# *Modeling Leads to Cause of Field Failures*

**David Trindade, Ph.D.**

Distinguished Engineer

Sun Microsystems, Inc.

**Joint Statistical Meetings** **August 2009**

Photo by Dave Trindade

# Agenda

Introduction to Field Reliability Issue

Responding to Field Failures

Data from the Field

Reliability Analysis and Modeling of Field Failures

Analysis Implications

Physical Mechanisms/Remediation/Confirmation

Summary

# Introduction

## Field Failures in New Servers

- In 1999, Sun Microsystems began experiencing a number of field failures in new servers.
- The failures were sudden, unexpected, and could cause the system to “panic” (stop running, possibly reboot automatically).

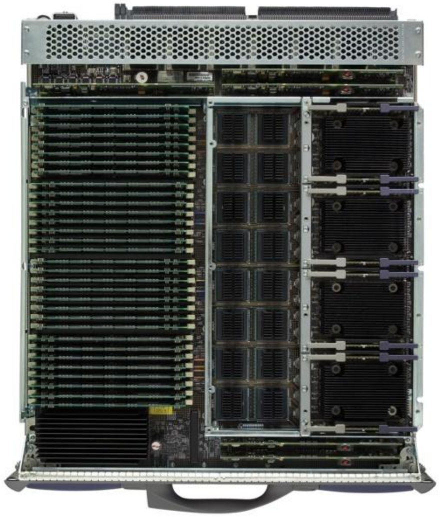
Engineers spent considerable efforts to restore systems to operation and prevent recurrence.

- Boards experiencing a failure were replaced with new boards and returned to Sun for analysis.
- Extensive data logging of conditions at the time of the failure were recorded for analysis.

Costs of field repairs escalated.

Customers demanded prompt resolution.

# Perspective on System Boards



Typical system board in a server

- Approximate size is 2’x2’ and weight ~30 lbs.
- Cost ~$100,000 per board
- Connects to chassis via sockets containing thousands of pins

Boards returned for failure analysis to factory

- Damage in transit was not uncommon.
- After analysis, over 95% of returned boards were classified as no trouble found (NTF). Remaining 5% were often determined to be damaged in transit.

# Actions to Identify Cause of Failures

Extensive stressing and testing of new and returned boards in systems

Physical failure analysis of returned boards

Observational visits to customer sites

Field environmental measurements

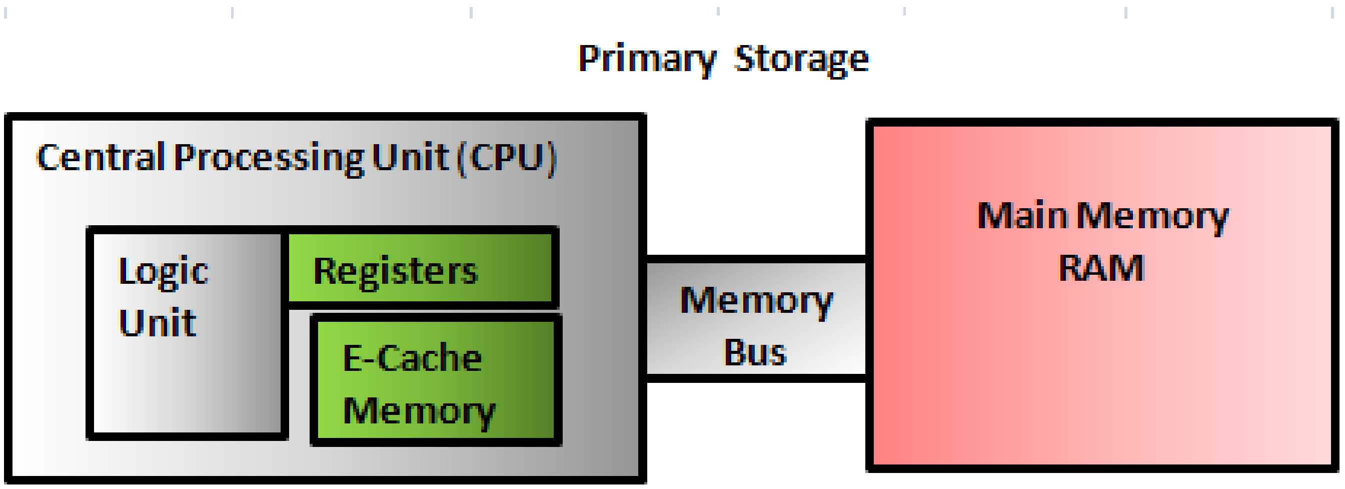
Analysis of system data logs (Explorer runs)

Consultation with suppliers

Frequent review and update meetings of teams of engineers and management

# Failure Mode: E-Cache Parity Errors

Months of work identified parity errors in e-cache (external, L2) SRAMS as problem location but determining exact cause was elusive.



*Source: Wikipedia: "Computer Data Storage"*

# Data Collection Team

A team was formed to collect data on field failures.

Data from major customers' datacenters were collected.

The importance of acquiring time dependent field data was emphasized.

# Field Data: Random Field Behavior?

Some customers experienced no failures.

Other customers saw high levels of failures for the **same** systems.

**Clues to Source?**

A customer in a concrete vault below ground level saw no failures.

Other customers in high altitude environments (observation stations) had more  frequent failures.

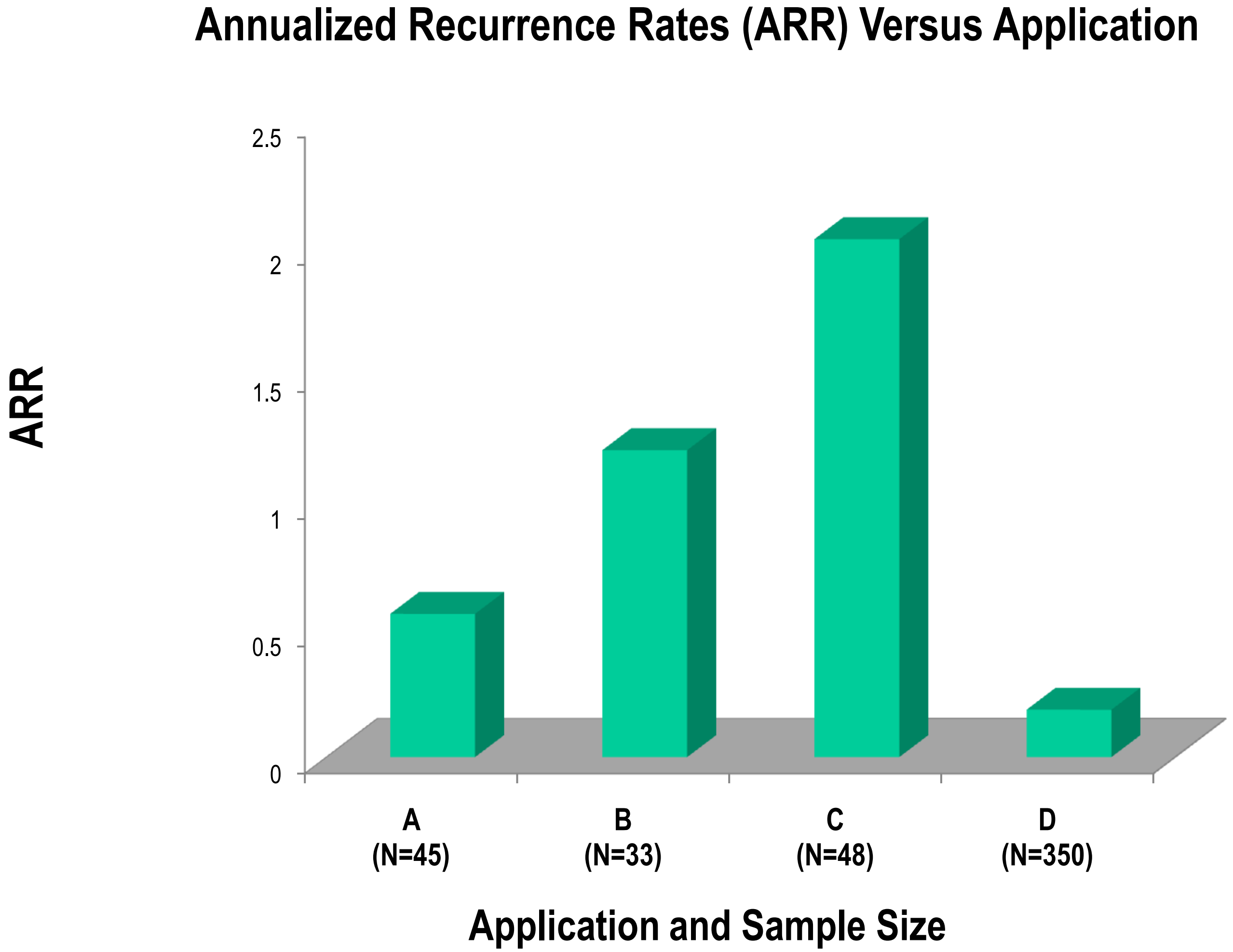
Was altitude or barometric pressure a factor?

# Datacenter Field Failures

In the **same** datacenter, customers running **different applications** on **identical** systems experienced widely different failure (recurrence) rates.

# Example of Application Dependence

## Single Datacenter, 476 Identical Systems

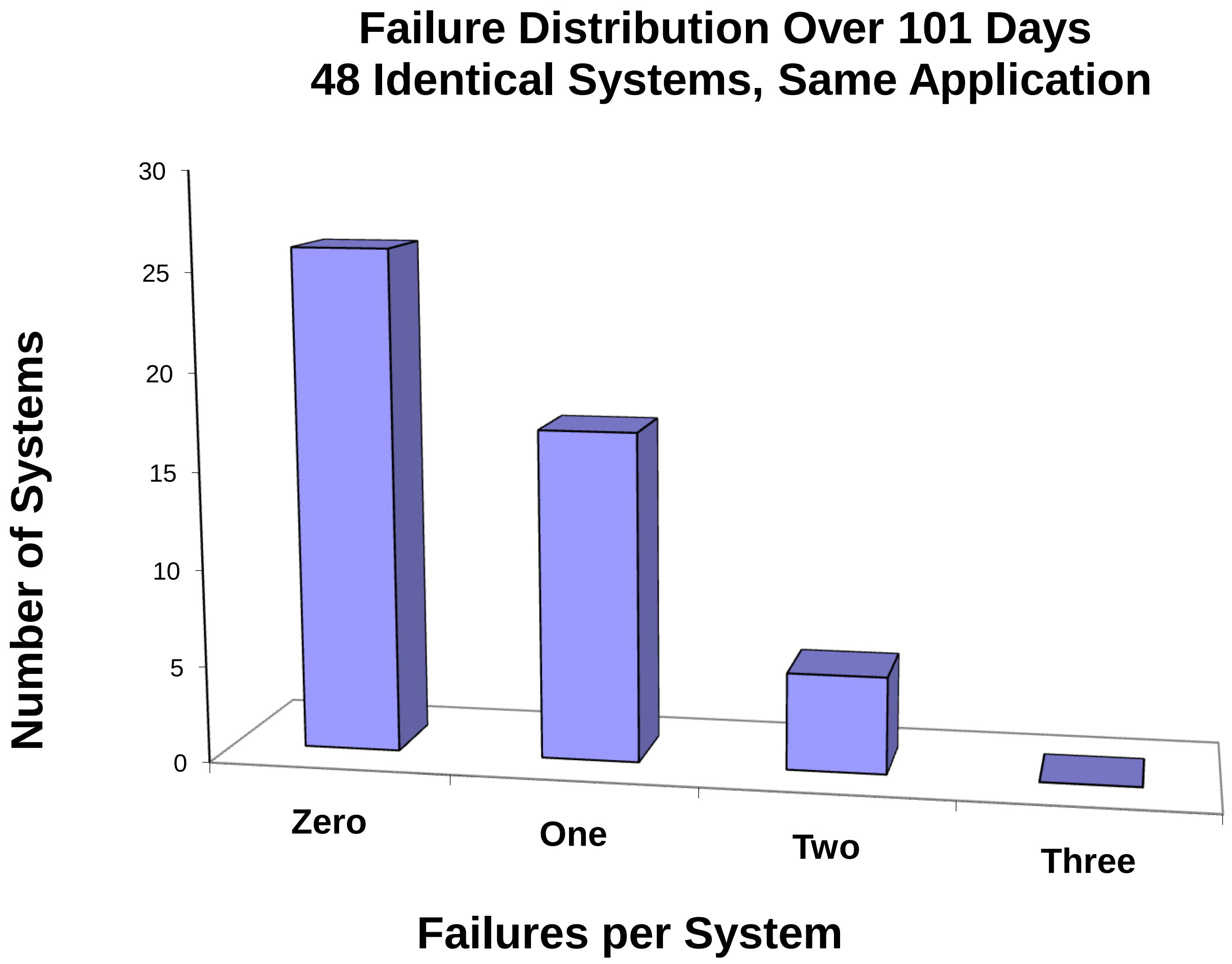


Nearly 11x difference in ARRs between applications C and D.

# Distribution of Failures Across Systems in a Datacenter

In the **same** datacenter, for **identical** systems running the **same applications** over the **same time period**, there could be systems with **no** failures, some with **single** failures, and some with **multiple** failures.

# Example of Failure Distribution



# Statistical Analysis and Modeling

Could statistical analysis and modeling of the data provide any **insights** into the cause?

How could the **application dependence** be explained?

Could the model agree with the field behavior and allow **prediction** of future failures?

Could we model the **distribution** of failures across systems in a datacenter?

# Do We Have a Renewal Process ?

**Critical question:**

For a renewal process, the times between failures are **independent and identically distributed (*i.i.d.*)** observations from a single population. Does such an assumption hold?

**Implication:**

A **renewal process** (*i.i.d.*), such as replacement of a failed component with one from same population, implies restoration of the system to "**like new**".

The assumption of a renewal process needs to be verified.

# Data Limitations

Unfortunately, age related data is typically not available for systems**.**

Field reliability data is often collected in a form that allows determination of a **mean time between failure**, ***MTBF***.

It is much easier to **count the numbers** of failures in a given time period (e.g., one month) for a group of systems operating during that time period than it is to obtain the system installation dates to **measure age** and the time dependent history of the failures.

Are there other ways to model the field behavior?

# Renewal Process: Single System

For a renewal process, the **single distribution** of **failure times between repairs** defines the expected pattern of repairs.

Let $X_i$ denote the **interarrival time** between the *i*th and the (*i*-1) repair.

Knowing the **probability distribution** (***pdf***) of $X_i$, we can theoretically find distributions for cumulative number of failures versus time, ***N*(*t*),** along with the average number of repairs versus system age, that is, the mean cumulative function, ***M*(*t*),** and the renewal or recurrence rate (ROCOF) ***m*(*t*)** = $dM(t)/dt$

# Poisson Model for Renewal Process

Suppose the interarrival times $X_i$ are *i.i.d.* with **exponential** probability density function (***pdf***) having **constant** failure rate intensity $\lambda$, that is,

$$f(x) = \lambda e^{-\lambda x}$$

Then, we can show that $N(t)$ has a **Poisson distribution** with **constant renewal rate** intensity $\lambda$. The **expected number** of repairs in time $t$ is $\lambda t$**.**

Note that $\lambda$ is a **rate** (i.e., repairs/time) that is multiplied by time ***t*** to give the **number** of repairs by time ***t***.

# Homogeneous Poisson Process Model (*HPP*)

Consequently, the probability of observing **exactly** *N*(*t*) = *k* failures in the **interval** (0,*t*) is given by the Poisson distribution

$$P\left[N(t)=k\right]=\frac{\left(\lambda t\right)^{k}e^{-\lambda t}}{k!}$$

We call this renewal process for which the interarrival times are exponentially distributed a ***homogeneous Poisson process*** (***HPP***).

**Multiple Systems:** By **multiplying** the calculated *HPP* Poisson distribution **probabilities** for a given failure rate by the **number** of systems, we can estimate the expected **distribution of failures** across many similar *HPP* systems.

# Case Study *HPP*

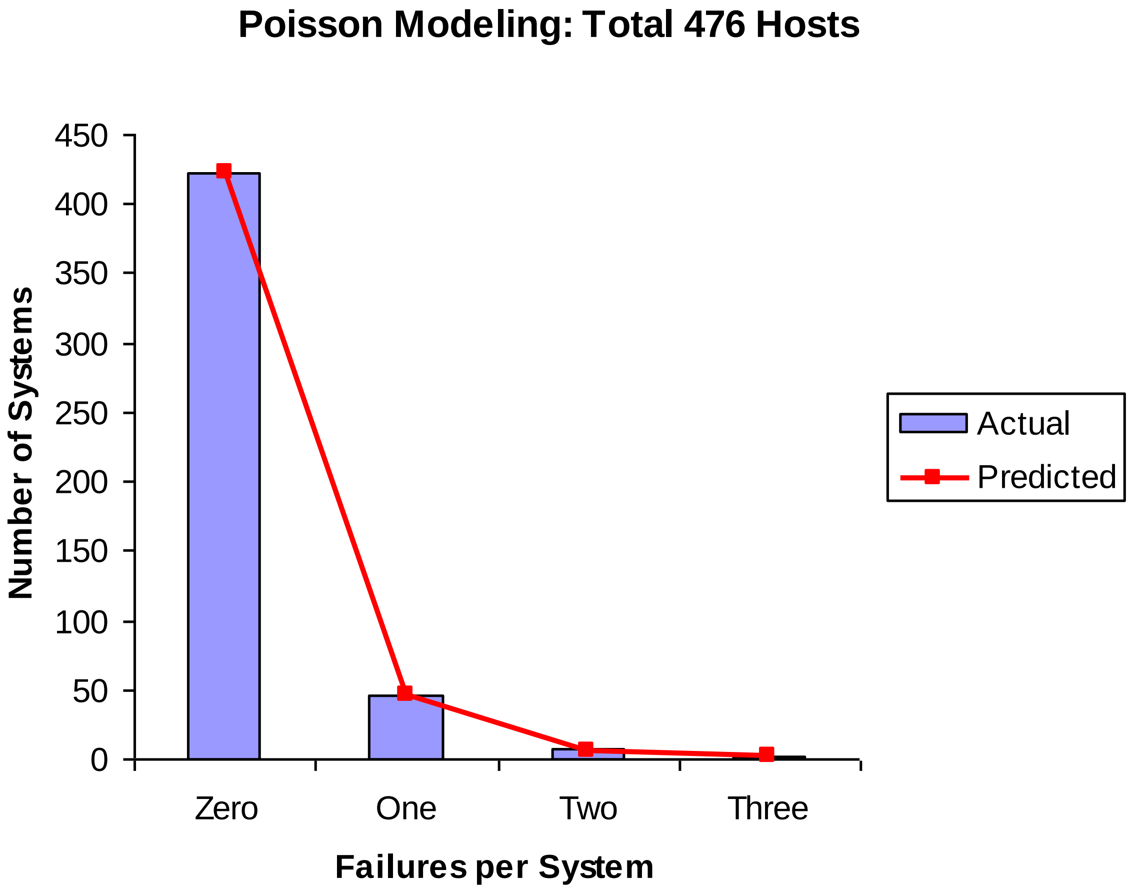
There were a total of 476 hosts in a large datacenter.

*For confidentiality, the specific customer, type of system (large), and applications are not identified.*

By determining an overall failure rate or *MTBF* over the previous few months, we checked for the suitability of an *HPP* model that could predict over the next 101 days how many of the 476 systems would have exactly no failures, one failure, two failures, and so on.  This prediction was then compared against actual failure counts across all systems.

# Model Confirmation

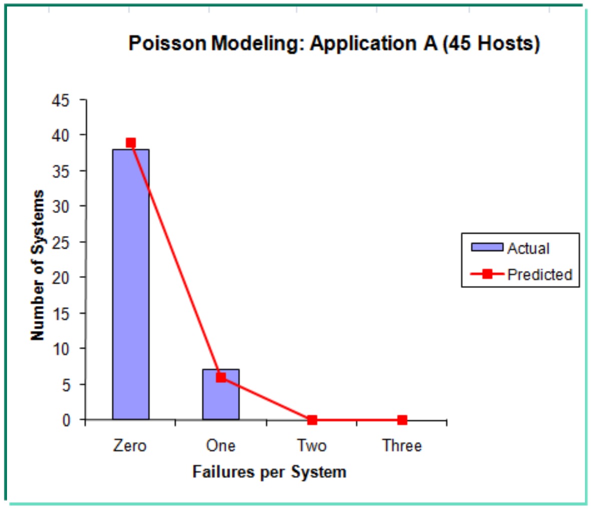
Comparison of Poisson Distribution Predictions Versus Actual Failures for a 101 Day Period

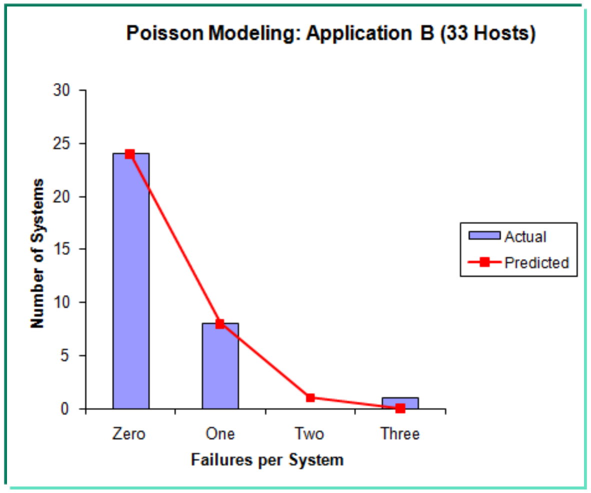


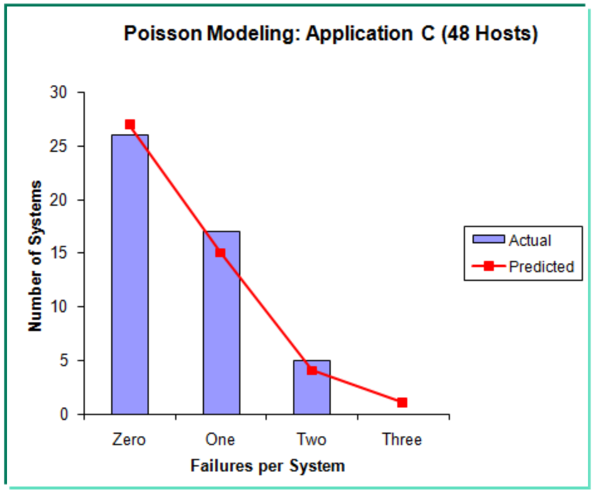
The model was in excellent agreement with observed results, confirming the *HPP*.

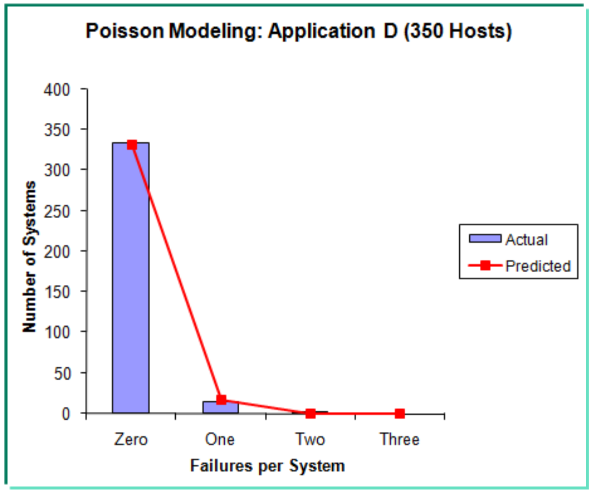
# Modeling Applications to HPP

When each application was checked against the HPP model, agreement again was excellent.


Poisson Modeling: Application A (45 Hosts)


Poisson Modeling: Application B (33 Hosts)


Poisson Modeling: Application C (48 Hosts)

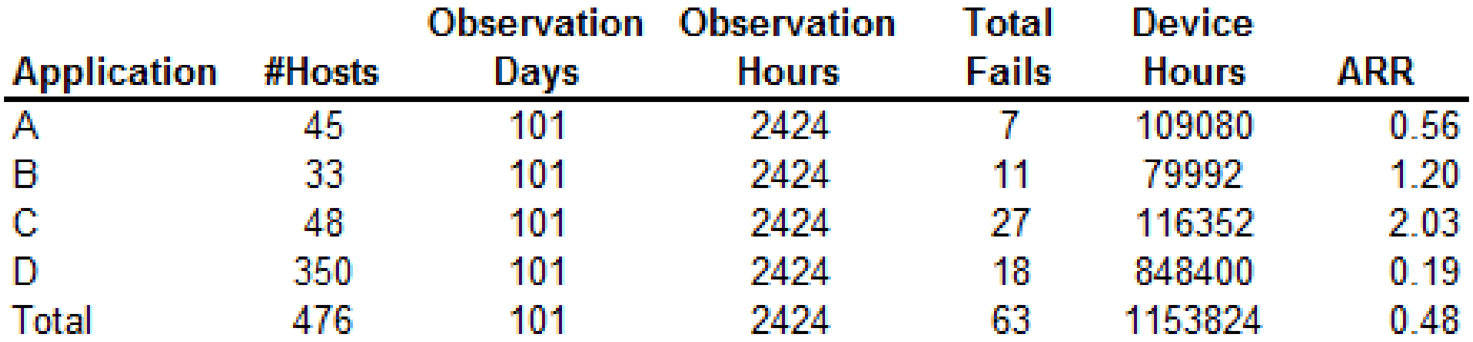

Poisson Modeling: Application D (350 Hosts)

# Failure Rate Estimates for Poisson Processes

Over a period of 101 days, there were a total of 63 failures among the 476 systems in the datacenter.  The overall annualized recurrence rate (***ARR***) is estimated as

$$\lambda = \frac{63}{476}\left(\frac{365}{101}\right) = 0.48 \text{ per system}$$

Similarly, we can estimate the  *ARR* separately for each application.



| Application | #Hosts | Observation Days | Observation Hours | Total Fails | Device Hours | ARR |
|---|---|---|---|---|---|---|
| A | 45 | 101 | 2424 | 7 | 109080 | 0.56 |
| B | 33 | 101 | 2424 | 11 | 79992 | 1.20 |
| C | 48 | 101 | 2424 | 27 | 116352 | 2.03 |
| D | 350 | 101 | 2424 | 18 | 848400 | 0.19 |
| Total | 476 | 101 | 2424 | 63 | 1153824 | 0.48 |

# Superposition *ARR* Estimate

The overall datacenter ARR arises from a **superposition** of four application dependent Poisson processes with intensities $\lambda_i$ , $i = 1,2,3,4$.

We can estimate the overall *ARR* by using a weighted formula (weights based on the number of systems – called hosts - running each application):

$$\lambda = \frac{\sum_i \lambda_i N_i}{\sum_i N_i} = \frac{0.56 \times 45 + 1.20 \times 33 + 2.03 \times 48 + 0.051 \times 350}{476} = 0.48$$

This result matches the previous estimate for the overall *ARR* for the 476 servers.

# Consequences and Implications

Since the results were consistent with a *HPP*, the implication was that the failure behavior for any system in the datacenter derived from a **renewal process** with a **constant, application dependent failure rate**.

Constant failure rates result from a **constant source**.

There was **no physical damage** to the SRAM by the cause. The “good as new” assumption for a renewal process seemed valid. Called “**soft errors**”.

Failure rates were also determined to vary with altitude.

This confirmed that only plausible source was **radiation from cosmic rays** causing single bit parity errors in the e-cache memory. Without error detection and correction, failures could occur and panic the systems.

# Physical Mechanisms for Soft Errors

- The radiation environment
  - **Alpha** particles
  - **High energy** cosmic rays
  - **Low energy** cosmic rays and $^{10}$B fission in boron-doped phosphosilicate glass (BPSG) dielectric layers of ICs
- Factors impacting soft error rates (SER)
  - Complexity Density
  - Lower voltage Higher speeds Lower cell capacitance
- The susceptibility to soft error rates for DRAM and SRAM has increased with **reduced dimensions** (higher densities) and **lowered operating voltages** of advancing technology.

# Failure Description and Remediation

The server **writes** to e-cache memory. Memory in e-cache can be saved eventually to permanent memory. If a cosmic ray causes a uncorrected parity error to occur in e-cache and an attempt is made to **read** data in e-cache or to write it to main memory, the parity error will be detected and the system will panic to prevent data corruption.

An effective **solution** was to incorporate **mirroring**, where every byte is duplicated and stored in two locations in *SRAM* along with a parity checker built into the *SRAM*.

(Note: The equally effective alternative of replacing parity protection with single-error correction, double-error detection error correction code, "***SECDED ECC***", was rejected as it would have required a change to the processor's pipeline.)

# Application Dependence Explained

If an application **writes** often to memory but **reads infrequently**, an e-cache error can be overwritten before a read cycle sees the error. Imagine an application updating minutes used by a cell phone user. Consequently, the failure rates will be low.

If an application **reads frequently**, then e-cache errors will be detected quickly and cause failures. The failure rates will be high.

# Best Practices For Systems Prior to Installation of Mirrored SRAMs

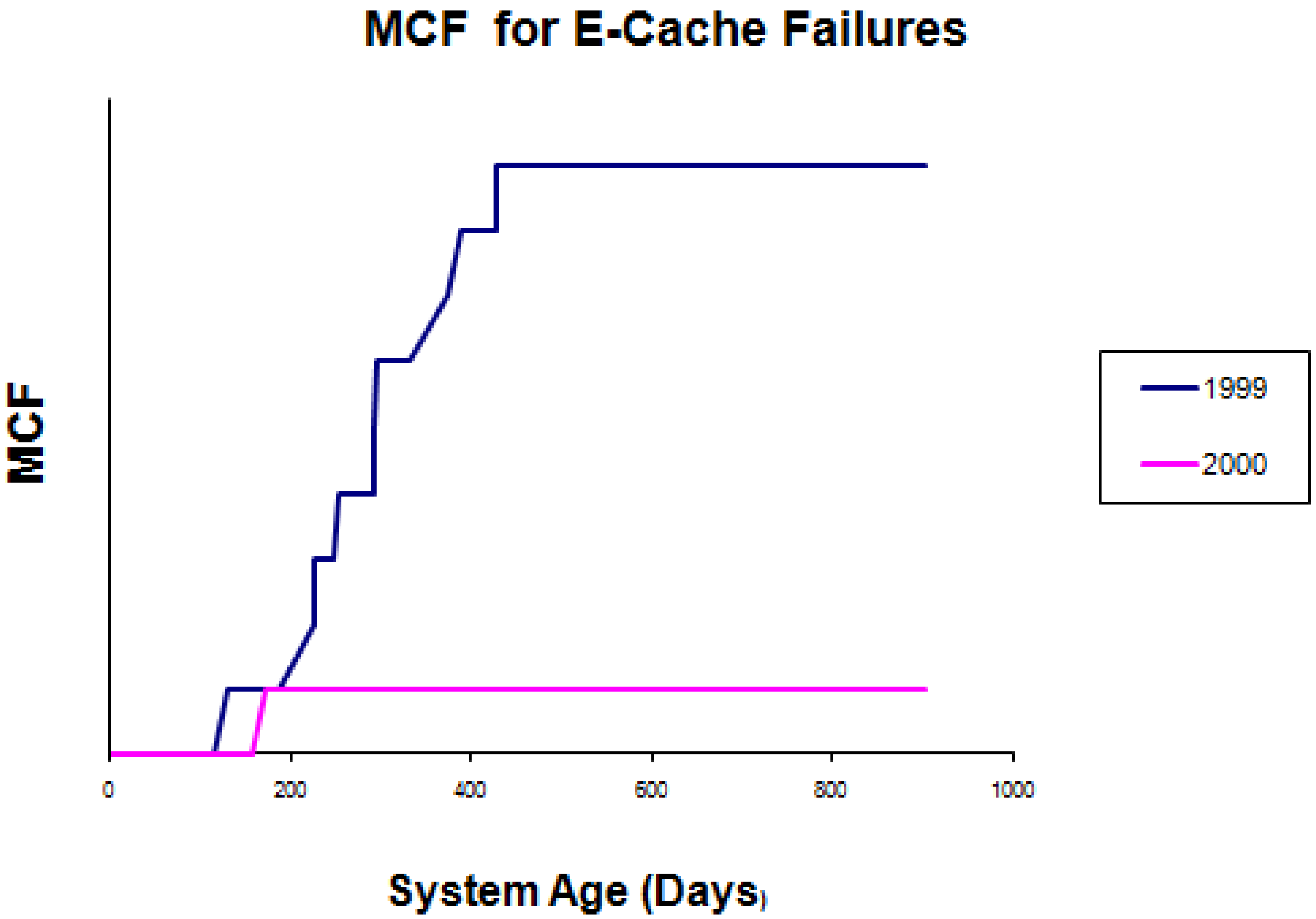
Instead of removing a failed board, the simplest action was simply to **reboot the system**.  No physical **damage** had occurred and the probability of a hit by a cosmic ray was **purely random**.

In addition, the costs of replacing boards and subsequent damage to the boards or systems (e.g., bent pins) could be avoided.

Spreadsheets were sent to the field for the service engineers to do the model fitting for any customer and illustrate the model consistency.

# Confirmation

Introducing mirrored *SRAMs* into systems stopped the failures.



# Summary

Field failures represent significant inconvenience to customers.

Field failures remediation efforts are costly to system manufacturers.

Complex systems make identification of causes difficult and challenging.

Statistical analysis and modeling can provide valuable insights into causes.

Undetected and uncorrected soft errors are a significant factor in system reliability, but there are approaches to alleviate the problem.

# Where to Get More Information

Google “soft error reliability” for a wealth of information on the topic.

Search Wikipedia under “soft error”, “CPU cache”, “cosmic rays”.

*SER-History, Trends, and Challenges* by J. Ziegler and H. Puchner, Cypress Semiconductor Corporation (2004)

“Radiation-Induced Soft Errors in Advanced Semiconductor Technologies”, R. Baumann, *IEEE Trans. On Device and Materials Reliability*, Vol. 5, No. 3, September 2005

For statistical analysis and modeling of reliability data see *Applied Reliability*, 2nd ed. by P. Tobias and D. Trindade, Chapman & Hall/CRC (1995)

Additional references on modeling and data analysis at www.trindade.com/publications.html

# Author’s Contact Information

Distinguished Principal Engineer
Sun Microsystems, Inc.
San Jose, CA
Email: david.trindade@sun.com
Work: 408-404-8989

For biography, see www.trindade.com/biography.html