

# **Universidade Federal de Pernambuco**

Graduação em Ciência da Computação

Centro de Informática

2016.2

## **Mineração de Processos de Middleware Proposta de Trabalho de Graduação**

**Aluno:** Rodolfo Santos Vera Cruz (rsvc@cin.ufpe.br)

**Professor:** Nelson Souto Rosa (nsr@cin.ufpe.br)

## **Contexto**

A utilização de softwares atualmente está bastante difundida. É muito fácil encontrar pessoas de diversas partes do mundo que possuem algum contato diário com serviços de software, como Google, sites de compra online ou até mesmo algum jogo para dispositivos móveis. Esta grande facilidade de acesso a esses serviços tem criado uma alta demanda para empresas do ramo. Por conta desta crescente demanda, e com o objetivo final de prestar um bom serviço para seus clientes, muitas empresas tem adotado a utilização de sistemas distribuídos, os quais conseguem atender uma grande quantidade de usuários com qualidade.

Junto com a crescente utilização dos sistemas distribuídos, vem uma grande preocupação a respeito da manutenção desses sistemas. Como mostrado em [1], mais de 90% do custo de um software está ligado a manutenção, e um dos motivos para este alto custo é a falta/deficiência de documentação e substituição de pessoas que possuem mais conhecimento a respeito do software.

Fazer a manutenção de um sistema distribuído é ainda mais difícil, pois ele consiste de vários pedaços de softwares espalhados em diferentes máquinas e que podem ter sido desenvolvidos por diferentes grupos, usando diferentes linguagens de programação [2]. Por este motivo, a utilização de ferramentas que auxiliem o monitoramento de sistemas distribuídos se faz cada vez mais útil e necessário.

## Objetivo

A fim de realizar um eficiente monitoramento de sistemas distribuídos, registrar as atividades realizadas pelo sistema torna-se uma importante e essencial tarefa. Este registro é feito através de ferramentas de trace [3][4][5]. Todas essas ferramentas procuram registrar as atividades realizadas por um sistema distribuído, a fim de facilitar a identificação de um mal funcionamento do mesmo, com o objetivo final de corrigir os problemas encontrados, melhorando cada vez mais o funcionamento do sistema como um todo.

O desenvolvimento dessas ferramentas de trace deve levar em consideração vários fatores, dentre eles, a baixa sobrecarga no sistema imposta pelo serviço de trace, e um registro ordenado dos eventos ocorridos, já que estes acontecem em diferentes máquinas.

O objetivo deste trabalho é desenvolver um dos módulos (Data Collection) de um sistema de trace para middleware [6][7], o qual pode ser observado na Figura 1- Esquema do sistema para fazer mineração de processos em middleware. Este módulo tem como função principal coletar os eventos realizados pela execução do middleware, ordená-los e formatá-los, gerando um 'log' de atividades do sistema para uma posterior análise.

O 'log' será gerado utilizando o padrão XES. O formato XES é comumente utilizado como um padrão para registro de eventos. Ele possui como algumas de suas principais características uma formatação simples que pode ser entendido por qualquer pessoa facilmente, e compatibilidade com qualquer domínio/aplicação [8].

Este 'log' no padrão XES vai servir como entrada para o framework ProM. O ProM é uma ferramenta de mineração de processos, pela qual se extrai informações do sistema através dos eventos registrados no 'log'. As principais vantagens do ProM são sua flexibilidade, pois ele suporta diferentes formatos de arquivo para entrada e saída de dados, e a facilidade que ele proporciona para o desenvolvimento de novas funções, permitindo uma fácil reutilização de seu código [9].



Figura 1- Esquema do sistema para fazer mineração de processos em middleware

## Cronograma

<b>Atividades</b>	<b>Agosto</b>	<b>Setembro</b>	<b>Outubro</b>	<b>Novembro</b>	<b>Dezembro</b>
Levantamento bibliográfico					
Escolha da abordagem a ser adotada					
Implementação e teste da abordagem escolhida					
Análise dos resultados					
Elaboração do relatório					
Elaboração da apresentação					

## **Possíveis Avaliadores**

Fernando Castor

## Referências

1. “How to save on software maintenance costs”, 2007. Disponível em <http://asq.org/public/wqm/how-to-save-on-software-maintenance-costs.pdf>. Acessado em Setembro de 2016.
2. Marcos K. Aguilera, Jeffrey C. Mogul , Janet L. Wiener , Patrick Reynolds , Athicha Muthitacharoen, “Performance debugging for distributed systems of black boxes”, Proceedings of the nineteenth ACM symposium on Operating systems principles, Outubro 2003, 19-22
3. Sigelman et al, “Dapper, a Large-Scale Distributed Systems Tracing Infrastructure”, Technical Report, 2010.
4. R. Fonseca, G. Porter, R. H. Katz, S. Shenker,, I. Stoica, “X-Trace: A pervasive network tracing framework,” In Proceedings of USENIX NSDI, Abril 2007, 271–284.
5. E. Anderson, C. Hoover, X. Li and J. Tucek, "Efficient tracing and performance analysis for large distributed systems", *Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2009. MASCOTS '09. IEEE International Symposium on. IEEE*, 1-10
6. Bernstein, P. A. (1996), 'Middleware: A Model for Distributed System Services', *Communications of the ACM* **39**(2), 86--98.
7. Vinoski, S. (2002), 'Where is middleware', *IEEE Internet Computing* **6**(2), 83--85.
8. Gunther, C., Verbeek, H.: XES v2.0 (2014), <http://www.xes-standard.org/>
9. van Dongen, B., de Medeiros, A., Verbeek, H., Weijters, A., van der Aalst, W. , “The ProM Framework: A new era in process mining tool support” . ICATPN 3536, 444–454 , 2005

## **Assinaturas**

Recife, Pernambuco. Brasil  
14 de Setembro de 2016

---

**Nelson Souto Rosa**  
**Orientador**

---

**Rodolfo Santos Vera Cruz**  
**Orientando**