

Um Estudo Empírico Exploratório em Confiabilidade de Sistemas Operacionais

Marcela Prince Antunes, Rivalino Matias Júnior

Faculdade de Computação - Universidade Federal de Uberlândia (UFU)
Av. João Naves de Ávila, 2121 - 38400-902 - Uberlândia/MG - Brasil

mahprince@comp.ufu.br, rivalino@fc.ufu.br

Abstract. *In this paper, we conduct an empirical exploratory study on OS reliability. We analyze more than 30,000 real OS failure data collected from different workplace environments. The results show that the main cause of OS failures is related to OS services. The Gamma and Weibull density functions presented the best fit to the OS failure data. OS kernel failures were more prevalent in enterprise workplaces than academics.*

Resumo. Neste artigo, conduzimos um estudo empírico exploratório sobre confiabilidade de sistemas operacionais (SO). Analisamos mais de 30.000 falhas reais de SO, coletadas em diferentes ambientes de trabalho. Os resultados mostram que a principal causa de falhas do SO está relacionada aos serviços de SO. As distribuições Gama e Weibull apresentaram o melhor ajuste aos dados de falhas de SO. As falhas do *kernel* foram mais predominantes em ambientes corporativos do que em acadêmicos.

1. Introdução

Cada vez mais pessoas são dependentes de software para as suas rotinas diárias [Lyu 2007]. Isto leva a situações onde falhas de software podem causar desde simples inconvenientes até eventos catastróficos [Leveson e Turner 1993]. Atualmente, a confiabilidade de software é uma preocupação não só para sistemas especializados, mas também para software de prateleira. Por exemplo, de acordo com [Li *et al.* 2008] os sistemas de navegação da marinha americana e de gerenciamento de energia da ABB [ABB 2002] utilizam sistemas operacionais MS Windows.

A confiabilidade de software é definida como a probabilidade do software executar livre de falhas em um dado ambiente e por um período de tempo definido [ANSI/IEEE Standard 1990]. Muitas pesquisas foram desenvolvidas nesta área considerando diferentes categorias de software. Contudo, estes trabalhos têm focado em aplicações do usuário que, em sua maioria, rodam sobre um software de sistema operacional (SO). Como essas aplicações dependem do SO, se o software de SO não garantir, pelo menos, o mesmo nível de confiabilidade esperado para as aplicações, a confiabilidade de todo o sistema é comprometida. Pois, para o usuário, não importa se uma falha ocorre no nível da aplicação ou do SO, nos dois casos a experiência é de uma falha no sistema. Assim, aumentar a confiabilidade do SO é um requisito importante para a confiabilidade dos sistemas de computação como um todo.

Os sistemas operacionais modernos enfrentam muitos desafios para garantir alta confiabilidade. Em termos de linhas de código estes crescem continuamente para suportar as inovações de hardware e de cargas de trabalho. A literatura (ex. [Murphy 2008] e [Yuan e Zhang 2011]) indica que falhas de software tendem a crescer de acordo

com o tamanho dos programas. A natureza altamente integrada dos subsistemas de um SO é outro desafio, pois torna a propagação de erros no *kernel* difícil de prevenir e isolar [Tanenbaum, Herder e Bos 2006]. A fim de melhorar a confiabilidade do SO, é essencial entender seus padrões de falhas observados sob cargas de trabalho reais. Neste artigo, apresentamos um estudo que caracteriza vários aspectos das falhas de SO, analisando dados de falhas de diferentes sistemas reais e perfis de uso. Como resultado, obtivemos funções de confiabilidade para diferentes grupos de sistemas e categorias de falhas. Estas funções são importantes para a análise quantitativa destes sistemas, pois na sua ausência muitos trabalhos adotam funções teóricas não estimadas de dados reais. Para o melhor do nosso conhecimento, este é o estudo acadêmico mais abrangente sobre caracterização de falhas de SO realizado até o momento, com mais de 30.000 falhas de SO analisadas.

As demais seções deste trabalho são organizadas como segue: a Seção 2 apresenta os trabalhos relacionados, a Seção 3 descreve os materiais e métodos utilizados, a Seção 4 discute os resultados encontrados e, finalmente, a Seção 5 apresenta nossa conclusão.

2. Trabalhos relacionados

Em [Swift, Bershad e Levy 2003], os autores relatam que extensões do *kernel* do sistema operacional (ex. *drivers* de dispositivo) correspondem a mais de 70% do código do *kernel* do Linux, enquanto o Windows XP apresenta mais de 35.000 *drivers* diferentes. Eles concluem que as extensões de terceiros foram uma das principais causas de falhas do Windows XP, onde os *drivers* foram responsáveis por 85% das falhas relatadas. Achados semelhantes foram encontrados no Linux [Chou *et al.* 2001], onde os erros de *drivers* foram predominantes em comparação com outras partes do *kernel*.

Em [Ganapathi e Patterson 2005] foram analisadas 1.546 falhas de sistemas operacionais coletadas em computadores com SO Windows em um ambiente acadêmico. Somente falhas no nível de *kernel* foram consideradas. Os autores concluem que as falhas de *kernel* são menos prevalentes do que as falhas de aplicações, porém com maior impacto em termos de inatividade não planejada. Em um novo estudo, [Ganapathi, Ganapathi e Patterson 2006] os autores analisaram 2.528 falhas de *kernel* do Windows XP. Os resultados corroboraram o primeiro estudo, bem como os resultados em [Swift, Bershad e Levy 2003], enfatizando que os *drivers* de dispositivo são os que mais contribuem com as falhas do sistema operacional.

Em [Kalyanakrishnam, Kalbarczyk e Iyer 1999], os autores analisaram dados de falhas de 70 servidores Windows NT, coletados ao longo de seis meses. Os resultados mostraram que, em média, o tempo de atividade dos servidores foi de 283,68 horas. Da mesma forma, em [Xu, Kalbarczyk e Iyer 1999] a confiabilidade de servidores Windows NT foi investigada. Foram coletados dados de falha de 503 servidores ao longo de quatro meses. Todos os eventos que levaram o sistema a reinicializar, sofrer erro ou parar foram considerados falhas de SO. Eles encontraram os principais tempos médios entre falhas (MTBF), sendo eles: hardware (92,74 horas), aplicações (31,52 horas), configuração (13,61 horas) e manutenção (5,92 horas).

Este trabalho difere dos citados anteriormente da seguinte forma. Os estudos anteriores investigaram as falhas de SO, em sua maioria, no nível de *kernel*. Diferentemente, consideramos falhas de SO em ambos os níveis, ou seja, *kernel* e

espaço do usuário, já que em ambos a experiência do usuário será afetada. Em resumo, a maioria dos trabalhos anteriores analisou a confiabilidade do SO pela perspectiva do fabricante (ex. [Murphy 2008]), programador (ex. [Swift, Bershad e Levy 2003]) ou administrador (ex. [Kalyanakrishnam, Kalbarczyk e Iyer 1999]). Nesses casos, os cálculos de confiabilidade não representam integralmente as percepções dos usuários em relação à confiabilidade do sistema operacional, já que consideraram como falhas do SO apenas aquelas que ocorreram no nível do *kernel*. Nossa abordagem abrange diferentes categorias de falhas de SO, as quais serão descritas na próxima seção.

3. Materiais e Métodos

3.1. Estratégia de coleta dos dados

Baseando-se em diferentes pesquisas (ex. [Bott 2013] e [StatCounter Global Stats 2013]) sobre sistemas operacionais de mercado, descobrimos que o Windows 7 (Win7) é atualmente o mais usado, tendo mais de 45% de participação no mercado, seguido pelos outros sistemas da família Windows (XP, Win8 e Vista). Assim, decidimos estudar as falhas deste SO. O Win7 fornece registros detalhados de falhas através do *Reliability Analysis Component* (RAC) [Microsoft 2007], que é ativado automaticamente durante sua instalação. Os dados coletados pelo RAC são gravados em um arquivo SDF (*SQL Compact Edition Database File*). Este arquivo contém todos os eventos de falhas registrados desde a data de instalação do sistema operacional. Nós coletamos o arquivo SDF de diferentes computadores através de dois métodos. No primeiro, copiamos manualmente o arquivo. No segundo, criamos um repositório online para receber arquivos SDF através de uma página web. Nos dois casos, para cada arquivo, preenchemos um formulário¹ para caracterizar o sistema pesquisado. Este formulário contém informações sobre o ambiente de trabalho e o perfil de uso (ver Tabela 2).

Como os arquivos SDF armazenam não apenas falhas de SO foi necessário filtrar seu conteúdo. Para isso, criamos três categorias de falhas de SO: aplicações, serviços e *kernel*. A primeira abrange as falhas causadas pelo funcionamento incorreto das aplicações do SO, que executam no espaço do usuário. A segunda categoria inclui falhas de componentes dos serviços do SO. A terceira categoria contém falhas causadas pelos subsistemas do *kernel* do SO, as quais geralmente exigem a reinicialização do sistema; são fortemente relacionadas às bibliotecas (ex. *ntdll.dll*). A fim de calibrar as regras de filtragem, realizamos uma classificação manual dos eventos de falhas de uma amostra dos arquivos coletados, nos baseando em três campos do arquivo SDF: *Event Identifier*, *Source Name* e *Product Name*. A Tabela 1 mostra um exemplo dessa classificação. O primeiro campo identifica unicamente o tipo de evento do Windows relacionado ao registro da falha. Note que para eventos de falha relacionados com aplicações do SO, os dois primeiros campos não são suficientes para distinguir aplicações de SO e de usuário. Assim, com o auxílio de um especialista, analisamos também um terceiro campo (*Product Name*). Por exemplo, na Tabela 1 dois eventos de falha possuem o *Event ID* e *Source Name* iguais a 1000 e “*Application Error*”, respectivamente; nestes casos ambos são analisados com relação ao seu *Product Name*. Um deles é um evento relacionado ao Firefox.exe que é considerada uma aplicação de usuário, enquanto o outro, explorer.exe, é considerada uma aplicação de SO.

¹ <http://hpdc.facom.ufu.br/osr-team/index.php>

Tabela 1- Exemplo do filtro utilizado

| Event ID | Source Name | Product Name | Categoria de falha de SO |
|----------|-----------------------|-----------------------|--------------------------|
| 20 | WindowsUpdate cliente | Atualização do Office | Serviços de SO |
| 1000 | Application Error | explorer.exe | Aplicações de SO |
| 1000 | Application Error | firefox.exe | <i>Não Aplicável</i> |
| 1002 | Application Hang | mmc.exe | Aplicações de SO |
| 1137 | Windows StartupRepair | Windows | <i>Kernel do SO</i> |

Baseando-se nos arquivos coletados a partir de 735 computadores, encontramos 30.815 falhas de SO. Estas falhas foram organizadas em seis grupos (G1 a G6). As falhas dos primeiros 5 grupos foram coletadas localmente, enquanto as falhas do G6 foram coletadas pelo formulário online. A Tabela 2 apresenta o perfil dos 6 grupos.

Tabela 2 - Descrição dos grupos

| | Ambiente | Perfil de Aplicação |
|-----------|------------------------------------|--|
| G1 | Acadêmico | Aplicativos de escritório e edição gráfica. |
| G2 | Acadêmico | Aplicativos de escritório, IDE de programação, multimídia, ciência e engenharia. |
| G3 | Acadêmico | Aplicativos de escritório, edição gráfica e IDE de programação. |
| G4 | Corporativo | Aplicativos de escritório, edição gráfica e IDE de programação. |
| G5 | Corporativo | Aplicativos de escritório, edição gráfica e IDE de programação. |
| G6 | Acadêmico, Corporativo, Doméstico. | Aplicativos de escritório, edição gráfica, IDE de programação, multimídia, ciência e engenharia, jogos, aplicativos ERP, antivírus, servidores: banco de dados, web, aplicações, e-mail, diretórios e arquivo/impressão. |

Os grupos G1 e G2 contêm falhas de SO de computadores de duas universidades. Nestes grupos, os computadores são usados por estudantes para executar aplicativos de propósito gerais (ex. edição de texto) e aulas de laboratório (ex. aplicativos de engenharia). G3 e G4 são de uma mesma universidade. No entanto, G3 contém falhas de SO de computadores de laboratórios de ensino, com perfil de uso semelhante ao do G1 e G2, enquanto G4 contém falhas de computadores utilizados em ambiente administrativo. As falhas do G5 vêm de um ambiente corporativo. G6 é o único grupo heterogêneo, ou seja, contém falhas de SO de diferentes ambientes, como empresas, universidades e computadores pessoais. Uma análise detalhada de cada grupo é apresentada na Seção 4.

3.2. Abordagem de Análise de Dados

Inicialmente, realizamos uma caracterização de cada grupo calculando suas estatísticas descritivas. Depois, realizamos o teste de melhor ajuste de dados (*Goodness-of-fit - GoF*) para avaliar a hipótese de que cada amostra de tempos de falha segue uma determinada distribuição de probabilidades [Rayner, Thas e Best 2009]. Este teste nos permite selecionar a função densidade que se ajusta melhor à amostra, a fim de usá-la para estimar as métricas de confiabilidade de interesse. Nós adotamos quatro testes GoF amplamente usados: Anderson Darling [Cullen e Frey 1999], Kolmogorov-Smirnov [Cullen e Frey 1999], Qui-Quadrado [Conover 1999] e Razão de verossimilhança [Mood, Graybill e Boes 1974]. Nem todos os quatro testes foram aplicados a todas as amostras, já que isso depende dos pressupostos de cada teste. Além disso, não é incomum obter resultados conflitantes entre os testes para uma mesma amostra devido às especificidades de cada algoritmo. Por isso, contrastamos os resultados de todos os testes e selecionamos o que mostrou mais concordância.

Aplicamos o teste GoF com funções amplamente usadas em estudos de confiabilidade. Consideramos quinze funções de densidade: Exponencial, Exponencial $c/2$ parâmetros, Gama, Gama generalizada, Gumbel, Maior Valor Extremo, Logística, Log-logística, Log-logística $c/3$ parâmetros, Lognormal, Lognormal $c/3$ parâmetros, Normal, Menor Valor Extremo, Weibull e Weibull $c/3$ parâmetros. Para cada grupo de falhas, testamos todas as amostras para identificar a melhor função densidade por computador. Dado que os computadores de um mesmo grupo (exceto o G6) apresentam o mesmo perfil de uso e carga de trabalho, nós também agrupamos suas amostras e fizemos o teste GoF para identificar a função densidade por grupo.

Com base no modelo de melhor ajuste obtido, calculamos um conjunto de métricas de confiabilidades de interesse. Estas métricas estão listadas na Tabela 3. As três primeiras são padrões em engenharia de confiabilidade [Trivedi 2001]. Com base na $R(t)$, estimamos a quarta métrica, MTBF, que é o tempo entre duas falhas subsequentes do SO. Esta é uma métrica fundamental para confiabilidade em sistemas reparáveis, como no nosso estudo. A quinta métrica indica o tempo de garantia t para uma dada confiabilidade demonstrada (ex. para $R=0,05$, $t \leq 3$ horas). A última métrica mostra o momento em que $x\%$ dos sistemas terão falhado. Por exemplo, uma vida B10% de 70h significa que 10% dos sistemas terão falhado após 70 horas de operação. Nós usamos $R = 0,95$ para a função tempo de garantia e 10% para a Vida Bx.

Tabela 3 - Métricas de confiabilidade calculadas

| Métrica | Descrição |
|-------------------------------|--|
| Confiabilidade $R(t)$ | Probabilidade que o sistema irá funcionar corretamente até o tempo t |
| Probabilidade de Falha $F(t)$ | Probabilidade que o sistema irá falhar após o tempo t |
| Taxa de Falha $\lambda(t)$ | Taxa de falha dado um tempo de sobrevivência t |
| Vida Média (MTBF) | Tempo médio entre falhas |
| Tempo de Garantia $t(R)$ | Momento no qual um valor de confiabilidade específico será alcançado |
| Vida Bx% | A hora em que $x\%$ dos sistemas da amostra terão falhado |

4. Resultados

4.1. Análise de Caracterização

O período de amostragem de cada grupo segue: G1 (20/03/2012 até 14/12/2012), G2 (16/07/2012 até 27/09/2012), G3 (13/12/2011 até 09/11/2012), G4 (24/10/2011 até 19/11/2012), G5 (24/10/2011 até 05/11/2012) e G6 (27/09/2010 até 06/03/2013). Em seguida, calculamos as estatísticas descritivas para cada grupo a fim de obter uma caracterização geral das falhas. Estes dados são apresentados na Tabela 4.

Tabela 4 - Caracterização de falhas por grupo

| | G1 | G2 | G3 | G4 | G5 | G6 |
|-------------------------------------|-------|-------|-------|-------|-------|-------|
| Período de amostragem (dias) | 269 | 72 | 332 | 391 | 378 | 891 |
| Total de computadores | 5 | 63 | 268 | 275 | 41 | 83 |
| Total de falhas de SO | 284 | 406 | 6844 | 19725 | 548 | 3008 |
| % Aplicação | 0,35 | 3,94 | 0,89 | 20,55 | 9,87 | 5,65 |
| % Serviço | 97,89 | 92,61 | 84,07 | 76,92 | 82,27 | 85,14 |
| % Kernel | 1,76 | 3,45 | 15,04 | 2,53 | 7,86 | 9,21 |
| Falhas por dia (média) | 1,06 | 5,64 | 20,61 | 50,45 | 1,45 | 3,38 |

Os serviços de SO foram responsáveis pelo maior percentual de falhas em todos os grupos, sendo os percentuais próximos. As falhas de *kernel* não apresentam os

menores percentuais como esperado, tendo valores similares aos de aplicações de SO. Os ambientes de trabalho representados pelos grupos G3 e G4 apresentam a pior confiabilidade, uma vez que têm o maior número de falhas por dia, seguidos pelos grupos G2, G6, G5 e G1. Apesar de terem diferentes perfis de uso e serem provenientes de diferentes departamentos, eles são da mesma organização. Nossa hipótese é que sua baixa confiabilidade pode ser resultado da política de administração aplicada à organização como um todo. Com relação às falhas de aplicações de SO, podemos notar que elas ocorreram mais nos grupos ligados aos ambientes corporativos (G4 e G5) do que nos outros. Por outro lado, as falhas de serviço de SO são prevalentes nos grupos relacionados aos ambientes acadêmicos (G1, G2 e G3). Em termos de falhas de *kernel* do SO, notamos uma maior prevalência em ambientes corporativos (G4 e G5) do que nos outros, exceto pelo G3, que parece ser uma exceção (*outlier*) nesta categoria de falha. Por causa do perfil heterogêneo do G6, nós o analisamos separadamente. Os padrões parecem seguir os encontrados nos outros grupos. Os serviços de SO mostram um maior percentual de falhas, seguido pelas falhas de *kernel* e aplicações de SO, nesta ordem. Isto corrobora os resultados encontrados nos outros grupos, onde as falhas de *kernel* do SO parecem ocorrer mais em ambientes corporativos. Analisando os perfis de utilização dos sistemas no G6, eles são mais relacionados ao ambiente corporativo, confirmando a hipótese anterior.

4.2. Modelagem de Falhas de SO

Um dos resultados mais importantes deste estudo são os modelos de melhor ajuste identificados para as categorias de falhas de SO investigadas. Estes modelos são úteis para estudos de confiabilidade de sistemas computacionais, tanto analiticamente quanto através de simulações. Primeiramente, realizamos os testes GoF para cada uma das três categorias de falhas por computador e para cada grupo individualmente. O método de estimação de parâmetros usado foi o método de máxima verossimilhança. A Tabela 5 apresenta os resultados. As quatro entradas sem modelo apresentaram amostras muito pequenas de falhas de SO para as categorias e grupos, inviabilizando a identificação de um modelo. Para os demais casos foi possível estimar um modelo e seus parâmetros. Note que, em geral, Gama e Weibull apresentaram a melhor qualidade de ajuste nas categorias, sendo Gama prevalente para serviços de SO e Weibull para aplicações e *kernel* de SO.

Tabela 5 - Modelos de melhor ajuste por categoria de falha e por grupo

| | Aplicações do SO | Serviços do SO | Kernel do SO |
|-----------|--|------------------------------------|--|
| G1 | - | Gama ($k=0,12; \mu=6,11$) | - |
| G2 | - | Weibull ($\beta=0,82; \eta=279$) | - |
| G3 | Weibull ($\beta=0,32; \eta=264$) | Gama ($k=0,16; \mu=7,50$) | Gama ($k=0,45; \mu=6,81$) |
| G4 | Log-logística ($\mu=-6,07; \sigma=2,1$) | Gama ($k=0,14; \mu=6,64$) | Weibull ($\beta=0,32; \eta=254$) |
| G5 | Weibull ($\gamma=0; \beta=0,33; \eta=382$) | Gama ($k=0,35; \mu=6,45$) | Weibull ($\beta=0,51; \eta=254$) |
| G6 | Weibull ($\beta=0,44; \eta=482$) | Gama ($k=0,30; \mu=6,71$) | Log-logística ($\mu=4,26; \sigma=1,1$) |

Posteriormente, identificamos o modelo de melhor ajuste para os grupos como um todo considerando as duas abordagens descritas na Seção 3.2. Em seguida, confrontamos os dados para maior confiança. Os resultados são apresentados na Tabela 6. A estimação de parâmetros baseou-se na máxima verossimilhança e na função de log-verossimilhança (Lk). A maioria das amostras (exceto G2) contém alguns tempos de falha com valor zero. Portanto, foi necessário usar o método PNZ (*percentage of non-*

zero) [Reliability HotWire 2008]. Como pode ser visto nas tabelas 5 e 6, os modelos Gama e Weibull foram predominantes, mostrando alta consistência nas análises intra e intergrupos. Baseado nos modelos estimados, computamos as métricas de confiabilidade apresentadas a seguir.

Tabela 6 - Modelos de melhor ajuste por grupo

| | G1 | G2 | G3 | G4 | G5 | G6 |
|---------------------------|-------|-------|--------|--------|-------|--------|
| Modelo | Gama | Gama | Gama | Gama | Gama | Gama |
| μ (hora) | 6,08 | 6,06 | 6,92 | 6,58 | 6,25 | 6,46 |
| k | 0,124 | 0,662 | 0,201 | 0,134 | 0,368 | 0,31 |
| Valor da Lk | -221 | -2255 | -14516 | -30793 | -2847 | -10309 |
| PNZ | 0,97 | - | 0,5 | 0,85 | 0,98 | 0,64 |
| Tamanho da amostra | 279 | 343 | 6559 | 19265 | 508 | 2926 |

4.3. Métricas de Confiabilidade

A partir dos modelos apresentados na Tabela 5, estimamos as métricas de confiabilidade apresentadas nas tabelas 7, 8 e 9.

Tabela 7 - Métricas de confiabilidade para aplicações do SO

| | G1 | G2 | G3 | G4 | G5 | G6 |
|-----------------|----|----|--------------------|----------------------|--------------------|--------------------|
| $R(t=8h)$ | - | - | 0,722 | 0,164 | 0,758 | 0,847 |
| $F(t=8h)$ | - | - | 0,278 | 0,836 | 0,242 | 0,152 |
| $\lambda(t)$ | - | - | 0,013/h | 0,026/h | 0,011/h | 0,009/h |
| MTBF (h) | - | - | 1825,88 | 35,43 | 2314,03 | 1265,05 |
| $t(R)$ hora | - | - | $R=0,95$ 0,025h | $R=0,55$ 1,9E-21h | $R=0,95$ 0,053h | $R=0,95$ 0,556h |
| $Bx\%$ hora | - | - | $x=10$ 0,240h | $x=45$ 1,9E-21h | $x=10$ 0,441h | $x=10$ 2,864h |

Tabela 8 - Métricas de confiabilidade para serviços do SO

| | G1 | G2 | G3 | G4 | G5 | G6 |
|-----------------|----------------------|--------------------|----------------------|----------------------|--------------------|--------------------|
| $R(t=8h)$ | 0,344 | 0,947 | 0,218 | 0,414 | 0,739 | 0,415 |
| $F(t=8h)$ | 0,656 | 0,053 | 0,782 | 0,586 | 0,261 | 0,585 |
| $\lambda(t)$ | 0,027/h | 0,006/h | 0,017/h | 0,022/h | 0,014/h | 0,014/h |
| MTBF (h) | 53,730 | 311,123 | 114,518 | 102,384 | 216,784 | 142,082 |
| $t(R)$ hora | $R=0,95$ 1,9E-11h | $R=0,95$ 7,380h | $R=0,40$ 5,0E-12h | $R=0,92$ 5,9E-12h | $R=0,95$ 0,011h | $R=0,57$ 0,000h |
| $Bx\%$ hora | $x=10$ 1,8E-07h | $x=10$ 17,796h | $x=60$ 5,0E-12h | $x=10$ 1,7E-08h | $x=10$ 0,291h | $x=43$ 0,000h |

Tabela 9 - Métricas de confiabilidade para kernel do SO

| | G1 | G2 | G3 | G4 | G5 | G6 |
|-----------------|----|----|--------------------|--------------------|--------------------|--------------------|
| $R(t=8h)$ | - | - | 0,868 | 0,708 | 0,841 | 0,880 |
| $F(t=8h)$ | - | - | 0,132 | 0,292 | 0,159 | 0,120 |
| $\lambda(t)$ | - | - | 0,009/h | 0,013/h | 0,011/h | 0,014/h |
| MTBF (h) | - | - | 411,709 | 1668,893 | 493,424 | 1,0E+99 |
| $t(R)$ hora | - | - | $R=0,95$ 0,928h | $R=0,95$ 0,006h | $R=0,95$ 0,732h | $R=0,95$ 2,834h |
| $Bx\%$ hora | - | - | $x=10$ 4,304h | $x=10$ 0,133h | $x=10$ 3,021h | $x=10$ 6,412h |

Em geral, os resultados obtidos corroboram com a análise de caracterização (ver Tabela 4). Para 8 horas de funcionamento, apenas dois grupos apresentam

confiabilidade, $R(t)$, superior a 50% na categoria de serviços de SO. Além disso, podemos comparar o $R(t=8)$ com outras métricas, como o MTBF. Nem sempre a confiabilidade mais alta apresenta o maior MTBF. Por exemplo, na Tabela 7, o G6 apresenta a maior confiabilidade durante 8 horas de operação, mas o G3 apresenta um MTBF significativamente maior. De maneira semelhante, na Tabela 9, o G3 apresenta maior confiabilidade durante 8 horas do que G4, mas o MTBF deste é mais de quatro vezes maior do que o daquele. Da mesma forma, computamos as métricas de confiabilidade por grupo e apresentamos na Tabela 10. Uma vez que estes resultados consideram todos os tipos de falhas de SO, eles oferecem uma visão mais geral das métricas de interesse. Por exemplo, para qualquer tipo de falha de SO, os sistemas de G2 e G5 são mais confiáveis; mesma conclusão pode ser tirada a partir dos valores de MTBF. Em alguns casos específicos, os valores alvo para $t(R)$ e para a função $Bx\%$ podem não ser encontrados devido aos dados da amostra. Por exemplo, na Tabela 7, o valor alvo de $t(R=0,95)$ não foi possível de alcançar devida a pouca confiabilidade das aplicações de SO do grupo G4.

Tabela 10 - Métricas de confiabilidade por grupo

| | G1 | G2 | G3 | G4 | G5 | G6 |
|--------------|----------|----------|----------|----------|----------|----------|
| $R(t=8h)$ | 0,345 | 0,921 | 0,294 | 0,357 | 0,743 | 0,458 |
| $F(t=8h)$ | 0,655 | 0,079 | 0,706 | 0,643 | 0,257 | 0,542 |
| $\lambda(t)$ | 0,028/h | 0,007/h | 0,017/h | 0,023/h | 0,014/h | 0,015/h |
| $MTBF (h)$ | 52,633 | 282,338 | 101,205 | 82,454 | 187,455 | 127,640 |
| $t(R)$ | $R=0,95$ | $R=0,95$ | $R=0,49$ | $R=0,85$ | $R=0,95$ | $R=0,63$ |
| hora | 3,7E-11h | 3,978h | 0,000h | 7,3E-15h | 0,025h | 0,000h |
| $Bx\%$ | $x=10$ | $x=10$ | $x=51$ | $x=15$ | $x=10$ | $x=37$ |
| hora | 2,7E-07h | 11,452h | 0,000h | 7,3E-15h | 0,398h | 0,000h |

A Figura 1 apresenta as curvas gerais de confiabilidade para os seis grupos. Estas curvas são calculadas através dos modelos da Tabela 6, cujas estimativas estão listadas na Tabela 10. G4 é o único que parece apresentar um comportamento de cauda longa. Podemos observar que a maioria das falhas de G2 diminui depois do período de 8 horas usado no cálculo de $R(t)$ (ver Tabela 10), o que explica sua alta confiabilidade em relação aos demais grupos. Comportamento semelhante pode ser observado no G5, que também apresenta estimativa consistente em todas as categorias de falhas analisadas. Assim, podemos concluir que G2 e G5 apresentam, respectivamente, a melhor experiência de usuário com relação à confiabilidade de SO em termos dos ambientes acadêmicos e corporativos estudados.

5. Conclusão

Verificamos que os serviços de SO foram os principais causadores de falhas de SO nos computadores pesquisados. A taxa de falha de aplicações de SO e *kernel* são baixas e semelhantes. Estes são achados importantes, pois em ambientes corporativos os usuários finais não lidam com serviços de SO diretamente, sendo estes mantidos pelos administradores. Estes resultados sugerem que a política de administração de serviços de SO é um fator importante para a confiabilidade de software como um todo. Também, há evidência de que falhas de *kernel* ocorreram mais em ambientes corporativos do que acadêmicos. A suspeita é que este fato esteja relacionado ao tempo de operação do SO.

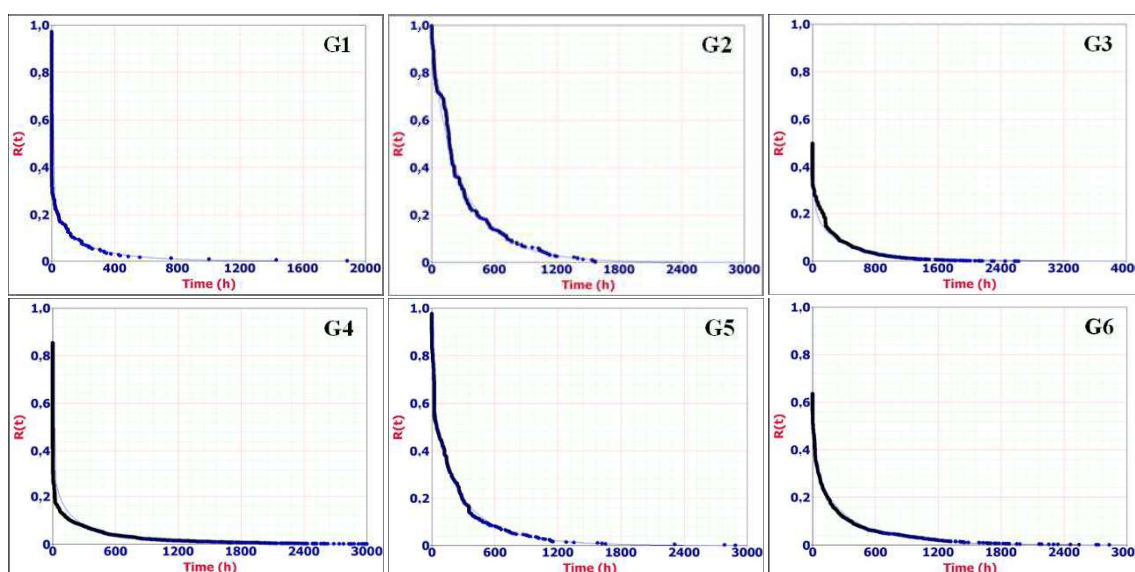


Figura 1 - Curva de confiabilidade por grupo

Comparando a taxa de falha da categoria *kernel* do G3 com G4, G5 e G6 (ver Tabela 9), observamos que a probabilidade de falha por hora em G3, $\lambda(t)$, é menor que nos outros grupos. Considerando que o tempo médio de operação do SO em laboratórios de ensino deva ser menor do que em ambientes corporativos, maiores seriam as chances de falhas de *kernel* neste último, já que uma maior duração de operação ininterrupta aumentaria as chances de propagações de erro no *kernel*. Por fim, estimamos as funções densidades de melhor ajuste para as três categorias de falhas investigadas. Estes resultados (ver Tabela 5), obtidos com base em dados reais, proporcionam modelos que podem ser usados em diferentes estudos na área de modelagem e simulação de confiabilidade de sistemas de computação.

Referências

- ABB. (2002). “The red line holding technologies together”, <http://www.abb.com/cawp/seitp255/359bcb94e511843bc1256c76005a8813.aspx>.
- ANSI/IEEE Standard Glossary of Software Engineering Terminology. (1990).
- Bott, E. (2013). “Latest OS share data shows Windows still dominating in PCs”, <http://www.zdnet.com/latest-os-share-data-shows-windows-still-dominating-in-pcs-7000013351/>
- Chou, A., Yang, J., Chelf, B., Hallem, S., and Engler, D. (2011) “An Empirical Study of Operating Systems Errors”, Proc. of the 18th ACM Symposium on Operating Systems Principles, p. 73 – 88.
- Conover, W.J. (1999). Practical Nonparametric Statistics, John Wiley & Sons, 3rd ed.
- Cullen, C. and Frey, H. C. (1999) Probabilistic Techniques in Exposure Assessment: A Handbook for Dealing with Variability and Uncertainty in Models and Inputs. Springer.
- Ganapathi, A. and Patterson, D. (2005). “Crash Data Collection: A Windows Case Study”, Proc. Of International Conference on Dependable Systems and Networks, p. 280-285.

- Ganapathi, A., Ganapathi, V. and Patterson, D. (2006) “Windows XP *Kernel* Crash Analysis”, Proc. of the 20th conference on Large Installation System Administration, p. 149-159.
- Kalyanakrishnam, M., Kalbarczyk, Z. and Iyer, R. (1999). “Failure Data Analysis of a LAN of Windows NT Based Computers”, Proc. of the 18th IEEE Symposium on Reliable Distributed Systems, p. 178-187.
- Leveson, N. G. and Turner, C. S. (1993). “An Investigation of the Therac-25 Accidents”, in Computer, p. 18-41.
- Li, P.L., Ni, M., Xue, S., Mullally, J.P., Garzia, M., Khambatti, M. (2008). “Reliability Assessment of Mass-Market Software: Insights from Windows Vista”, Proc. of the 19th International Symposium on Software Reliability Engineering, p. 265-270.
- Lyu, M. R. (2007). “Software Reliability Engineering: A Roadmap”, in Future of Software Engineering, p.153 - 170.
- Microsoft. (2007). “Reliability Analysis Component”, [http://technet.microsoft.com/en-us/library/cc774636\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc774636(v=ws.10).aspx)
- Mood, A.M., Graybill, F.A. and Boes, D. C. (1974). Introduction to the Theory of Statistics, McGraw-Hill, 3rd ed.
- Murphy, B. (2008). “Reliability Estimates for the Windows Operating System”, <http://www.dcl.hpi.uni-potsdam.de/meetings/mshpsummit/slides/brendan.murphy.pdf>
- Rayner, J. C. W., Thas O. and Best, D. J. (2009). Smooth Tests of Goodness of Fit: Using R, John Wiley & Sons, 2nd ed.
- Reliability HotWire. (2008). “Life Data Analysis with Zero-Time (Out-of-The-Box) Failures”, <http://www.weibull.com/hotwire/issue83/hottopics83.htm>.
- StatCounter Global Stats. (2013). “Top Operating Systems Worldwide”.
- Swift, M. M., Bershad, B.N., Levy, H.M. (2003). “Improving the reliability of commodity operating systems”, Proc. of the 19th ACM Symposium on Operating Systems Principles, p. 207-222.
- Tanenbaum, A., Herder, J. N. and Bos, H. (2006). “Can We Make Operating Systems Reliable and Secure?”, in Computer, p. 44-51.
- Trivedi, K.S. (2001). Probability and Statistics with Reliability, Queueing, and Computer Science Applications, Wiley-Interscience, 2nd ed.
- Xu, J., Kalbarczyk, Z. and Iyer, R. K. (1999). “Networked Windows NT System Field Failure Data Analysis”, in Pacific Rim International Symposium on Dependable Computing, p. 178 - 185.
- Yuan, D. and Zhang, C. (2011). “Evaluation Strategy for Software Reliability based on ANFIS”, in International Conference on Electronics, Communications and Control, p. 3738-3741.