Arquitetura para Prevenção e Tolerância a Falhas em Sistemas-de-sistemas Virtuais

Francisco Henrique Cerdeira Ferreira[1], Rodrigo Pereira dos Santos[1]

[1]Programa de Pós-Graduação em Informática
Universidade Federal do Estado do Rio de Janeiro (UNIRIO) - Rio de Janeiro, Brasil

{francisco.ferreira,rps}@uniriotec.br

Abstract. *The interest in systems-of-systems (SoS) has been increasing as this approach allows the development of systems with capabilities that are possible only through a set of constituent systems working together. However, these types of systems run under high uncertainty, since constituent systems are independent and their behavior over time is unknown. Thus, a SoS must be able to readapt itself in the face of situations that cause failures. To this end, this PhD work proposes a fault avoidance and fault tolerance architecture for dealing with misbehavior of constituent systems in SoS.*

Resumo. *O interesse em sistemas-de-sistemas (SoS) tem aumentado uma vez que esta abordagem permite a construção de sistemas com capacidades que são possíveis somente por meio de um conjunto de sistemas constituintes trabalhando juntos. No entanto, esses tipos de sistemas operam sob grande incerteza, uma vez que seus sistemas constituintes são independentes e seu comportamento ao longo do tempo é desconhecido. Assim, um SoS deve ser capaz de se readaptar diante de situações que causam falhas. Este trabalho de doutorado propõe uma arquitetura para prevenção e tolerância a falhas para lidar com o mau comportamento de sistemas constituintes em SoS.*

1. Introdução

Nos últimos anos, os sistemas-de-sistemas (SoS) vêm ganhando popularidade em diversas áreas como militar, logística, transporte, saúde, dentre outras. SoS são arranjos de sistemas, chamados de constituintes, integrados em um sistema maior, capaz de prover capacidades possíveis apenas com essa integração [Department of Defense, 2008]. Considere, por exemplo, um sistema que se comunica com outro a partir de uma *Application Programming Interface (API*), que provê dados sobre a previsão do tempo e com outra aplicação capaz de fornecer informações em tempo real sobre o trânsito. Esse sistema, desenvolvido a partir da integração de outros sistemas, seria capaz de identificar potenciais pontos de alagamento em uma cidade e indicar rotas de trânsito para que os motoristas evitem locais de risco.

Dentre os desafios relacionados ao desenvolvimento de SoS, pode-se citar a sua arquitetura dinâmica, que é influenciada pelo nível de liberdade dos sistemas constituintes, que podem deixar o SoS sem aviso prévio e alterar suas regras e nível de contribuição de acordo com o interesse de seus desenvolvedores. Com isso, SoS operam em um ambiente de incertezas, e um dos principais desafios no desenvolvimento desses sistemas é a detecção e recuperação de falhas decorrentes do comportamento não previsível dos sistemas constituintes [Andrews *et al*., 2013]. Tais falhas podem representar a descontinuidade ou alteração de uma operação realizada pelo SoS,

colocando em risco o cumprimento das missões para o qual o SoS foi projetado. Diante disso, estudos surgiram com o objetivo de aumentar a confiabilidade em SoS, mas esse ainda é um problema em aberto [Eddaoui *et al.,* 2018] e, como forma de contribuir para essa área, esse trabalho apresenta uma arquitetura para prevenção e tolerância a falhas em SoS.

O restante do artigo está organizado da seguinte forma: a Seção 2 traz a apresentação do problema; a Seção 3 descreve a solução proposta nesta pesquisa; a Seção 4 apresenta o projeto para a avaliação da solução; na Seção 5, são relatadas as atividades realizadas; e a Seção 6 conclui este artigo.

2. Apresentação do Problema

Segundo Maier (1996), SoS são caracterizados pela (i) independência operacional de seus constituintes, (ii) independência gerencial constituintes, (iii) distribuição geográfica, (iv) desenvolvimento evolucionário e (v) comportamento emergente. SoS também são classificados quanto à coordenação dos elementos, conforme descrito na Tabela 1.

Tabela 1. Tipos de SoS [Dahmann e Baldwin, 2008]

Tipo de SoS	Definição
Direcionados	Criados e gerenciados para cumprir uma missão global. Os sistemas constituintes estão subordinados a uma autoridade central.
Reconhecidos	Objetivos, recursos e uma gerência designada são reconhecidos por todas as partes envolvidas. Os constituintes mantêm independência em termos de objetivos, financiamento, desenvolvimento e sustentação.
Colaborativos	Os sistemas constituintes possuem ciência de sua participação em um SoS e contribuem voluntariamente para atingir um objetivo global. Não existe uma autoridade central e há um acordo mútuo entre as partes para colaboração.
Virtuais	Não existe uma autoridade central e nem acordo entre as partes. É um ambiente descoordenado onde os sistemas constituinte não têm ciência de que contribuem com uma missão de um SoS.

O presente trabalho tem como objeto de estudo os SoS Virtuais, descrito na Tabela 1. Nesse tipo de SoS, além da autonomia dos sistemas constituintes (para se autorregular, alterar os níveis de colaboração ou mesmo tornar-se indisponíveis por vontade própria), existem outros fatores que podem causar falhas. Por exemplo, um determinado sistema constituinte do SoS sofrer ataques, flutuações na rede (i.e., atrasos, *throughput*, tempo de resposta) ou mesmo falha ou sobrecarga no hardware. Nesses casos, é necessário que o SoS se readapte para permanecer operando.

Estudos propõem soluções para tolerância a falhas durante a fase de modelagem do SoS [Andrews *et al*., 2013; Garro e Tundis, 2015]. Tais abordagens exigem que os candidatos a sistemas constituintes sejam definidos já no início do projeto. Esse tipo de abordagem é adequado para SoS com algum tipo de autoridade central, responsável por coordenar ações e que tenha influência sobre os sistemas constituintes, ou com SoS colaborativos, onde existe um acordo mútuo. Todavia, essas soluções não são

convenientes para SoS Virtuais, onde os constituintes podem não possuir ciência da sua participação para atingir objetivos globais e seu comportamento ao longo do tempo não é evidente.

Devido à arquitetura dinâmica, garantir que um SoS opere de maneira confiável requer mecanismos de prevenção e tolerância a falhas em tempo de execução [Dabholkar *et al*., 2012]. Isso pode ser alcançado, por exemplo, por meio da substituição de um sistema constituinte que esteja ocasionando falhas no SoS por outro com as mesmas características. Na web, por exemplo, o site *ProgrammableWeb*[1] possui uma coleção de mais de 20 mil[2] APIs relacionadas a diversos domínios, tais como previsão do tempo, mapas, comércio eletrônico e saúde. Algumas APIs fornecem o mesmo tipo de serviços e, portanto, são potenciais candidatos a compor um SoS que os utilize.

Hamer (2011) sugere que uma possível solução para a seleção de sistemas constituintes em tempo real é a introdução de um mediador na arquitetura do SoS, que recebe as requisições e as encaminha para provedores de serviço em condições de atendê-las de acordo com os requisitos definidos. No entanto, essa abordagem apresenta uma vulnerabilidade importante que é a adição de um ponto de falha, já que se o elemento mediador se tornar indisponível, haverá perda de comunicação entre os sistemas constituintes, visto que as mensagens passam por esse elemento obrigatoriamente. Além disso, adiciona um potencial ponto de gargalo, pois o mediador deverá receber a requisição, processá-la e encaminhá-la ao sistema requisitado.

Técnicas de predição de falhas em SoS também têm sido utilizadas [Wang *et al*., 2018]. Nesse caso, são utilizados dados de comportamentos anteriores do SoS para a identificação de falhas futuras. Por conta da potencial degradação de SoS Virtuais devido a comportamentos não previstos dos sistemas constituintes, faz-se necessário o monitoramento em tempo real de métricas associadas aos constituintes que influenciam a confiabilidade do SoS de forma geral. Contudo, apesar de diversos fatores afetarem o desempenho de sistemas constituintes, nem todos podem ser medidos de forma direta, como utilização de CPU e memória, por exemplo, já que é necessário acesso direto ao sistema operacional onde estão instalados os sistemas constituintes. Todavia, métricas relacionadas às condições de rede são possíveis de serem obtidas de maneira não intrusiva e podem indicar degradações nos sistemas constituintes. Por exemplo, um tempo de resposta alto, além de indicar possíveis instabilidades de rede na qual os sistemas constituintes estão conectados, pode também significar alta utilização de CPU.

Contudo, alguns SoS exigem mais do que a disponibilidade de seus constituintes, mas também que eles atendam a outros requisitos referentes a desempenho, econômicos e sociais. Por isso, a mera substituição de sistemas constituintes não garante que o SoS opere de acordo com todos os requisitos definidos. Nesse sentido, o objetivo desse trabalho é propor uma arquitetura tolerante a falhas para SoS Virtuais. Esta arquitetura prevê a substituição de um sistema constituinte falho por outro que cumpra os requisitos necessários para compor o SoS. Com isso, pretende-se minimizar os efeitos negativos decorrentes de falhas em sistemas constituintes, como a perda de capacidade do SoS em entregar uma ou mais funcionalidades, ou mesmo o colapso do sistema. A Seção 3 descreve a arquitetura proposta.

---

[1] https://www.programmableweb.com

[2] Dado de 29 de janeiro de 2019.

3. Proposta de Solução

A arquitetura proposta contempla a inclusão de um elemento no SoS chamado Supervisor, que possui dois módulos: um de tolerância a falhas, para lidar com situações de falhas imediatas; e outro de prevenção a falhas, para situações que podem incorrer em falhas.

O Supervisor é responsável por registrar quais os serviços dos sistemas constituintes que são necessários para que o SoS cumpra sua missão. Além disso, o Supervisor recebe mensagens de *feedback* que descrevem a qualidade da interação entre os sistemas constituintes. Para isso, os constituintes deverão possuir uma interface com o Supervisor A Figura 1 apresenta uma visão preliminar da proposta.

Em um primeiro momento, é necessário que sejam registrados no Supervisor os candidatos a sistemas constituintes. Como os parâmetros de integração podem variar entre os sistemas constituintes, é necessário também que o Supervisor registre tais parâmetros (e.g., *endpoints*, variáveis, autenticação) para cada sistema constituinte.

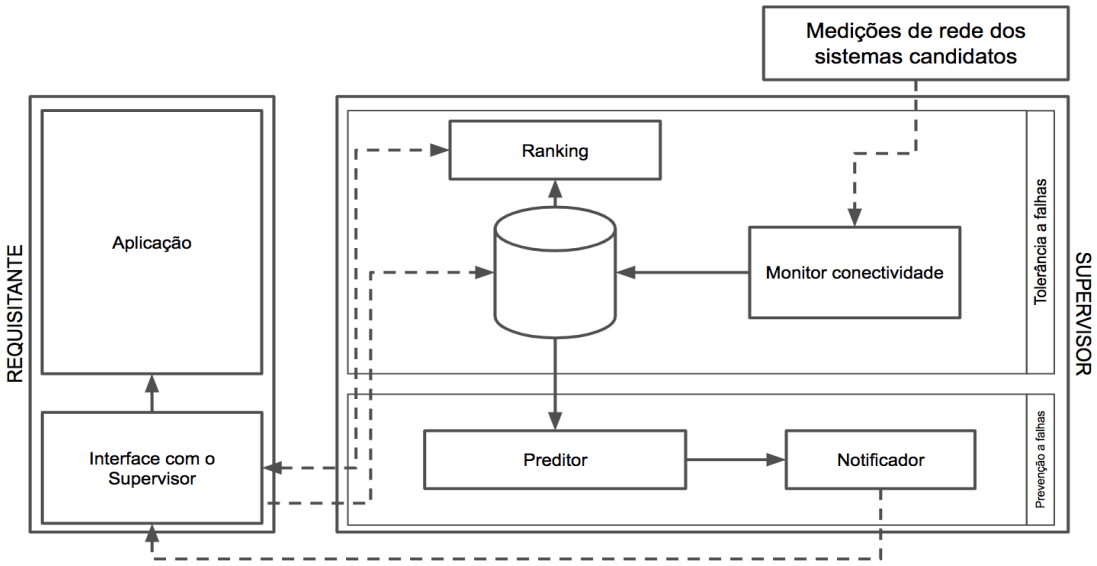

Figura 1. Arquitetura conceitual preliminar

O mecanismo de *feedback* foi proposto para capturar uma medida de qualidade de interação entre os constituintes do SoS. Essa medida deve representar fatores que influenciem na confiabilidade do SoS, que ainda estão sendo identificados nesta pesquisa de doutorado. O *feedback* é necessário pois descreve a qualidade da interação entre os sistemas constituintes que pode variar em função do serviço provido pela Internet, que é o de melhor-esforço, no qual não existe garantia de entrega de pacotes e nem uma preocupação com flutuações na rede [Shenker, 1995]. Por esse motivo, medidas de qualidade de rede como *throughput*, latência, variação do atraso e perda de pacotes podem variar de acordo com a localização dos sistemas constituintes. Sendo assim, o *feedback* ajudará a definir qual a melhor alternativa para um requisitante baseado em características de comunicação anteriores entre requisitante e requisitado. Um *ranking* será montado com as melhores opções de constituintes para o SoS.

O Supervisor irá monitorar medidas de conectividade direta entre o próprio Supervisor e os constituintes. Isso servirá para que o Supervisor tenha medidas de sistemas constituintes mesmo que eles não estejam se comunicando. Apesar dessas medidas não indicarem a qualidade da interação entre os constituintes, pode indicar um

eventual comportamento de falha, possibilitando ao Supervisor, a partir do módulo de prevenção de falhas, notificar o requisitante sobre um comportamento indesejável. Uma vez identificada uma situação de falha, a aplicação requisitante poderá buscar uma nova alternativa pela consulta ao *ranking* mantido pelo Supervisor. Nessa arquitetura, o Supervisor não representa um ponto de falha para o SoS ainda que ocorra problemas com o mesmo, já que a comunicação entre os elementos do SoS é feita diretamente. Neste caso, o Supervisor é responsável pela indicação de constituintes substitutos em uma situação de falha.

É importante destacar que confiabilidade em SoS possui um espectro mais amplo e não se limita a fatores de desempenho. É possível citar, além dos elementos mencionados no presente trabalho, outros aspectos, como os econômicos (e.g., custos de utilização de um constituinte) e sociais (e.g., familiaridade do usuário com recursos providos pelo constituinte, tais como recursos de mapas, onde a visualização pode variar de acordo com o sistema constituinte). Nesse sentido, outros elementos estão sendo verificados por meio de um mapeamento sistemático da literatura para que sejam posteriormente incorporados à arquitetura proposta.

4. Projeto de Avaliação da Solução

Pretende-se conduzir experimentos na plataforma *PlanetLab*[3], com o objetivo de tratar questões técnicas que podem levar a problemas de confiabilidade. O PlanetLab é uma plataforma com mais de 1.500[4] computadores distribuídos por todos os continentes e permite a condução de testes e experimentos em escala global. A principal vantagem de utilizar o *PlanetLab* para pesquisas é que os computadores estão geograficamente distribuídos e fazem parte de diferentes redes com regras próprias de roteamento, qualidade de serviço e segurança. Com isso, é possível simular um ambiente real para um SoS e evitar que políticas locais interfiram ou mascarem os resultados obtidos.

Para problemas que não são essencialmente técnicos, pretende-se também realizar um levantamento com desenvolvedores de sistemas com o objetivo de validar e refinar a arquitetura proposta e, em uma etapa posterior, avaliar a solução por meio da condução de estudos de caso.

5. Atividades Realizadas

Foi iniciado um mapeamento sistemático da literatura com o objetivo de identificar quais abordagens têm sido utilizadas para tratar questões relacionadas à confiabilidade em SoS. O planejamento e a execução do mapeamento seguem o processo proposto por Kitchenham e Charters (2007). O objetivo do mapeamento é identificar os fatores que exercem influência na confiabilidade em SoS, mapear as abordagens que têm sido utilizadas para a avaliação da confiabilidade em SoS e as técnicas e ferramentas utilizadas para atacar o problema. Espera-se que esse mapeamento forneça elementos para o refinamento da proposta.

6. Considerações finais

SoS são estruturas complexas que estão sujeitas a falhas que muitas vezes podem não ser previsíveis. Em SoS Virtuais, esses problemas podem ser ainda maiores em função

[3] https://www.planet-lab.org/

[4] Dado de 29 de janeiro de 2019.

da autonomia dos sistemas constituintes e pela falta de coordenação. Nesse sentido, este trabalho propõe uma técnica de tolerância a falhas para SoS Virtuais com a introdução de um elemento para lidar com prevenção e tolerância a falhas.

Pretende-se desenvolver um ferramental para implementar o modelo proposto. Embora esteja fundamentada em um mapeamento da literatura, a proposta será avaliada por especialistas a fim de embasar a construção deste ferramental. Com isso, pretende-se reduzir as possibilidades de degradação do SoS em função da mudança de comportamento nos sistemas constituintes. Como contribuição científica, pretende-se somar ao corpo de conhecimento uma arquitetura para lidar com falhas em SoS. Como contribuição tecnológica, pretende-se disponibilizar uma ferramenta para lidar com prevenção e tolerância a falhas em SoS virtuais.

Referências

Andrews, Z., Fitzgerald, J., Payne, R. and Romanovsky, A. (2013) "Fault modelling for systems of systems". In *Proceedings of the 11th International Symposium on Autonomous Decentralized Systems*, Mexico City, Mexico, pp. 59-66.

Dabholkar, A., Dubey, A., Gokhale, A., Karsai, G., and Mahadevan, N. (2012) "Reliable distributed real-time and embedded systems through safe middleware adaptation". In *Proceedings of the 31st IEEE Symposium on Reliable Distributed Systems*, Irvine, EUA, pp. 362–371.

Dahmann, J. and K. Baldwin. (2008) "Understanding the Current State of US Defense Systems of Systems and the Implications for Systems Engineering". In *Proceedings of the 2nd IEEE Systems Conference*, Montreal, Canada, pp. 1-7.

Department of Defense (2008) "Systems engineering guide for system of systems", Disponível em <https://www.acq.osd.mil/se/docs/se-guide-for-sos.pdf>. Acessado em 29 de dezembro de 2018.

Eddaoui, I., Itmi, M., Hami, A. E., Hmina, N., and Mazri, T. (2018). “A deterministic approach for systems-of-systems resilience quantification”. *International Journal of Critical Infrastructures* 14(1):80-99.

Garro, A. and Tundis, A. (2015) "On the reliability analysis of systems and SoS: The RAMSAS method and related extensions". *IEEE Systems Journal* 9(1):232-241.

Hamer, P. and Skramstad, T. (2011) "Service-oriented architecture for resilient complex systems". *In Proceedings of the 2011 IEEE Symposium on Reliable Distributed Systems*, Madri, Espanha, pp. 62–66.

Kitchenham, B. and Charters, S. (2007) “Guidelines for performing systematic literature reviews in software engineering”. Keele University Department of Computer Science, Technical Report EBSE 2007-001, 2007.

Maier, M. (1996) “Architecting Principles for Systems-of-Systems”. *INCOSE International Symposium*, *6*(1), pp. 565–573.

Shenker S. (1995) "Fundamental design issues for the future Internet". *IEEE Journal on Selected Areas in Communications* 13(7):1176-1188.

Wang, H., Wang, L., Yu, Q., Zheng, Z., and Yang, Z. (2018) "A proactive approach based on online reliability prediction for adaptation of service-oriented systems". *Journal of Parallel and Distributed Computing* 114:70-84.