



Defining and Measuring **Red Storm** Reliability, Availability, and Serviceability (RAS)

Jon Stearley

Sandia National Laboratories

May 18, 2005

Cray Users Group 2005 Conference

See <http://www.cs.sandia.gov/~jrstear/ras/>



Outline

Problem:

Can't agree on terms!?!?

Proposal:

State model

Definitions

Measurements

A primary goal of this work is foster discussion:

ASK QUESTIONS at any time!!!



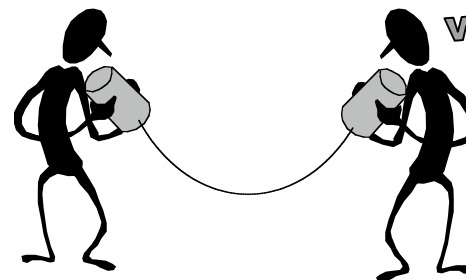
Expensive Supercomputers (but poor RAS?)

The lack of standardized RAS definitions and measurements:

- Obscures meaningful discussion of the real issues
- Delays real RAS improvements
- Increases costs
(in all phases: procurement, operation,
and end-of-life determination)



**My supercomputer
is SO RELIABLE!!!**



that depends on
what you mean by
"fault"...



e.g. Procurement

“Failure of single component will not cause the full system to become unavailable...”

(Red Storm, Purple, Thunder, Q)

“MTBI for full system shall be greater than 50 hours... for a single application”

“MTBI for full system (reboot) shall be greater than 100 hours...”

(over how many samples?)

(Red Storm)

“100 hour capability jobs (90% of system) will successfully complete 95% of the time...” **(= 79 days of failure-free computing?)**

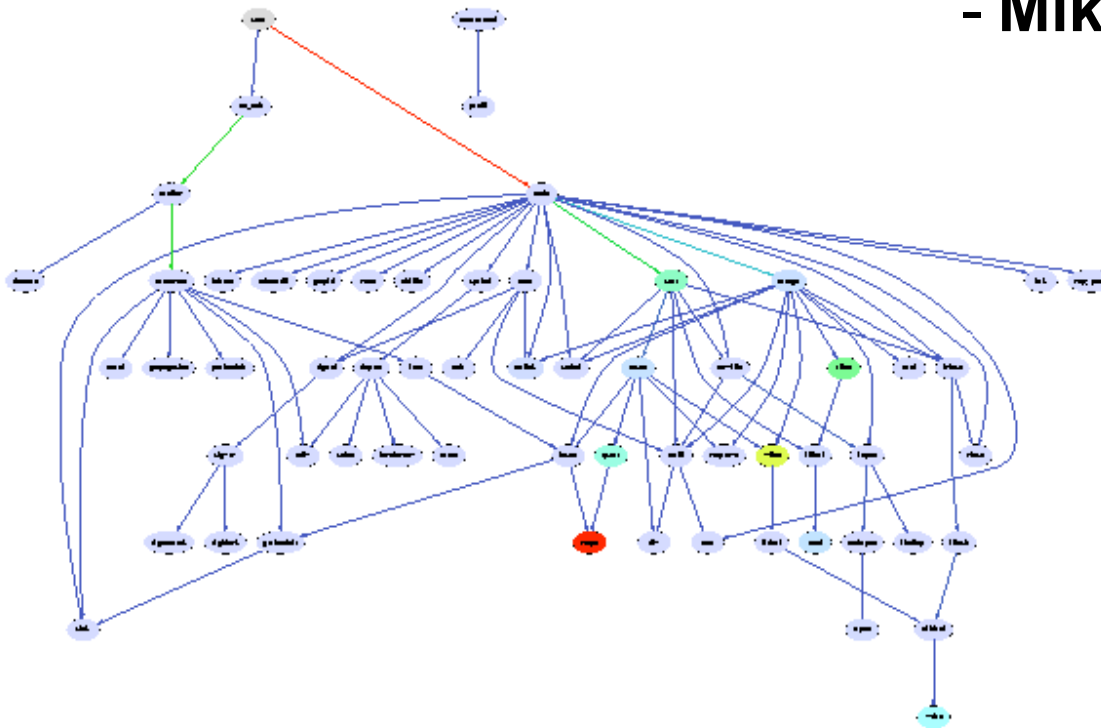
“Over any 4 week period, the system will have an effectiveness level of at least 95%...” **(Purple)**



System State

“A computer is in one of two situations. It is either known to be bad or it is in an unknown state.”

- Mike Levine (PSC)



Is the
“system”
“up”
(yet)?



e.g. Operation

Excerpt from <http://www.nerdc.gov/nusers/status/AvailStats/>:

System Availability Details

FY05 - FEBRUARY 2005										
System	Scheduled		Un-Scheduled			Overall Avail %	Sched Avail %	^MTBI (Hours)	^^MTR (Hours)	^^^MTBF (Day:Hour:Min)
	H/W	S/W	H/W	S/W	Other					
Parallel	99.61%	99.22%	99.87%	99.21%	100.00%	97.92%	99.07%	226	4.8	9 05:46
Storage	99.26%	99.39%	99.67%	99.91%	100.00%	98.23%	99.58%	207	3.4	8 13:00
File Servers	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%			
Math/Vis Servers	100.00%	99.82%	100.00%	99.85%	100.00%	99.66%	99.84%	1208	4.00	25 11:48

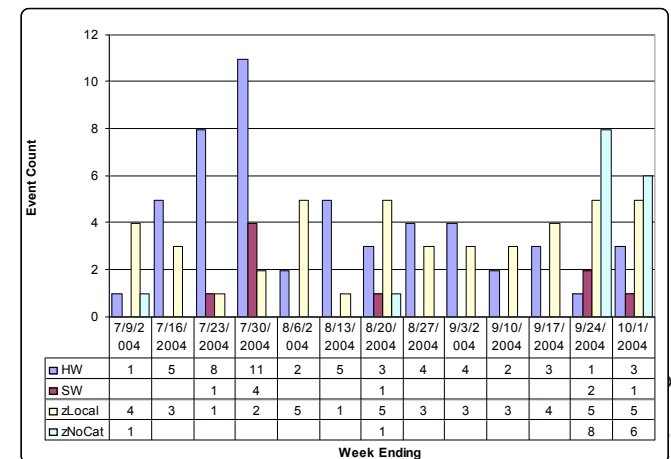
^Mean Time Between Interruptions = Total wall clock hours/total number of downtime periods

^^Mean Time To Restoral = Total downtime hours/total number of downtime periods

^^^Mean Time Between Failures = Total wall clock hours - Total downtime hours/Total downtime hours -1

Excerpts from LLNL ASC White “Six Sigma Report”

Estimated MTTI				
sector	week ending item	8/27/04	9/3/04	
snow+white+frost+ice				
	# failures TOTAL	7	7	
	# failures HW	4 57%	4 57%	
	# failures SW	0 0%	0 0%	
	# failures LOCAL	3 43%	3 43%	
	# nodes	624	624	
	# hours	168	168	
	# node-hours	104832	104832	
	MTBF (hr)	24	24	
	MTBF (hr/node)	14976	14976	





Accurate Comparisons?

Systems	CPUs	Reliability & Availability
ASCI Q	8,192	MTBI: 6.5 hrs. 114 unplanned outages/month. ◆ HW outage sources: storage, CPU, memory.
ASCI White	8,192	MTBF: 5 hrs. (2001) and 40 hrs. (2003). ◆ HW outage sources: storage, CPU, 3 rd -party HW.
NERSC Seaborg	6,656	MTBI: 14 days. MTTR: 3.3 hrs. ◆ SW is the main outage source. Availability: 98.74%.
PSC Lemieux	3,016	MTBI: 9.7 hrs. Availability: 98.33%.
Google	~15,000	20 reboots/day; 2-3% machines replaced/year. ◆ HW outage sources: storage, memory. Availability: ~100%.

MTBI: mean time between interrupts; MTBF: mean time between failures; MTTR: mean time to restore

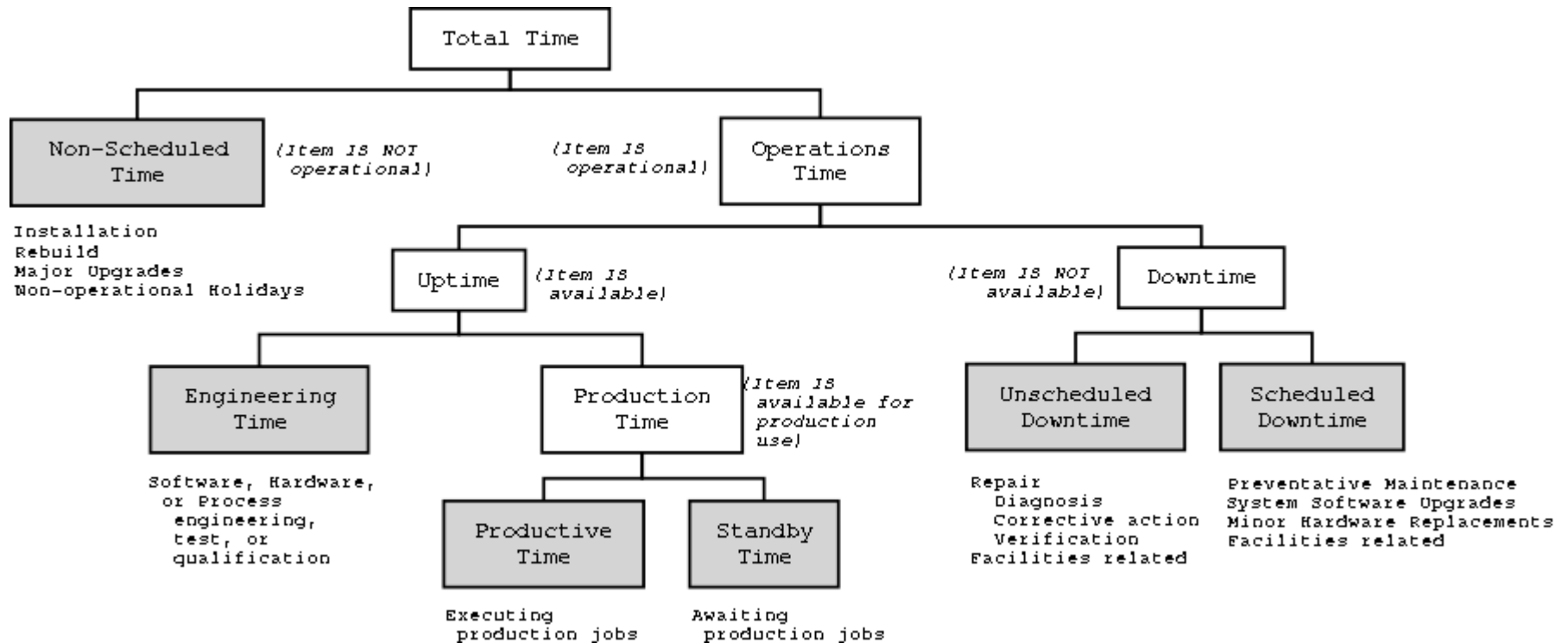
Source: Daniel A. Reed, UNC (via Chung-Hsing Hsu, LANL)



State Model

(adapted from SEMI-E10)

- Items are **always** in one of the **six** basic states (grey boxes).
- Time is hierarchically categorized (white boxes).

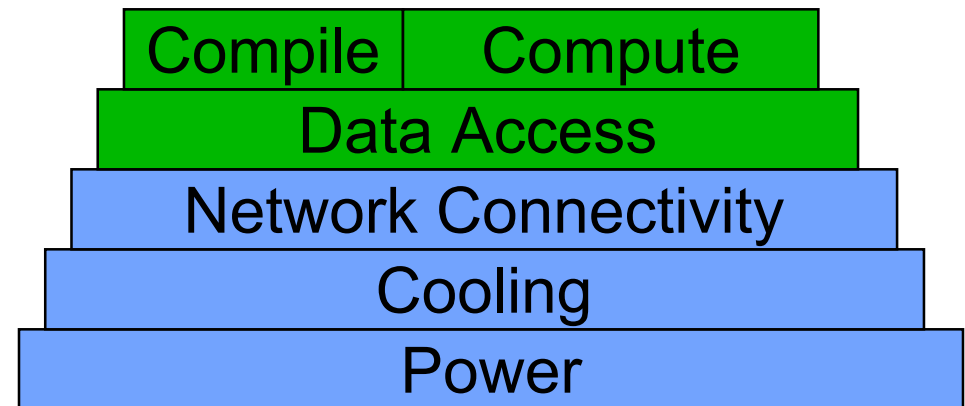


Definitions: Reliability

Reliability (IEEE):

The probability that an *item* will:

- **function** without failure
- under **stated conditions**
- for a specified **amount of time**.



$$R(t) = e^{-\lambda t}, \lambda = 1/\text{MTBF?}$$

Is constant failure rate really appropriate?



item – an all-inclusive term to denote any level of unit (e.g. component, system, etc.)



Cause vs Effect: Failure vs Interrupt

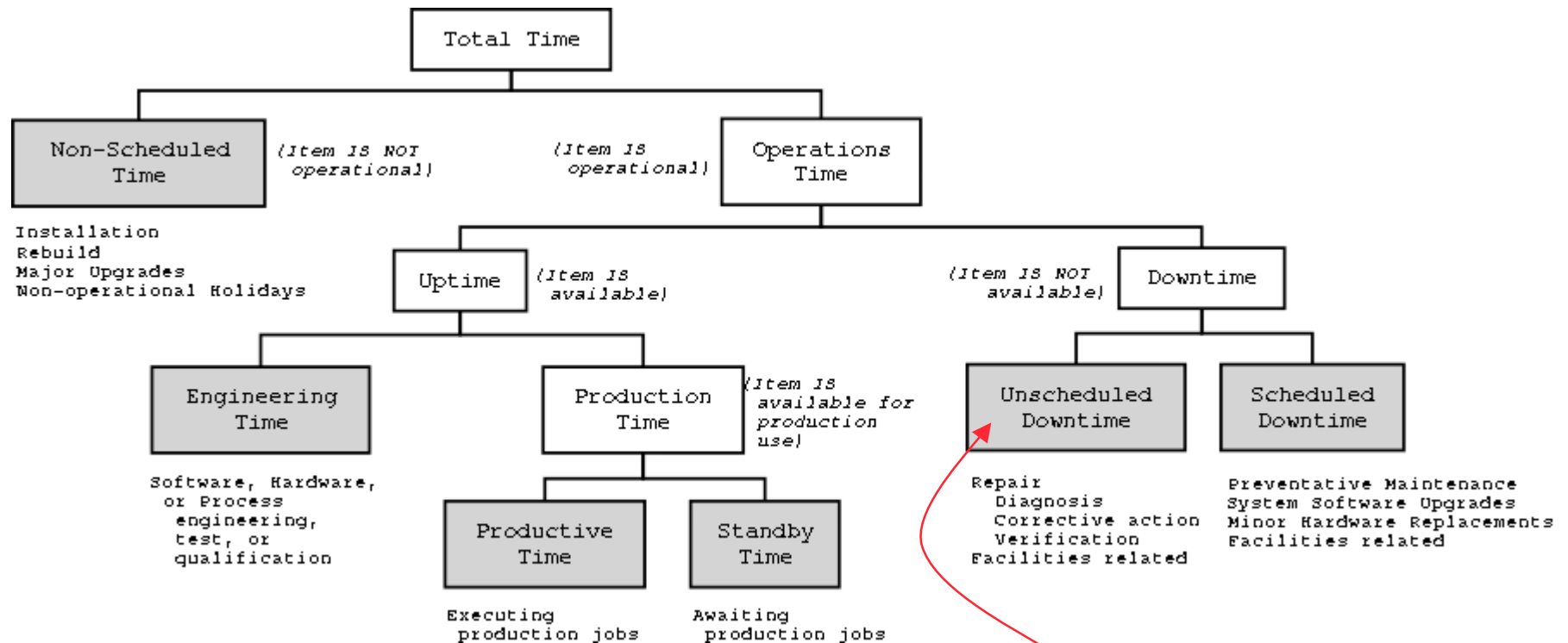
Failure – the termination of the ability of an item to perform a required function.
External corrective action is required into order to restore this ability (e.g. manual reboot, repair, replacement).

Interrupt – the suspension of a process to handle an event external to the process.
(yes, this ISO9000 definition is somewhat vague...)

**Failures regard items; interrupts regard work.
Failures *may* cause interrupts.**



State Model



“failure” = any transition into
Unscheduled Downtime

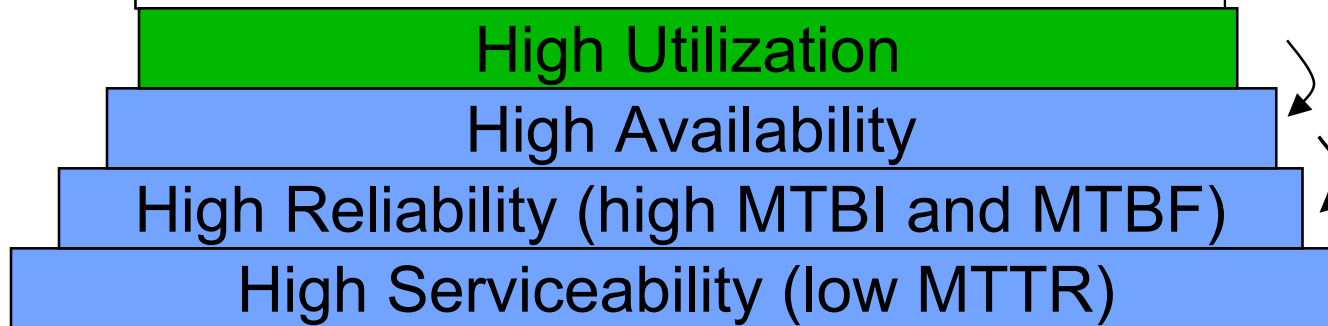
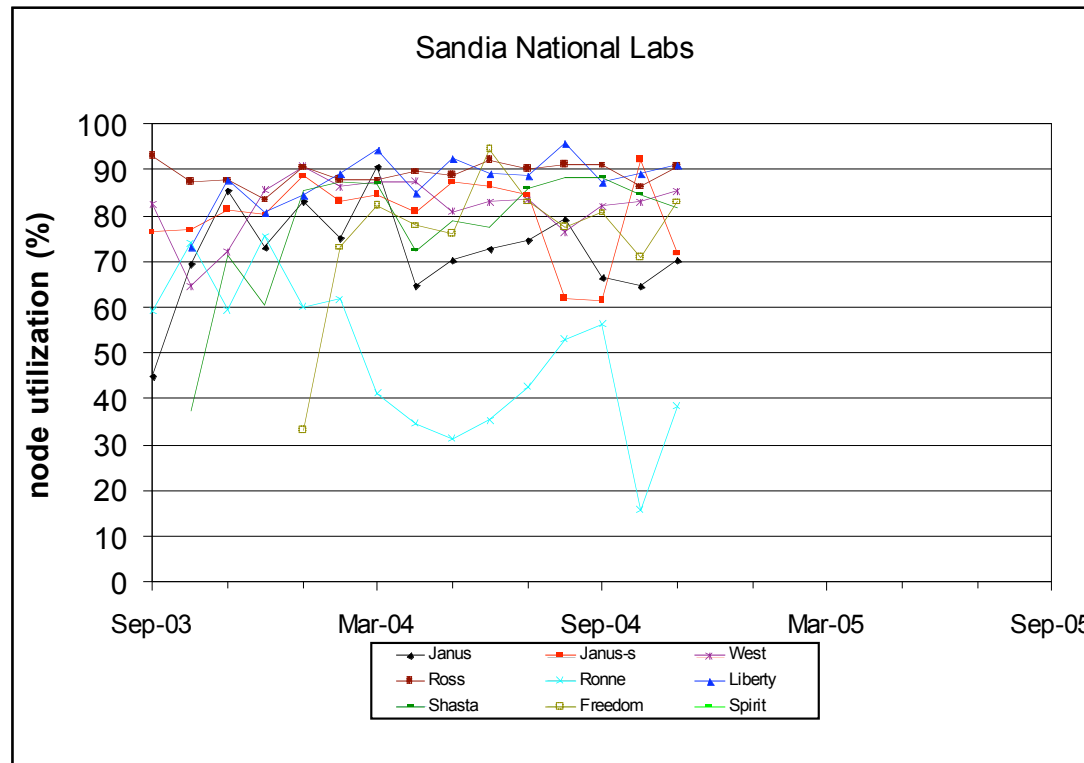
A general model, requiring system-specific details for application.

See <http://www.cs.sandia.gov/~jrstear/ras/>



Utilization

(a common aggregate metric)



implies?



See <http://www.cs.sandia.gov/~jrstear/ras/>

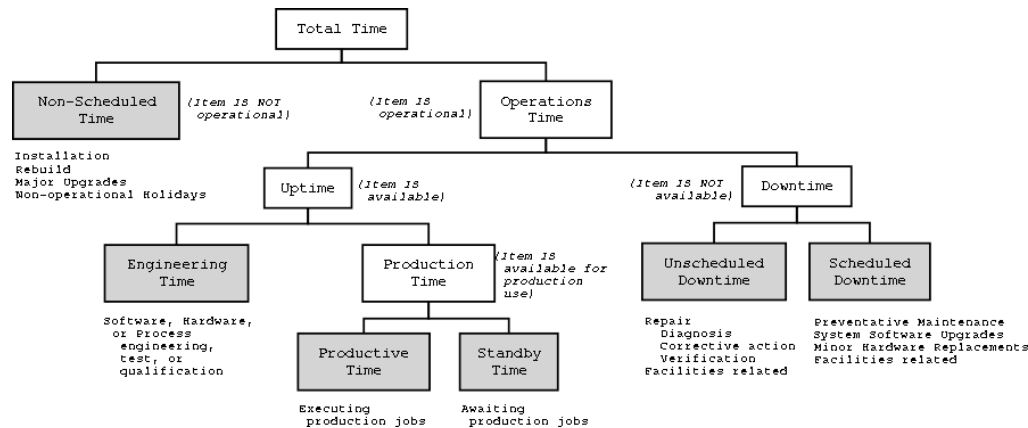


Utilization

Non-RAS:

$$\text{Production Time System Utilization (\%)} = \frac{\text{productive nodehours}}{\text{production nodehours}} * 100$$

This has NO RAS information! It is entirely a function of workload and queuing configuration.



RAS:

$$\text{Total System Utilization (\%)} = \frac{\text{productive time} * 100}{\text{total time}}$$

This IS a useful aggregate RAS metric. High total system utilization DOES indicate high reliability (MTBI, MBTF), availability, and serviceability (MTTR).

See <http://www.cs.sandia.gov/~jrstea/ras/>





Mean Time Between Job Interrupts

Common:

$$MTBI = \frac{\textit{total time}}{\textit{number of interrupts}}$$

Easy to calculate
(but assumes downtime is negligible)

Proposed:

Job Interrupt - The unexpected interruption of an active job.

$$MTBI_{Job} = \frac{\textit{production time}}{\textit{number of job interrupts}}$$

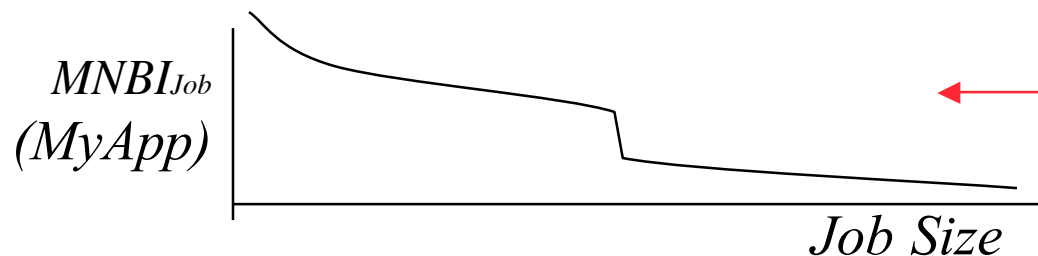
More precise (but no work info)

More specific

$$MNBI_{Job} = \frac{\textit{productive nodehours}}{\textit{number of job interrupts}}$$

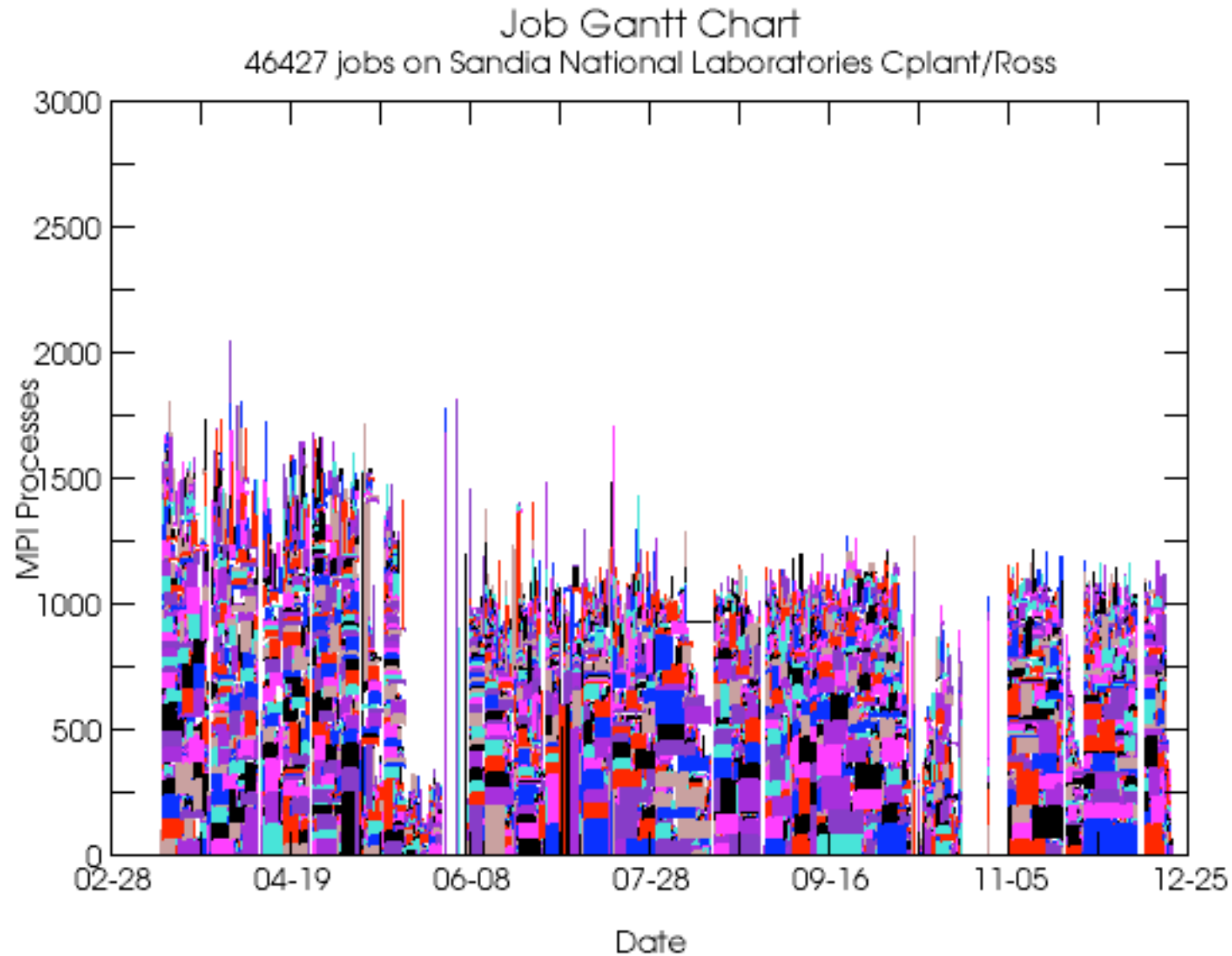
Includes workload information

Conceptual plot of how long an application is likely to run before experiencing an interrupt, as a function of job size.





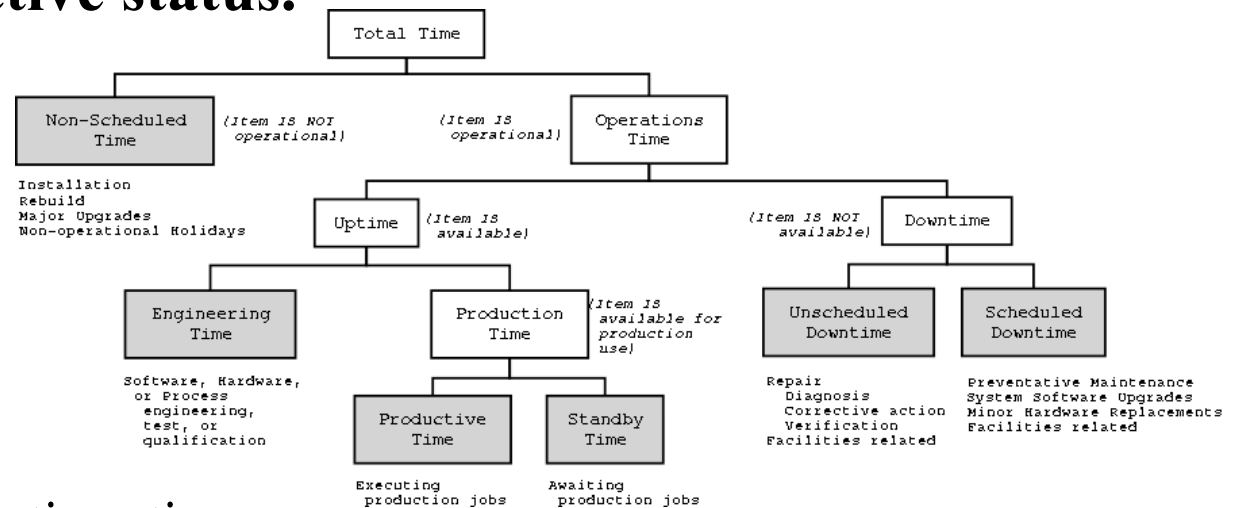
Workload information is vital!





Mean Time Between Node Failures

Node Failure – an event requiring that the node (a component) enter a downtime status before it can enter a productive status.



$$MTBF_{Node} = \frac{\textit{production time}}{\textit{number of node failures}}$$

$$MNBF_{Node} = \frac{\textit{productive nodehours}}{\textit{number of node failures}}$$

← Includes workload information



Red Storm

“Intended Function”

Red Storm is a *production* supercomputer, whose function is manifested via the following services:

Service Name	Description
login	Users can log in to the system.
compile	Users can compile applications.
job	Batch and interactive jobs work (submission, wait, shell-execution, application-execution, and cleanup [1]) work correctly, as are all batch queue functions (jobs can be submitted, queried, removed, and are being appropriately scheduled and executed).
io	Users and jobs can utilize the high performance file system.
scale	At least a certain number of nodes are up (e.g. 95% of the nodes in the section).

Table 1: Critical System Services

The system is...

- “up” = all the above services are working.
- in “degraded” mode = a useful subset of the above are working.
- “down” = none of the above services are working.

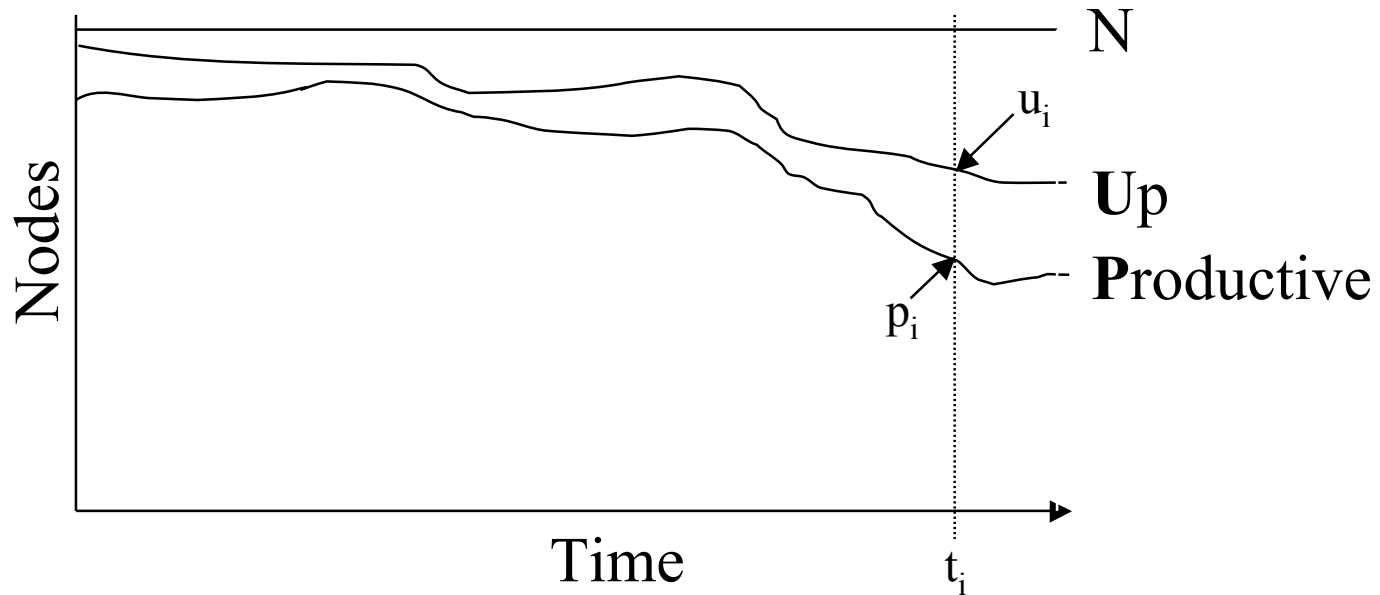
“Service Interrupt” = an interruption in any of the above services.

“System Failure” = an event requiring that the system (the majority of components) enter a downtime status before any component may enter a productive status.
= the system must go down before it can come up.



Red Storm

Low-Level State Criteria



Item	State	Variable	Value
nodes	up	u_i	= SELECT COUNT(processor_id) FROM processor WHERE status='up';
	productive	p_i	= SELECT COUNT(DISTINCT processor_id) FROM partition_allocation;
	standby	s_i	= $u_i - p_i$
	downs	d_i	= $(u_i - u_{i+1}) > 0 ? (u_i - u_{i+1}) : 0$ (an estimate only - see text!)
jobs	interrupted	j_i	= SELECT COUNT(partition_id) FROM job_accounting WHERE destroy_time $\geq t_{i-1}$ AND cleaned_by='ras';

Table 2: Counting compute and login nodes at time t_i

See <http://www.cs.sandia.gov/~jrstea/ras/>



Next Steps

- **Ongoing discussion and revision
(including at DARPA HPCS working meeting in June)...**
- **Actual implementation for Red Storm
(expect to begin tracking state in June)**
- **Application to another SNL platform
(e.g. current production Linux clusters - in process...)**
- **Review failure and interrupt distributions
(towards selecting a model for reliability calculation)**

Interested? Contact me at jrstear@sandia.gov

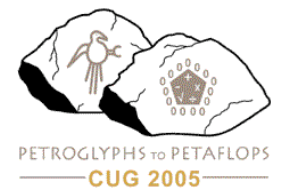
See <http://www.cs.sandia.gov/~jrstear/ras/>



Extra Slides...



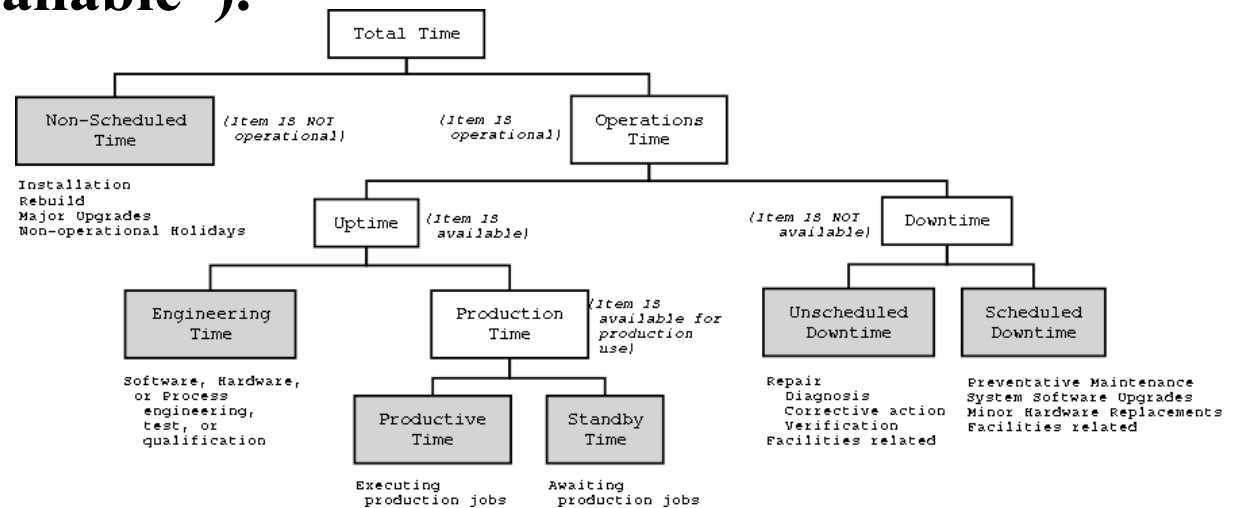
See <http://www.cs.sandia.gov/~jrstear/ras/>





Availability

Availability - the fraction of a time period that an item is in a condition to perform its intended function upon demand (“available”).



$$Total\ Availability_{System} (\%) = \frac{uptime}{total\ time} * 100$$

$$Scheduled\ Availability_{System} (\%) = \frac{uptime - downtime}{scheduled\ uptime} * 100$$

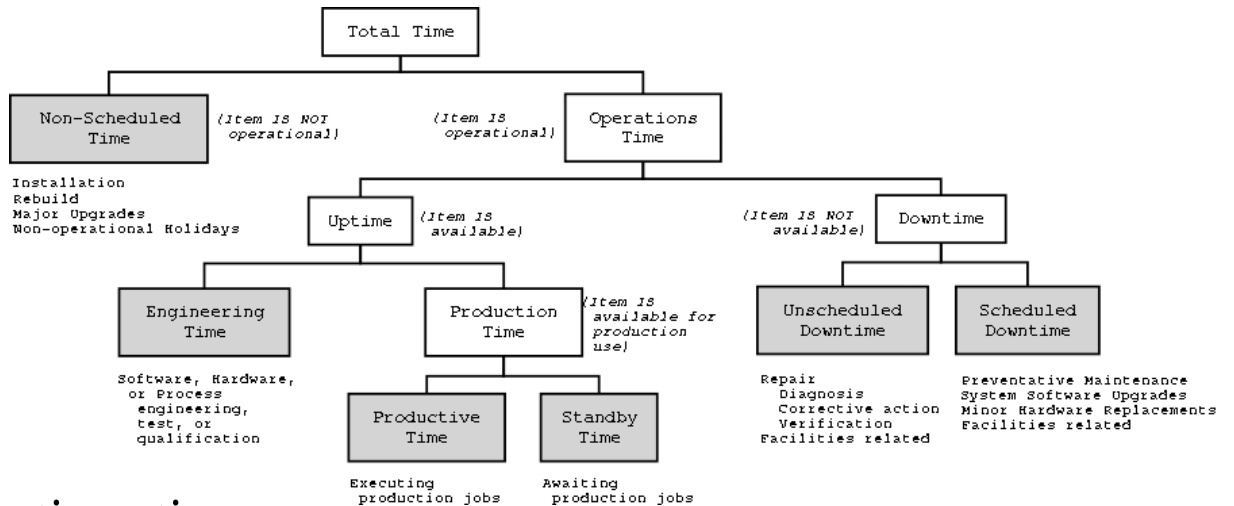
Quantitative expectations exist

See <http://www.cs.sandia.gov/~jrstear/ras/>



Mean Time Between System Failures

System Failure – an event requiring that the system (the majority of components) enter a downtime status before any component may enter a productive status.



$$MTBF_{System} = \frac{\textit{production time}}{\textit{number of system failures}}$$

$$MNBF_{System} = \frac{\textit{productive nodehours}}{\textit{number of system failures}}$$

← Includes workload information

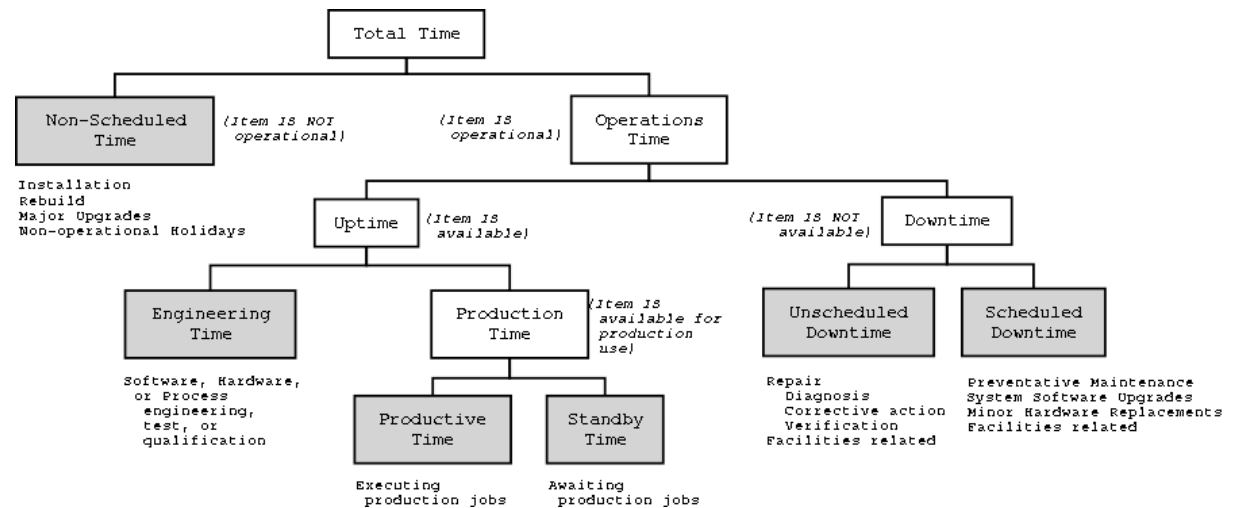
See <http://www.cs.sandia.gov/~jrstear/ras/>





Mean Time Between Service Interrupts

Service Interrupt – any event which disrupts full service to users (for any reason).



$$MTBI_{Service} = \frac{\text{production time}}{\text{number of service interrupts}}$$

$$MNBI_{Service} = \frac{\text{productive nodehours}}{\text{number of service interrupts}}$$

← Includes workload information





Serviceability

Serviceability -

The probability that an item will be retained in, or restored to, *a condition to perform its intended function* within a specified period of time.

(A.K.A. “maintainability” in other communities)

Higher serviceability reduces the time spent in repair and maintenance (thus increasing availability and uptime ratio respectively)



Repair vs Maintain?

Repair – the act of restoring an item to a condition to perform a required function

Maintainance – the act of sustaining an item in or restoring it to a condition to perform a required function, usually during scheduled downtime.

MTTR – mean time to repair

MNTR - mean nodehours to repair (lost work potential)

MTTB - mean time to boot system

MTTR affects availability
(MTBI and MTBF affect availability and reliability)

See <http://www.cs.sandia.gov/~jrstear/ras/>



Exponential Random Variable

Components which exhibit a constant failure rate are appropriately modeled as exponential random variables, which have a time-to-failure pdf of $f(t)=\lambda e^{-\lambda t}$ (and thus a cdf of $F(t)=1-e^{-\lambda t}$), where λ is the constant “failure rate”. Using this model:

$$\text{Reliability} = R(t) = e^{-\lambda t} \quad \text{and} \quad \text{MTBF} = 1/\lambda$$

A *system* MTBI of 50 hours ($\lambda=0.02$) would correspond to a reliability of 0.368. In other words, there is a 36.8% chance that the *system* will not experience an interrupt within 50 hours.

If we model Red Storm as a series *system* of 10,000 nodes which fail independently of each other, and we require a *system* MTBI of 50 hours, this corresponds to a per-node MTBI of 500,000 hours. If the requirement is reduced to a job running on 40% of the nodes, this corresponds to a node MTBI of 200,000.