.client: scheduler job id is 263596
2015-05-18 14:48:20.103 (INFO ) [0xfffab08c330] 29518:tatu.runjob.client: scheduler reservation id is 31
2015-05-18 14:48:20.109 (INFO ) [0xfffa99434e0] 29518:tatu.runjob.monitor: monitor started
2015-05-18 14:48:23.317 (INFO ) [0xfffa99434e0] 29518:tatu.runjob.monitor: task record 951942 created
2015-05-18 14:48:23.320 (INFO ) [0xfffab08c330] VST-00420-11531-32:29518:ibm.runjob.client.options.Parser: set local
socket to runjob_mux from properties file
2015-05-18 14:48:25.992 (INFO ) [0xfffab08c330] VST-00420-11531-32:1141998:ibm.runjob.client.Job: job 1141998 started
2015-05-18 14:48:27.782 (INFO ) [0xfffab08c330] VST-00420-11531-32:1141998:ibm.runjob.client.Job: exited with status 0
2015-05-18 14:48:27.783 (INFO ) [0xfffab08c330] tatu.runjob.client: task exited with status 0
2015-05-18 14:48:28.672 (INFO ) [0xfffa99434e0] 29518:tatu.runjob.monitor: tracklib completed
2015-05-18 14:48:28.673 (INFO ) [0xfffa99434e0] 29518:tatu.runjob.monitor: monitor terminating
2015-05-18 14:48:28.676 (INFO ) [0xfffab08c330] tatu.runjob.client: monitor completed
```

When things go wrong… running

- You'll see RAS events appear in your .error file it's not always the sign of trouble
- Exit codes are more meaningful
  - 0 is normal
  - 15 is usually hitting walltime and dying with SIGTERM
  - 9 is usually hitting walltime and requiring SIGKILL, but may be a system issue
  - anything else - unless running Nek - is worth reporting
- RAS stands for Reliability, Availability, and Serviceability
- Few are a sign of a serious issue, most are system noise
  - Messages have a severity associated with them
    - INFO
    - WARN
    - ERROR
    - FATAL
  - Only FATAL RAS events will lead to the termination of your application
  - Still worth watching as they may be the sign of an application issue

When things go wrong… running

Remember:

The VelociRAStor says **ONLY FATAL RAS EVENTS ARE FATAL**

Thank you and Questions

Thanks for listening!

Any questions?