

Rastreamento e Visualização de programas Java usando JVMTI *

Geovani Ricardo Wiedenhof, Benhur Stein

Universidade Federal de Santa Maria - Laboratório de Sistemas de Computação
Faixa de Camobi, Km 9 -CEP 97105-900-Santa Maria-RS-Fone/Fax:(55) 220 8523
{grw,benhur}@inf.ufsm.br

Introdução

A utilização da linguagem Java no desenvolvimento de aplicações paralelas e distribuídas é crescente. Naturalmente, surge a necessidade de meios para a depuração do código Java desenvolvido. As ferramentas tradicionais de depuração de aplicações sequenciais são em geral insuficientes para essas tarefas. Uma ferramenta complementar é a visualização da execução da aplicação. Para isso, registra-se os principais eventos ocorridos durante a execução da aplicação e depois apresenta-se os mesmos de forma gráfica. Este trabalho descreve uma ferramenta de rastreamento de programas Java que está sendo desenvolvida no Laboratório de Sistemas da Computação (LSC) localizado na Universidade Federal de Santa Maria (UFSM). Este faz uso da interface de recuperação de eventos JVMTI “*Java Virtual Machine Tool Interface*”, para monitorar e registrar os principais eventos gerados pela JVM durante a execução de uma aplicação Java. Depois desse registro, os eventos são visualizados. A próxima seção apresenta as ferramentas utilizadas para a realização do registro, recuperação e visualização de programas Java. Segue uma descrição mais detalhada do trabalho, os resultados obtidos até o momento e os trabalhos futuros previstos para o aprimoramento da ferramenta.

Ferramentas

A depuração de programas Java através da visualização ocorre a partir da monitoração dos eventos gerados pela JVM e do armazenamento de informações desses eventos. Após a execução, esses eventos devem ser convertidos para o formato do visualizador com objetivo final de visualização. Para isso foram utilizadas as ferramentas descritas nas próximas sub-seções.

Interfaces de Recuperação de Eventos Java

O Java possui, a partir da versão 1.5 da *JDK* (disponível desde setembro de 2004), um módulo de depuração e monitoramento chamado JVMTI [Sun 2004], que permite interceptar diversos eventos ocorridos na JVM, como criação e destruição de fluxos de execução, chamada e retorno de métodos, alocação e liberação de memória. Essa interface

*Financiamento: CNPq e FAPERGS

substitui a anterior, chamada JVMPI, “*Java Virtual Machine Profiler Interface*”, que foi usada para o desenvolvimento de uma versão anterior deste trabalho [SIL 2002].

A JVMTI difere da JVMPI por possuir um conjunto maior de funções e notificações de eventos. A JVMTI também possui uma forma diferente de informar e permitir a seleção desses eventos, o que a torna mais eficiente que a predecessora.

Registro e Recuperação de dados

Para o registro e recuperação das informações dos eventos monitorados pela JVMTI foi utilizada a biblioteca libRastro[LIB 2004]. Essa biblioteca permite gerar rastros de execução genéricos com baixa intrusividade na execução do programa rastreado. A biblioteca suporta o rastreamento de programas com múltiplos fluxos de execução e/ou de programas distribuídos em diversas máquinas. A biblioteca oferece ainda recursos para a sincronização de datas registradas por relógios de máquinas independentes. Essas características tornam a libRastro uma biblioteca adequada às necessidades em questão.

Pajé

Para visualizar os rastros gerados, foi utilizada a ferramenta genérica de visualização Pajé [KER 2003]. Essa ferramenta é baseada na visualização *pós-mortem*, ou seja, é feita a conversão para o formato Pajé após terem sido gerados os rastros de execução.

Descrição do Trabalho

Com a monitoração dos eventos através da JVMTI, e o registro com a biblioteca libRastro foi criada uma biblioteca que une as funcionalidades em uma interface para o rastreamento de programas Java. Para realizar esse rastreamento, não é necessário a mudança do código da aplicação, basta executar o programa passando um parâmetro que ocasiona a carga e execução da JVMTI e da biblioteca. A biblioteca pode ser configurada para seleção de eventos e métodos que serão monitorados.

No lançamento da aplicação, pode ser passado como opção para a execução dois arquivos com informações de seleção. O primeiro arquivo contém a seleção de eventos a serem monitorados. A criação e destruição de threads, a entrada e a saída de métodos são alguns exemplos de eventos que podem ser monitorados. Através do segundo arquivo passado é feita a seleção dos métodos a serem monitorados na entrada e saída. Caso seja habilitada a geração dos rastros, na criação de fluxos de execução é registrado o identificador da mesma, bem como outras informações relevantes para posterior visualização, na entrada e saída de métodos é registrado o identificador do método e o identificador do objeto que o executa.

O objetivo da utilização dos arquivos com as seleções é melhorar a posterior visualização dos rastros gerados e ter uma menor interferência na execução normal da aplicação. Quanto maior o volume dos rastros gerados (para se obter uma visualização mais completa) maior será a influência da geração dos rastros na execução do programa.

Após a geração dos arquivos de rastros através da biblioteca, um outro programa é usado para converter os eventos para o formato do visualizador Pajé. Esse programa

oferece a opção de escolher qual o tipo de visualização desejada, por fluxo de execução ou por objeto. A visualização por fluxo mostra os eventos, entrada e saída de métodos, associados ao fluxo de execução que os executou. O segundo tipo de visualização, por objeto, permite mostrar o fluxo que está executando cada método do objeto, dando uma noção do nível de execução concorrente no interior de cada objeto. Essa escolha ajuda na posterior depuração da aplicação.

Resultados

Esta seção apresenta alguns resultados preliminares obtidos com a utilização da interface de rastreamento de programas Java desenvolvida. A aplicação utilizada como exemplo é uma simulação de construção de um computador por peças e a impressão de cada peça. Os exemplos de peças são: o próprio computador, monitor, disquete e teclado. Os resultados são mostrados nas figuras 1 e 2.

A figura 1 mostra a visualização da execução da aplicação, selecionando-se somente os métodos definidos na aplicação, através dos arquivos de configurações. Nessa figura foi incluída algumas setas com o número especificado e o nome do método que o fluxo está executando. A seta 1 mostra os fluxos de execuções criados dentro do java, e a seta 2 mostra o fluxo criado pela aplicação, a “main”. Na aplicação o método “imprimeComputador” chama, por exemplo o método “imprimeMonitor”, que por sua vez invoca o método “imprimePeca”. Abaixo foi incluído o código Java do método “imprimeComputador”.

```
public void imprimeComputador()  
    super.imprimePeca();  
    teclado.imprimeTeclado();  
    monitor.imprimeMonitor();  
    disquete.imprimeDisquete();
```

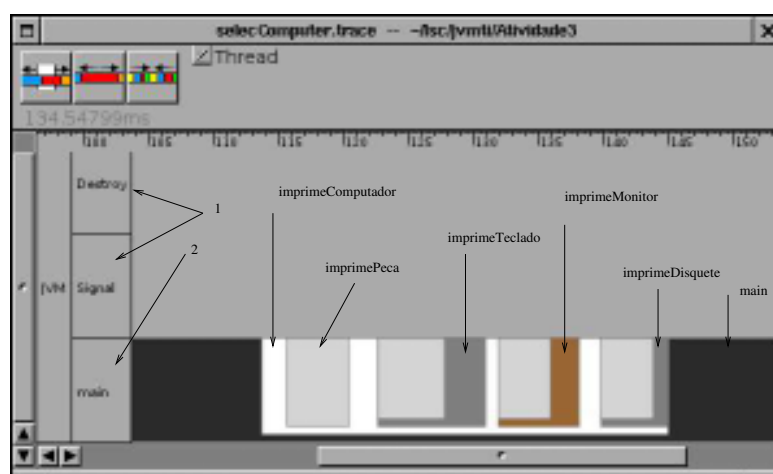


Figura 1: Visualização de uma aplicação no Pajé com seleção

A figura 2 mostra a visualização no Pajé da execução da aplicação de impressão

do computador com a monitoração de todos métodos que são executados, inclusive os métodos internos da biblioteca Java.

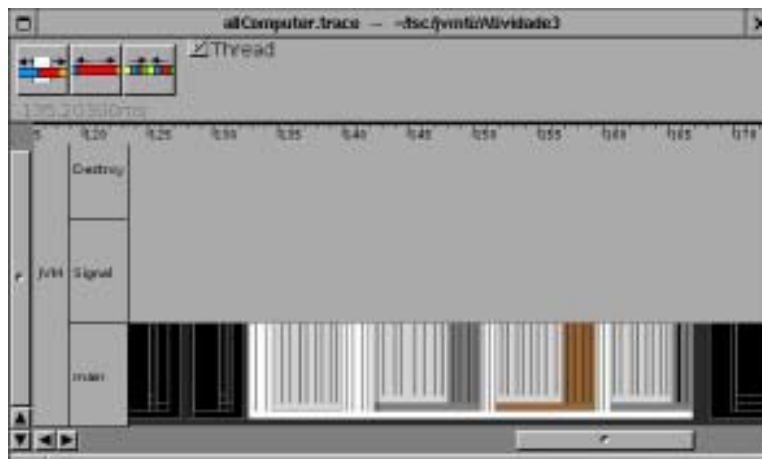


Figura 2: Visualização de uma aplicação no Pajé sem seleção

Conclusão

Este texto descreve o trabalho que está sendo feito com o objetivo de rastrear aplicações Java. Esse trabalho utiliza a interface JVMTI para recuperação dos eventos gerados pela JVM, a biblioteca libRastro para o registro e recuperação dos principais eventos e o Pajé para a visualização dos rastros gerados. Com isso, obteve-se alguns resultados satisfatórios para observações e controle das execuções.

Como trabalhos futuros, além dos eventos que estão sendo monitorados, pretende-se incluir o monitoramento de outros eventos oferecidos pela JVMTI, como a sincronização de fluxos de execução. Espera-se também realizar o rastreamento das comunicações RMI.

Referências

- [KER 2003] KERGOMMEAUX, J. C. de; OLIVEIRA STEIN, B. de. Flexible performance visualization of parallel and distributed applications. **Future Generation Computer Systems**, 2003.
- [LIB 2004] LIBRASTRO. <http://www.inf.ufsm.br/lsc>.
- [SIL 2002] SILVA, G. J. da; STEIN, B. Uma Biblioteca Genérica de Geração de Rastros de Execução para Visualização de Programas. **Anais do I Simpósio de Informática da Região Centro, Santa Maria**, 2002.
- [Sun 2004] Sun Microsystems. **Java virtual machine tool interface**. <http://java.sun.com/j2se/1.5.0/docs/guide/jvmti/jvmti.html>.