Reliability Estimation and Analysis of Linux Kernel

[1]Sanjeev Kumar Jha, [2]Dr. A.K.D.Dwivedi, [3]Dr. Amod Tiwari
[1,2]DOEACC Society Ministry of C&IT, Govt. of India
[3]Associate Professor, PSIT Kanpur, India

Abstract:
Latest stable version of Linux kernel used by all Linux based operating system is 2.6 series. The purpose of this study is to develop an optimized reliability model suitable for this version of the kernel and analyze its reliability by fitting this model.
By using goodness of fit test best distributions (first two) on the basis of their rankings are selected. Collected failure data are modeled by these two distributions and parameters are estimated by using the method of maximum likelihood functions, which is considered as best method of estimation. Data are extracted in raw format from different sources than after appropriate preprocessing and using SQL queries it is stored in MYSQL Database under Linux Operating System. On the basis of both selected distributions reliability models are constructed and appropriate formula is derived for estimating reliability by using constructed models. Ultimately reliability estimates using derived models are compared and result is shown.

Keywords
Goodness of Fit, Kernel, Likelihood Function, Linux, Operating System, Open source software (OSS), Software reliability model.

I. Introduction

Kernel is heart of any operating system with its different modules responsible for monitoring and management of the entire computer system. The Linux kernel is an example of operating system kernel used by the Linux family of Unix-like operating systems. It is one of the widely used examples of free and open source software. Due to its robust nature, security and economical feasibility it is used all over the world as an operating system by corporate as well as government organizations.

The Linux kernel [1] is released under the GNU [2] General Public License and is developed by contributors worldwide.

Torvalds developed Linux when he was student of Helsinki University in 1991. There are different versions of Linux kernel. Version 1.0 was released on 14 March 1994. This release of the Linux kernel only supported single-processor i386-based computer systems. Portability became a concern, and so version 1.2 (released on 7 March 1995) gained support for computer systems using processors based on the Alpha, SPARC, and MIPS architectures. Version 2.6.0 was released on 18 December 2003(5,929,913 lines of code). The 2.6 series of kernel is still the active series of stable kernels as of December 2010. The development for 2.6.x changed further towards including new features throughout the duration of the series. Among the changes that have been made in the 2.6 series are: integration of µClinux(Micro Controller Linux) into the mainline kernel source, PAE (Physical Address Extension) support, support for several new lines of CPUs, integration of ALSA (Advanced Linux Sound Architecture)[3] into the mainline kernel sources, support for up to 232 users (up from 216), support for up to 229 process IDs (up from 215), substantially increased the number of device types and the number of devices of each type, improved 64-bit support, support for file systems of up to 16 terabytes, in-kernel preemption, support for the Native POSIX Thread Library, User-mode Linux integration into the mainline kernel sources, SELinux integration into the mainline kernel sources, Infiniband support, and considerably more. Several file systems like FUSE, JFS, XFS, ext4 etc. were added throughout the 2.6.x releases. On 20 October 2010 - Linux 2.6.36 was released (13,499,457 lines of code) and in December 2010 Linux 2.6.37 came into existence. Developer's community is considering this version as a base for future kernel development.

In India an operating system, BOSS [4] developed by C-DAC under Ministry of Information technology, Government of India is also based on kernel 2.6 version. It is derived from Debian Linux and is developed for enhancing the use of Free/ Open Source Software throughout India. BOSS4.0 is based on kernel 2.6.32-486.It supports multiple Indian languages. The accessibility of BOSS Linux will have a constructive impact on the digital divide in India as more people can now have access to software in their local language to use the Internet and other information and communications technology (ICT) facilities.

Now a day Linux is one of the most widely ported operating system kernels, running on a diverse range of systems from the iPAQ (a handheld computer) to the IBM Z/Architecture (a massive mainframe server that can run hundreds or even thousands of concurrent Linux instances). It has been ported to various handheld devices such as Tux Phone, Apple's iPod and iphone.

Due to high usability of Linux kernel especially 2.6 version (with its all subversions), it is considered in this paper for reliability study. As very few researches are available on reliability of Linux kernel thus this research and other researches of this type will be useful for researchers and other stockholders of open source operating system.

To find reliability two types of data can be used: time between failures and fault count. The main input parameter to the "time between failures" models is the intervals of successful operations. A probability distribution model whose parameters are estimated by using appropriate mathematical technique reflects the pattern of these intervals.

In case of "fault count" the input parameter of study is the number of faults in a specified period of time rather than the times between failures. In this model normally the failure rate, defined as the number of failures per hour, is used as the parameter of a Probability Distribution Function (pdf). Like the first class, as the fault counts drop, the reliability is expected to increase.

Linux kernel source code is distributed among different modules having their own roles. Thus failure or unexpected result of kernel can be due to normal errors or severe errors in any of these modules. In Linux, a "panic" is an unrecoverable system error detected by the kernel contrary to similar errors detected by user space code. It is possible for kernel code to indicate such a condition by calling the panic function located in the header file sys/system.h. However, most panics are the result of unhandled processor exceptions in kernel code, such as references to invalid memory addresses. These are typically indicative of a bug somewhere in the call chain leading to the panic. They can also indicate a failure of hardware, such as a failed RAM cell or errors in arithmetic functions in the processor caused by a processor bug overheating/damaged processor, or a soft error [5]. This study is based on time between failures and is concerned with bug arrival of Linux 2.6 kernel with major modules as ACPI, Drivers, File System, IO/Storage, Memory Management, Networking, Platform Specific Hardware, Power Management, Process Management, SCSI Drivers, Timers, v4l-dvb and virtualization.

Bug data for Linux kernel 2.6 versions is collected from year 2004 to December 2010.It has sufficient number of bug data. Depending upon trend of the collected data appropriate model is constructed and reliability is estimated by using this model.

The rest of this Paper is organized as follows. Section II provides some mathematical background regarding reliability estimation. Section III concentrates on failure data analysis and the reliability modeling process. Section IV and V concentrates on Goodness of Fit and Research Methodology. Section VI concentrates on reliability estimation and Section VII concludes the Chapter with a summary.

II. Background for Estimating Reliability

There are two approaches for prediction of software reliability: early stage reliability and later stage reliability. In early stage reliability, reliability is estimated during design phase where as in later stage reliability is estimated during operational stage.

Software Reliability is defined as the probability of failure-free software operation for a specified period of time in a given environment. It depends upon failure data or bug data of that software. Failure behavior can be reflected in various ways such as Probability Density Function (pdf) and Cumulative Distribution Function (cdf). pdf denoted as f(x), shows the relative concentration of data samples at different points of measurement scale, such that the area under the graph is unity. CDF, denoted as F (x), is another way to present the pattern of observed data under study. CDF describes the probability distribution of the random variable, X, i.e. the probability that the random variable X assumes a value less than or equal to the specified value t. In other words, F (t) =P [X ≤ t]. [6]

The probability that the random variable X takes on a value in the interval [a, b] is the area under the pdf (probability density function) from a to b, or:

$$P(a \le x \le b) = \int_a^b f(x)dx \ \text{ and } \ f(x) \ge 0 \text{ for all } x \qquad (1)$$

In terms of life data analysis equation (1) describes the probability of a failure occurring between two different points in time.CDF measures the area under the pdf curve up to a given time.

Thus $F(t) = P[X \le t] = \int_{0,-\infty}^{t} f(x)dx \Rightarrow f(t) = F'(t)$.

Therefore, f (t) is the rate of change of F (t). If the random variable T denotes the failure time, F (t), or unreliability, is the probability that the system will fail by time t. Consequently, the reliability R (t) is the probability that the system will not fail by time t

i.e.: F (t) =Q (t) =1-R (t) where R (t) is the reliability function. The reliability function can then be related to the pdf in the following manner: Q (t) +R (t) =1

$$R(t) = 1 - Q(t) = 1 - \int_{0,-\infty}^{t} f(x)dx \qquad (2)$$

Another function that can be derived from the pdf is the failure rate function. The failure rate function (also known as the hazard rate function) is defined by:

$$\lambda = \frac{f(t)}{1 - \int_{0,-\infty}^{t} f(x)dx} = \frac{f(t)}{R(t)} \qquad (3)$$

The mean life, or MTTF, is another widely used function that can be derived directly from the pdf. The arithmetic mean or expected value is defined by:

$$\grave{i} = m = \int_t^{\infty} x.f(x)dx \qquad (4)$$

It is apparent from above equations that pdf [7,8] is sufficient for calculating reliability estimates.

III. Data Collection and Preprocessing of Data

The approach to the reliability estimates of Linux kernel 2.6 consists of three steps: bug collection, bug preprocessing and bug analysis. Bug collection is associated with collecting a data related to failure of a product. In the bug-collection step, the online bug-repository systems are used to collect the failure data. For this purpose web site http://www.bugzilla.kernel.org is used. Data is collected for kernel version 2.6, which is latest available stable version of the Linux kernel. This version of Linux kernel came into existence in year 2004. Bugs are collected from 4/1/2004 i.e. initial year of the kernel release to December 2010. Data are extracted directly from the web site. Bugs reported might be duplicates, provide incomplete information, or may not represent real defects.

Therefore, during the bug preprocessing such noises are removed from the bugs gathered in the first step. Finally, in the third step, the preprocessed data is stored in Mysql database. Initially data was in csv (comma separated value) format. Mysql is an open source [8] data base system. It is freely available and very secure. Total of 2462 records are stored in Mysql table. Main fields of the table are as given below:

Field	**Data Type**	**Purpose**
Bug_Id	Int	Identification Number (Primary Key)
Opendate	Date	Date of submission of the bug
Bug_severity	Varchar	How severe is this bug
Product	Varchar	Module of the kernel
Component	Varchar	Sub module of the kernel
cf_kernel_ version	Varchar	Version of the kernel

From existing table time to failure data is extracted by using a sql query.

Sql Query: SELECT a.product, a.opendate, b.opendate as basedate, round (datediff (a.opendate, b.opendate)) as TimetoFailureDAY, round (datediff (a.opendate, b.opendate)) as TimetoFailureWEEK FROM `bugdatakernellinux` as a, `bugdatakernellinux` as b and b.opendate in (select min (opendate) from `bugdatakernellinux`)).

Structure of extracted data is as shown in following Table.

Table 1: Failure Data

Product	Open Date	Base Date	Time-to failure Day	Time-to failure Week
File System	8/6/2010	1/4/2004	2406	344

File System	3/5/2010	1/4/2004	2252	322

Observing the volume of bug-reports and the varying years of operations among the products, weekly and monthly frequency of bug is calculated and their plot is shown in Fig. 1

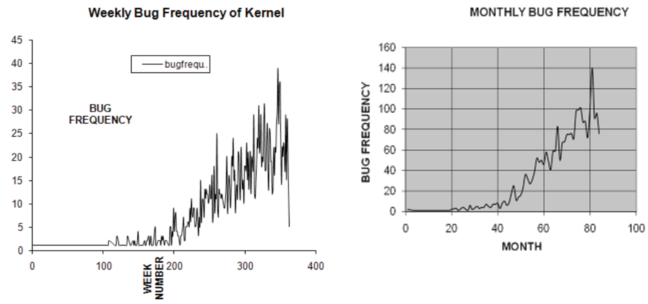

Fig. 1 : Weekly and Monthly Bug Frequency

Bug-Data is collected for the period January 2004 to December 2010, thus the number of weeks for which data is collected are 362 (7 years). In Fig.2 x-axis shows week number from 1 to 362 in the first part and month number from 1 to 84 in second part of the diagram where as y axis shows relative bug frequency. For simplification and clarification purpose the same data is shown weekly as well as monthly.

It is apparent from the diagram that initially up to 20th month i.e. during year 2004-2005 bug frequency is minimum that was official release phase of earlier versions of kernel-2.6 series that is 2.6.1, 2.6.11 and 2.6.15 obviously during this period overall kernel reliability should be maximum. After 40th month there is a sharp increase in bug frequency but does not follow straight-line pattern and it is maximum in 81st [Year 2009-2010] month that is 139 and hence reliability should be minimum during this phase. Particularly in 346th week it is 39.This period represents release phase of advanced version of the kernel 2.6 series that is 2.6.33 onwards. Obviously during this phase there was maximum posting of errors on repository web sites. It may be due to high usability as well as increased awareness among users of open source community.

As kernel has different modules and each module is responsible for a particular defined set of tasks. These tasks also include interaction with hardware. Thus another reason may be due to synchronization between corresponding hardware and kernel modules. To identify these patterns module wise study and analysis of kernel is needed that will be discussed in next paper. Here main target is to study reliability trend of entire kernel version 2.6.

Further by visualizing only graph it is very difficult to judge the exact distribution to be fitted in the collected data hence goodness of fit test is performed to identify the distribution to be fitted. It is elaborated in next section.

IV. Goodness of Fit Test and Parameter Estimation [7, 10 - 12]

The failure properties of a component are best described by statistical distributions, the most commonly used life distributions are as given below:

- 1 and 2 parameter exponential distributions.
- 1, 2 and 3 parameter Weibull distributions.
- Normal distribution.
- Lognormal distribution.
- Generalized Gamma (i.e. G-Gamma)
- Gamma distribution.
- Logistic distribution.
- Log logistic distribution.
- Gumbel distribution.

A Goodness of Fit Test is used is used to identify whether a particular distribution is suitable for a given data or not. In goodness of fit test Null hypothesis H0: sample is taken from a population having given distribution is tested against alternative hypothesis H1: sample is not drawn from a given population having given distribution. On the basis of this test best-fitted distribution can be identified.

There are different methods for goodness of fit like Probability Plots, Correlation Coefficient, and Method of Maximum Likelihood [13], Kolmogorov-Smirnov (KS) Test and Chi Square Test [11]. Method of Maximum Likelihood Estimation is considered as best method and can be used in maximum number of cases. Estimators obtained by this method having minimum variance and maximum efficiency. Thus in this research, Method of Maximum Likelihood is used for parameter estimation as well as goodness of fit.

A. Maximum Likelihood Estimation

Maximum likelihood estimation is used to estimate distribution parameters for a set of data by maximizing the value of Likelihood function. This Likelihood function is largely based on the probability density function (pdf) for a given distribution.

As an example, consider a generic pdf:

$f(x_i;\grave{e}_1,\grave{e}_2,\grave{e}_3,\grave{e}_3....\grave{e}_k)$ where x represents the data (times-to-failure) and θ1, θ2... θk are the parameters to be estimated.

For complete data, the likelihood function [12 - 13] is a product of the pdf functions, with one element for each data point in the data set:

$$L = \prod_{i=1}^{n} f(x_i;\grave{e}_1,\grave{e}_2,....\grave{e}_k)$$

where n is the number of failure data points in the complete data set, and xi is the ith failure time. It is often mathematically easier to manipulate this function by first taking the logarithm of it. This log-likelihood function then has the form:

$$\ln L = \sum_{i=1}^{n} \ln f(x_i;\grave{e}_1,\grave{e}_2,....\grave{e}_k) \qquad (5)$$

Parameters are estimated by using following partial derivatives

$$\frac{\partial \ln L}{\partial \grave{e}_j} = 0, \ j = 1,2...,k \qquad (6)$$

These parameters can be obtained by solving above equations.

The distribution with the largest L value is the best fit statistically.

The log-likelihood function is used for goodness of fit because it is much easier to calculate log likelihood function than likelihood function. Using the log-likelihood function does not affect the validity of the results [7].

For mathematical calculations different mathematical tools like Weibull++, R software, SPSS, Mat Lab etc. are available.

V. Research Methodology Used

In this research Weibull++ [7], which is a well-known tool for life data analysis, is used for mathematical and statistical calculations.

Parameters are estimated by using method of maximum likelihood. Parameters for all life data distributions on the basis of given data are estimated and presented in following table:

Table 2: Parameter Estimates

Distribution	Parameter
Exponential 1	μ = 4.873E-04
Exponential 2	μ = 4.873E-04, γ(Gamma)=1
Normal	Mean (μ)=2052.101, Std(σ)=369.2867
Lognormal	Lmean =7.602529 , LStd =0.278619
Weibull2	β=7.223641, α=2192.193
Gamma	μ=4.585892, K=20.92044
Logistic	μ=2090.126, σ=203.7073
Log logistic	μ=7.635804, σ=0.109968
Gumbel	μ=2216.230, σ=269.8285

- By using above parameters, value of Log Likelihood function is calculated and presented in Table 3.

Table 3: Log Likelihood Value

Distribution	LKV (ln (L))	Rank
Gumbel	-17741.3	1
Weibull2	-17914.1	2
Logistic	-18003.3	3
Normal	-18047.2	4
Loglogistic	-18292.3	5
Gamma	-18487.3	6
Lognormal	-19064.1	7
Exponential 2	-21237.5	8
Exponential 1	-21238.7	9

A distribution having maximum LKV is considered as best distribution to be fitted for given data. Thus from Table 3 it is clear that Gumbel and Weibull distribution is best suited and hence may be considered for reliability estimation.

A. Construction of Reliability Model Using Gumbel Distribution [7 , 14]

The Gumbel distribution's pdf is skewed to the left. The Gumbel distribution is appropriate for modeling strength, which is sometimes skewed to the left (few weak units in the lower tail, most units in the upper tail of the strength population). The Gumbel distribution could also be appropriate for modeling the life of products that experience very quick wear-out after reaching a certain age. The distribution of logarithms of times can often be modeled with the Gumbel distribution.

The pdf of the Gumbel distribution is given by:

$$f(T) = \frac{e^{z-e^{z}}}{ó}, f(z) \geq 0, ó > 0$$ where: $$z = \frac{T-ì}{ó}$$ and

μ=location parameter and σ=scale parameter. Here T represents Time to Failure. The effect of μ and σ on Gumbel Distribution is shown in Fig. 2.

$$F(z) = \int_0^z \frac{e^{z-e^{z}}}{ó} ó dz = \int_1^y \frac{y.e^{-y}}{y} dy = \left[-1/e^{y}\right]_1^y = 1-e^{-y} = 1-e^{-e^{z}}$$

Here dT=σdz

$$F(z) = 1-e^{-e^{z}} \qquad (7)$$

Hence unreliability function at time T is given by

$$Q(T) = 1-e^{-e^{z}} .$$

Thus Reliability Function

$$R(T) = 1\text{-}Q(T) = e^{-e^{z}} \qquad (8)$$

where $z = \frac{T-\mu}{\sigma}$

Likelihood function is given by

$$L(\mu, \sigma|Ti) = \prod_{i=1}^{n} f(T_i; ì, ó) \qquad (9)$$

Log Likelihood Function is obtained by taking log from both sides of equation (9).

That is $$\ln(L) = \sum_{i=1}^{n} \ln(f(Ti; ì, ó)) = \sum_{i=1}^{n} \ln\left(\frac{e^{z_i} - e^{z_i}}{ó}\right)$$

$$= \sum_{i=1}^{n} (z_i - e^{z_i}) - \sum_{i=1}^{n} ó = \sum_{i=1}^{n} \frac{Ti-\mu}{\sigma} - \sum_{i=1}^{n} e^{(Ti-\mu)/\sigma} - n\sigma$$

$$\ln(L) = \frac{1}{\sigma}\left[\sum Ti - n\mu\right] - e^{-\left(\frac{\mu}{\sigma}\right)} \sum_{i=1}^{n} \frac{Ti}{\sigma} - n\sigma \qquad (10)$$

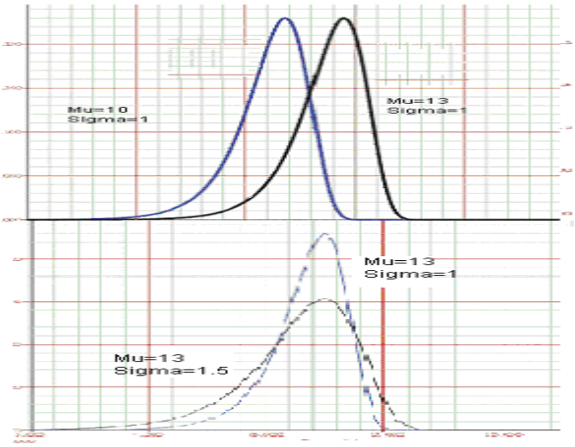

Fig. 2: Effect of μ and σ on Gumbel Distribution

MLE's and goodness of fit is evaluated on the basis of this equation. For MLE's partial derivative of equation (10) with respect to μ and σ are used.

V.2 Using Weibull Distribution [15,16]

The Weibull distribution is one of the most widely used lifetime distributions in reliability engineering. It is a versatile distribution that can take on the characteristics of other types of distributions, based on the value of the shape parameter, β.

Two parameter Weibull distribution is given by

$$f(t) = \frac{\beta t^{\beta-1}}{\alpha^{\beta}} e^{-\left(\frac{t}{\alpha}\right)^{\beta}} \qquad (11)$$

where α represents scale parameter and β represents the shape parameter. The effect of the scale parameter is to squeeze or stretch the distribution. Fig. 3 shows the Weibull pdf for several values of the shape parameter when α =1.

Here cdf=Unreliability Function=Q (t) =F (t)

$$F(t) = \int_0^t f(t)dt = \int_0^t \frac{\beta t^{\beta-1}}{\alpha^\beta} e^{-\left(\frac{t}{\alpha}\right)^\beta} dt$$

Put $\left(\frac{t}{\alpha}\right)^\beta = y \Rightarrow \frac{1}{\alpha^\beta}\beta\, t^{\beta-1}dt = dy$

$$F(t) = \int_0^y e^{-y}dy = 1 - e^{-\left(\frac{t}{\alpha}\right)^\beta}.$$

Thus Reliability Function

$$R(t) = 1 - Q(t)$$

$R(t) =$ [illegible] (12)

Log Likelihood Function is given by

$$\ln(L) = \sum \ln(f(Ti;\hat{a}\ \acute{a})) = \sum \ln\left(\frac{\beta Ti^{\beta-1}}{\alpha^\beta} e^{-\left(\frac{Ti}{\alpha}\right)^\beta}\right)$$

$$= \sum_{i=1}^{n}\left(\ln\beta + (\beta-1)\ln(Ti) - \beta\ln(\alpha) - \left(\frac{Ti}{\alpha}\right)^\beta\right) \quad (13)$$

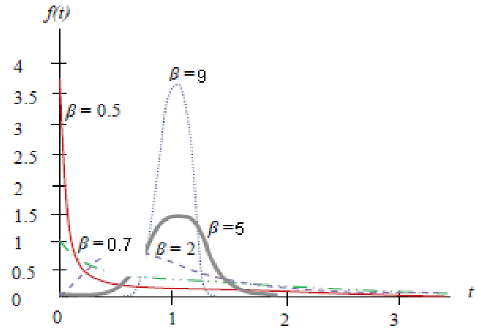

Fig. 3: Weibull cdf with different values of β when α =1

MLE's and goodness of fit is evaluated on the basis of this equation. For MLE's partial derivative of equation (13) with respect to β and α are used. For goodness of fit test log likelihood function is used. In coming sections reliability is calculated by using both models and final result is compared.

VI. Reliability Estimation

It is clear from the goodness of fit section that best distributions appropriate for collected sample are Gumbel distribution and Weibull distribution with two parameters. Thus the above-derived models are applied on the collected sample. Sample Data under study on which these models are to be applied is given below .The data is extracted from Mysql table – (some records)

Table 4: Time to Failure Data

PRODUCT	TIMETOFAILURE DAY (Ti)
File System	2406
File System	2252
Other	1008
File System	1605
Other	1748

VI.1 Reliability Evaluation by using both models

Parameters are estimated by using Weibull++ software and verified by likelihood function derived in preceding sections and SPSS 17.0 software.

Table 5: Parameter Estimation

Distribution	Parameter I	Parameter II
Gumbel	μ=2216.230129	σ=269.8285418
Weibull	β=7.13784899	α=2190.92971

Gumbel pdf is $f(T) = \frac{e^{zi - e^{Z_i}}}{\sigma}, f(z) \geq 0, \sigma > 0$

where $Zi = \frac{Ti - \mu}{\sigma} = \frac{Ti - 2216.30129}{269.8285418}$ (14)

Weibull pdf is

$f(t) = \frac{\beta t^{\beta-1}}{\alpha^\beta} e^{-\left(\frac{t}{\alpha}\right)^\beta}$ where β=7.13784899 and α=2190.92971.

Thus f (t)

$$= \frac{(7.13784899)T^{6.13784899}}{(2190.92971)^{7.13784899}} e^{-\left(\frac{T}{2190.92971}\right)^{7.13784899}} \quad (15)$$

By applying equation (14) and (15) pdf of both the distribution is calculated and graph is shown in table 6 and corresponding pdf graph is shown in Fig. 4.

Table 6: pdf table

Month No.	f(t)-Gumbel	f(t)-Weibull
5	0.00000157	0.0000000001
10	0.00000274	0.0000000087
15	0.00000477	0.0000001307
20	0.00000832	0.0000008481
25	0.00001447	0.0000035489
30	0.00002516	0.0000113099
35	0.00004365	0.0000299054
40	0.00007543	0.0000688724
45	0.00012950	0.0001420910
50	0.00021979	0.0002665110
55	0.00036563	0.0004563211
60	0.00058736	0.0007089989
65	0.00088781	0.0009830149
70	0.00120678	0.0011811245
75	0.00136317	0.0011765324
80	0.00111509	0.0009126333
85	0.00051963	0.0005064347
86	0.00046806	0.0004720789

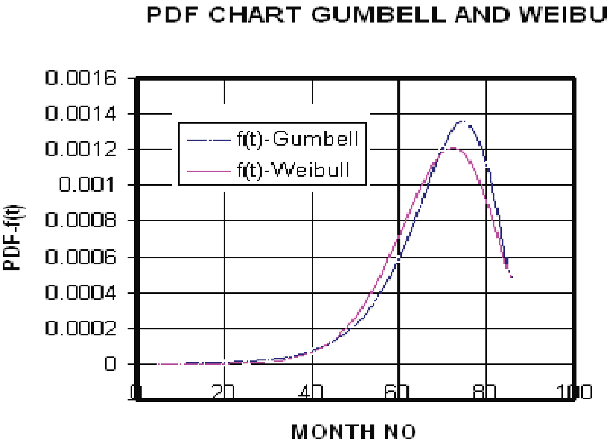

Fig. 4: pdf-Gumbel and Weibull

Graphs in Fig. 4 shows almost same pattern, which indicates best goodness of fit of the data and its randomness. Probability is maximum during 74th and 75th month of the survey.

That is P [T=74 or 75] is maximum. It means that probability that time to failure is 74 or 75] is maximum. It further implies that during this phase [2009-2010] there was maximum number of fault counts. It may be due to high awareness and usability as well as growth of development in kernel module and hardware components. There may be lack of synchronization between hardware components and kernel modules during this phase.

Reliability can be estimated by using equation (8) and (12)

From equation (8)

Reliability-Gumbel = $e^{-e^{z}}$

$$= e^{-e^{\frac{Ti-2216.230129}{269.8285418}}} \qquad (16\ )$$

From equation (12)

Reliability-Weibull = $e^{-\left(\frac{t}{á}\right)^{â}}$

$$= \exp\left(-\left(\frac{Ti}{2190.9271}\right)^{7.13784899}\right) \qquad (17)$$

Reliability is estimated by using equation (16) and (17). Table 7 shows the individual reliabilities for different time periods for Kernel 2.6 series by using both the models.

Table 7: Reliability Estimates by using both models

Year	Month	Month No.	Reliability using Gumbel Distribution	Reliability using Weibull Distribution
2004	May	5	0.999575778	0.999999999
2004	October	10	0.999260476	0.999999668
2005	March	15	0.998710977	0.999992294
2005	August	20	0.997753633	0.999932152
2006	January	25	0.996086676	0.999641465
2006	June	30	0.993186954	0.998618967
2006	November	35	0.988151393	0.995713158
2007	April	40	0.979432754	0.988636424
2007	September	45	0.964415176	0.973345026
2008	February	50	0.93878029	0.943446497
2008	July	55	0.895704178	0.89009713
2008	December	60	0.825273336	0.803342744
2009	May	65	0.715461476	0.676231328
2009	October	70	0.557785972	0.512169031
2010	March	75	0.361376624	0.332142546
2010	August	80	0.169547964	0.172506868
2010	December	85	0.04531613	0.065692132

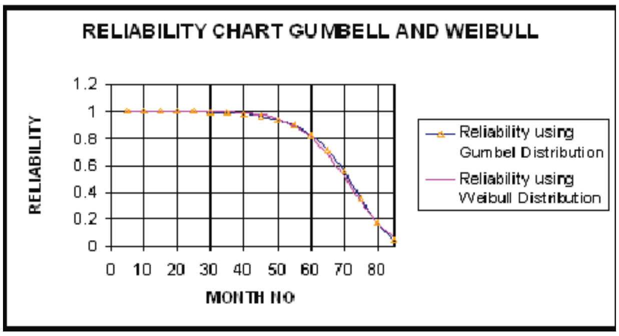

Fig. 5: Overall Reliability of Linux Kernel 2.6 Series using Gumbel and Weibull Distribution

Fig. 5 compares the reliabilities of Linux kernel 2.6 series by using both the methods .As expected both graph converges to a single line. It indicates that reliability calculated by both the methods at different interval of time does not vary significantly. Thus it indicates that calculation strategy can be accepted. Further it is clear from the above diagram and table that kernel 2.6 series follows a better reliability pattern. On and average during first 3/4th of the survey both the method gives reliability estimates of more than 80% which is good enough for any product.

VII. Conclusions

Data related to failure reports are the main source for understanding the failure distribution, classifying failures, and building accurate dependability models. The quality of analysis heavily depends on comprehensive and accurate recording of these data and accurate determination of distribution to be fitted. The dearth of a commonly accepted data format for archiving bug reports and efficient tool/technique for goodness of fit adds to the complexity and inaccuracy of failure data analysis. As there are multiple distributions for life data analysis with different set of parameters there is confusion in selection of appropriate distribution. In this experiment, by using goodness of fit test and Likelihood Function two best distributions for the collected data is selected. These tests suggest two distributions Gumbel and Weibull as two top ranker distributions for this experiment.

By fitting both the distributions it has been found that pdf graph of both the distributions are almost same which indicates correctness of goodness of fit.

There are different methods for estimating parameter of the distribution. In this experiment method of maximum likelihood is used. Method of maximum likelihood is considered as best method. MLE's are unbiased as well as it has minimum variance and hence most efficient. Thus in this study for parameter estimation method of Maximum Likelihood is used.

Reliability is estimated by using both distributions and reliability graph shows converging reliability estimates by both the methods. 3/4th of graph shows more than 80% reliability, which is a good sign of OSS success all over the world.

Further, decreasing trend of reliability in later phase of survey may be due to low reliability of a specific module or due to low reliability of specific higher kernel version or due to awareness and increased usability of OSS (exhaustive usage).

As kernel consists of different modules hence one avenue of future research is to investigate the reliability growth of these modules individually and their relationship and their combined effect on overall reliability of kernel. In this research Bug_severity is not considered as important parameter. But a bug having blocking/high severity may affect reliability more than a bug having low severity. Thus another avenue for future research may be reliability estimation by considering this factor.

This research and future research of this category may be useful for persons involved in OSS Development (Regarding future strategy of programming and designing), Government agencies using OSS (Selection of OSS), IT consultants (for strategic planning) and Researchers of any domain (Selection of Model/Distribution).

References

[1] Rémy Card, Franck Mével, "The Linux Kernel Book"

[2] GNU Public License, [Online] Available : "http://en.wikipedia.org/wiki/GNU_General_Public_License"

[3] Physical_Address_Extension". [Online] Available : http://en.wikipedia.org/wiki/Physical_Address_Extension"

[4] BOSS, [Online] Available : http://bosslinux.in/

[5] Soft Error, [Online] Available : "http://en.wikipedia.org/wiki/Soft_error"

[6] S.C. Gupta, V.K. Kapoor ,"Fundamental of Mathematical Statistics"

[7] Weibull++, Reliability Function, [Online] Available : "http://www.weibull.com","http://www.reliasoft.com"

[8] Sanjeev Kumar Jha, Dr. A.K.D.Dwivedi, Dr. Amod Tiwari, "Reliability Models and Open Source Software :An Empirical Study"

[9] Christian Walck, "Hand-book on STATISTICAL DISTRIBUTIONS for experimentalists"

[10] Snedecor, George W., Cochran, William G. (1989), Statistical Methods, Eighth Edition, Iowa State University Press.

[11] Engineering Statistics Hand Book, [Online] Available : http://www.itl.nist.gov/div898/handbook/eda/section3/eda35f.htm

[12] [Online] Available : http://www.stat.rice.edu/~dobelman/textfiles/Distribution Handbook.pdf

[13] F. S. G. RICHARDS, "A Method of Maximum-likelihood Estimation", http://www.jstor.org/pss/2984037

[14] Gumbel Distribution, [Online] Available : http://mathworld.wolfram.com/GumbelDistribution.html

[15] Cobra Rahmani, Harvey Siy, Azad Azadmanesh "An Experimental Analysis of Open Source Software Reliability"

[16] Ying ZHOU , Joseph DAVIS "Open source software reliability model: an empirical approach "

[17] Bug Repository: [Online] Available : bugzilla.org,sourceforge.net

[18] Kernel Source: [Online] Available : linuxkernel.org

Sanjeev Kumar Jha, Senior Systems Analyst, DOEACC Society Chandigarh B.O: Lucknow, Ministry of CO&IT Government of India. Bachelors and Masters Degree in Statistics [Honors] from Patna University and presently doing his PhD in Computer Science from Singhania University. He has attended 2 international Conferences. Area of research interest is Reliability and Open Source Software

Dr. A.K.D. Dwivedi, Director, DOEACC Society Chandigarh, Ministry of C&IT Government of India. M.Tech from Allahabad University and PhD from Gorakhpur University. Attended many international and national Conferences. Published many papers in international and national journals. Executed many government projects successfully. Area of research interest is Signal Processing, Reliability and Open Source Software.

Dr. Amod Tiwari, Associate Professor, PSIT Kanpur. PhD from IIT Kanpur. Attended many international and national Conferences. Published many papers in international and national journals. Area of research interest is Image Processing, Reliability and Open Source