Official Use Only

Exceptional service in the national interest

Sandia National Laboratories

A State Machine Approach to Disambiguating Supercomputer Event Logs

Jon Stearley, Robert Ballance, Lara Bauman

USENIX Workshop on
Managing Systems Automatically and Dynamically (MAD'12)
Oct 7, 2012

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Ambiguous - all events of interest

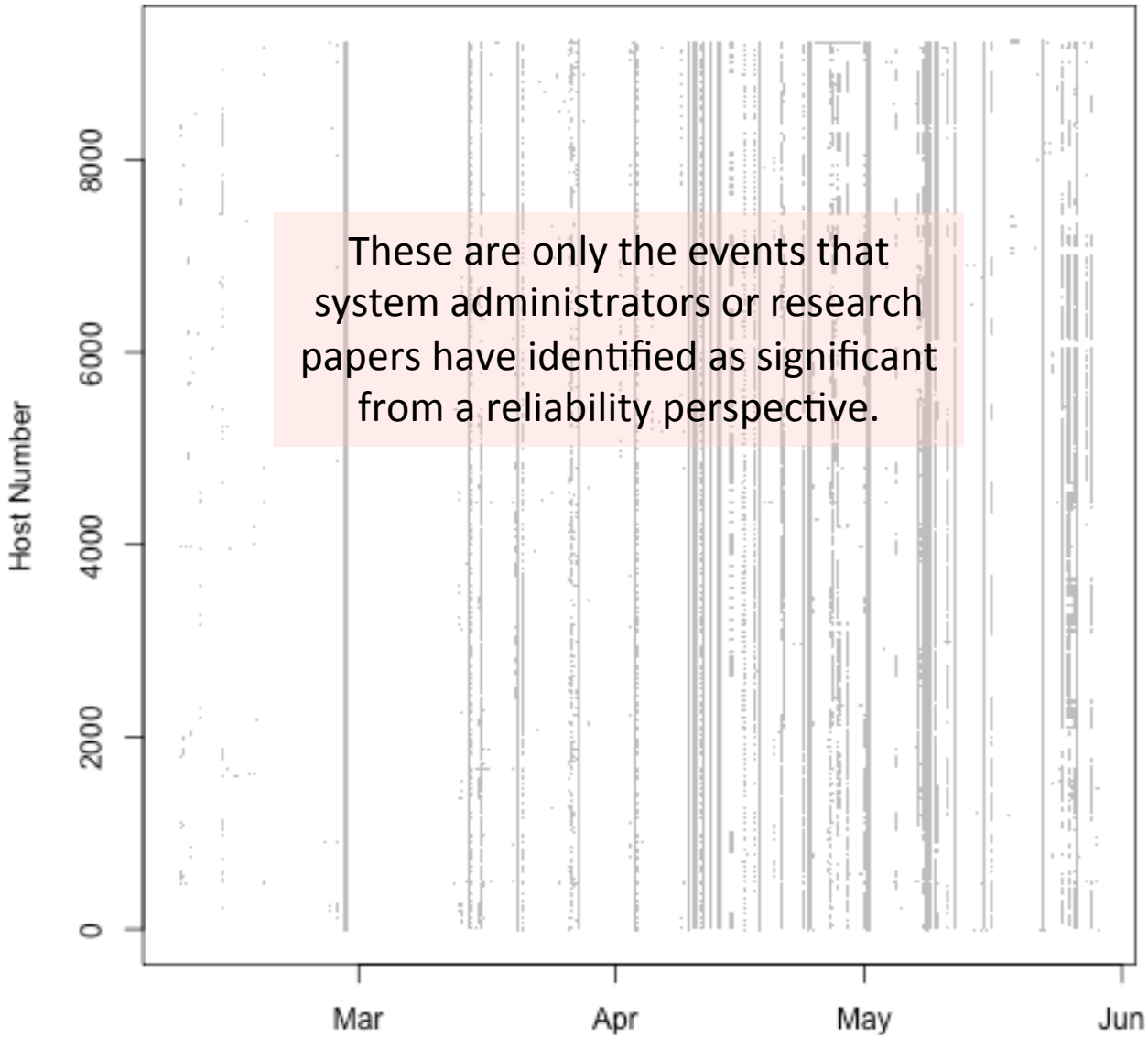
Sandia National Laboratories

One dot per event
(708,510 total)

Which of these dots indicate “failures”?

How would you calculate mean time to failure?

Or model failure distribution, etc?



For accurate understanding of our systems,
event disambiguation is essential!

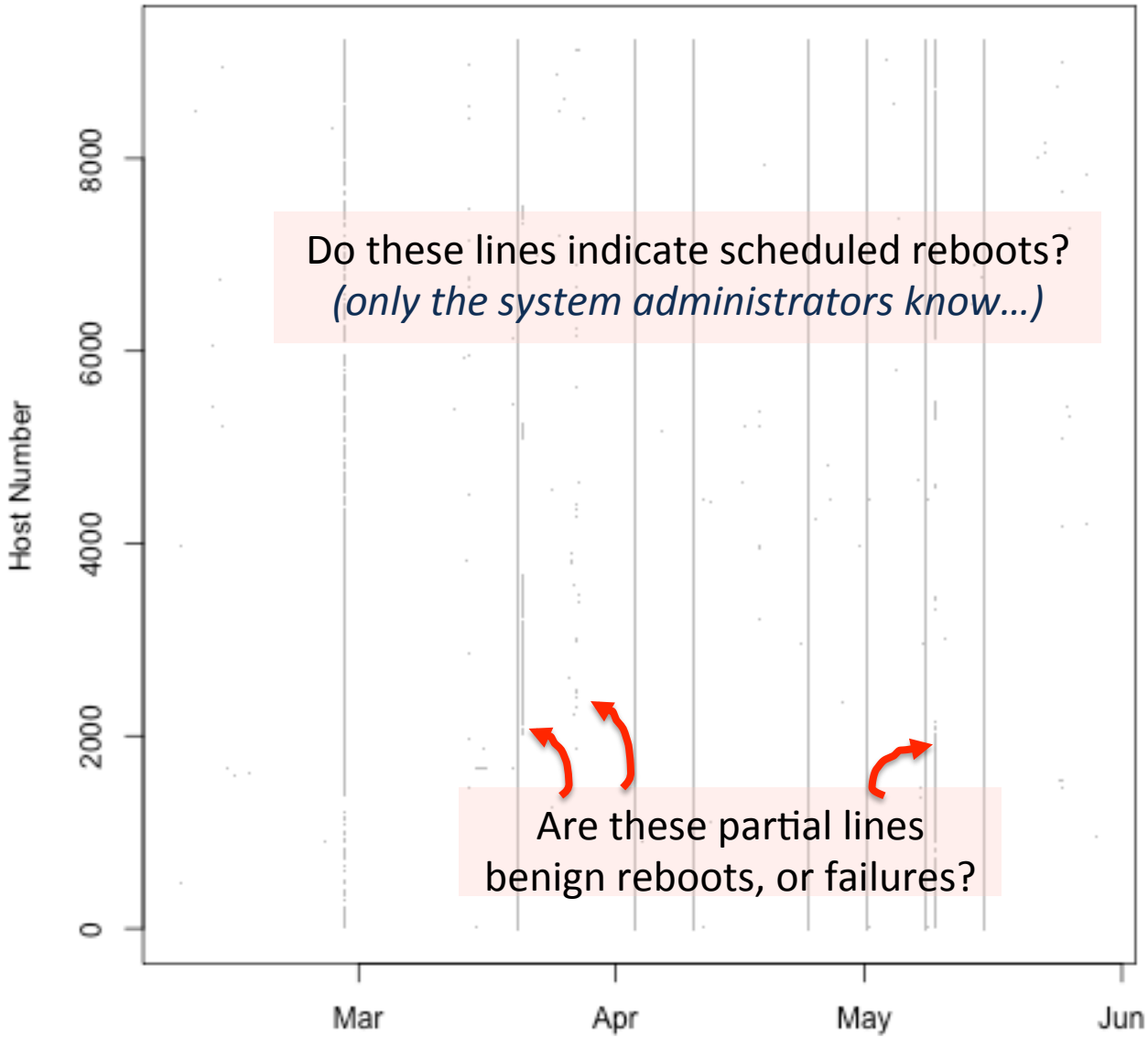
Ambiguous - CRAYHEARTSTOPALERT

One dot per event
(70,126 total)

Which of these dots indicate "failures"?

How would you calculate mean time to failure?

Or model failure distribution, etc?



For accurate understanding of our systems,
event disambiguation is essential!

Disambiguated - CRAYHEARTSTOPALERT

One dot per event

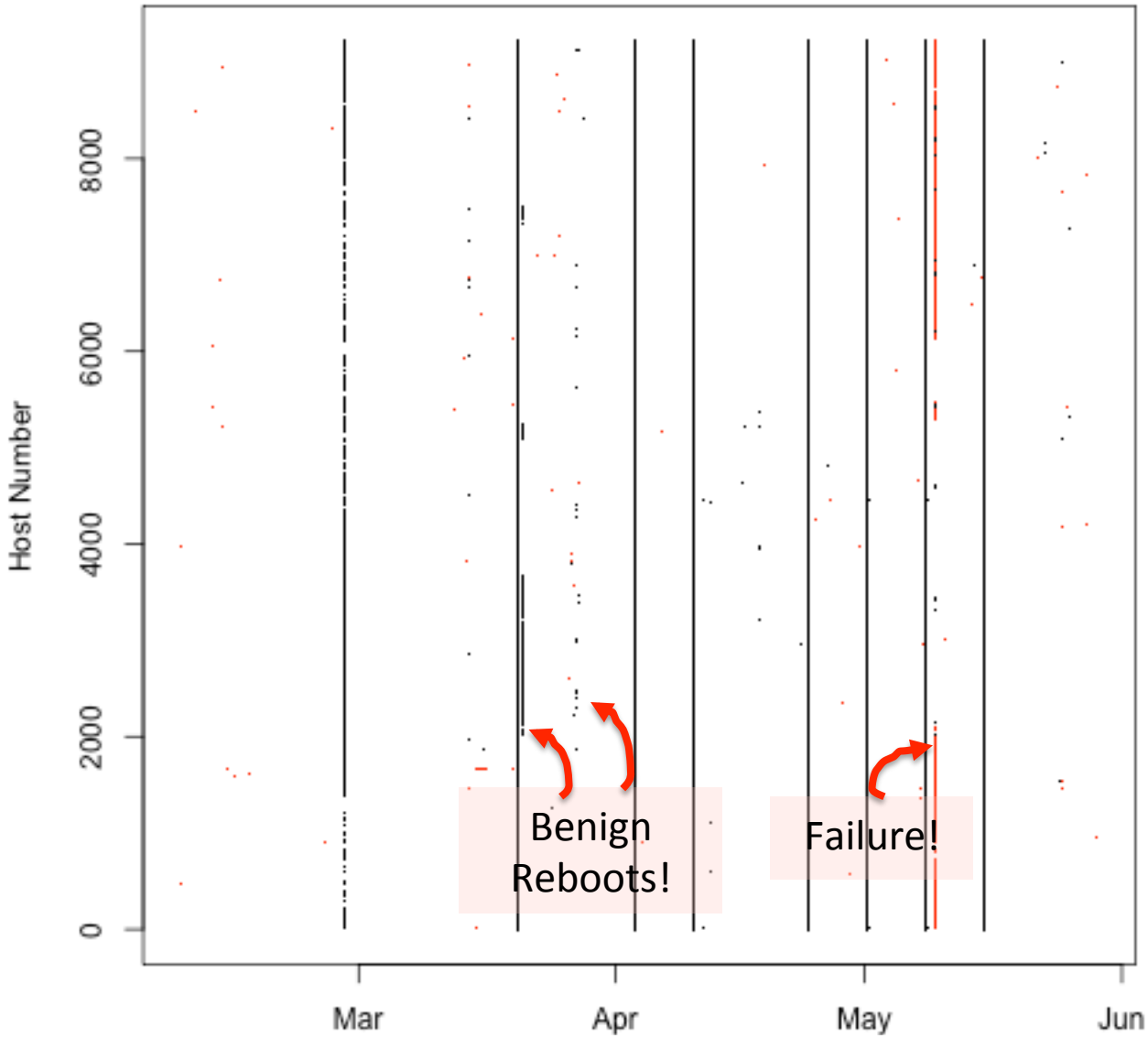
(70,126 total)

Which of these dots indicate “failures”?

(1,784 total, at 77 distinct times)

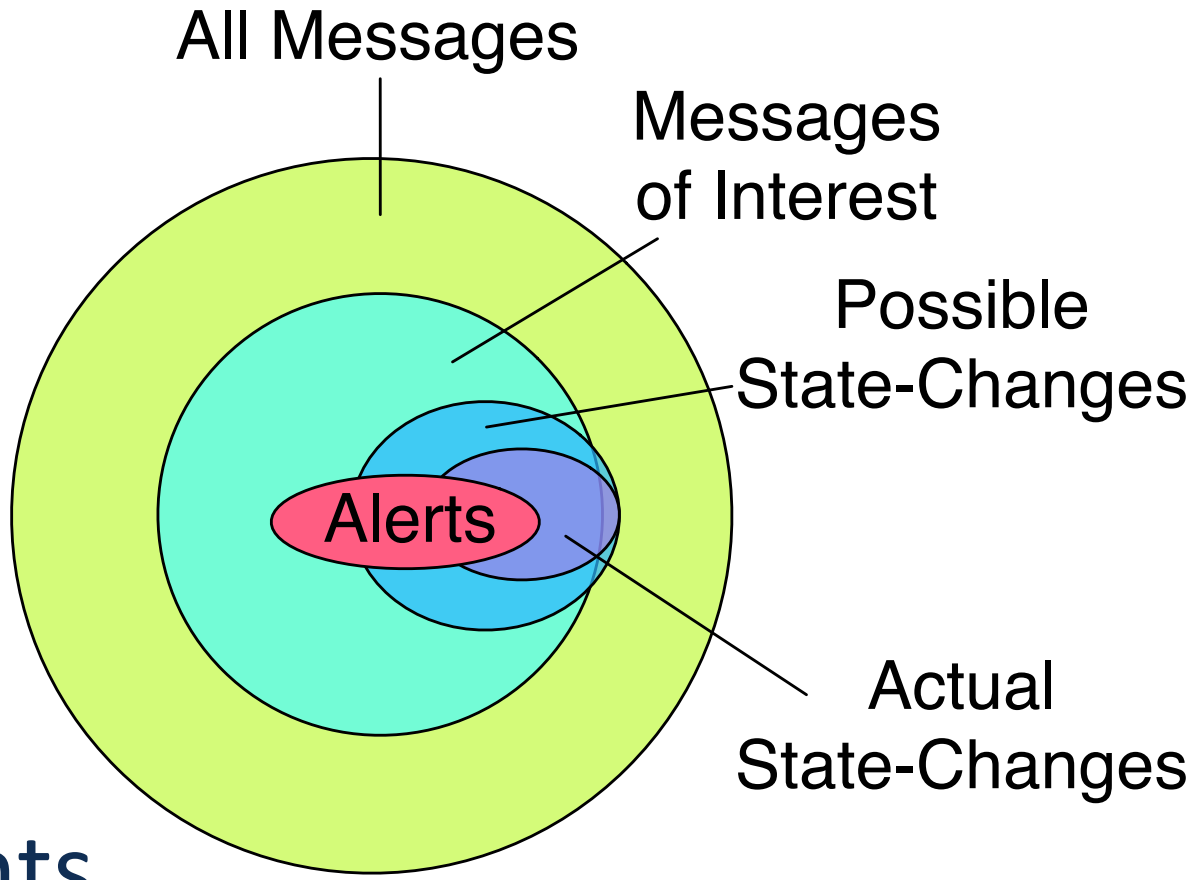
How would you calculate mean time to failure?

Or model failure distribution, etc?



For accurate understanding of our systems,
event disambiguation is essential!

Event Classes



Example Events

Event Type	Splunk syntax
example message, ... indicates truncation due to space limitation	

vs CRAYHEARTSTOPWARN="warn" (and not "considered dead" yet)

CRAYHEARTSTOPALERT **sourcetype=cray ec_heartbeat_stop "considered dead"**

```
2012-05-29 12:55:45 ec_heartbeat_stop src:::c13-1c2s1 seqnum:0x0 svc:::c13-1c2s1n1(131276)[alert node
heartbeat fault] Cause:ec_null Text:node c13-1c2s1n1 considered dead, last heartbeat 00025e67, HT cave
indicates a HT link is down, HT lockup criteria met, HT lockup reset is on, socket 0 core 0 bank 0 status
0xb600000000000185 addr 0x0000000000000400 ...
```

MOABRSVSTARTSYS **sourcetype=moabstats etype=rsvstart rsvid=PreventMaint.* OR rsvid=system* OR rsvid=*-sys.***

```
05:24:50 1338290685:2551224 rsv PreventMaint.2 RSVSTART 1338290660 1338290660 1538290660 8894 8894
0.000000 0.000000 92,93,2,3,9212,...(every nid appears individually, we have truncated 9000+ nids here) -
RSV=%=PreventMaint.2=; Other - -
```

States

Operational State	SLURM	MOAB	CRAY
Up	Up	Up	Up
Suspect	Draining	Down	Suspect
SoftDown	Drained, Drained*	Down	Admindown
HardDown	Draining*, Down	Down	Down
SchedDown	(not tracked as a distinct state)		

Production Uptime

Up

All evidence suggests that host is providing correct service.

Suspect

Evidence of possible problems.

Unknown

Insufficient information.

Unscheduled Downtime

HardDown

Harshly terminated service.

SoftDown

Gracefully terminated service.

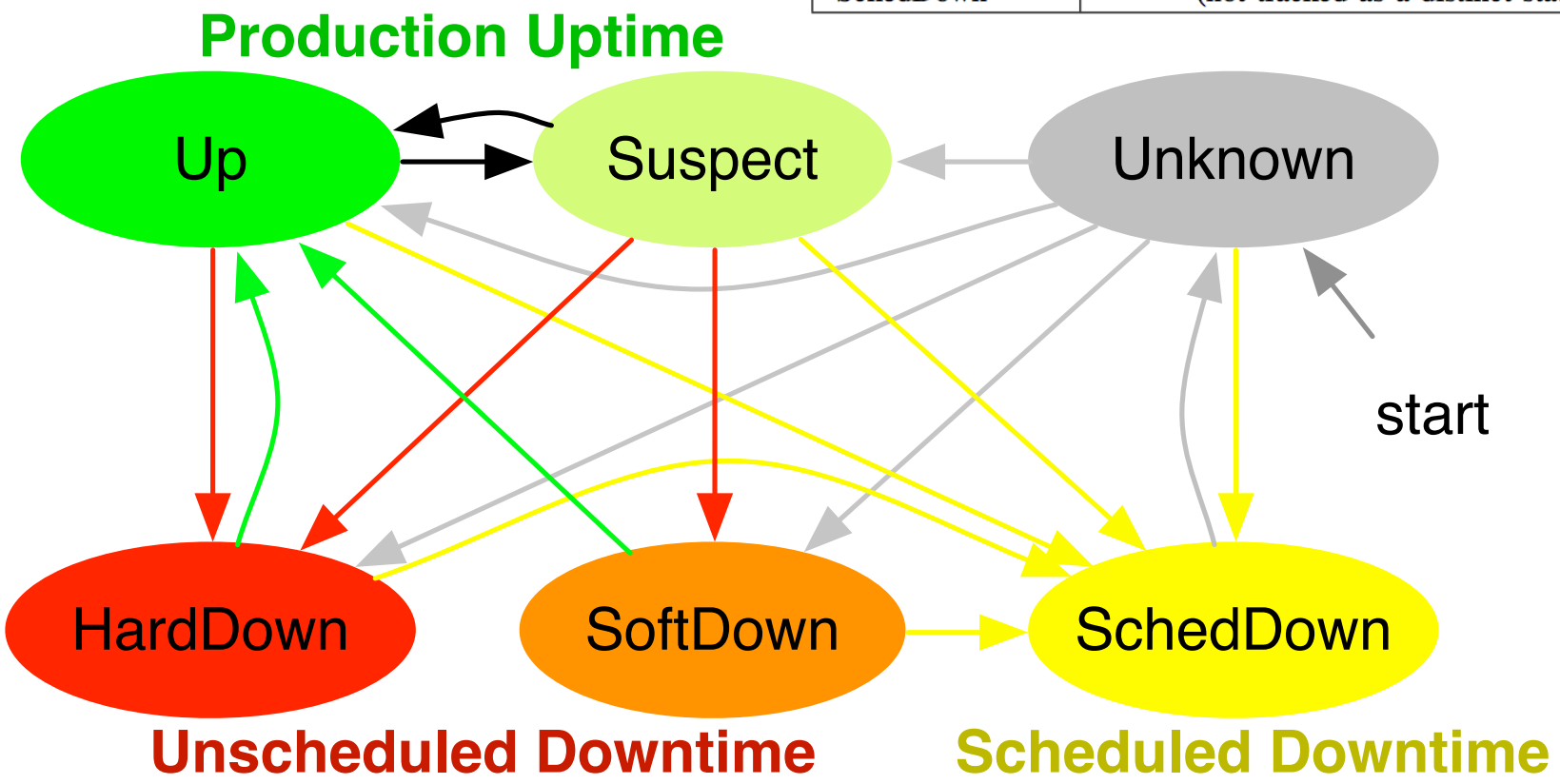
Scheduled Downtime

SchedDown

Expected hiatus in service.

State Machine

Operational State	SLURM	MOAB	CRAY
Up	Up	Up	Up
Suspect	Draining	Down	Suspect
SoftDown	Drained, Drained*	Down	Admindown
HardDown	Draining*, Down	Down	Down
SchedDown	(not tracked as a distinct state)		

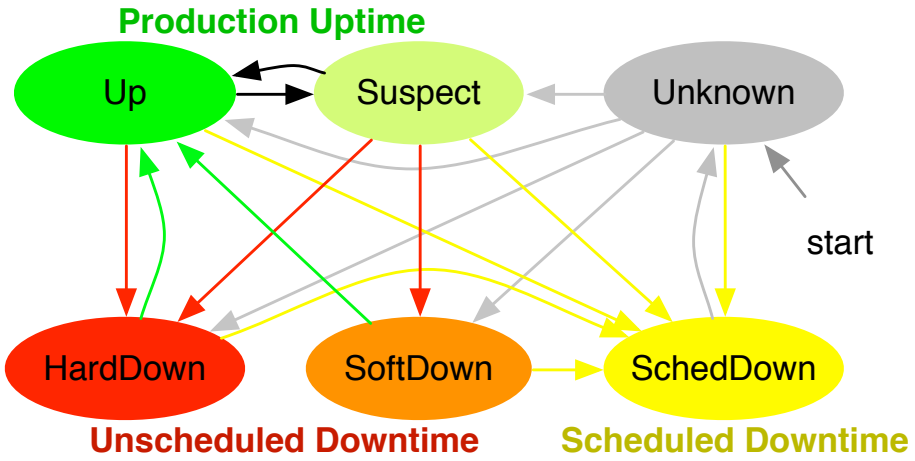


Consider a 10k host supercomputer:

Track the state of each host separately, using the above state machine.

Transition Rules

Operational State	SLURM	MOAB	CRAY
Up	Up	Up	Up
Suspect	Draining	Down	Suspect
SoftDown	Drained, Drained*	Down	Admindown
HardDown	Draining*, Down	Down	Down
SchedDown	(not tracked as a distinct state)		



"eventtype=cos_**X**-**Y**"

Defines a transition rule from **X** to **Y**.

Examples:

"cos_**Up**-**Suspect**=MOABUNSCHEDULEABLE | CRAYHEARTSTOPWARN | CRAYSUSPECT

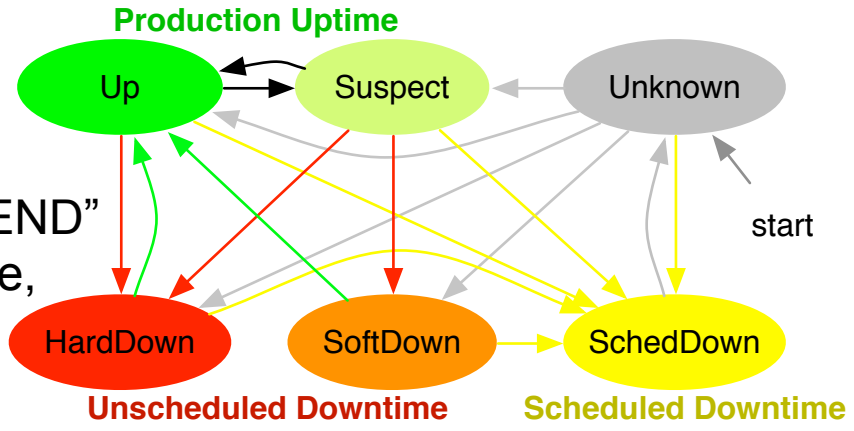
If a host in **Up** emits any of these messages, count it as having transitioned to **Suspect**

"cos_**HardDown**-**Up** = MOABJOBSTART | MOABSCHEDULEABLE | CRAYBOOTED | CRAYAVAIL | CRAYHEALTHUP

If a host in **HardDown** emits any of these messages, count it as having transitioned to **Up**

Transition Table

"cos_**SchedDown-Unknown**=MOABRSVSYSEND"
That's the only way out (ignore everything else, such as CRAYHEARTSTOPALERT)!



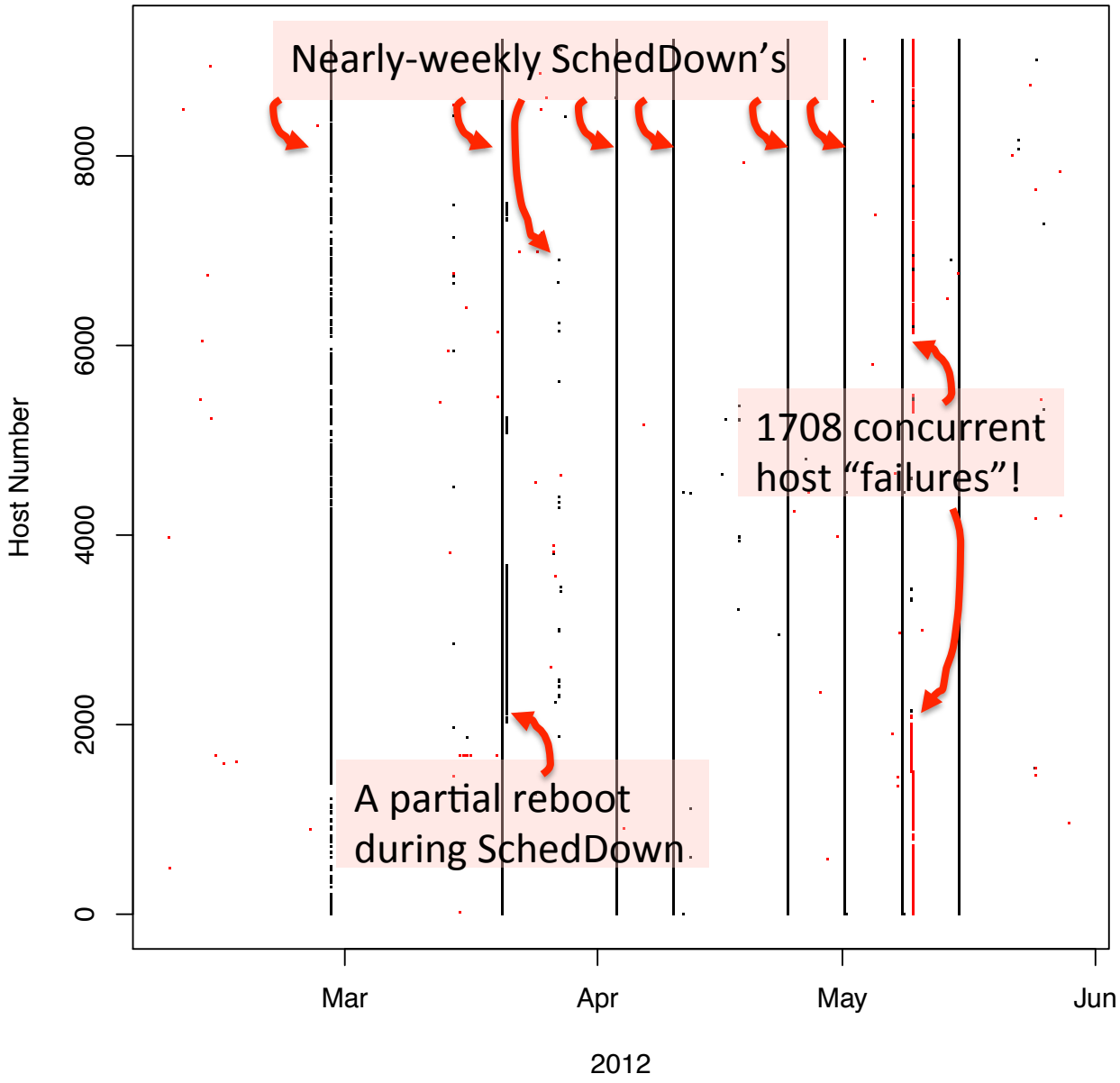
New State	Event Type	Source	Interpretation
Up	MOABJOBSTART	Moab	A Job is starting.
	MOABSCHEDULEABLE	Moab	Scheduler deems the host to be available for scheduling.
	CRAYBOOTED	Syslog	A host has completed its boot-up process.
	CRAYAVAIL	Cray	Cray management software deems the host as available for scheduling.
	CRAYHEALTHUP	Syslog	The "node health checker" has set a compute host to the available state, e.g. after the host has entered a Suspect state and extensive checks were run on the system before risking starting additional jobs on it.
Suspect	MOABUNSCHEDULEABLE	Moab	Scheduler deems the host to be unavailable for new jobs.
	CRAYHEARTSTOPWARN	Cray	A host heartbeat has gone silent, indicating that the host might be dead.
	CRAYSUSPECT	Syslog	The "node health checker" has set a compute host to suspect because the host failed a health test.
SoftDown	CRAYADMINDOWN	Syslog	The "node health checker" (or system administrator) has marked a host as warranting additional attention before it should be trusted to run new jobs.
HardDown	CRAYHEARTSTOPALERT	Cray	A host heartbeat has gone silent for long enough that the host is considered to be dead, or there are directly observable signs that the host has died, such as well-known error codes in the heartbeat memory location. Cray management software will automatically kill dependent jobs, halt scheduling, or similar actions depending on the criticality of the declared dead host. See Table III for an example.
	KERNEL_PANIC	Cray	A host kernel has panic'd (died). Most such events come through the Cray logs, there are are some hosts for which kernel panics can only be seen via syslogs.
SchedDown	MOABRSVSTARTSYS	Moab	A system reservation has started. Typically used to designate the start of a system maintenance period. See Table III for an example.
Unknown	MOABRSVENDSYS	Moab	A system reservation is complete.

Disambiguated - CRAYHEARTSTOPALERT

Red = “Failure”!
(A transition to Unscheduled Downtime)

Black = Benign!
(A normal reboot)

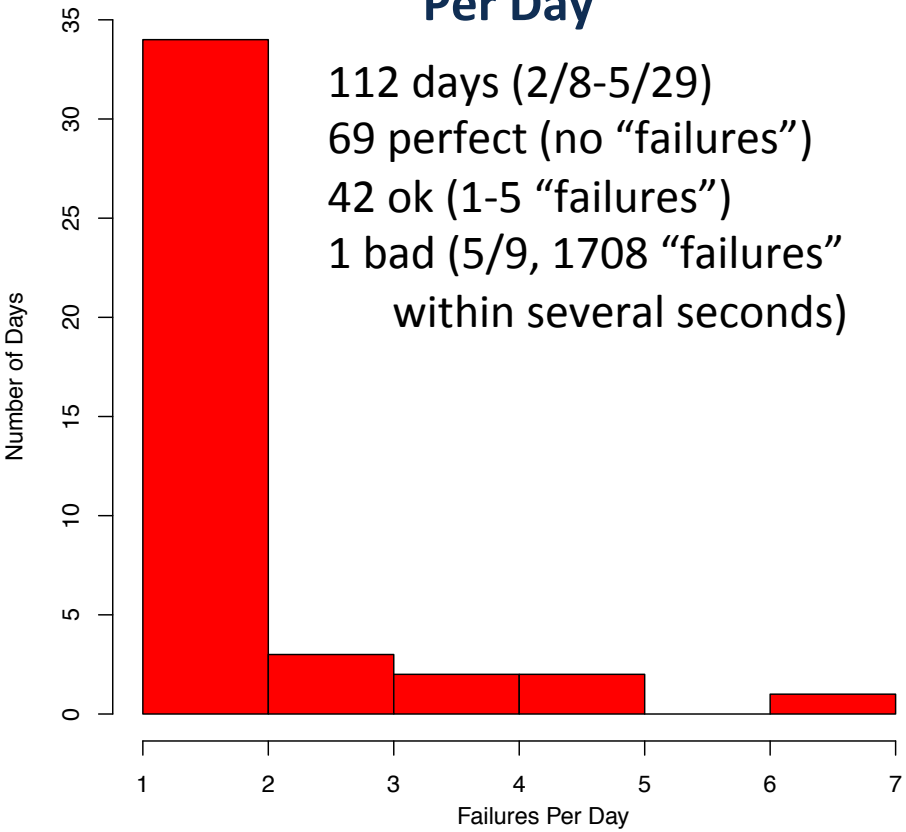
How else would you disambiguate the partial reboot and 1708 event?



Summary Statistics

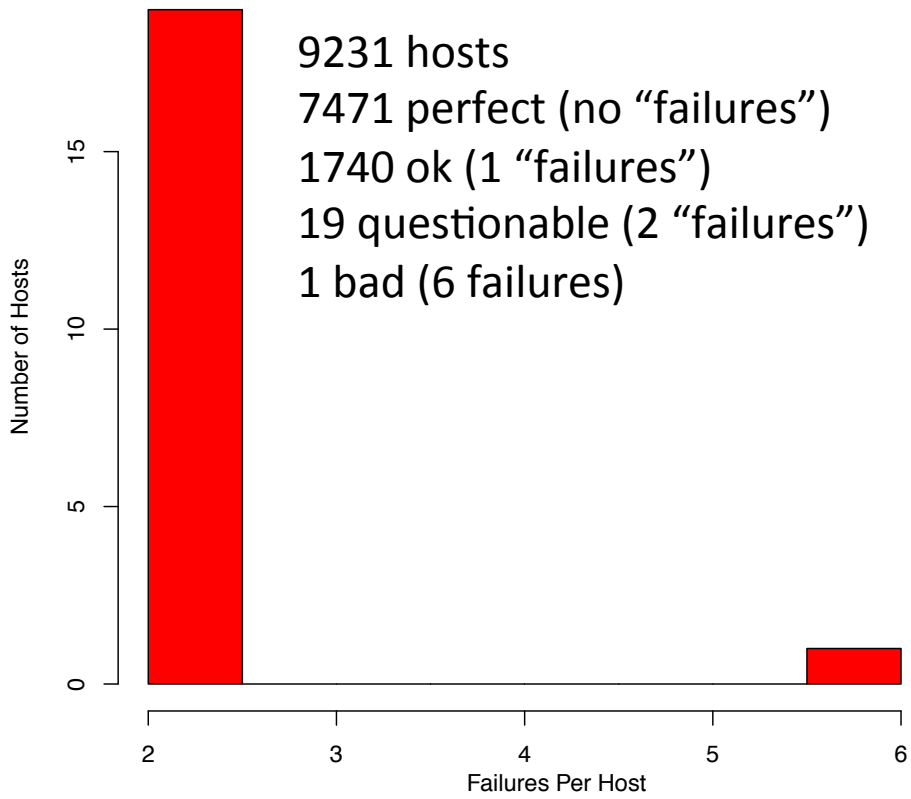
Call a CRAYHEARTSTOPALERT event indicating a transition to HardDown a “failure”:

Per Day



112 days (2/8-5/29)
69 perfect (no “failures”)
42 ok (1-5 “failures”)
1 bad (5/9, 1708 “failures” within several seconds)

Per Host



9231 hosts
7471 perfect (no “failures”)
1740 ok (1 “failures”)
19 questionable (2 “failures”)
1 bad (6 failures)

MTT“F”:

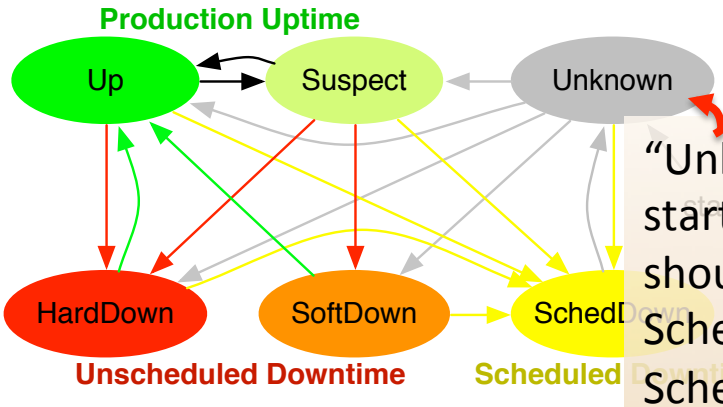
If we count 5/9 as a single “failure”: $\frac{112 \text{ days} * 9231 \text{ hosts}}{77 \text{ “failures”}}$ = 37 years/“failure”/host

If we count 5/9 as having 1708 “failures”: $\frac{112 \text{ days} * 9231 \text{ hosts}}{1784 \text{ “failures”}}$ = 1.6 years/“failure”/host

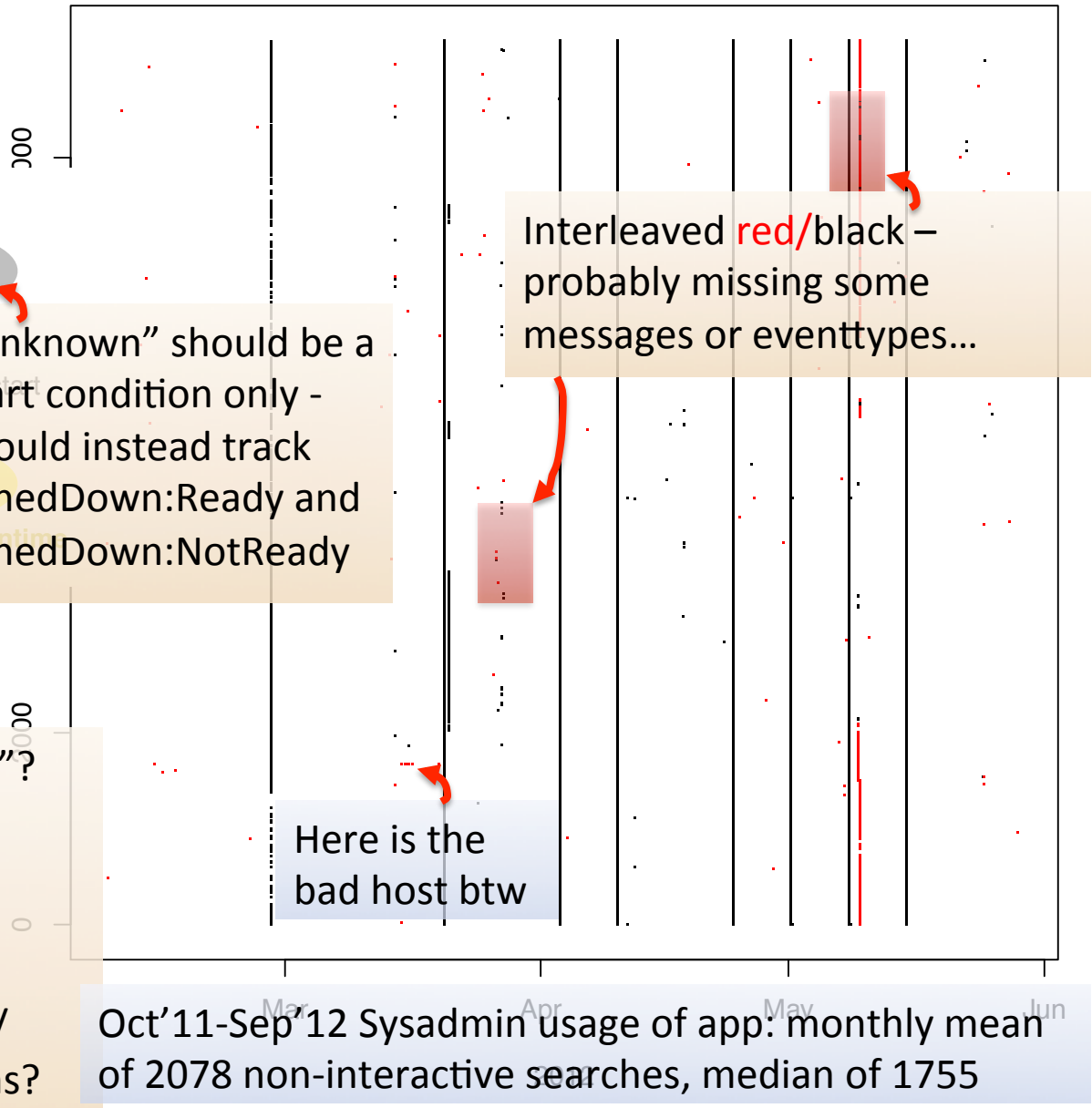
If we naively count all 70126 CRAYHEARTSTOPALERTS: = 1.5 days/“failure”/host

Future Work

How about other event types?
And other systems?



"Unknown" should be a start condition only - should instead track SchedDown:Ready and SchedDown:NotReady



Interleaved red/black – probably missing some messages or eventtypes…

Here is the bad host btw

How does this relate to job "failures"?

How does it help detection and prediction efforts?

What does it tell us about HPC fault/ error/failure modes and distributions?

Oct'11-Sep'12 Sysadmin usage of app: monthly mean of 2078 non-interactive searches, median of 1755

The End

Sandia National Laboratories

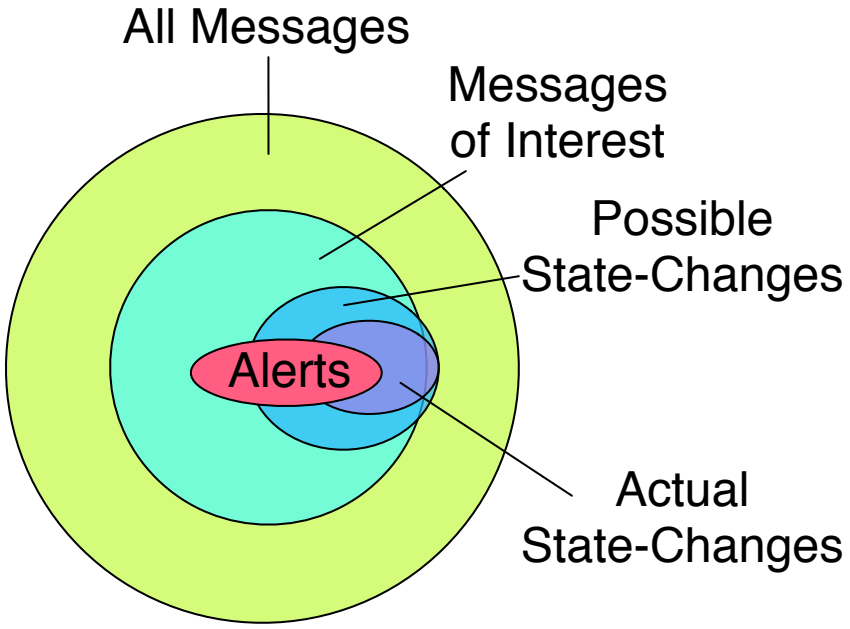
For accurate understanding of our systems,

event disambiguation is essential!

The Splunk “app” is called “HPC” and will appear at http://splunk-base.splunk.com/apps soon…

(extra slides follow)

Stack of Splunk Mechanisms



Splunk Mechanism	Example Splunk-HPC *app* objects
Dashboards	DownHosts, NMTTI, SMTTI, JMTTI, ...
Saved Searches	hoststate, SystemHardDownAlert, ...
Custom Commands	stateMachine.py
Lookups	hostlist.py, hostnames.csv, ...
Eventtypes	cos_HardDown-Up, OOM, MCE, ...
Indexes	hpc_SystemA, hpc_SystemB, ...