Simple Plots for Monitoring the Field Reliability of Repairable Systems

David Trindade, Ph.D., Sun Microsystems Inc.
Swami Nathan, Ph.D., Sun Microsystems Inc.

Key Words: Field Reliability analysis, Repairable Systems, Computer Systems, Mean Cumulative Function, Failure Rate

SUMMARY & CONCLUSIONS

Repairable systems such as computers or servers consist of processing hardware, operating systems (OS) and application software, storage, and a variety of third party subsystems. Reliability practitioners often attempt to model such systems with sophisticated statistical methods such as non-homogeneous Poisson processes (NHPP) and maximum likelihood parameter estimation. However, management as well as support engineers who maintain the system are easily intimidated by such techniques. Monitoring the reliability of complex systems does not necessarily require intricate models. This paper illustrates a few simple plots that aid in tracking the field reliability of a group of systems. Reliability is ascertained as a function of time without resorting to complex stochastic techniques and while maintaining statistical rigor. Analyses based on the mean cumulative function (MCF) are simple and easily understandable by decision makers. Called time dependent reliability (TDR) at Sun Microsystems, such methods have been successfully used to estimate, monitor, and improve the reliability of repairable systems such as workstations, servers, and storage arrays.

1. *INTRODUCTION*

Repairable systems such as those in datacenters are inherently hardware and software entities whose reliability is influenced by various characteristics that are unique to the local deployment, operation and environment. Included are computer servers, networking equipment like routers, storage devices like arrays, storage area networking equipment and a variety of software services. Systems are often installed in a staggered fashion and are removed or upgraded periodically. Hence, the reliability analysis is complicated by the highly multicensored (left and right) nature of the data. The analysis is made more difficult because of the effects of constant changes in the software and firmware from patches and upgrades to newer versions, training, environmental issues, changes in operators and systems administrators, and so on.

Systems and equipment manufacturers commonly report reliability in terms of summary statistics, such as MTBF. However, there are several critical assumptions involved in stating an MTBF. The primary consideration is that the repairable systems have recurrence events that follow a single distribution for the times between events; that is, the failures follow a renewal process. A further assumption is that the times between events are independent and exponentially distributed with a constant rate of occurrence, resulting in a homogeneous Poisson process (HPP). Consequently, strict reliance on the MTBF without full understanding of the implications can result in missing developing trends and drawing erroneous conclusions [3,7].

The notion of a Mean Cumulative Functions (MCF) [1,2,3,4] is used to present effective field data analysis methodologies. These extremely useful concepts have been in existence for nearly two decades, but the literature remains highly theoretical for the average practitioner, and reported applications have been limited [5,6]. Though the discussion is presented here in the context of datacenter systems, the techniques covered are completely general in their applicability.

2. *CUMULATIVE PLOTS & MEAN CUMULATIVE FUNCTION*

A cumulative plot is a plot of the number of failures versus time for a single system. Time can be power on hours, days since installation, calendar time, or even number of cycles. Cumulative plots reveal trends in the occurrence of failures or repair events.

Figures 1, 2 and 3 show three different cumulative plots.

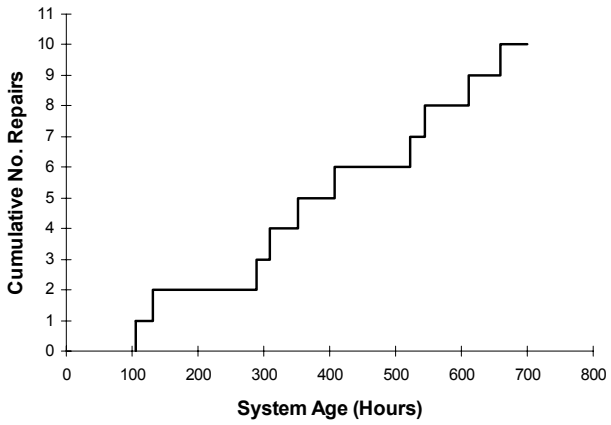

Figure 1: Trendless system: Near linear growth in failures vs. time

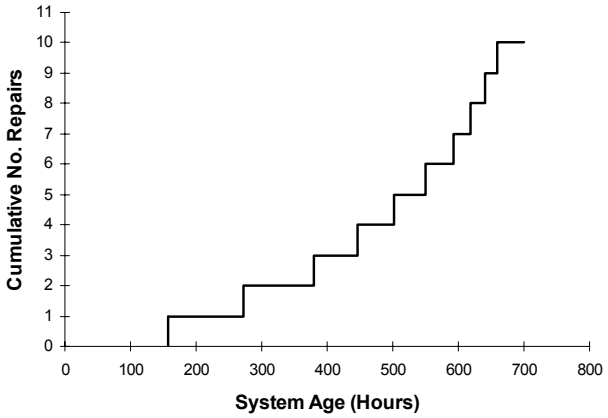

Figure 2: Worsening system: Curves upwards. Step rate increases.

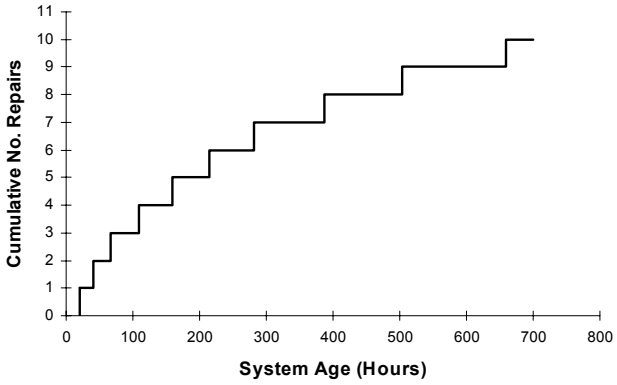

Figure 3: Improving system: starts flattening out. Step rate decreases.

In all three plots, the system has had 10 failures in 700 hours, resulting in an MTBF of 70 hours. Even though they have the same MTBFs, there is clearly a difference in the behavior of the three systems. Cumulative plots are far more revealing and yet simple.

Mean Cumulative Function (MCF) is the average of the cumulative plots across all the individual systems at risk at any point in time.

Figure 4. shows the cumulative plots of a population of 4 machines in a commercial datacenter. We take a vertical slice at a particular age, e.g., at 100 days. Note machine A has had 3 fails, machine B has had 2 fails and machines C and D have had one fail each. We can thus compute the average number of failures for this population at 100 days. We can take these vertical slices at any point in time and construct another plot of the average number of failures versus age. The average number of failures against time is called the mean cumulative function or MCF. Figure 6 shows the MCF for the above population where the steps have been replaced with connecting lines.

The Mean Cumulative Function is simply the average number of failures that can be expected at various ages of a system in the population. It enjoys all properties of a cumulative plot such as displaying the average trends of the population of systems in the datacenter. For example, if a new system is installed in this datacenter, we can expect about 1.75 repair actions (on the average) in the first 100 days of operation.

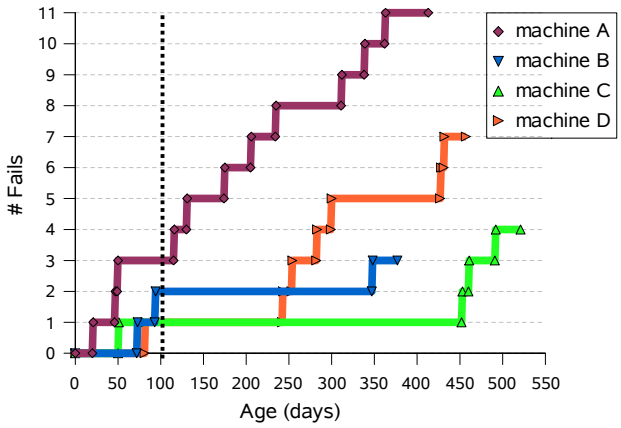

Figure 4: Cumulative plots of a population of 4 machines

2.1 Incorporating Multicensoring

Machines are installed at different points in time, and so at a given date, different machines have different ages, i.e., there is strong right censoring. For example, if machine A is installed on 1/1/03 and machine B is installed on 3/1/03, then on 1/1/04 machine A is 365 days old and machine B is 301 days old. So after 301 days machine B stops contributing information to the MCF, and machine A stops contributing after 365 days. This is an example of right censoring.

There are also cases where the failure data is available only after a certain date. For example, a company takes over a service contract on a given date, and there is no failure history prior to that date. This induces left censoring because information is not available before a certain age. In the above example, if data collection started on 4/1/03, in the absence of any failure history between 1/1/03 to 4/1/03, then machine A contributes no information in the first 90 days of operation and machine B contributes no information in the first 31 days of operation. This is an example of left censoring. The MCF plot handles both right and left censoring by accounting for the number of machines at risk at a particular age when failures occur.

Figure 5 [3] shows the life of three machines with failure and censoring times. Table 1 illustrates the calculation of the MCF incorporating censoring. When the first failure happens at 33 hours, we have one failure out of three machines and so the MCF is 1/3. At 135 hours we have another failure out of three machines, and so the fails/machine is 1/3 and the MCF is 2/3 (cumulative fails/machine). At 300 hours, System 3 is censored and so the number of systems at risk changes to two machines. The fails/machine at the next two failure points becomes ½. At 500 hours, System 2 gets censored and the number of systems at risk becomes one. The approach for incorporating left censoring is quite similar, except that the number of systems at risk will increase when

the left censoring time is over. It is evident that this procedure can be very easily implemented in a spreadsheet such as Staroffice© or Excel©

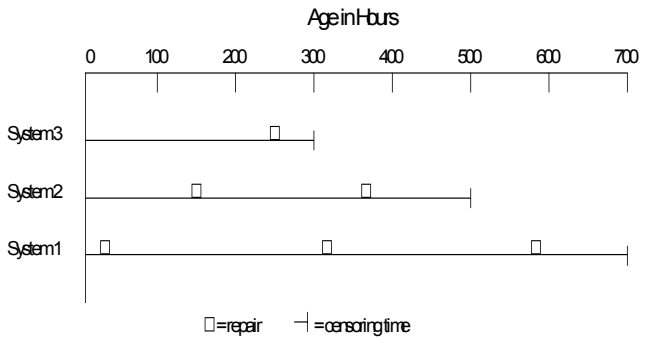

Figure 5 : Example MCF calculation with right censoring

Time (Hrs)	*Number of Systems at Risk*	*Fails/machine*	*(MCF)*
33	3	1/3	1/3
135	3	1/3	2/3
247	3	1/3	3/3
300-|	3		3/3
318	2	1 / 2	3/3+1/2
368	2	1 / 2	3/3+2/2
500-|	2		3/3+2/2
582	1	1/1	3/3+2/2+1/1
700 -|	1		3/3+2/2+1/1

Table 1 : Step by Step calculation of MCF vs. Age

2.2 Identifying Anomalous Machines

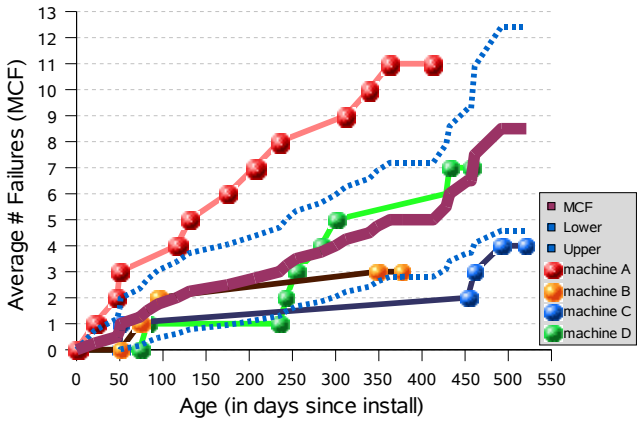

Figure 6 :Mean Cumulative Function with confidence intervals.

One can calculate confidence intervals [2] on the MCF and plot them alongside the MCF. If the cumulative plot of a system goes well beyond the upper confidence level, we can, for engineering purposes, say that the machine is experiencing a significantly higher number of failures than the population at large. This simple technique to identify anomalous behavior has proved invaluable to support engineers in the field by helping them focus attention on anomalous machines rather than taking population wide upgrade actions.

Figure 6 shows the MCF and 95% confidence intervals for the 4 machines shown in Figure 4. It can be clearly seen that machine A has had a significantly higher number of failures. Even though machine D has had the next highest number of failures, it is still operating within bounds for this population. Support engineers can look at this plot and immediately focus all attention on remediating machine A.

The confidence intervals are to be understood more as an envelope of pointwise confidence intervals on the mean rather than a prediction or tolerance interval on the MCF. Thus, if a machine falls within the bounds, it is definitely not an anomalous machine. However, if it falls above the upper confidence bound then visual interpretation and heuristics are used to determine if the machine is experiencing a higher number of failures than the population. This heuristic approach has worked quite well in most practical situations (especially in smaller sample sizes) without resorting to exact computation of prediction intervals and tests of outliers.

3. *CALENDAR TIME ANALYSIS*

Computer systems in commercial datacenters undergo several changes during their operation. Software patches are added to fix bugs, upgrades to newer versions of software, memory, addition of disks, cables etc. These are effects that are not age dependent effects in the “bathtub curve sense” but are a result of “external events” affecting the configuration and operation of the machines. Since machines are installed at various points in time, at a given calendar date, they all have different system ages. Hence, the impact of any calendar time effect may not be as evident because of the averaging across systems in the MCF vs. age plot.

To detect calendar time effects, we can plot the cumulative adjusted fails/machine in a manner quite similar to the MCF vs. age plot (see figures 7a,7b). Essentially we start from the date of install of the first machine and calculate the fails/machine at all dates until the current date and plot the cumulative average of the fails/machine against date, which we'll refer to as the calendar time function (CTF).

When the slope of the MCF (recurrence rate) is plotted for two systems in Figure 7a, we can see two distinct spikes. However, when we plot the CTF recurrence rate as a function of calendar time in Fig 7b, we can see that the spikes align over a common date, indicating that the phenomenon is not age related but calendar time related. The procedure for calculating recurrence rates is discussed in the next section.

The CTF plot of the average fails/machine does not have the notion of confidence intervals for the following reason. A confidence bound reflects statistical uncertainty. A new machine will go through 50 days and 100 days of operation, and hence, we can say that this new machine will

experience about 1.75 fails by the time it reaches 100 days of operation. However a new machine will never see January 2003, and thus, the notion of confidence bounds is meaningless in calendar time. With the CTF, we are looking for common reliability effects across systems associated with specific dates.

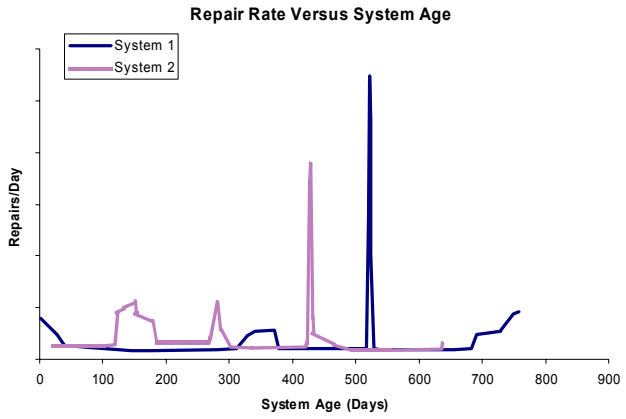

Figure 7a: Recurrence Rate vs. System Age

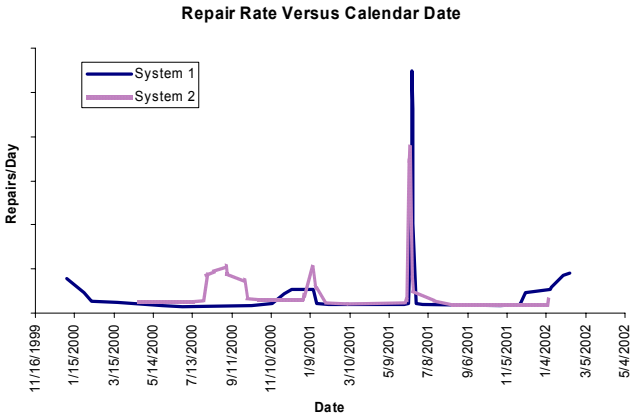

Figure 7b: Recurrence Rate vs. Calendar Time

4. *RECURRENCE RATE*

The recurrence rate (RR) is the slope of the MCF curve. This RR plot magnifies the trends in the MCF or CTF curves and identifies portions of age or calendar time where the rate of occurrence of failures is increasing or decreasing. The recurrence rate against calendar time has been exceptionally useful in showing trends in the rate of failures at customer datacenters following software patch installations, certain hardware upgrades, changes in datacenter environments, or changes in operating personnel.

The recurrence rate is estimated by numerical differentiation [8] of the cumulative average number of failures vs. age or calendar time. The degree of smoothness of the curve is controlled by the number of points used in calculating the tangent at each point in the MCF or CTF. Spreadsheets have a slope function which can be used to calculate the recurrence rate. For a 5 point slope, we can calculate the slope of the first five (x,y) pairs and plot the slope value against the 3rd point. Then calculate the slope of data points 2 through 6 and plot it against the 4th point and so on. It should be noted that if multiple failures occur on the same age, then they should be combined for purposes of slope determination. The slope is essentially change in Y over change in X, and so the different X values (or system ages) in the slope calculation should be unique to avoid division by zero problems.

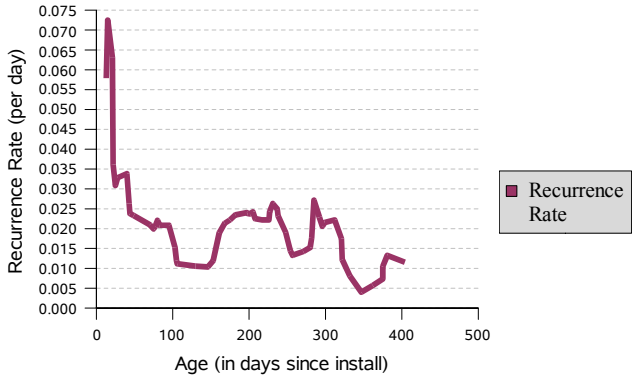

Figure 8 : Example of recurrence rate vs. age

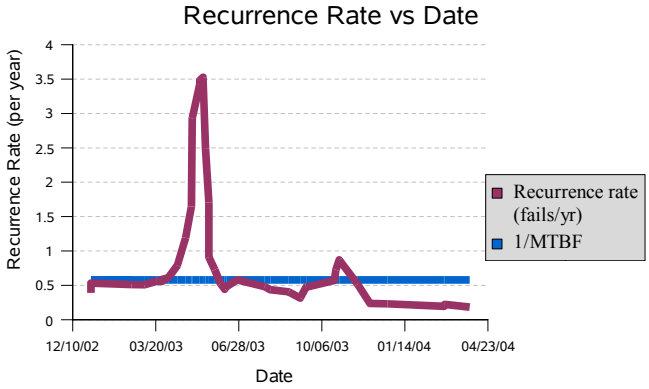

Figure 9 : Example of Recurrence Rate vs. Date

Figure 8 has an example of a recurrence rate vs. system age. We can see that the recurrence rate is quite high in the first 50 days of operation, and then the rate of failures improves with age, becoming fairly trendless. This plot immediately provides clues as to potential failure modes such as early life problems in the case of hardware or "learning curve" issues with system administrators or operating personnel. Learning curve issues are fairly common in datacenter systems due to the constant evolution in software and hardware.

"Spikes" in the recurrence rate plot can occur in many cases if there is a 'clustering' of failures i.e., multiple failures in a short period of time. This usually points to poor diagnostics, imperfect repair or spares that are 'dead on arrival'.

Figure 9 shows an example of recurrence rate vs. calendar time. One can see that the rate of occurrence of failures increased from April-Jun 03 and then fell. The rate remained fairly stable for about 6 months and decreased by about 50% in December. It has remained at this constant low rate of failure in 2004. This plot provides extremely valuable information to the customer who owns the datacenter as well as the support engineers who maintain the datacenter. The spike in May was found to be a result of one misbehaving machine experiencing multiple failures in a rather short window of time.

MTBF Considerations

When the recurrence rate is flat or nearly flat, the rate of occurrence of failures is fairly constant, and one can apply the conventional notion of MTBF. Under all other situations, the rate of failures depends on time (calendar and/or system age), and the traditional notion of single MTBF to describe all periods has very limited applicability.

5. *FAILURE CAUSE PLOTS*

One of the standard plots used to depict failure causes is the Pareto chart. A Pareto chart can depict the number of occurrences of various root causes. Figure 10 is an example of such a chart. One can see the top 5 causes of failures of a population of servers at a particular datacenter over a two year period. The actual causes have been masked for legal reasons. It is evident that the Pareto chart is static in time and does not address which causes have been remediated and which causes are trending badly in recent times. Failure cause plots as a function of age or calendar time are far more revealing because they are dynamic, showing time evolution of causes.

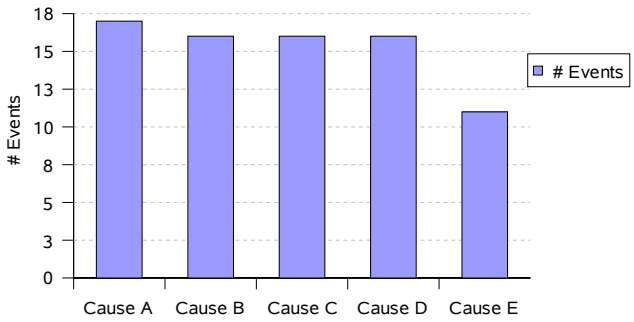

Figure 10 : Example Pareto Chart showing failure cause frequencies

Figure 11 shows the calendar time plot for the Pareto chart in Figure 10. The difference is quite evident, with each failure event mapped in time by cause.

Although the Pareto chart shows Cause A as the top contributor, the time dependent plot clearly reveals that the issue cropped up only in September and has been on a strong increasing trend. The cause was diagnosed to a collection of interoperability issues related to 3rd party software. There was a change in the software version in early September that immediately resulted in a host of failures. This issue was consequently promptly addressed. In the Pareto Chart, causes B,C and D have equal contributions. However, in the calendar time plot, one can see clearly different trends in the occurrence of these events. Cause E on the other hand did not occur for more than 7 months in this population, and then a rash of events can be seen.

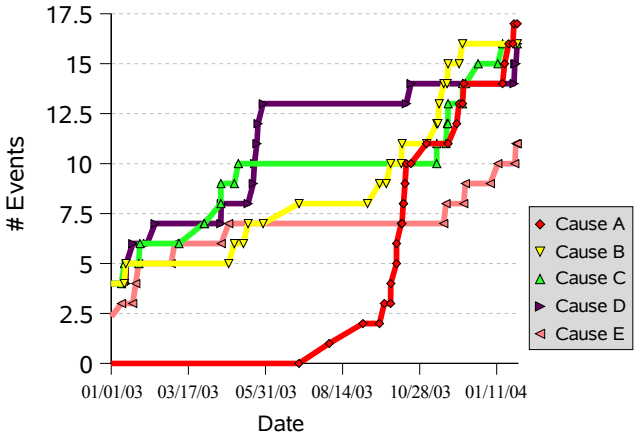

Figure 11: Calendar Time Plot

These plots can be created both against calendar time as well as system age to detect either type of trends. Note that this plot is a raw count of events and is not normalized by the number of machines like the MCF. For greater understanding, the failure cause plots can be plotted akin to MCFs by normalizing by the number of machines. Thus one can show the MCF at the system level and show the failure cause MCFs that add up to the system MCF in one plot.

The keen observer can also note that one can do these failure cause plots for the anomalous machines (as identified by the technique in the previous section) to zoom into the reasons for failure.

6. *ADDITIONAL USES OF MCFs*

One of the most useful capabilities of plotting MCF versus system age is that it can be used for comparing different populations of machines. It provides a way to correctly compare the cumulative fails/machine at various ages across different subsets of machines. The subsets could be:

1. Machines by customer datacenter
2. Machines belonging to the same customer but located at different datacenters
3. Machines of different vintages i.e., machines manufactured in 2002 vs. 2003.
4. Machines performing different functions i.e., production machines vs. development machines.

The versatility of this approach has been strengthened by recent works on incorporating missing data which is a common problem in field data analysis [9]

7. *FITTING MCFs with NHPP*

Although we mentioned earlier that NHPP model fitting involves sophisticated procedures, under some circumstances, one might require a parametric model for prediction and extrapolation purposes. In those situations, one can use a power law model or other NHPP models to fit an appropriate function through the non parametric MCF. This enables one to then conduct parameter estimation and provide estimates of number of failures over a time window with confidence intervals.

Figure 12 shows an example of fitting a power law model through an MCF. The function used was MCF = a $(AGE)^b$, where a and b are constants determined from model fitting. Further information can be found in [3,4].

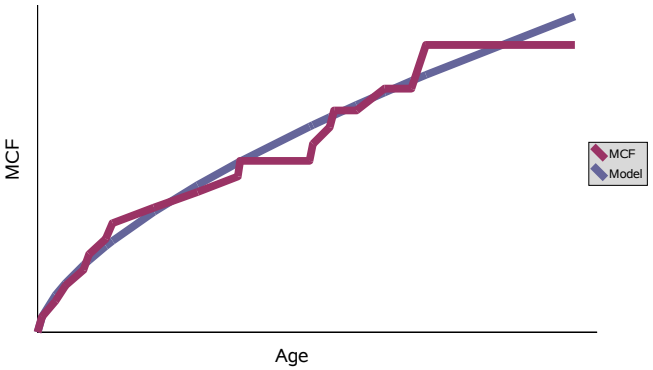

Figure 12: Fitting a Power Law Model through an MCF

CONCLUSIONS

The analysis of repairable systems does not have to be complicated. Measuring and monitoring repairable system reliability can be greatly facilitated by the use of simple but very powerful graphical techniques presented in this paper. These procedures have been applied very effectively at Sun Microsystems for monitoring the reliability of Sun equipment at many customers, These methods have allowed Sun engineers to quickly identify trends, anomalous systems, unusual behavior, the effects of hardware and software changes, maintenance practices, installation actions, and so on. Customers presented with TDR analysis reports utilizing the procedures discussed in this paper have responded very favorably to these revealing charts.

REFERENCES

1. Nelson W., "Graphical Analysis of System Repair Data", *Journal of Quality Technology*, **17**, 140-146.
2. Nelson, W., *Recurrence Events Data Analysis for Product Repairs, Disease Recurrences and Other Applications*, ASA-SIAM series on Statistics and Applied Probability, 2003.
3. Tobias, P.A., Trindade, D.C., *Applied Reliability*, Chapman & Hall/CRC, 1995.
4. Meeker, W.Q., Escobar, L.A., *Statistical Methods for Reliability Data*, Wiley Interscience, 1998.
5. Lawson, J.S., Wesselmann, C.W., Scott, D.T.,"Simple Plots Improve Software Reliability Prediction Models", *Quality Engineering*, Vol 15. No. 3. pp411-417, 2003.
6. Usher, J.S., "Case Study: Reliability Models and Misconceptions", Quality Engineering, Vol. 6, No. 2, pp 261-271.
7. Elerath, Jon G., "Specifying Reliability in the Disk Drive Industry : No More MTBFs", *Annual Reliability and Maintainability Symposium*, 2000, Los Angeles, CA, U.S.A.
8. Trindade, D.C., "An APL Program to Numerically Differentiate Data", IBM TR Report 19.0361, January 12, 1975
9. Rutledge, R., "An Approach to include missing data in MCFs", Sun Microsystems Internal White Paper, 2003.

BIOGRAPHIES

David Trindade, Ph.D.
Sun Microsystems, Inc.
6005 Assisi Court
San Jose, CA 95138

e-mail: david.trindade@sun.com

Dr. David Trindade is a Distinguished Engineer at Sun Microsystems. Formerly he was a Senior Fellow at AMD. His fields of expertise include reliability, statistical analysis, and modeling of components, systems, and software, applied statistics, especially design of experiments (DOE), and statistical process control (SPC). He is co-author (with Dr. Paul Tobias) of the book *Applied Reliability*, 2nd ed., published in 1995. He has authored many papers and presented at many international conferences. He has a BS in Physics, an MS in Statistics, an MS in Material Sciences and Semiconductor Physics, and a Ph.D. in Mechanical Engineering and Statistics. He has been an adjunct lecturer at the University of Vermont and Santa Clara University.

Swami Nathan, Ph.D.
Sun Microsystems Inc.
Mailstop USCA14-204
Santa Clara, CA-95054

e-mail: swami.nathan@sun.com

Dr. Swami Nathan is a staff engineer in Sun Microsystems. His field of interest is on field data analysis, statistical analysis and reliability/availability modeling of complex systems. He received his B.Tech from Indian Institute of Technology, and M.S. and Ph.D. in reliability engineering from the University of Maryland, College Park. He has authored over a dozen papers in peer reviewed journals and international conferences and holds 2 patents.