Online Diagnostics at Scale

Don Maxwell (ORNL)
Jeff Becklehimer (Cray)
April 30, 2012

- Diagnostic definitions
- Diagnostic process
- Motivation and Experiences
- Finding Problems
- Resolving Problems

Diagnostic Definitions

- Offline Diagnostics
  - No operating system booted
  - Manufacturing screens
  - Cray
    - SMW necessary
    - memtest, cpuburn, mtstat, bist, xtcablecheck

- Online Diagnostics
  - Runs under operating system
  - Acceptance tests – Batch, Intel MPI Benchmarks (IMB)
  - Applications that stress the machine

Diagnostic Process

- Commodity Part Manufacturer
  - Screening Process To Eliminate Weak Parts
    - Thermal Extremes
    - Electrical Extremes
    - Application Results
- Supercomputer Vendor
  - Schedules Sometimes Dictate Parts Arrival at Site
  - Screening/Burn-In Process To Eliminate Weak Parts
    - Not as Extreme Extremes
    - Different Set of Applications to Eliminate Weak Parts
- Consumer
  - Screening/Burn-In Process To Eliminate Weak Parts
  - Yet Another Set of Applications
  - Acceptance Process

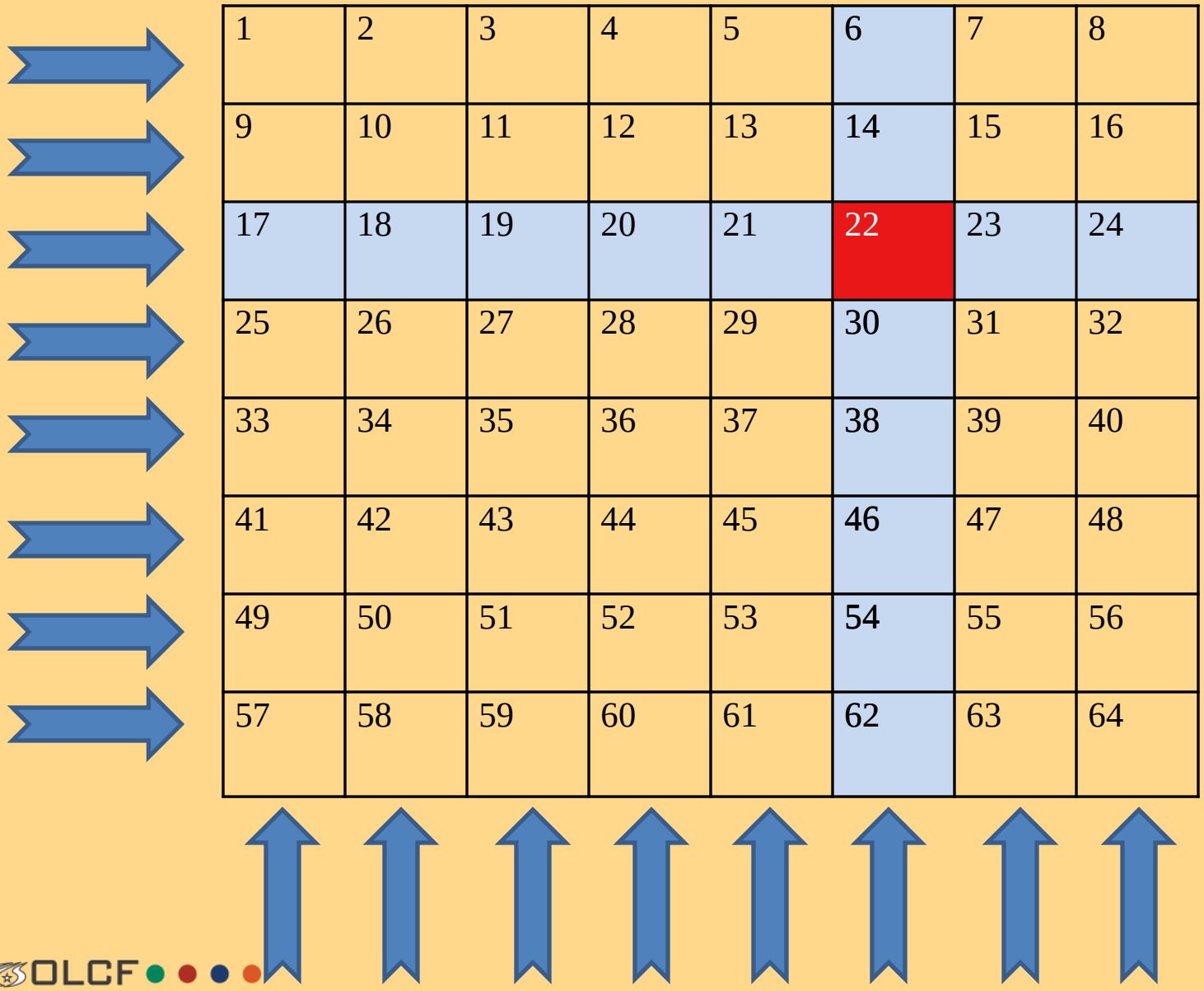
Motivation

- Why do we care about diagnostics?
  - Reliability
  - Stability
  - Metric requirements
  - User experience
- HPC Pitfalls
  - Volume of parts
  - Early parts
    - Schedules
    - Experience in the market minimal
  - Diagnostic scaling

Experiences

- Early-life failures
  - Node failures
  - Segmentation faults for particular codes
    - LSMS/Madness/S3D
  - Soft errors
- Most problems found during installation and acceptance
- Initial search for ongoing diagnostic to minimize user impact
  - Short-running application proven to find issues
  - Run in nodehealth/NodeKARE?
    - No MPI
  - Prologue/Epilogue
    - Apps not short enough
  - Batch mode
    - Problems
      - Application must scale higher than per node due to ALPS limitations
      - Cycles away from users
        - » Can only run a small number of months after acceptance

Finding Soft Errors



MySQL Table for Tracking Diag Runs

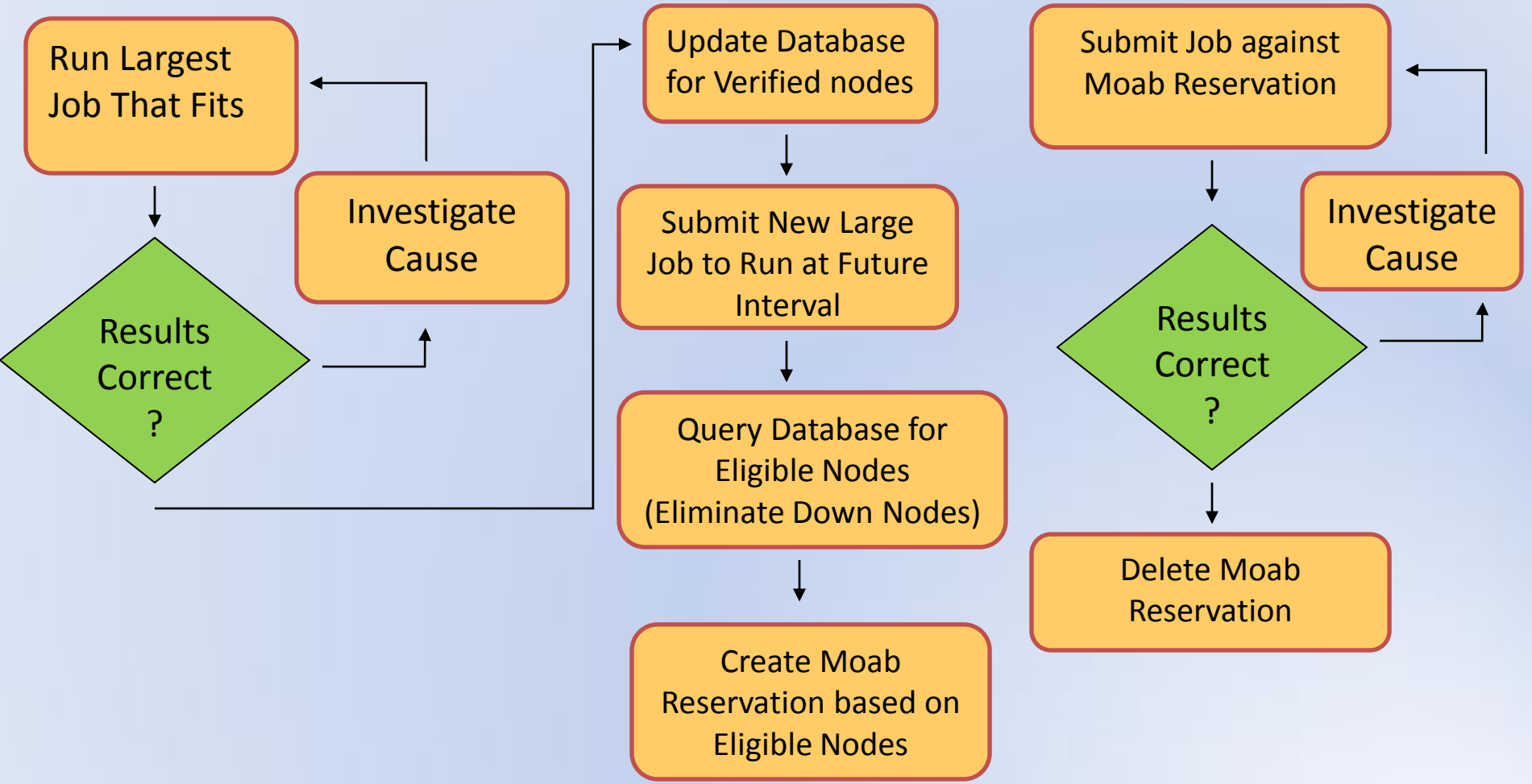
- Accounting to cover the entire machine
- Compare (now - last runtime) > run frequency
- Moab reservations used to get jobs through machine
- Experimentation with large jobs vs. small jobs

Field	Type
last_runtime	Datetime
processor_id	int(10) unsigned
job_id	varchar(80)
Hostname	varchar(80)

| last_runtime | processor_id | job_id | hostname |

| 2012-04-20 02:23:49 | 4516 | 1075439 | jaguarpf |

Diagnostic Run Flowchart



Segmentation Faults

- Good news
  - Console output to detect problem
- Bad news
  - Cannot distinguish a hardware issue from a user error

A definite hardware problem

[2011-11-15 10:38:48][c8-4c2s0n0]lmp_jaguar_pgi[8292] general protection ip:918202 sp:7fffffff83a0 error:0 in lmp_jaguar_pgi[400000+883000]
[2011-11-16 03:39:08][c8-4c2s0n0]lmp_jaguar_pgi[9425] general protection ip:918202 sp:7fffffff8390 error:0 in lmp_jaguar_pgi[400000+883000]
[2011-11-16 03:49:37][c8-4c2s0n0]lmp_jaguar_pgi[9633] general protection ip:918202 sp:7fffffff8340 error:0 in lmp_jaguar_pgi[400000+883000]

Don't care - software development issue

[2012-04-25 05:48:17][c11-5c1s4n3]namd2-topo-pme[17779] general protection ip:45ffd35 sp:2aab75cf6d90 error:0 in namd2-topo-pme[4000000+a20000]
[2012-04-25 05:48:17][c9-1c2s2n2]namd2-topo-pme[17690] general protection ip:45ffd35 sp:2aab75ef7d90 error:0 in namd2-topo-pme[4000000+a20000]
[2012-04-25 05:48:17][c9-6c1s5n1]namd2-topo-pme[18973] general protection ip:45ffd35 sp:2aab75ef7d90 error:0 in namd2-topo-pme[4000000+a20000]

Segmentation Faults Monitored by SEC

- SEC (Simple Event Correlator)
  - http://simple-evcorr.sourceforge.net
  - In use at OLCF since 2006 to monitor systems
    - "**Real Time Health Monitoring of the Cray XT Series Using the Simple Event Correlator (SEC)**," CUG 2007

Suppress user segfaults based on context

```
type= Suppress
ptype= RegExp
pattern= \[([:\- 0-9]+)\]\[([0-9A-z-]+)\](\S+)\[\d+\][:]* (?:general protection|segfault)
context= SUPPRESS_APP_$3
```

Set the suppress context when 2 faults from the same app within a given window of time

```
type= SingleWithThreshold
ptype= RegExp
continue= takenext
pattern= \[([:\- 0-9]+)\]\[([0-9A-z-]+)\](\S+)\[\d+\][:]* (?:general protection|segfault)
desc= Segfault App $3
action= create SUPPRESS_APP_$3 180; \
        reset Segfault Node $2 App $3; \
        delete SEGFAULT_$2_$3
window= 30
thresh= 2
```

Segmentation Faults Monitored by SEC

If not suppressing, add data to context for later reporting if conditions are met. Set expiration in case conditions are not met.

```
type= Single
ptype= RegExp
continue = takenext
pattern= \[([:\- 0-9]+)\]\[([0-9A-z-]+)\](\S+)\[\d+\][:]* (?:general protection|segfault)
context= !SUPPRESS_APP_$3
desc= Segfault Node $2 App $3
action= add SEGFAULT_$2_$3 $1 Node $2 Segfaulted App $3; \
        set SEGFAULT_$2_$3 200
```

Wait window of time for 3 segfaults from the same app on the same node and report.

```
type= SingleWithThreshold
ptype= RegExp
continue = takenext
pattern= \[([:\- 0-9]+)\]\[([0-9A-z-]+)\](\S+)\[\d+\][:]* (?:general protection|segfault)
context= SEGFAULT_$2_$3
desc= Segfault Node $2 App $3
action= report SEGFAULT_$2_$3 /bin/mail -s "%%DEST_HOST%% SegFault Threshold Exceeded $2"
%%DEST_EMAIL%%; \
        delete SEGFAULT_$2_$3
window= 180
thresh= 3
```

Removing Failed Parts

- Gemini Interconnect
  - Dynamic routing provides capability for replacement of parts on running machine
  - Software bugs cause reroute to fail leading to reboot
- Warmswap Procedure
  1. Verify system reroute
     - rtr –stage-routes
  2. Idle the module
     - Moab hostlist-based reservation
     - Check ALPS to ensure the module is idle
  3. xtwarmswap/Remove/Repair/Replace module
  4. xtbounce/Route/Boot module
  5. Run test jobs against Moab reservation
  6. If successful, remove Moab reservation – otherwise, return to hardware engineers

Future

- Reduce application to run on the order of seconds
- Ensure application can run in a scalable manner
- Integrate margining extremes to verify parts
- Incorporate into prologue/epilogue