RAS Benchmarking at Sun: Four Views of Mount Fuji

Richard Elling

Performance, Availability, and Architecture Engineering Group

Sun Microsystems, Inc.

November 8, 2005

Summary

- R^3 benchmarks have been done for a number of systems
- Each benchmark provides a view into the RAS capabilities of a system
- No benchmark stands alone
- No benchmark is perfect
- We have had success showing incremental and generational improvement in product design
- In practice, benchmarks results follow progression similar to grief:
  - > Disbelief
  - > Anger
  - > Acceptance
- Proving useful to product development teams

Why Views of Mount Fuji?

"It struck me that it would be good to take one thing in life and regard it from many viewpoints, as a focus for my being, and perhaps as a penance for alternatives missed."

Roger Zelazny,
24 Views of Mount Fuji, by Hokusai

Availability Benchmark Approach

Availability, by itself, is difficult to translate into a single benchmark or system requirement. We decompose availability into:

- Rate
  - > How often do faults occur?
- Robustness
  - > Do faults cause system outages?
  - > Can the system be repaired online?
- Recovery
  - > How quickly can we return to nominal operation?
- R^3 benchmarks all of these factors.

Rate

- Rate is driven by
  - How many parts are used
    - Redundancy increases rate
    - High levels of integration tend to reduce rate – Moore's Law is a big win!
- The lower the rate, the more reliable the component
- Think Telcordia, MIL-217, etc.
- But we won't go there today... no rate ratholes, please!
- Need relative weight of FITs for each FRU

Example Rate Analysis

Components	Relative Predicted FITs (%)
Disks	10
Power Supplies	15
CPU/Memory boards	20
Other PCBs	20
Memory	25
Fans	5
Miscellaneous and cables	5

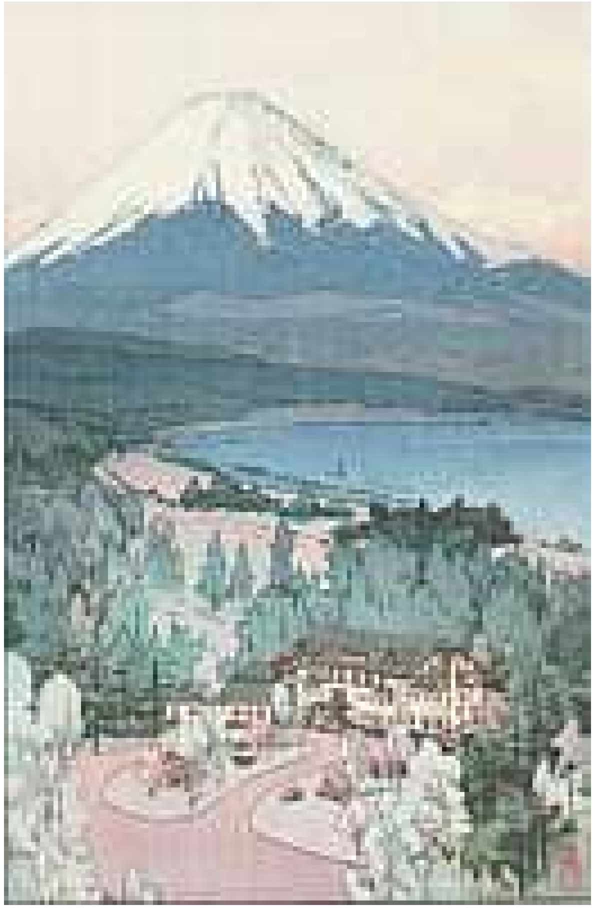
Robustness

- Robustness increases with redundancy
  - N+1, 2N, RAID, mirroring, spare banks and bits
- If something fails, there is a spare
- Error detection and correction
  - Parity with retry, CRC, SEC-DED, SSC-DSD
- Failure prediction based on correctable error counts
  - De-allocate FRUs that have high levels of correctable errors
- Benchmarks used: MRB-A, FRB-A, SCB-M

Recovery

- How quickly can a system automatically return to operation after a fault or maintenance event
  - After either hardware or software faults
- Recovery time drivers
  - POST, OBP, BIOS, Boot loader
  - Fault detection methods
  - OS and service shut down and start up times
  - Membership arbitration and data synchronization
- Benchmarks used: SRB-A, SRB-X

Fault Robustness Benchmark - A



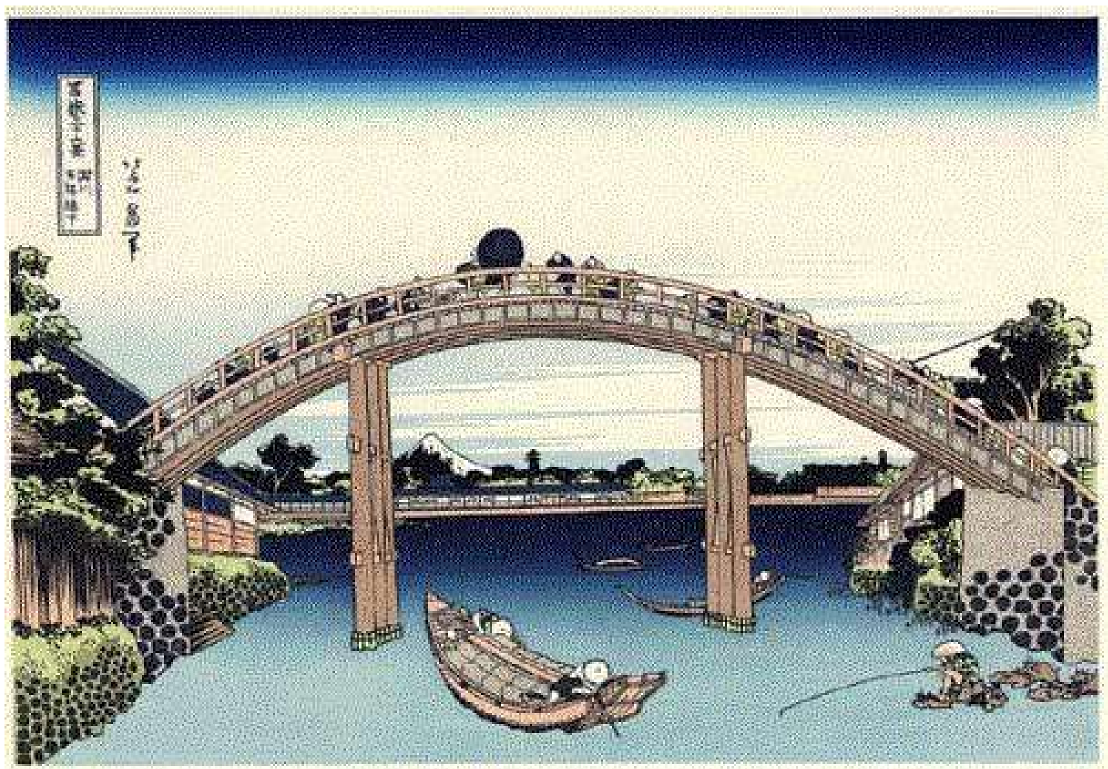
Fault Robustness Benchmark - A

- Rewards systems where faults do not cause disruption of service
  - It is a numeric scalar between 1 and 100
    - 1 = any single failure will cause a disruption
    - 100 = no single failure will cause a disruption
- Rewards redundant systems
  - When less reliable parts are made redundant
  - Not when reliable parts are made redundant
  - Attempts to optimize cost/redundancy trade-off

Example FRB-A Analysis

Components	Relative Predicted FITS (%)	FRB Class Scalar	
		Less Robust	Mirroring, DR, CPU Offlining
Disks	10	1	100
Power Supplies	15	100	100
CPU/Memory boards	20	1	100
Other PCBs	20	1	1
Memory	25	10	10
Fans	5	100	100
Miscellaneous and cables	5	1	1
Score		FRB-A=23.1	FRB-A=52.8

Maintenance Robustness Benchmark



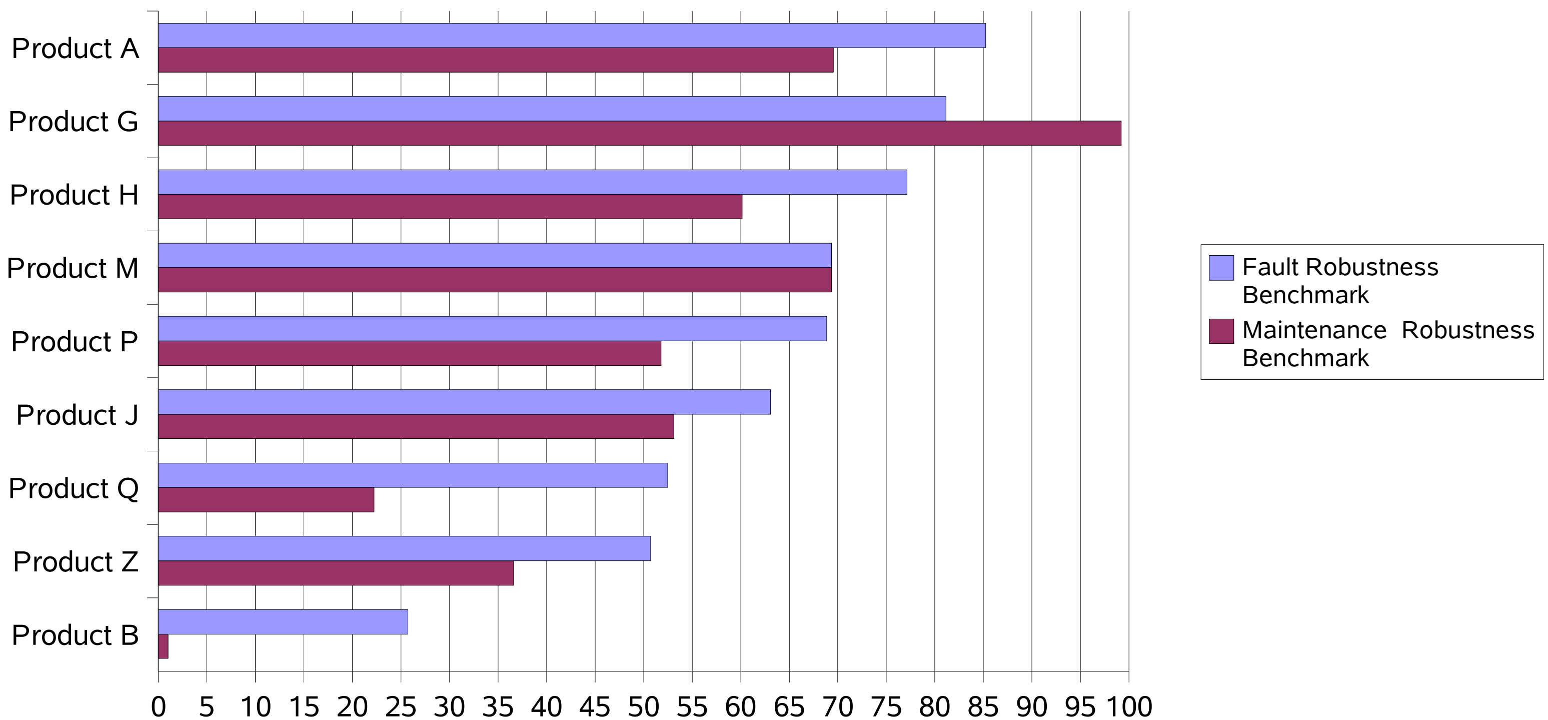
Maintenance Robustness Benchmark - A

- Rewards systems where maintenance does not cause disruption
  - It is a numeric scalar between 1 and 100
    - 1 = all maintenance actions result in a system outage
    - 100 = all FRUs can be replaced without an outage
- Rewards hot swap

Example MRB-A Analysis

Components	Relative Predicted FITS (%)	MRB Class Scalar	
		Less Robust	Mirroring, DR, CPU Offlining
Disks	10	1	100
Power Supplies	15	100	100
CPU/Memory boards	20	1	100
Other PCBs	20	1	1
Memory	25	10	100
Fans	5	100	100
Miscellaneous and cables	5	1	1
Score		**MRB-A=23.1**	**MRB-A=75.3**

R^3 FRB-A and MRB-A Results



Data sorted by Fault Robustness Benchmark Results

System Complexity Benchmark - M

System Complexity Benchmark - M

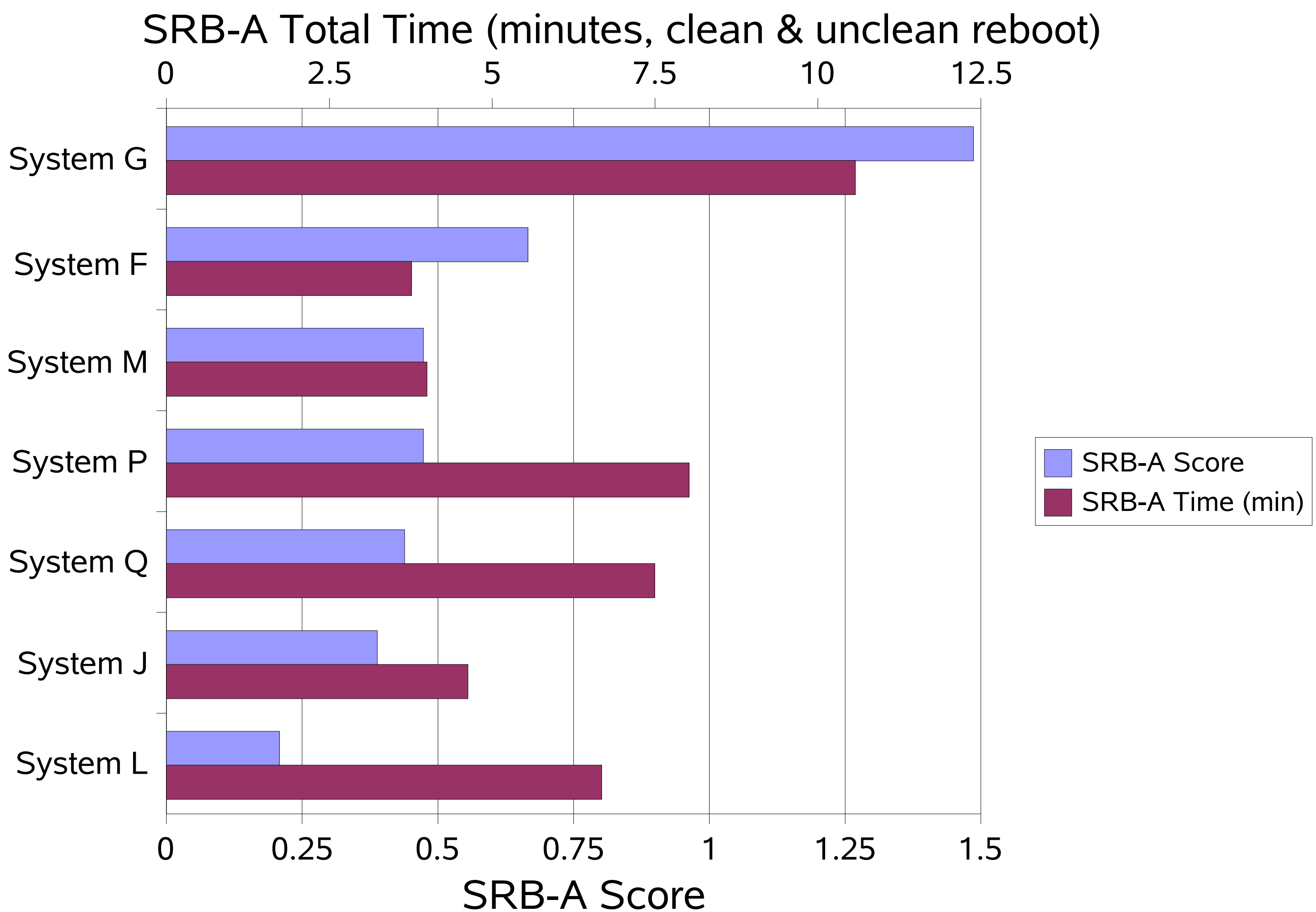
- Measures mechanical complexity for servicing system
- Unbounded score in range 1 - ∞
  - > High score (complexity) is bad
- Rewards systems with:
  - > Hot pluggable FRUs
  - > Require no tools
- Penalizes:
  - > Buried FRUs
  - > Cabling rats nest
  - > Loose fasteners – where does this screw go?

System Recovery Benchmark - A

System Recovery Benchmark - A

- Measures hardware and OS recovery time
  - > Clean shutdown
  - > Clean boot
  - > Unclean boot (OS abort and dump) and recovery
- A scale factor is divided by the total time in minutes
  - > Work in progress
  - > Normalized for system size
  - > SF = 0.1 * #CPUs + 0.4 * GBytes DRAM + 0.05 * # I/O channels + 0.45 * #LUNs
- Rewards systems with fast fault detection, correction and reboot

SRB-A Score and Time Measurement



System Recovery Benchmark - X

System Recovery Benchmark - X

- Recovery benchmark for clusters
- Today, more characterization than benchmark
- Used for generational improvement of entire cluster stack
  - > Initially, many opportunities for improvement in all software and hardware layers
  - > Today, becoming highly optimized

Conclusion

- R^3 benchmarks have been done for a number of systems
- Each benchmark provides a view into the RAS capabilities of a system
- No benchmark stands alone
- No benchmark is perfect
- We have had success showing incremental and generational improvement in product design
- In practice, benchmarks results follow progression similar to grief:
  - > Disbelief
  - > Anger
  - > Acceptance
- Proving useful to product development teams

PA^{2}E – RAS Engineering

pae-ras@sun.com