

TFPS: Um Sistema de Pré-processamento de Traces para Auxiliar na Visualização de Programas Paralelos

Denise Stringhini¹
Phillippe O. A. Navaux²

Universidade Federal do Rio Grande do Sul
Instituto de Informática - II
Curso de Pós-graduação em Ciência da Computação
Caixa Postal 15064. Porto Alegre, RS, 91501-970, BRASIL

RESUMO

Este artigo apresenta a ferramenta TFPS - *Trace File Preprocessor System*, que tem como principal objetivo a visualização de programas paralelos desenvolvidos em PVM. O propósito desta visualização é a análise de desempenho de tais programas, assim como a observação de seu comportamento geral. Apesar de demonstrar aspectos físicos da máquina virtual montada pelo PVM, a ferramenta se concentra na visualização de aspectos lógicos da aplicação paralela.

O diferencial em relação a outras ferramentas deste tipo é a utilização do arquivo de *trace* antes da fase de animação. Em geral, esta fase da visualização se utiliza do *trace* apenas para extrair os eventos que serão demonstrados. A proposta do TFPS é a extração de características próprias da aplicação antes da fase de animação. Estas características são usadas para a montagem das janelas de visualização. O objetivo é apresentar janelas praticamente "personalizadas", de acordo com as características próprias da aplicação que será analisada.

ABSTRACT

This paper presents the TFPS - Trace File Preprocessor System tool, which has as a main goal the visualization of parallel programs developed in PVM. The proposal of this visualization is the performance analysis of such programs as well as the observation of its general behavior. In spite of demonstrate the physical aspects of the PVM virtual machine, the tool concentrates on the visualization of logical aspects of the parallel application.

The difference in relation to the other tools is the use of the trace file before the animation phase. In general, this visualization's phase uses the trace only to extract the events that will be demonstrate. The TFPS' proposal is to extract the application own characteristics before the animation phase. This characteristics are used to create the visualization views. The goal is to present the views practically "personalized", in agree with the own characteristics of the application that will be analyzed.

¹ Mestre em Ciência da Computação (CPGCC-UFRGS, 1997). Professora - ULBRA - Área de interesse: processamento paralelo. E-mail: string@inf.ufrgs.br

²Doutor em Informática (Grenoble, 1979). Professor Pesquisador - CPGCC-UFRGS - Área de interesse: processamento paralelo. E-mail: navaux@inf.ufrgs.br

1. INTRODUÇÃO

A ferramenta desenvolvida tem como principal objetivo facilitar e tornar mais intuitiva a compreensão e a avaliação de desempenho de programas paralelos através da visualização lógica dos programas. Para isto, fornece uma quantidade maior de informações em cada janela de visualização e as organiza de forma que, à primeira vista, o programador já tenha uma idéia de quais são os principais problemas da aplicação que está desenvolvendo. Tem, então, a possibilidade de executar uma animação da execução do programa em várias janelas, que fornecem diferentes parâmetros de avaliação.

A ferramenta TFPS, desenvolvida para aplicações PVM [GEI 94], é constituída de duas partes distintas e independentes: o pré-processador TFP (*Trace File Preprocessor*) e a TFP View, que é a ferramenta de visualização. As próximas seções descrevem brevemente cada componente.

2. TFP - TRACE FILE PREPROCESSOR

O pré-processador lê um arquivo de *trace* gerado pela ferramenta PVM/TAPE [MAI 95] e coleta todas as informações necessárias à montagem das janelas idealizadas. O algoritmo é simples e consiste em acumular dados de execução em uma única leitura e processá-los. Estes dados são organizados de forma que possam ser demonstrados de forma gráfica nas janelas.

Um exemplo do tipo de informação coletada é o **grafo de envio de mensagens**, através do qual é possível que se saiba o destino das mensagens de cada processo, ou seja, com quais outros processos existe comunicação.

Um exemplo do tipo de informação processada é a **quantidade de mensagens colocadas na rede de comunicação por processo e por máquina**. Esta informação é utilizada nas janelas para mostrar processos/máquinas de forma ordenada. Maiores detalhes sobre o pré-processador podem ser encontrados em [STR 96].

3. TFP VIEW

A TFP View é composta por três janelas independentes: *Spacetime Diagram*, *Kiviat Diagram* e *Processes Mapping Diagram*. Está ainda em fase de implementação o protótipo da quarta janela que é a *Processes Graph*.

O *Spacetime Diagram* (figura 3.1) é muito comum em ferramentas de visualização. Entretanto, algumas melhorias foram acrescentadas a partir do pré-processamento de *traces* aqui apresentado. A principal, é a ordenação de processos no eixo vertical. Os processos estão agrupados nas máquinas em que executaram, as quais são identificadas pela legenda e ordenadas por maior quantidade de mensagens colocadas na rede. Por sua vez, os processos são ordenados internamente a cada grupo também por quantidade de mensagens colocadas na rede.

O *Processes Mapping Diagram* é um diagrama que demonstra o mapeamento dos processos na máquina virtual do PVM e pode ser visualizada na figura 3.2.

A terceira, o *Kiviat Diagram*, como o próprio nome já diz, é um diagrama Kiviat, onde as máquinas PVM da aplicação estão dispostas em um círculo. Cada uma é representada por um raio e um ponto neste raio consiste no percentual de mensagens colocadas na rede pelo qual a máquina é responsável. O polígono formado por estes

pontos durante a animação descreve a utilização da rede pelas máquinas. Estas, estão também ordenadas no círculo por maior quantidade de mensagens colocadas na rede. Isto facilita a interpretação, já que fará com que o polígono tenda para o lado direito, caso haja utilização desigual.

Já a *Processes Graph*, um grafo de processos, consiste numa representação completamente independente da máquina virtual. Através dela, é possível a visualização da aplicação como um todo. Sua montagem é baseada numa lista de adjacências obtida durante a leitura do *trace* e seu desenho é realizado através de um algoritmo específico para desenho de grafos, o *Forced Directed Placement* [FRU 91].

Para ambos os exemplos do conjunto de janelas da **TFP View** abaixo (figuras 3.1 e 3.2) foi utilizada a ferramenta *ANDES* [KIT 95], que é uma ferramenta para avaliação de programas paralelos onde a técnica utilizada é baseada em programas sintéticos. Como trata-se de uma execução do tipo SPMD, todos os processos ficaram com a mesma string de identificação. Isto aconteceu porque o TFP retira esta string dos nomes dos programas fontes, fornecidos pela própria ferramenta de monitoração. A parte numérica do identificador corresponde à ordem de criação do processo na máquina em que executou.

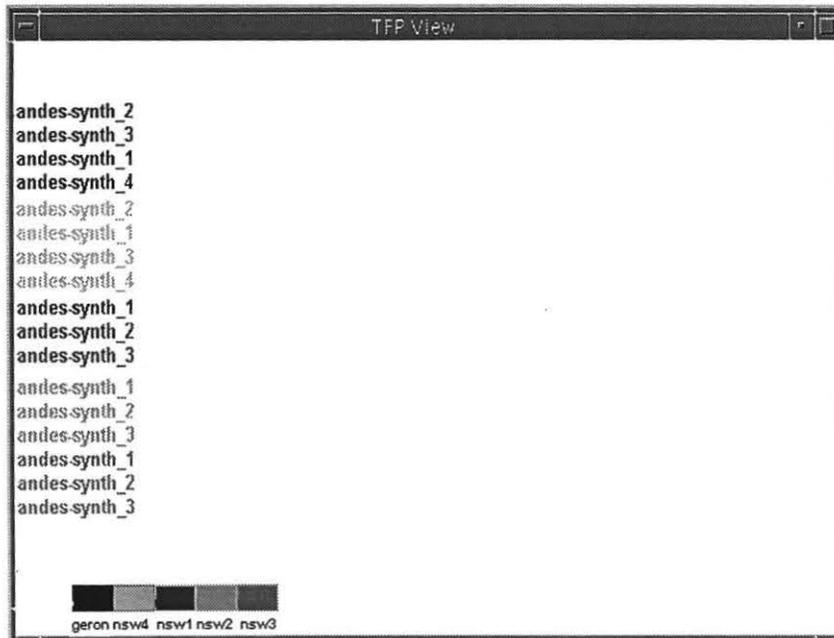


FIGURA 3.1: *Space-time Diagram*

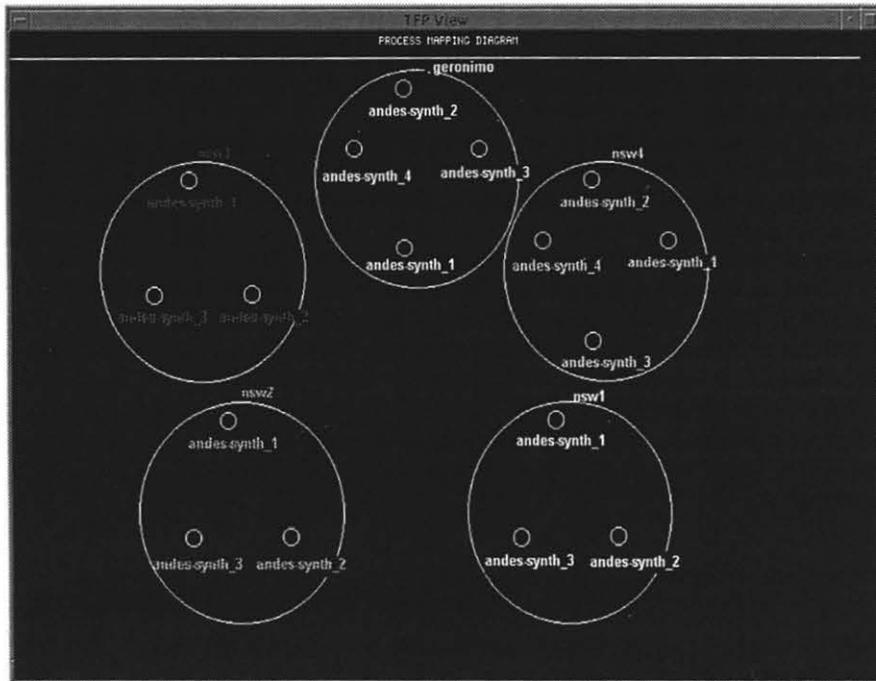


FIGURA 3.2: Processes Mapping Diagram

4. TRABALHOS RELACIONADOS

Para finalizar, apresenta-se a tabela 4.1 com uma comparação resumida entre três ferramentas analisadas, além da TFP View. São elas: ParaGraph [HEA 91], XPVM [GEI 96] e PVaniM [STA 95]. Esta tabela leva em consideração apenas os tipos de janelas implementadas pelo protótipo do TFP.

TABELA 4.1 - Comparação entre tipos de janelas entre algumas ferramentas.

Janelas	TFPS	ParaGraph	XPVM	PVaniM
Space-time	Ids. dos processos alfanuméricos. Ordem no eixo x. Mapeamento.	Ids. numéricos. - -	Ids. alfanum. - Mapeamento.	Ids. numéricos. - -
Kiviat	Utilização da rede. Ordem no círculo.	Utilização do processador. -	Não tem.	Não tem.
Processes Mapping	Localização gráfica dos processos.	Não tem.	Localização textual dos processos.	Localização textual dos processos.
Grafo de Processos ³	Aproximação dos vizinhos.	Dispostos em círculo ou pelo usuário.	Não tem.	Dispostos em círculo.

³Título em português por não participar do protótipo implementado que funciona em conjunto com o ParaGraph.

5. CONCLUSÃO

O TFPS demonstrou uma utilidade a mais para os arquivos de *trace*, comumente gerados para análise e até mesmo depuração de programas paralelos. O pré-processador se interpõe entre a fase de monitoração e a de animação, sendo um passo a mais na análise *post-mortem* de programas paralelos. Com isso, não influi nem no tempo de execução da aplicação, nem no tempo de execução da animação, ou seja, não há o indesejável compartilhamento de recursos que pode ser fatal em ferramentas de análise de desempenho de programas paralelos.

Através da TFP View foram aperfeiçoadas duas janelas já utilizadas em ferramentas de análise: *Spacetime Diagram* e *Kiviat Diagram*, que sofreram alterações que possibilitam uma análise mais intuitiva e direta da aplicação paralela. Além disso, foram criadas duas novas janelas em função das informações obtidas através do pré-processamento: o *Processes Mapping Diagram* e a *Processes Graph* (ainda não incluída, mas já validada), que visam demonstrar os aspectos lógicos da aplicação e relacioná-los aos aspectos físicos.

6. BIBLIOGRAFIA

- [FRU 91] Fruchterman, T. M. J. e Reingold, E. M. *Graph Drawing by Forced-directed Placement*. Soft. Pract. and Exp., vol 21(11), 1129-1164, Novembro 1991
- [GEI 94] Geist, A.; Beguelin, A.; Dongarra, J.; Jiang, W.; Manchek, R.; e Sunderam, V. *PVM*. MIT Press, 1994
- [GEI 96] Geist, G. A.; Kohl, James; Papadopoulos, Philip. *Visualization, Debugging, and Performance in PVM*. Disponível por ftp em netlib2.cs.utk.edu/pvm3/xpvm (nov. 1996).
- [HEA 91] Heath, Michael T. e Etheridge, Jennifer A. *Visualizing the Performance of Parallel Programs* IEEE Software, vol. 8(5):29-39, Maio 1991
- [KIT 95] KITAJIMA, João Paulo F. *ANDES: a Tool for Evaluating Parallel Systems*. In: Simpósio Brasileiro de Arquitetura de Computadores E Processamento de Alto Desempenho, 7., 1995, Canela, RS. *Anais...* Porto Alegre: Instituto de Informática/UFRGS, 1995. p. 367-381.
- [MAI 95] Maillet, Eric *TAPE/PVM an Efficient Performance Monitor for PVM Applications - User Guide*. LMC-IMAG, 19 pp., Grenoble, França, Technical Report, Março 1995
- [STA 95] Stasko, John; Topol, Brad e Sunderam, Vaidy *PVaniM 2.0- - Online o Postmortem Visualization Support for PVM* disponível por ftp: ftp.cc.gatech.edu/pub/people/stasko
- [STR 96] Stringhini, Denise; Navaux, Philippe O. A. *Pré-processamento de Traces para Auxiliar a Visualização de Programas Paralelos*. In: Simpósio Brasileiro de Arquitetura de Computadores e Processamento de Alto Desempenho, 8., 1996, Recife, PE. *Anais...* Recife: Departamento de Informática/UFPE, 1996. p. 129-138.