Processo de desenvolvimento de *software* focado em sistemas distribuídos autonômicos

Pedro Felipe do Prado[1], Luis H. V. Nakamura[1], Júlio C. Estrella[1], Regina H. C. Santana[1], Marcos J. Santana[1], Omar A. C. Cortes[2]

[1]Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo

São Carlos – SP – Brasil

[2]Instituto Federal do Maranhão – São Luís – MA – Brasil

{pfprado,nakamura,jcezar,rcs,mjs}@icmc.usp.br, omar@ifma.edu.br

Abstract. *This paper presents a software development process focused on autonomic distributed systems. It is based on the following areas of knowledge: unified process, distributed systems, autonomic computing, operations research and performance evaluation. Through a case study, it is a demonstration of concept as well as demonstrated how to use the proposed development process.*

Resumo. *Este artigo apresenta um processo de desenvolvimento de software focado em sistemas distribuídos autonômicos. Ele é baseado nas seguintes áreas de conhecimento: processo unificado, sistemas distribuídos, computação autonômica, pesquisa operacional e avaliação de desempenho. Por meio de um estudo de caso, é realizada a prova de conceito, bem como demonstrado como usar o processo de desenvolvimento proposto.*

1. Introdução

A complexidade dos **Sistemas Distribuídos** (SDs) tem crescido consideravelmente nos últimos anos, devido aos seguintes fatores: (i) maior número de parâmetros configuráveis (dinâmicos e estáticos); (ii) aumento do uso de Web services, fazendo com que os sistemas estejam cada vez mais abertos e distribuídos entre várias empresas; (iii) aumento da heterogeneidade dos sistemas, com *softwares* advindos de diferentes fabricantes, fazendo com que seu gerenciamento, manutenção e detecção de problemas seja cada vez mais difícil.

A **Computação Autonômica** (CA), surge como uma forma de solucionar esses problemas, visando prover capacidades de autogerenciamento para os SDs. A CA possui os seguintes aspectos: autoconfiguração, auto-otimização, autocura e autoproteção. Sendo assim, cada necessidade de autogerenciamento do SD é atendida por algum dos aspectos da CA.

Apesar de vários *frameworks*, modelos de arquitetura e aplicações da CA nos mais variados SDs estarem presentes nos trabalhos relacionados, não foi encontrado nenhum processo de desenvolvimento de software focado em sistemas distribuídos autonômicos.

Sendo assim, a proposta deste trabalho é mostrar um processo de desenvolvimento de *software* focado em sistemas distribuídos autonômicos, explicando as áreas de conhecimento que o compõe, bem como um estudo de caso, que serve como prova de conceito e também um guia, de como usá-lo.

2. Processo de desenvolvimento de *software* focado em sistemas distribuídos autonômicos

Foram definidas as seguintes áreas de conhecimento para integrar o processo proposto: Processo Unificado de Desenvolvimento de *Software* (PU), Sistemas Distribuídos (SD), Computação Autonômica (CA), Pesquisa Operacional (PO) e Avaliação de Desempenho de Sistemas Computacionais (ADSC). Cada área contribui de alguma forma para o desenvolvimento de um Sistema Distribuído Autonômico (SDA). Uma ideia inicial desse trabalho foi publicada por Prado et. al. (2015).

O PU tem sido usado há mais de trinta anos tanto na indústria, como no meio acadêmico. Algumas de suas vantagens são: baseado em componentes, centrado na arquitetura, flexibilidade para ser adaptado de acordo com as necessidades da empresa e/ou sistema a ser desenvolvido, flexibilidade em seus processos, podendo ser empregado para criação desde sistemas pequenos até sistemas de grande porte. De fato, trata-se de um *framework* genérico, não possuindo uma estrutura rígida. Dessa forma, ele contribui por meio de suas fases e fluxos de trabalho, facilitando a criação e manutenção do SDA.

Os SDs são o foco principal do processo. Deve-se ter conhecimento das características genéricas dos SDs, além das específicas do SDA que será desenvolvido. Alguns dos principais desafios dos SDs são: segurança, tratamento de falhas, balanceamento de carga, qualidade de serviço, entre outros.

A CA surge como uma maneira de solucionar os problemas do SDA que será desenvolvido. Esses problemas podem ser tanto os genéricos dos SDs, como também problemas específicos do SDA em questão. É importante aprender a relacionar os aspectos da CA com os problemas inerentes aos SDs. Por exemplo, o tratamento de falhas pode ser solucionado com o aspecto de autocura da CA.

A PO serve para formalizar matematicamente algum problema do SDA. Ela pode ser usada em três situações diferentes: na otimização off-line (tudo que é feito para melhorar o desempenho do SDA enquanto ele não está em execução), no "serviço" que é fornecido pelo SDA (por exemplo, o "serviço" de um servidor Web é atender requisições HTTP) e também na implementação de um ou mais aspectos da CA, visando o autogerenciamento do SDA.

Por fim, a ADSC pode ser usada em todas as fases do processo, comparando diferentes soluções para o SDA.

3. Estudo de caso: otimização *off-line*

Para este estudo de caso, foi escolhido o problema de seleção de Web services baseada em atributos de qualidade de serviço. Esse problema foi abordado em maiores detalhes em Prado et. al. (2012) e Prado et. al. (2013). De maneira informal, cada grupo de funcionalidades relacionadas é denominado um Web service abstrato. Por exemplo, pode-se pensar em um Web service abstrato para a compra de uma passagem aérea, outro para reservar um quarto de hotel, outro para alugar um carro e assim por diante. Para cada Web service abstrato, existe uma série de Web services concretos, que são aplicações que realmente implementam essas funcionalidades. Entretanto, cada Web service concreto possui seus próprios atributos de qualidade de serviço: **custo** e **tempo médio de execução** (TME). Uma forma de armazenar os atributos de QoS é usando Web semântica, como foi proposta no trabalho de Nakamura et. al. (2014).

Sendo assim, a criação de um plano de composição, trata da seleção de quais Web services concretos serão alocados para cada Web service abstrato, tendo em vista obter o melhor tempo médio de execução total, respeitando uma restrição de orçamento. Trata-se de uma aplicação do problema da mochila com múltiplas escolhas (do inglês, *Multiple-Choice Knapsack Problem*).

$$\text{Maximizar} \quad \sum_{i=1}^{m} \sum_{j \in N_i} t_{ij}\, x_{ij}$$

$$\text{Sujeito a:} \quad \sum_{i=1}^{m} \sum_{j \in N_i} c_{ij}\, x_{ij} \leq \Omega,$$

$$\sum_{j \in N_i} x_{ij} = 1,\ i = 1, \ldots, m$$

$$x_{ij} \in \{0,1\}, \quad i = 1, \ldots, m, \qquad j \in N_i$$

Figura 1: Modelo matemático do problema.

De maneira formal, o modelo matemático criado pode ser observado na Figura 1. Em termos da seleção de Web services baseada em QoS, as *m* classes são os Web services abstratos, os itens a serem selecionados são os Web services concretos, o benefício de cada Web service concreto é o TME normalizado (entre 0 e 1, quanto maior melhor), o custo é o custo de cada Web service concreto e a capacidade da mochila é o orçamento definido. A variável t_{ij} representa o TME do Web service concreto *j* atribuído ao Web service abstrato *i*. A variável c_{ij} representa o custo de atribuir o Web service concreto *j* ao Web service abstrato *i*. A restrição da primeira linha garante que o custo total dos Web services concretos selecionados não ultrapasse o orçamento definido e a restrição da segunda linha garante que exatamente um Web service concreto seja selecionado para cada Web service abstrato.

Descobriu-se que o custo e o TME dos Web services são atualizados com frequências diferentes. O custo, é atualizado mensalmente, já o TME é atualizado a cada hora. Sendo assim, surgiu a ideia de executar o algoritmo BE para todos os fluxos de composição definidos, visando eliminar do espaço de busca os planos de composição que não atendem a restrição definida (ultrapassam o orçamento definido). Os fatores e níveis definidos foram: (i) número de Web services abstratos: 2 e 5; (ii) número de Web

services concretos por Web service abstrato: 25 e 50; (iii) orçamento: 25% e 50% do custo total possível.

Na Tabela 2, temos os resultados obtidos para os fluxos de composição com dois Web services abstratos. A coluna "**A**" representa a quantidade de Web services abstratos, a coluna "**C**" representa a quantidade de Web services concretos por Web service abstrato, a coluna "**O**" representa o orçamento selecionado e a coluna "**R**" representa a porcentagem de pontos válidos obtidos. Por exemplo, se o tamanho do espaço de busca é cem, mas apenas dez pontos cumpriram a restrição de orçamento, então a coluna "**R**" apresentará o valor 10%. Na Tabela 3, temos os resultados obtidos para o fluxo de composição com cinco Web services abstratos.

Tabela 1 – Resultados 2 WSs abstratos

A	C	O (%)	R (%)
2	25	25	9,28
2	50	25	15,68
2	25	50	47,04
2	50	50	54,12

Tabela 2 – Resultados 5 WSs abstratos

A	C	O (%)	R (%)
5	25	25	1,80
5	50	25	2,26
5	25	50	41,52
5	50	50	46,08

4. Conclusões e trabalhos futuros

Neste artigo foi apresentado como usar a PO durante a otimização off-line, visando reduzir o tamanho do espaço de busca que será percorrido quando o SDA estiver em execução. Outros estudos de caso também podem ser realizados, usando PO e avaliação de desempenho para otimizar o SDA durante sua execução e também para implementar diferentes mecanismos da CA para criar um sistema com a capacidade de autogerenciamento.

Referências

Nakamura, L. H. V.; Prado, P. F.; Libardi, R.; Nunes, L.H.; Santana, M. J.; Santana, R. H. C. "Fast Selection of Web services with QoS using a Distributed Parallel Semantic Approach". International Conference on Web Services (ICWS), 2014. DOI: 10.1109/ICWS.2014.100

Prado, P. F.; Nakamura, L. H. V.; Estrella, J. C.; Santana, M. J.; Santana, R. H. C. "Different Approaches for QoS-aware web services composition focused on e-commerce systems". 13th Symposium on Computer Systems (WSCAD-SSC). DOI: 10.1109/WSCAD-SSC.2012.14

Prado, P. F.; Nakamura, L. H. V.; Estrella, J. C.; Santana, M. J.; Santana, R. H. C. "A performance evaluation study for QoS-aware web services composition using heuristic algorithms". International Conference on Digital Society (ICDS), 2013.

Prado, P. F.; Nakamura, L. H. V.; Santana, M. J.; Cortes, O. A. C.; Santana, R. H. C. "Cobapas: Combinatorial Optimization based Approach for Autonomic Systems". International Conference on Autonomic and Autonomous Systems (ICAS), 2015.