Implemetação de um formato de rastro binário Pajé

Vinícius A. Herbstrith[1], **Lucas Mello Schnorr**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{vaherbstrith,schnorr}@inf.ufrgs.br

Resumo. *Este artigo apresenta um formato binário para o arquivo de rastro Pajé, almejando a otimização da leitura e escrita de arquivos neste formato. A proposta inclui a implementação de dois conversores e um componente de leitura binário para a ferramenta de visualização PajeNG.*

1. Introdução

Com o desenvolvimento de aplicações paralelas e distribuídas surge o interesse da análise do seu desempenho através de eventos selecionados e representativos que são registrados durante sua execução. Tais informações de análise são registradas em arquivos de rastro de execução [Stein 2001]. Se torna necessário o registro de tempo de cada evento, sua identificação e outros dados que sejam de interesse para determinado evento.

O formato Pajé [Schnorr 2014b] é largamente utilizado para a análise e visualização de rastros. O formato se destaca no atributo extensibilidade, possibilitando seu uso nos mais diversos contextos de rastreamento, com diferentes necessidade, sem ser necessário modificar os componentes da ferramenta Pajé. No entanto, a maior desvantagem do formato é a sua estrutura textual, com uma maior intrusão durante o registro do rastro e também uma leitura lenta do arquivo de rastro no momento da análise. Tendo em vista a escalabilidade e o sempre crescente volume de dados que ocorre nos dias de hoje [Geller 2011], é de grande interesse superar esta desvantagem do formato Pajé.

Este artigo apresenta um formato binário para o arquivo de rastro Pajé, almejando a otimização da leitura e escrita de arquivos Pajé. As seções 2.1 e 2.2 apresentam o formato binário alternativo que foi proposto para Pajé, fazendo uso da biblioteca de rastros libRastro [da Silva et al. 2003] e Poti [Schnorr 2014c] para sua geração.

2. Proposta e implementação

A proposta inclui a implementação de dois conversores e um componente de leitura binário para a ferramenta de visualização PajeNG. Estes três componentes são detalhados na Figura 1, onde temos os dois conversores (*poti_reader* e *conversor*), desenvolvidos dentro da biblioteca Poti com a linguagem C, e o leitor binário (*PajeBinaryReader*), desenvolvido na ferramenta PajeNG em C++. Os componentes em cor cinza são aqueles que já fazem parte da PajeNG.

A biblioteca Poti foi modificada para executar chamadas às funções da libRastro. O rastro binário em Pajé segue portanto o mesmo padrão de formato gerado pela libRastro. A finalidade da ferramenta `poti_reader` é fazer a leitura do arquivo binário, usando as funções de decodificação da libRastro, e retornar o equivalente em formato textual.

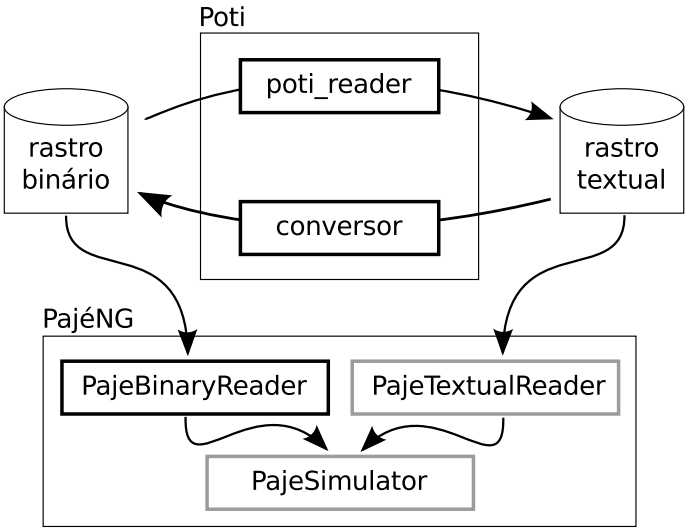


Figura 1. Componentes Poti implementados.

A ferramenta `conversor` faz a tradução de arquivos do formato textual Pajé para o novo formato binário. Na implementação deste, foram usados Flex e Bison, para fazer a análise léxica e sintática da entrada textual, e as funções da nova biblioteca Poti para gerar o arquivo de saída em formato binário equivalente.

A leitura do novo formato é feita dentro da biblioteca PajéNG [Schnorr 2014a], com o novo componente `PajeBinaryReader` que foi adicionado à mesma. A configuração da estrutura do novo formato binário foi feita com base nas definições usadas na simulação de rastros feita na PajéNG.

Com o uso desta estrutura e a natureza binária do arquivo, temos um ganho de desempenho considerável. Todavia, conversões dos dados da libRastro para o formato *string* padrão usado na simulação da PajéNG são necessárias, impactando negativamente na simulação.

2.1. Codificação em binário do cabeçalho

Na Figura 2 temos a definição de um cabeçalho no formato binário correspondente. No campo `type` é usado um número qualquer que deve ser obrigatoriamente diferente dos definidos no cabeçalho do arquivo Pajé (999, como no exemplo desta figura) para distinguir a declaração de evento dos demais eventos registrados.

As informações de campos e formatos Pajé é representada pelas definições que são usadas na biblioteca PajéNG em um vetor de inteiros. No exemplo, o primeiro valor do vetor é a identificação do evento (10, no exemplo), seguido por tuplas nome do campo e tipo de dado. No exemplo a primeira tupla da sequência é Time e `date`, sendo representada por 1 e 5 no formato binário, com as outras tuplas vindo a seguir.

```
%EventDef PajePushState 10
%       Time date
%       Container string
%       Type string
%       Value string
%EndEventDef
```



```
0 type: 999 ts: 1412540235.483523846 (id1=1,id2=3)
 u_int32_ts-> (10) (1) (5) (4) (0) (3) (0) (9) (0)
```

Figura 2. Correspondência textual/binário da definição de um evento Pajé.

2.2. Codificação em binário de eventos registrados

Na Figura 3 temos um exemplo da representação com a libRastro para um evento datado. A codificação é semelhante àquela da definição do cabeçalho.

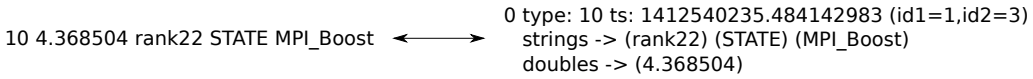
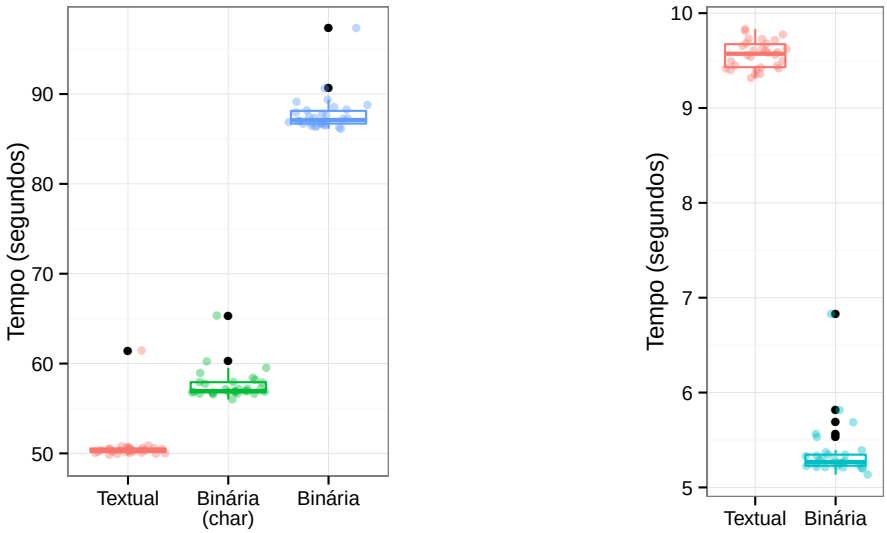


Figura 3. Correspondência textual/binário de um evento Pajé.

As informações de tempo e de identificação, geradas pela libRastro, não são utilizadas. Diferentemente da definição de cabeçalho, no evento a informação do campo `type` se refere à identificação do evento que foi definada no cabeçalho.

3. Resultados sobre o desempenho do `PajeBinaryReader`

O `PajeBinaryReader` é o componente principal da nossa proposta, pois impacta diretamente a análise de arquivos Pajé. Os resultados apresentados aqui mostram o desempenho obtido deste componente quando este funciona em cooperação com os demais componentes já existentes da biblioteca PajeNG. Para fins de comparação, usamos como referência o leitor já existente chamado `PajeTextualReader`. O programa `pj_validate` foi utilizado em todas as medidas. Ele lê e simula um arquivo de rastro dado como entrada. Foram feitas 30 execuções com um arquivo de rastro de 776 MB para cada experimento apresentado na Figura 4. Todos os experimentos foram executados em uma máquina com as seguintes características: processador Intel i7-4770 CPU (3.40GHz), HD 1TB SATA II (3.0Gb/s), 8GiB DIMM DDR3 (1.6GHz).



(a) Com simulação **(b) Sem Simulação**

Figura 4. Resultados dos diferentes experimentos

Pode se notar nos resultados da Figura 4a uma perda de desempenho de aproximadamente 75% do formato binário puro (87 segundos em média) em relação ao formato

textual padrão (51 segundos em média) quandos os testes compreendem a execução da simulação dos eventos que são lidos. Tal perda é atribuída à necessidade de converter todos os dados para o formato *string* que é usado pela simulação da PajeNG.

Como uma forma de obter um melhor resultado, foram feitos testes usando apenas a estrutura de caracteres da libRastro para salvar todas as informações de eventos registrados. Embora esta mudança apresente um desempenho melhor que o do formato binário original proposto, ainda sim perdemos um tempo considerável em conversões para o formato *string*, tendo um desempenho de por volta de 15% inferior à leitura tradicional como pode ser visto ainda na Figura 4a comparando o tempo Textual com o tempo do experimento Binário (char).

No entanto, como pode ser visto na Figura 4b, se compararmos apenas os tempos da leitura completa do arquivo, ou seja, a passagem do disco para a memória das informações de eventos, o formato binário tem um ganho de 40% sobre o formato textual padrão.

4. Conclusões

Neste artigo apresentamos o formato binário Pajé. Embora os resultados iniciais obtidos na leitura e simulação com o novo formato tenham sido negativos, ainda existem otimizações importantes visando o ganho de desempenho a serem feitas.

Com a continuação do trabalho na forma de alterações na estrutura interna de simulação da ferramente PajéNG, buscando tirar proveito da estrutura provida pela libRastro e eliminando conversões entre tipos de dados, esperamos um ganho expressivo de desempenh na simulação do rastro.

Outra otimização possível é a remoção de informações que não são usadas para a análise de rastro, mas que fazem parte da estrutura da libRastro. Com tais remoções, também se espera mais ganho em desempenho para o novo formato binário.

Referências

da Silva, G. J., Schnorr, L. M., and Stein, B. (2003). Jrastro: A trace agent for debugging multithreaded and distributed java programs. In *Proceedings of the 15th Symposium on Computer Architecture and High Performance Computing*, pages 46–54. Los Alamitos: IEEE Computer Society.

Geller, T. (2011). Supercomputing's exaflop target. In *Comm. of the ACM*, page 54(8).

Schnorr, L. M. (2014a). Pajeng. `https://github.com/schnorr/pajeng`.

Schnorr, L. M. (2014b). Pajé trace file format. `http://paje.sf.net`.

Schnorr, L. M. (2014c). Poti. `http://github.com/schnorr/poti`.

Stein, B. (2001). Depuração de programas paralelos. In *I Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul*, pages 151 – 176, Gramado, RS. Sociedade Brasileira de Computação (SBC).