

Implementação de um formato binário Pajé

Vinícius A. Herbstrith

Orientação: Lucas M Schnorr



ERAD/RS 2015

Gramado, 24 de abril de 2015

Contexto

- Sistemas de computação grandes e complexos
 - Alto desempenho
 - Computação distribuída

Contexto

- Sistemas de computação grandes e complexos
 - Alto desempenho
 - Computação distribuída
- Grandes Aplicações paralelas e distribuídas
 - Espaço: Milhares de processos
 - Tempo: grande número de eventos

Contexto

- Tianhe-2
 - 3.120.000 cores $\rightarrow$ 33.86 Pflops



Contexto

- Tianhe-2

- 3.120.000 cores $\rightarrow$ 33.86 Pflops



- Cimatec01

- 17200 cores $\rightarrow$ 0.41 Pflops



Contexto

- Tianhe-2

- 3.120.000 cores → 33.86 Pflops



- Cimatec01

- 17200 cores → 0.41 Pflops



- Boinc computação voluntária

- 674.060 computadores → 7.495 Pflops



Análise de desempenho

- Objetivo do paralelismo
 - Obter o maior desempenho possível das aplicações

Análise de desempenho

- Objetivo do paralelismo
 - Obter o maior desempenho possível das aplicações
- Workflow



Análise de desempenho

- Objetivo do paralelismo

- Obter o maior desempenho possível das aplicações

- Workflow



- Coleta de dados

- Amostragem, perfilagem, rastro

Dados de Aplicações

- Ondes3D: Propagação de ondas sísmicas em 3D
 - 50s rodando → 100k eventos

Dados de Aplicações

- Ondes3D: Propagação de ondas sísmicas em 3D
 - 50s rodando → 100k eventos
- LU.A.32: Resolução por gauss-seidel
 - 4.79s rodando → 7 milhões de eventos aproximadamente

Dados de Aplicações

- Ondes3D: Propagação de ondas sísmicas em 3D
 - 50s rodando → 100k eventos
- LU.A.32: Resolução por gauss-seidel
 - 4.79s rodando → 7 milhões de eventos aproximadamente
- Naïve Particle Simulator(BSP based)
 - 6.26s rodando → 200 milhões de eventos

Motivação

- Escalabilidade
- Intrusão do rastro

Outline

- Formato Paje
- Formato binário Paje
 - Librastros, Poti, PajeNG
- Primeira implementação
- Segunda implementação

Formato Paje

■ Formato do arquivo paje

```
%EventDef SendMessage 21
%      Time date
%      ProcessId int
%      Receiver int
%      Size int
%EndEventDef
```

```
%EventDef UnblockProcess 17
%      Time date
%      ProcessId int
%      LineNumber int
%      FileName string
%EndEventDef
```

```
21 3.233222 5 3 320
17 5.123002 5 98 sync.c
```

Formato Paje

- Prós
 - Extensibilidade

Formato Paje

- Prós

- Extensibilidade

- Contras

- Estrutura textual
 - Maior intrusão

Formato binário Paje: Librastro + Paje

- Librastro
 - Biblioteca genérica de rastro binário

Formato binário Paje: Librastros + Paje

■ Librastros

■ Biblioteca genérica de rastros binários

```
%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)
```

```
10 4.368504 rank22 STATE MPI_Boost
```



```
0 type: 10 ts: 1412540235.484142983 (id1=1,id2=3)
```

```
strings -> (rank22) (STATE) (MPI_Boost)
```

```
doubles -> (4.368504)
```

Formato binário Paje: Poti + librastro

- Poti
 - Biblioteca de criação de rastros em Formato Paje

Formato binário Paje: Poti + librastro

- Poti
 - Biblioteca de criação de rastros em Formato Paje
- Integração com a librastro
 - Criação de arquivos rst(formato librastro)
 - Conversão de arquivos Paje Textual para o novo formato binário
 - Conversão de arquivos rst para Paje Textual

Formato binário Paje: PajeNG + librastro

- PajeNG
 - Ferramenta de visualização de rastros no formato Paje

Formato binário Paje: PajeNG + librastro

- PajeNG
 - Ferramenta de visualização de rastros no formato Paje
- Integração com a librastro
 - Leitura e interpretação do arquivo rst gerado pela Poti

Primeira implementação

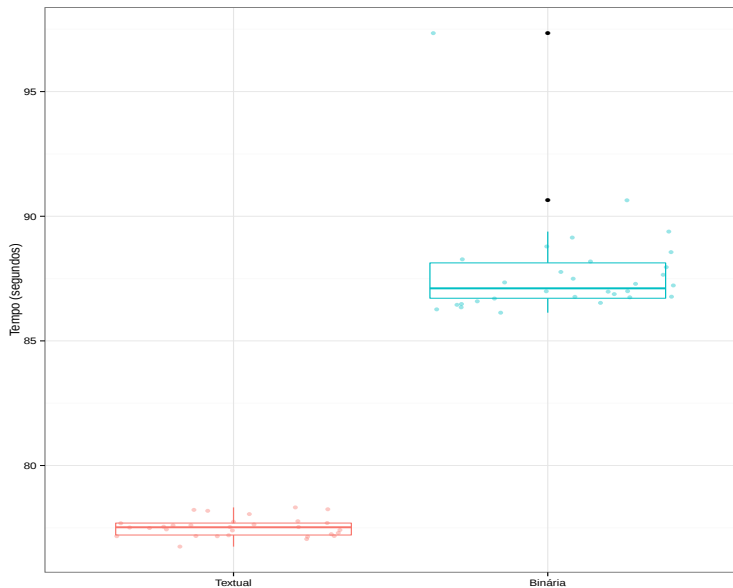
- Mínimo de alterações nas bibliotecas librastró e PajeNG

Primeira implementação

- Mínimo de alterações nas bibliotecas libastro e PajeNG
- PajeNG
 - Leitura dos eventos do arquivo .rst e sua conversão para a classe PajeTraceEvent

PajeTraceEvent Evento rst
paje_line *valueLine ← strings : (rank22) (STATE) (MPI_Boost)
 doubles: (4.368504)

Desempenho da leitura e simulação



Segunda Implementação

■ Alteração libastro

10 4.368504 rank22 STATE MPI_Boost



0 type: 10 ~~ts: 1412540235.404142903~~ (id1=1,id2=3)
strings -> (rank22) (STATE) (MPI_Boost)
doubles -> (4.368504)

Segunda Implementação

■ Alteração librastró

10 4.368504 rank22 STATE MPI_Boost $\longleftrightarrow$ 0 type: 10 ~~ts: 1412540235.404142903~~ (id1=1,id2=3)
strings -> (rank22) (STATE) (MPI_Boost)
doubles -> (4.368504)

■ Alteração PajeNG

■ Nova classe: PajeRastroTraceEvent

PajeRastroTraceEvent

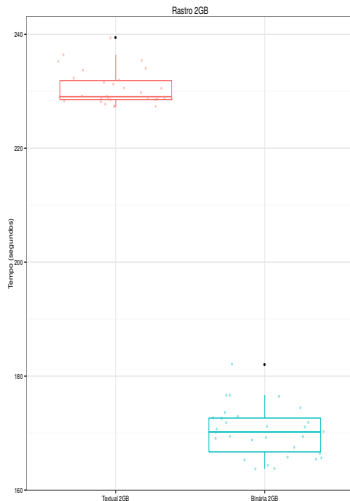
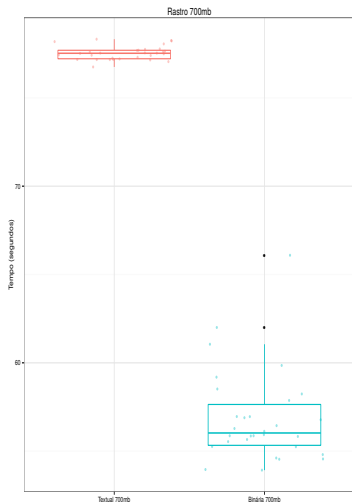
char v_string[]
double v_double[]



Evento rst

strings : (rank22) (STATE) (MPI_Boost)
doubles: (4.368504)

Desempenho da leitura e simulação



Conclusões

- Ganho de 22% no tempo de leitura e simulação do formato binário contra o textual.

Conclusões

- Ganho de 22% no tempo de leitura e simulação do formato binário contra o textual.
- Trabalho futuro
 - Experimentos sobre a intrusão

Obrigado pela atenção

- Contato: vaherbstrith@inf.ufrgs.br