



Universidade Federal de Pernambuco
Graduação em Ciência da Computação
Centro de Informática



Instrumentação de Middleware para Geração de Log

Proposta de Trabalho de Graduação

Aluno: Eduardo Peixoto Macedo
Orientador: Nelson Souto Rosa

Recife, Março de 2016

Índice

Introdução	3
Objetivo.....	4
Cronograma	5
Referências.....	6
Possíveis Avaliadores.....	7
Assinaturas	8

Introdução

Em Ciência da Computação, a instrumentação de código é utilizada para medição de desempenho, diagnóstico de erros e monitoramento da execução e comportamento do programa. Em termos de programação, para instrumentar código, o desenvolvedor adiciona certas funcionalidades ao código de forma que elas adicionem ao programa as seguintes habilidades:

- *Code Tracing*: geração de mensagens informativas em tempo de execução;
- Depuração (*debugging*): rastreamento, detecção e resolução de erros de programação durante o desenvolvimento;
- Análise Dinâmica: mensuração e monitoramento do comportamento dinâmico da aplicação, como utilização de recursos, frequência ou duração de chamadas de funções e utilização de instruções específicas;
- Contadores de Desempenho: a aplicação dispõe de componentes capazes de medir e gerar dados de desempenho.
- Log de Eventos: componentes que registram a ocorrência de determinados eventos durante a execução da aplicação.

A instrumentação do código se mostra essencial durante e após o desenvolvimento de uma aplicação, uma vez que o desenvolvedor está sempre precisando monitorar dados sobre a execução da aplicação. Medição de desempenho e detecção de erros, são apenas alguns exemplos de como o desenvolvedor utiliza essa técnica.

Outra importante funcionalidade é o *code tracing*. Fazer a aplicação gerar um registro de mensagens em tempo de execução permite que seja feita uma análise dinâmica do código, e portanto, um estudo bem mais detalhado do seu comportamento. Com essa habilidade, tanto o programador quanto o usuário da aplicação têm mais uma poderosa ferramenta de análise e monitoramento.

Objetivo

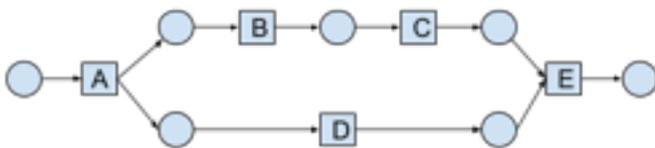
Este documento apresenta uma proposta de estudo teórico e prático sobre instrumentação de código de middleware [1]. Em particular, o estudo dará ênfase às funcionalidades de Code Tracing e Logging de sistemas de middleware.

O objetivo geral do trabalho é elaboração de um mecanismo de instrumentação de middleware escritas em Java. A meta inicial é tornar o projeto capaz de acrescentar estaticamente a funcionalidade de geração de log ao código Java. Posteriormente será feito um estudo para evoluir o mecanismo ao ponto de sermos capazes de instrumentar o middleware dinamicamente.

A geração do log (ou Trace) resultará num conjunto de registros que incluem chamadas de função, utilização de instruções, entre outros. O conjunto como um todo pode ser visto como um arquivo que representa o comportamento do middleware nas circunstâncias da execução que gerou o log. Obter essa representação se mostra interessante para que seja possível entender o funcionamento e estrutura de um aplicação, que no caso desse trabalho é o middleware.

O log será gerado no formato definido pelo XES. O XES é um padrão de log de eventos baseado em XML. Foi criado para fornecer um formato amplamente reconhecido e facilitar a troca de dados de log entre aplicações.

O log servirá então como entrada para um processo de mineração baseado no PROM [3]. Dado que não conhecemos o funcionamento de um middleware, o objetivo é interpretar as informações no log e construir uma representação de como ele se comporta. Por exemplo, a figura 1a mostra um exemplo de aplicação que executa as atividades A,B,C,D e E. Em uma determinada execução é gerado o trace mostrado na figura 1b. O que faremos é dar o trace como entrada para o PROM para que ele gere um modelo parecido com o da figura 1a.



(a) Um processo modelado por uma Rede de Petri



(b) Um *trace*. As linhas pontilhadas representam tempo de espera e as linhas cheias representam tempo de serviço

Figura 1: Um processo e um exemplo de *trace*

Cronograma

Atividades	Abril				Maio				Junho				Julho	
Levantamento de material e estudo	X	X	X	X	X	X								
Concepção da proposta					X	X	X	X	X	X				
Testes e experimentos									X	X	X	X		
Elaboração do relatório									X	X	X	X		
Preparação da apresentação											X	X	X	

Referências

- [1]. Bernstein, P. A. (1996), 'Middleware: A Model for Distributed System Services', *Communications of the ACM* 39(2), 86–98.
- [2]. Vinoski, S. (2002), 'Where is middleware', *IEEE Internet Computing* 6(2), 83--85.
- [3]. Leemans, S. J.; Fahland, D. & van der Aalst, W. M. (2015), Using Life Cycle Information in Process Discovery, in 'Business Process Intelligence'.

Possíveis Avaliadores

- Ricardo Massa (rmfl@cin.ufpe.br)

Assinaturas

Eduardo Peixoto Macedo
(Aluno)

Nelson Souto Rosa
(Orientador)