Um Estudo Exploratório Sobre Padrões de Falhas de Sistemas Operacionais

Caio Augusto R. dos Santos, Rivalino Matias Jr. (orientador)

Faculdade de Computação – Universidade Federal de Uberlândia (UFU)
Uberlândia – MG – Brazil

{caioarsantos,rivalino}@ufu.br

Abstract. *Given the intrinsic dependency of user programs on the operating system (OS) software, OS failures can severely impact even the most reliable applications. This research carried out an exploratory study on OS failure patterns, based on 7,007 real failure records collected from different computers running a mass-market operating system. We introduced an OS failure pattern detection protocol, which allowed us to detect and characterize failure patterns that showed consistent across different computers from the same and varied workplaces investigated.*

Resumo. *Devido à dependência intrínseca de programas de usuário em relação ao sistema operacional (SO), falhas no sistema operacional podem afetar, severamente, até mesmo as aplicações mais confiáveis. Esta pesquisa realizou um estudo exploratório sobre padrões de falha de SO, com base em 7.007 registros de falha reais coletados de diferentes computadores executando um sistema operacional de mercado. Foi desenvolvido um protocolo de detecção de padrões que permitiu detectar e caracterizar padrões de falha de SO que se mostraram consistentes em diferentes computadores dos mesmos e de variados ambientes de trabalho investigados.*

1. Problema de Pesquisa e Motivação

Devido ao aumento da dependência dos seres humanos por sistemas computacionais, uma falha em suas operações pode acarretar desde simples inconvenientes até grandes desastres. Atualmente, a preocupação com a confiança no funcionamento de sistemas computacionais não é exclusiva de sistemas críticos e altamente especializados, mas também ocorre com sistemas de propósito geral. Por exemplo, o sistema de navegação utilizado pela marinha norte-americana executa sob um sistema operacional convencional, o Microsoft Windows [Li *et al.* 2008]. Esse cenário tem se tornado cada vez mais frequente, no qual sistemas computacionais especializados, com requisitos de alta confiabilidade, utilizam partes de *software* de propósito geral, em especial sistemas operacionais. Nesses casos, de nada adianta ter aplicações que, garantidamente, possuem alta confiabilidade, se estas executam sob um sistema operacional que não oferece níveis compatíveis de confiança em seu funcionamento.

Nesse sentido, este trabalho teve por objetivo contribuir para uma melhor compreensão sobre o comportamento de falhas que ocorrem no sistema operacional, investigando a existência de padrões em suas ocorrências. Também, examinou-se a existência de dependência entre essas ocorrências, uma importante propriedade no

contexto da modelagem e avaliação quantitativa da confiabilidade de *software*. De acordo com [Goseva-Popstojanova e Trivedi 2000], uma das principais suposições adotadas pela maioria dos modelos de confiabilidade de *software* é a de que falhas são eventos independentes. Ainda de acordo com os autores, essa suposição de independência é, muitas vezes, realizada para adequar tais modelos para formas matematicamente tratáveis e, desse modo, simplificar a computação dos parâmetros dos modelos. Contudo, tais suposições podem não representar a real forma de ocorrência das falhas analisadas, prejudicando sobremaneira a avaliação da confiabilidade do *software*.

Em primeiro lugar, buscou-se responder à seguinte pergunta de pesquisa: **Falhas de sistema operacional seguem padrões**? Foram encontradas fortes evidências da existência de padrões nas amostras de falha analisadas, de modo que a pesquisa avançou abordando demais questões de interesse, tais como: "Quais componentes do SO mais contribuíram para a composição desses padrões?" "Quais falhas foram recorrentes nos padrões identificados?" "Quais tipos de correlação entre falhas estão presentes nos padrões identificados?" As causas de falha do SO também foram consideradas nesse trabalho, o que possibilitou responder outra importante pergunta de pesquisa: **As causas de falha do SO também seguem padrões**? Encontramos fortes evidências que indicam padrões consistentes também entre as causas de falha do SO, nos permitindo avançar buscando respostas para outros questionamentos correlatos.

Até onde sabemos, este foi o primeiro estudo a propor um protocolo para detectar padrões de falha em sistemas operacionais. A identificação de padrões em falhas de SO é importante, tendo repercussão em diferentes áreas: *i*) a *dependabilidade* do sistema computacional como um todo pode ser melhorada, pois o conhecimento dos padrões de falha e suas correlações permite a criação de mecanismos de prevenção de falha mais eficazes; *ii*) o processo de *teste de software* se beneficia na medida em que pode concentrar esforços nas causas de falha que, individualmente ou combinadas, acarretam maior impacto na confiabilidade do sistema; *iii*) a *modelagem de sistemas* pode oferecer resultados mais realísticos, levando-se em conta as possíveis interações entre falhas e causas de falha que são reveladas nos padrões de falha detectados.

2. Materiais e Métodos

Dado que essa pesquisa focou na detecção de padrões de falha de SO, as amostras de falha analisadas foram obtidas de sistemas computacionais reais. O Microsoft Windows 7 (Win7) foi escolhido para este trabalho, visto que é atualmente o sistema operacional mais utilizado (ex. [NetMarketShare 2017]), além de fornecer registros detalhados de falhas por meio do *Reliability Analysis Component* (RAC) [Microsoft 2007]. A amostra de trabalho foi composta de 7.007 registros de falha de SO obtidos de 566 computadores executando o Win7. Esses registros foram obtidos de quatro ambientes de trabalho com diferentes perfis de utilização (G1, G2, G3 e G4). Os dados de falha dos grupos G1 e G2 foram provenientes de computadores de uma mesma universidade. G1 contém falhas de computadores de laboratórios de ensino de cursos de graduação e G2 reúne registros de falha coletados de computadores do departamento administrativo da universidade. G3 possui registros de falha obtidos de computadores de um ambiente corporativo. G4 é o único grupo heterogêneo, ou seja, contém registros de falha de SO de diferentes ambientes de trabalho (corporativo, acadêmico, doméstico). Maiores detalhes sobre esses grupos podem ser obtidos em [Dos Santos e Matias 2016b] (ver pág. 20-21).

Diversos trabalhos (ex. [Swift *et al.* 2003]) que investigaram a confiabilidade de SO analisaram apenas falhas que ocorreram no núcleo (*kernel*) do SO. Em [Matias *et al.* 2013] discutiu-se a importância de se considerar não apenas falhas no espaço do *kernel*, mas também falhas de SO que ocorrem no espaço do usuário, dado que SOs modernos possuem partes executando em ambos os níveis; portanto, trabalhos que limitam suas análises em apenas um nível não avaliam a real confiabilidade do SO. Por isso, este estudo considerou falhas de SO que ocorreram tanto no espaço do *kernel* quanto no espaço do usuário. Utilizando a abordagem de classificação de falhas de SO proposta em [Matias *et al.* 2013], as falhas da amostra de trabalho foram organizadas em três categorias: Aplicação de SO (SO_{APP}), Serviço de SO (SO_{SVC}) e *Kernel* (SO_{KNL}). As duas primeiras contêm registros de falha de programas do SO que executam no espaço do usuário, enquanto a terceira contém falhas de subsistemas do SO que executam no espaço do *kernel*. Todas as falhas foram agrupadas em 113 tipos de acordo com o campo *ProductName* dos respectivos registros do RAC.

A categoria SO_{SVC} apresentou a maior quantidade de eventos de falha (4.248 falhas). Com o propósito de compreender o motivo dessa prevalência, os registros de falha dessa categoria foram examinados individualmente e observou-se que 23 dos seus 55 tipos de falha estão relacionados com o serviço *Windows Update* (SWU). A quantidade de eventos de falha dos 23 tipos correspondeu a 87,97% (3.737 falhas) do número total de falhas (4.248 falhas) da categoria SO_{SVC}. Portanto, essas falhas representam 53,33% de todas as falhas de SO da amostra de trabalho. Cada um dos 23 tipos de falha do SWU diz respeito a um tipo de atualização de *software* que falhou enquanto aplicada a um dado componente de *software*. Por exemplo, o tipo de falha do SWU identificado como AW7 (atualização do Windows 7) representa atualizações de *software* aplicadas aos arquivos do sistema operacional Win7. Note que essa falha é registrada quando o serviço de atualização (SWU) falha ao tentar atualizar arquivos (ex. bibliotecas) do SO. O mesmo foi observado para atualizações de diferentes tipos de *software*. Também, foi possível examinar as causas dessas falhas por meio dos *Error Codes* presentes no campo *Message* dos registros do RAC. *Error Code* é um valor hexadecimal associado com um evento de falha, sendo usado para identificar a causa de uma falha no Win7. A interpretação dos *Error Codes* foi realizada por meio da documentação oficial do fabricante do SO investigado. A Figura 1 apresenta uma visão geral da estratificação da amostra de trabalho. Primeiramente, as falhas foram divididas em categorias, depois em tipos, e para o tipo serviço *Windows Update* (SWU) foram criados subtipos e, por fim, as causas das falhas foram identificadas por meio dos respectivos *Error Codes*.

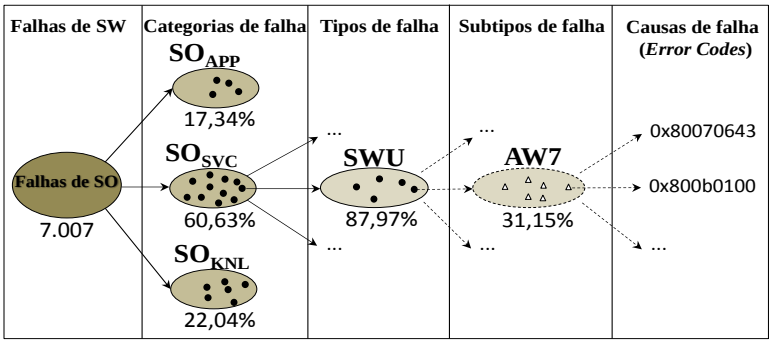

Figura 1. Estratificação da amostra de falhas de sistema operacional analisada.

Pela ausência de trabalhos na literatura sobre padrões de falha de SO, foi necessário definir uma abordagem própria, na forma de um protocolo (ver pág. 35-46 de [Dos Santos e Matias 2016b]), para detectar e caracterizar esses padrões na amostra de trabalho. Uma taxonomia própria foi criada (ver pág. 33-35 de [Dos Santos e Matias 2016b]) para definir os vários conceitos utilizados no protocolo proposto. Neste trabalho, *padrões de falha de SO* são combinações de eventos de falha que se repetem, sistematicamente, na amostra de trabalho.

No protocolo proposto, primeiramente as falhas de SO que ocorreram no maior número de grupos da amostra são identificadas. Essas falhas são utilizadas como referência para a detecção de possíveis padrões de falha, sendo chamadas de *falhas de referência*. A partir das falhas de referência foram identificados os eventos de falha de SO que ocorreram antes e/ou depois delas, ou seja, que podem estar relacionadas com as suas causas ou efeitos. As falhas antes e/ou depois de uma falha de referência são delimitadas pelo tempo de suas ocorrências, ou seja, apenas as falhas que ocorreram em um determinado intervalo de tempo antes e/ou depois da falha de referência são consideradas. Esse *intervalo de busca* é configurável no algoritmo de busca. A associação das falhas de referência com suas *falhas antes* e/ou *falhas depois* é denominada *combinação de eventos*. Quando uma combinação de eventos de falha de SO é observada, de forma consistente, em diferentes computadores do mesmo grupo e de grupos diferentes, então se considera que essa combinação de eventos seja, possivelmente, um padrão de falha de SO. Apenas as combinações de eventos de falha de SO que apresentam maior consistência na amostra de trabalho foram consideradas padrões de falhas neste estudo. A Figura 2 ilustra essas etapas. Abordagem similar foi adotada para investigar padrões nas causas das falhas de SO analisadas, em especial das falhas do serviço *Windows Update* (SWU), haja vista sua prevalência na amostra de trabalho. De forma correspondente à abordagem anterior, primeiramente as causas das falhas de referência são identificadas e posteriormente identificam-se as causas das falhas imediatamente antes e/ou imediatamente depois dessas causas mais recorrentes.

3. Resultados

Foram encontrados 45 padrões de falhas de SO. Esses padrões ocorreram em no mínimo três dos quatro grupos de computadores analisados neste trabalho. Dada a diversidade dos grupos e a quantidade de computadores pesquisados, considerou-se que tais achados são padrões genuínos de falhas do Win7. Verificou-se que AW7 foi o subtipo de falha do *Windows Update* que mais contribuiu para a composição desses padrões. Este subtipo representa falhas em atualizações de *software* aplicadas exclusivamente aos componentes do sistema operacional. Posteriormente, as causas de falha desse subtipo foram analisadas. Observou-se que a maioria dessas falhas está relacionada com fatores que o serviço *Windows Update* depende (ex. espaço em disco e o serviço RPC - *Remote Procedure Call*). O gerenciamento proativo desses fatores objetivando aumentar a confiabilidade do sistema operacional analisado é uma constatação desse estudo. Com respeito às falhas relacionadas com aplicações de SO (SO_{APP}), os resultados indicaram a prevalência de três programas, nesta ordem de importância: explorer.exe, mscorsvw.exe e rundll32.exe. Esses programas fazem parte da infraestrutura do Win7 e executam suas tarefas no espaço do usuário. Com relação às falhas de *kernel* (SO_{KNL}), os resultados obtidos neste estudo corroboram e são consistentes com os resultados reportados em

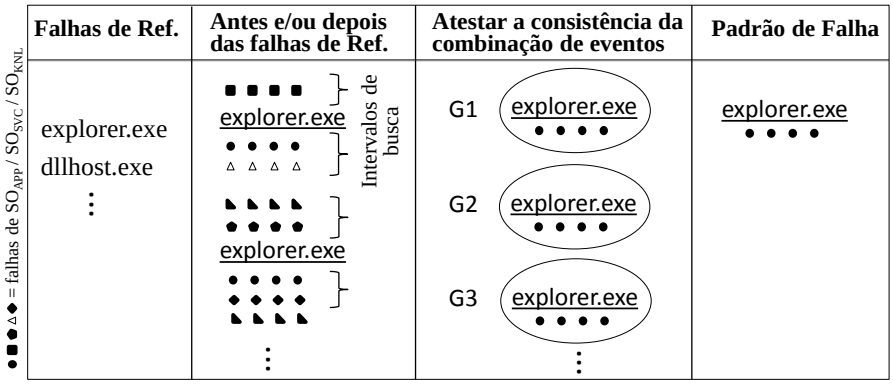

Figura 2. Visão geral das etapas do protocolo de detecção de padrões de falha.

outros trabalhos, ao observar que os *device drivers* foram a principal causa de falha de *kernel* na amostra de trabalho, em especial os *device drivers* de gerenciamento de energia e vídeo. Tendo em vista que as falhas de atualização, relacionadas com o serviço *Windows Update*, foram prevalentes na amostra de trabalho, verificou-se que a maioria dessas falhas teve como causa principal o mau funcionamento do *.Net Framework*. Este achado é um exemplo original de causalidade entre falhas de *software*, relação esta ainda pouco abordada na literatura de engenharia de confiabilidade de *software*.

Por fim, foram encontrados padrões de ocorrência entre diferentes falhas de SO. Por exemplo, falhas que se repetiram após um dado intervalo de tempo (ex. 1 hora, 4 horas) ou durante um intervalo muito curto de tempo (menor do que 1 segundo); também houve casos em que diferentes falhas ocorreram ao mesmo tempo como consequência de uma causa raiz comum. Esses e outros padrões observados são discutidos em detalhes em [Dos Santos e Matias 2016b] (ver pág. 64-74).

4. Conclusão da Pesquisa

Os resultados obtidos indicaram que as falhas de *kernel* representaram apenas 17,34% de todas as falhas de SO da amostra de trabalho. Portanto, a fim de serem mais efetivos, estudos em confiabilidade de SO não deveriam se concentrar apenas em falhas de *kernel*, como tem sido a norma na literatura. Os resultados também mostram fortes evidências de correlação entre falhas; propriedade estatística muitas vezes negligenciada por estudos em confiabilidade de *software*, os quais assumem que falhas de *software* ocorrem de forma independente. As evidências sugerem que esta suposição de independência não é verdadeira para as falhas do *software* de SO investigado neste trabalho. Esse resultado tem implicações importantes no tocante às suposições adotadas em modelos de confiabilidade de *software* de forma geral, em especial aqueles aplicados na análise de confiabilidade de SO. Apesar dos resultados quantitativos deste trabalho serem específicos para o Win7, os algoritmos (ver pág. 35-46 de [Dos Santos e Matias 2016b]) que compõem o protocolo proposto são genéricos suficiente para serem aplicados a qualquer SO de interesse.

5. Contribuição para a Literatura

Os resultados deste estudo foram a base para quatro artigos. O primeiro, [Dos Santos e Matias 2015], ganhou o prêmio de *Best Paper* na categoria "Sistemas Operacionais" do Simpósio Brasileiro de Engenharia de Sistemas Computacionais em 2015. O segundo

artigo, [Dos Santos e Matias 2016a], uma extensão do primeiro, foi publicado no principal periódico internacional de sistemas operacionais, o ACM SIGOPS Operating System Review. O terceiro artigo, [Dos Santos e Matias 2017a], foi publicado na edição 2017 do ACM Symposium on Applied Computing (*OS Track*). Por fim, um quarto artigo, [Dos Santos e Matias 2017b], foi publicado no periódico internacional Elsevier Journal of Systems and Software.

References

Dos Santos, C.A.R. e Matias, R. (2015) "An Empirical Study on Failure Causes in a Commercial Off-the-Shelf Operating System", Proc. 5th Brazilian Symp. on Computing Systems Engineering.

Dos Santos, C.A.R. e Matias, R. (2016a) "Exploratory Analysis on Failure Causes in a Mass-Market Operating System", ACM SIGOPS Operating Systems Review - Special Topics, 50(1), pp. 18-30.

Dos Santos, C.A.R. e Matias, R. (2016b) "Um Estudo Exploratório Sobre Padrões de Falhas de Software de Sistemas Operacionais", Dissertação (Mestrado em Ciência da Computação). Universidade Federal de Uberlândia, Uberlândia-MG.

Dos Santos, C.A.R. e Matias, R. (2017a) "An Empirical Study on Patterns of Failure Causes in a Mass-Market Operating System", Proc. 32th ACM Symposium on Applied Computing. *No prelo*.

Dos Santos, C.A.R. e Matias, R. (2017b) "Failure Patterns in Operating Systems: An Exploratory and Observational Study", Elsevier Journal of Systems and Software.

Goseva-Popstojanova, K. e Trivedi, K.S. (2000) "Failure Correlation in Software Reliability Models", IEEE Trans. on Reliability, 49(1), pp. 37-48.

Li, P.L., Ni, M., Xue, S., Mullally, J.P., Garzia, M. e Khambatti, M. (2008) "Reliability Assessment of Mass-Market Software: Insights from Windows Vista", Proc. 19th Int'l Symp. on Software Reliability Engineering, pp. 265-270.

Lyu, M. (2007) "Software *Reliability* Engineering: A Roadmap", Proc. 29th Int. Conf. Softw. Eng., Future Softw. Eng., pp.153 -170.

Matias R., Oliveira G. e Araújo L. (2013) "Operating System Reliability from the Quality of Experience Viewpoint: An Exploratory Study", Proc. 28th ACM Symposium on Applied Computing, pp.1644-1649.

Microsoft (2007) "Reliability Infrastructure", https://technet.microsoft.com/pt-br/library/cc732712(v=ws.10).aspx, Março.

NetMarketShare (2017) "Desktop Operating System Market Share", http://www.netmarketshare.com/operating-system-market-share.aspx?qprid=10&qpcustomd=0, Março.

Swift, M. M., Bershad, B. N. e Levy, H. M. (2003) "Improving the Reliability of Commodity Operating Systems", Proc. 19th ACM Symposium on Operating Systems Principles, pp. 207-222.