Binary Pajé Trace Format Proposal

Scientific Initiation scholar: Vinícius A. Herbstrith
Professor adviser: Lucas M Schnorr

WSPPD 2015
Porto Alegre, August 21th, 2015

Context

- Computation systems are getting bigger and more complex
  - High performance
  - Distributed computing

Context

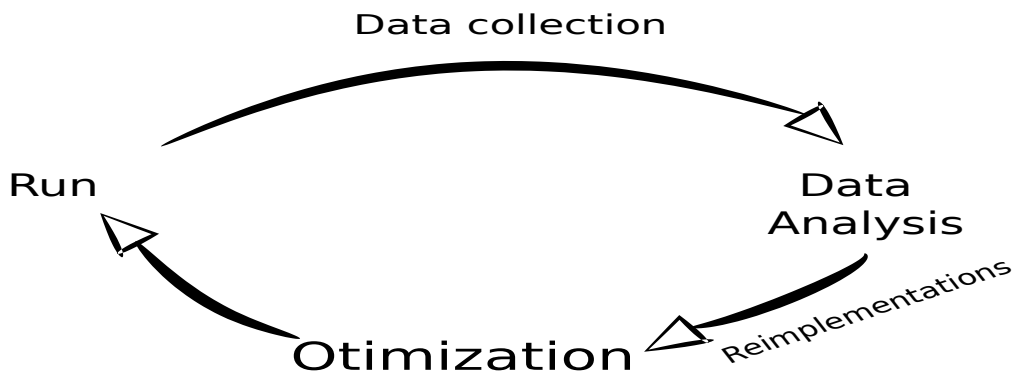
- Computation systems are getting bigger and more complex
  - High performance
  - Distributed computing
- Parallel and distributed applications also get bigger
  - Space: Millions of processes
  - Time: Big number of events

Performance Analysis

- Parallelism Objective
  - Achieve the best possible performance for the application

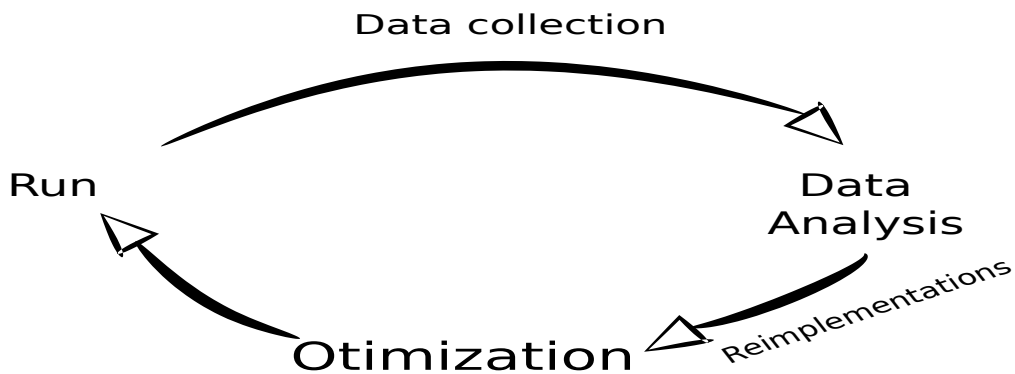
Performance Analysis

- Parallelism Objective
  - Achieve the best possible performance for the application
- Workflow



Performance Analysis

- Parallelism Objective
  - Achieve the best possible performance for the application
- Workflow



  -
- Data collection
  - Sampling, profilling, tracing

Application Collected Data

- Ondes3D: 3D Sismic propagation
  - 50s running $\rightarrow$ 100k events

Application Collected Data

- Ondes3D: 3D Sismic propagation
  - 50s running $\rightarrow$ 100k events
- LU.A.32: Gauss-Seidel method for solving linear equations
  - 4.79s running $\rightarrow$ 7 million events

Application Collected Data

- Ondes3D: 3D Sismic propagation
  - 50s running $\rightarrow$ 100k events
- LU.A.32: Gauss-Seidel method for solving linear equations
  - 4.79s running $\rightarrow$ 7 million events
- Naïve Particle Simulator(BSP based)
  - 6.26s running $\rightarrow$ 200 million events

Motivation

- Pajé Format
  - Achieve scalability through better performance on file reading

Outline

- Pajé trace file format
- Binary Pajé trace file format
  - Librastro, Poti, PajeNG
- Experiments and performance analysis
- Conclusion

Pajé trace file format

- Pajé trace file format

```
%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
```

Pajé trace file format

- Pros
  - Extensibility

Pajé trace file format

- Pros
  - Extensibility
- Cons
  - Textual nature
  - Higher intrusion

Binary Pajé trace file format

- Two version:

Binary Pajé trace file format

- Two version:
  - Standard version
    - Direct translation of the Pajé trace format

Binary Pajé trace file format

- Two version:
  - Standard version
    - Direct translation of the Pajé trace format
  - Reference version
    - Makes use of references from integer values to string values

Binary Pajé trace file format: libRastro + Pajé

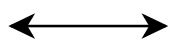
- LibRastro
  - Generic tracing library to register the behavior of any computer program

Binary Pajé trace file format: libRastro + Pajé

- LibRastro
  - Generic tracing library to register the behavior of any computer program

Header Conversion

```
%EventDef PajeDefineContainerType 0
%       Alias string
%       Type string
%       Name string
```

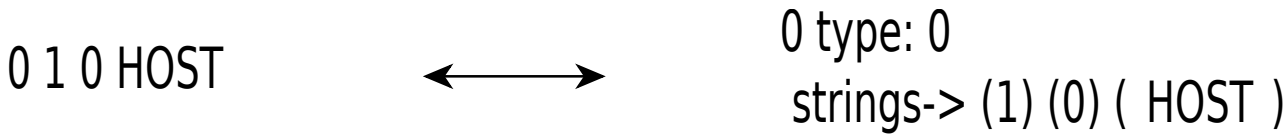


```
0 type: 999
 u_int32_ts-> (0) (11) (0) (3) (0) (2) (0)
```

Pajé binary trace file format: libRastro + Pajé

- LibRastro
  - Generic tracing library to register the behavior of any computer program

Event Conversion on standard version

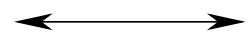


Binary Pajé trace file format: libRastro + Pajé

- LibRastro
  - Generic tracing library to register the behavior of any computer program

Event Conversion on references version

```
0 1 0 HOST
```



```
0 type: 888
 strings-> ( HOST )
0 type: 888
 strings-> (0)
0 type: 888
 strings-> (1)
0 type: 0
 u_int16_ts-> (2) (1) (0)
```

Binary Pajé trace file format: Poti + libRastro

- Poti
  - Library to create trace files in the Pajé file format

Binary Pajé trace file format: Poti + libRastro

- Poti
  - Library to create trace files in the Pajé file format
- libRastro integration
  - Create .rst files(libRastro format)
  - Convert Pajé text files to the new binary format
  - Convert Pajé binary files to Paje text

Binary Pajé trace file format: PajeNG + libRastro

- PajeNG
  - Visualization tool for the analysis of Pajé trace files

Binary Pajé trace file format: PajeNG + libRastro

- PajeNG
  - Visualization tool for the analysis of Pajé trace files
- libRastro integration
  - Reading and decodification of .rst files generated on Poti
  - New class: PajeRastroTraceEvent
  - New class: PajeRastroReader
    - Standard: direct read
    - Reference: resolve the integer and string references

Experiments

- Computer:
  - Intel i7-4770 CPU (3.40GHz)
  - HD 1TB SATA II (3.0Gb/s)
  - 8GiB DIMM DDR3(1.6GHz)

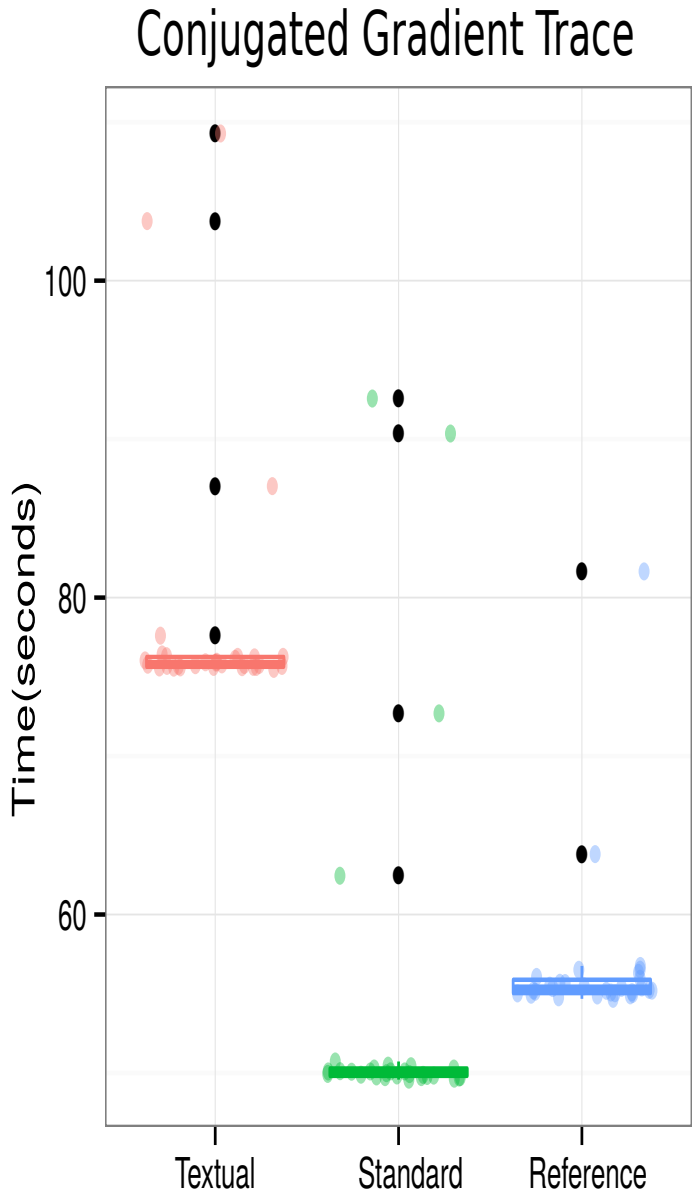
Experiments

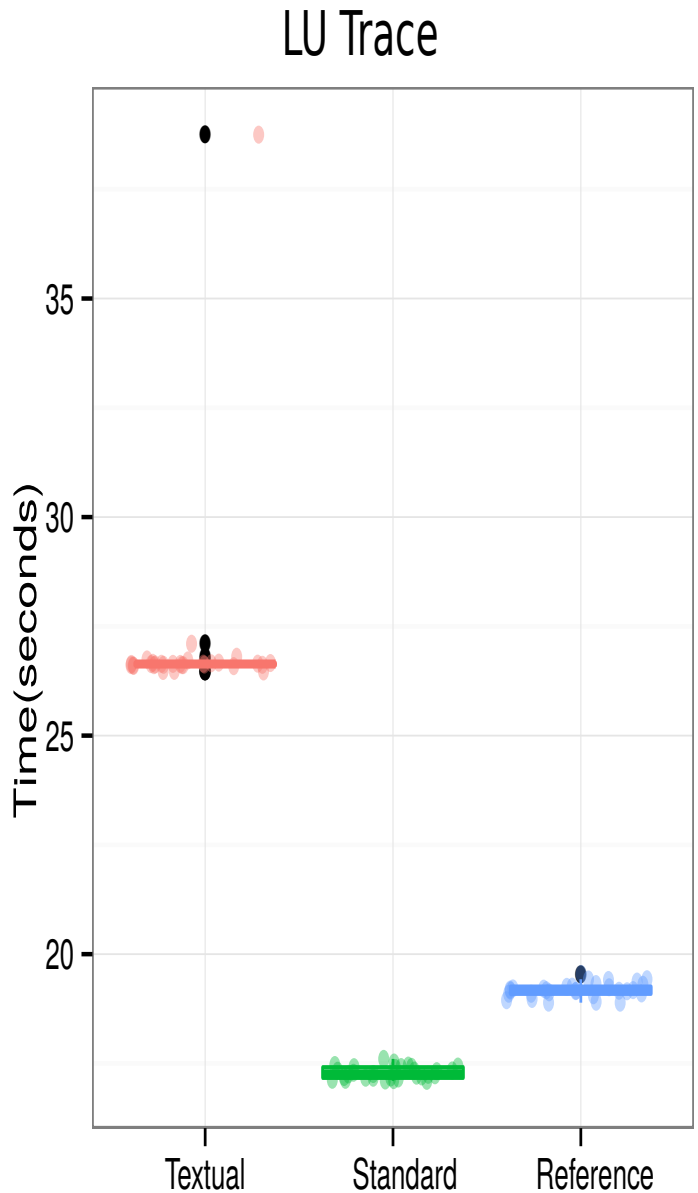
- Computer:
  - Intel i7-4770 CPU (3.40GHz)
  - HD 1TB SATA II (3.0Gb/s)
  - 8GiB DIMM DDR3(1.6GHz)
- Trace files:
  - LU decomposition application which generated a 0.77 GB trace file
  - Conjugated gradient application which generated a 2.1 GB trace file

Experiments

- Computer:
  - Intel i7-4770 CPU (3.40GHz)
  - HD 1TB SATA II (3.0Gb/s)
  - 8GiB DIMM DDR3(1.6GHz)
- Trace files:
  - LU decomposition application which generated a 0.77 GB trace file
  - Conjugated gradient application which generated a 2.1 GB trace file
- 30 executions for each trace file in it's textual, binary standard and binary reference version

Performance analisys





Conclusion

- 32% time reduction for the binary standard version
- 27% time reduction for the binary reference version

Conclusion

- 32% time reduction for the binary standard version
- 27% time reduction for the binary reference version
- File size:
  - 48% decrease on the the LU decomposition trace file
  - 43% decrease on the conjugated gradient trace file

Conclusion

- 32% time reduction for the binary standard version
- 27% time reduction for the binary reference version
- File size:
  - 48% decrease on the the LU decomposition trace file
  - 43% decrease on the conjugated gradient trace file
- Future work
  - Experiments on instrusion
  - Compression algorithms

Thank you for your attention

- Contact: vaherbstrith@inf.ufrgs.br