Speeding up Trace Compass

TRACE COMPASS

Loïc Prieur-Drevon

December 10, 2015

École Polytechnique de Montréal

Laboratoire **DORSAL**

Agenda

Context

- State System
- State History Tree (SHT)

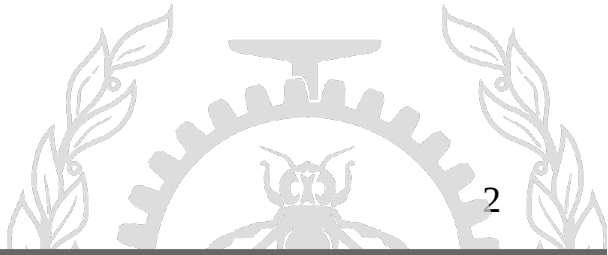
Approach

- Tools
- Known Bugs

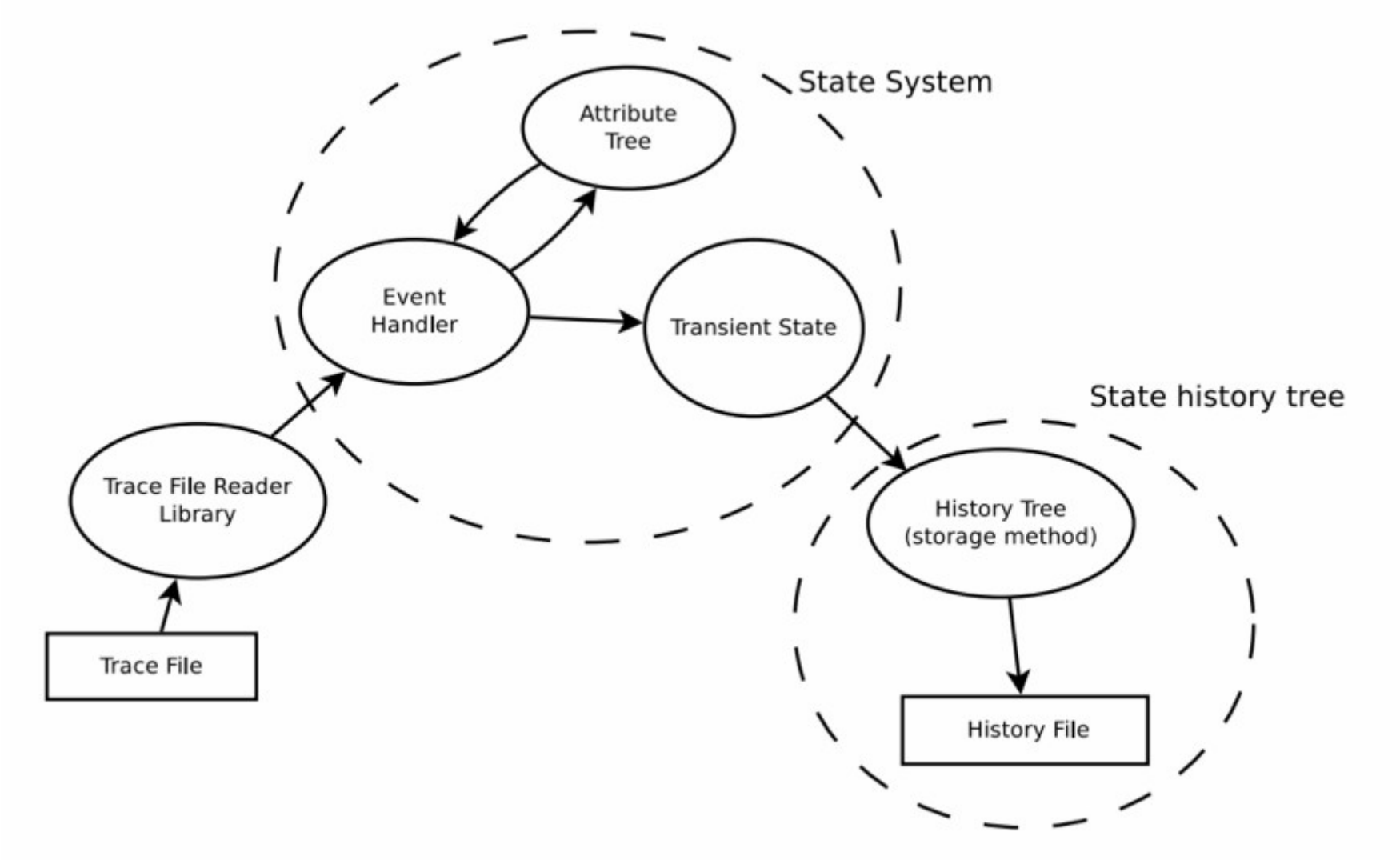
In-progress

- Overlapping SHT

Future Work

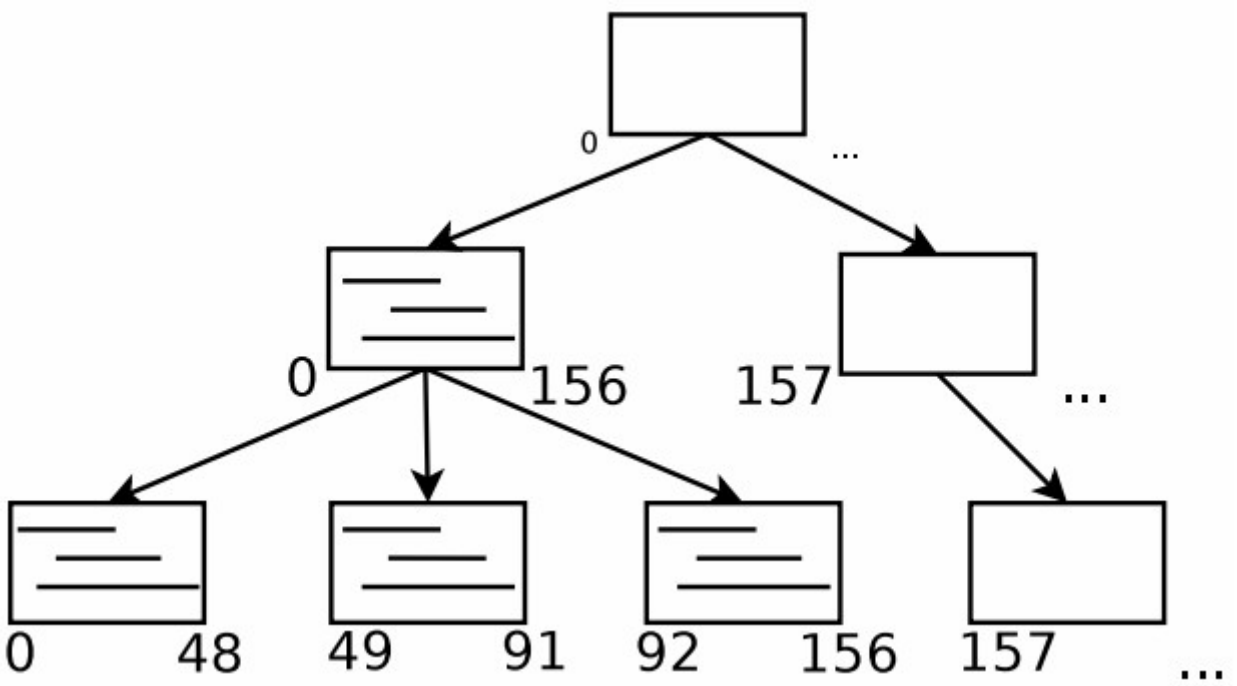


State Sytem



Source: State History Tree : an Incremental Disk-based Data Structure for Very Large Interval Data, Alexandre Montplaisir

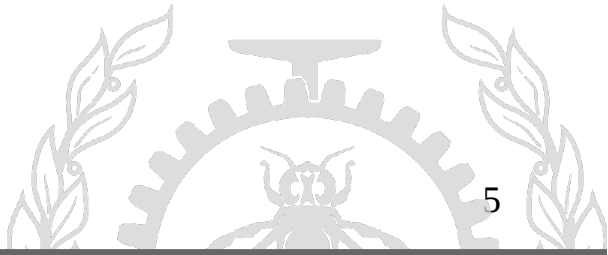
State Sytem : State History Tree (SHT)



Source: State History Tree : an Incremental Disk-based Data Structure for Very Large Interval Data, Alexandre Montplaisir

Approach

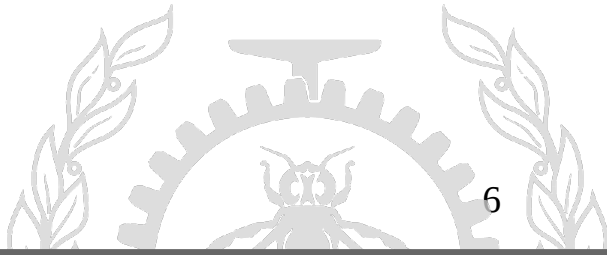
1. State System Tools
2. Know State System Issues
3. Literature Review
4. Work upwards
   - From Data Structure
   - To Trace Compass UI



SHT Benchmark

- Trace Size
- SHT Size
- SHT Build Time
- Number of Attributes
- Number of Nodes
- Number of Intervals
- Depth

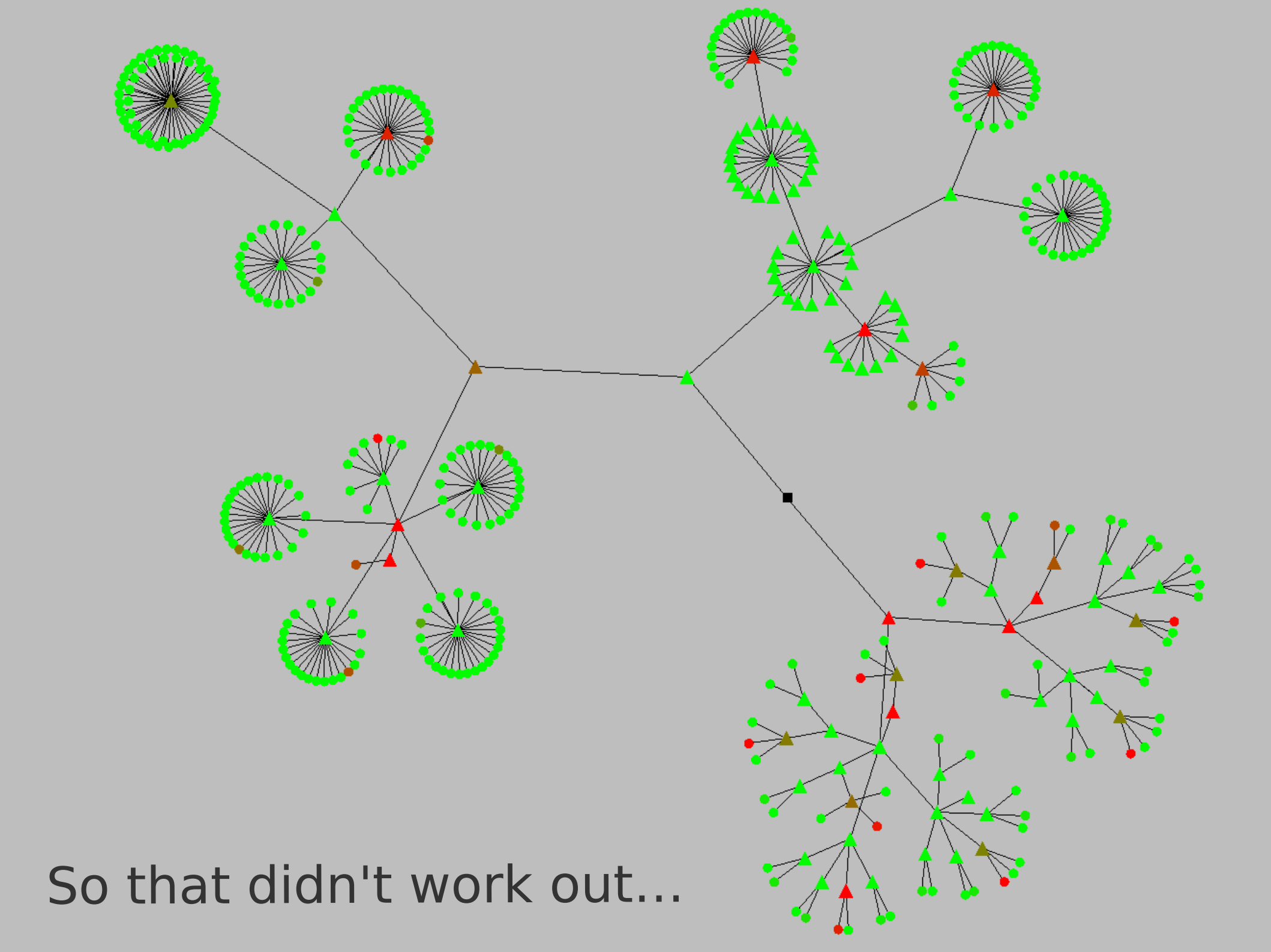
- Node Fill
- Number of NullStateValues
- Quark Locality
- Query :
  - Time
  - Nodes Searched



Visualization

Library : GraphStream

Demo : http://secretaire.dorsal.polymtl.ca/~lprieur/

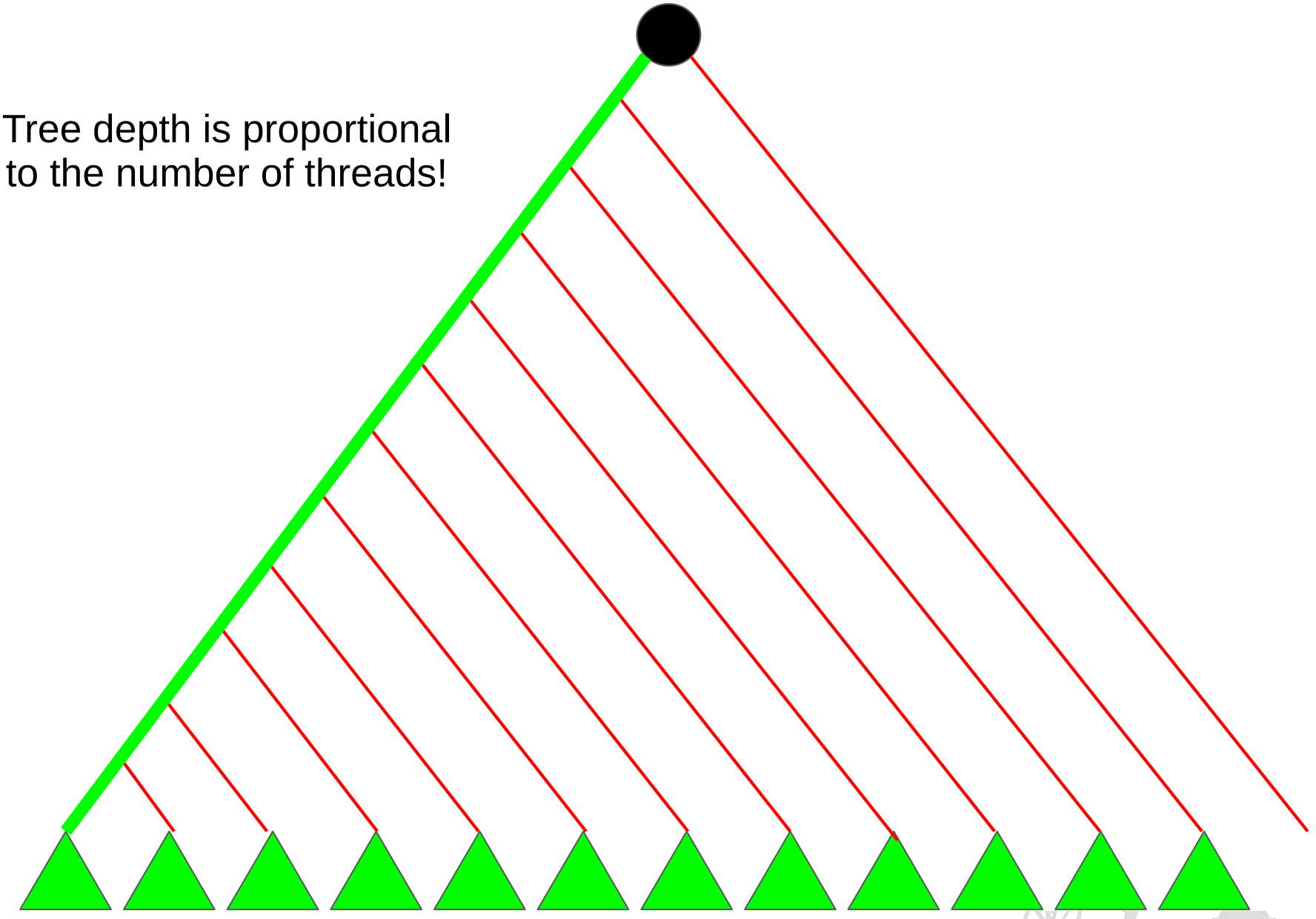

So that didn't work out...

StdOut

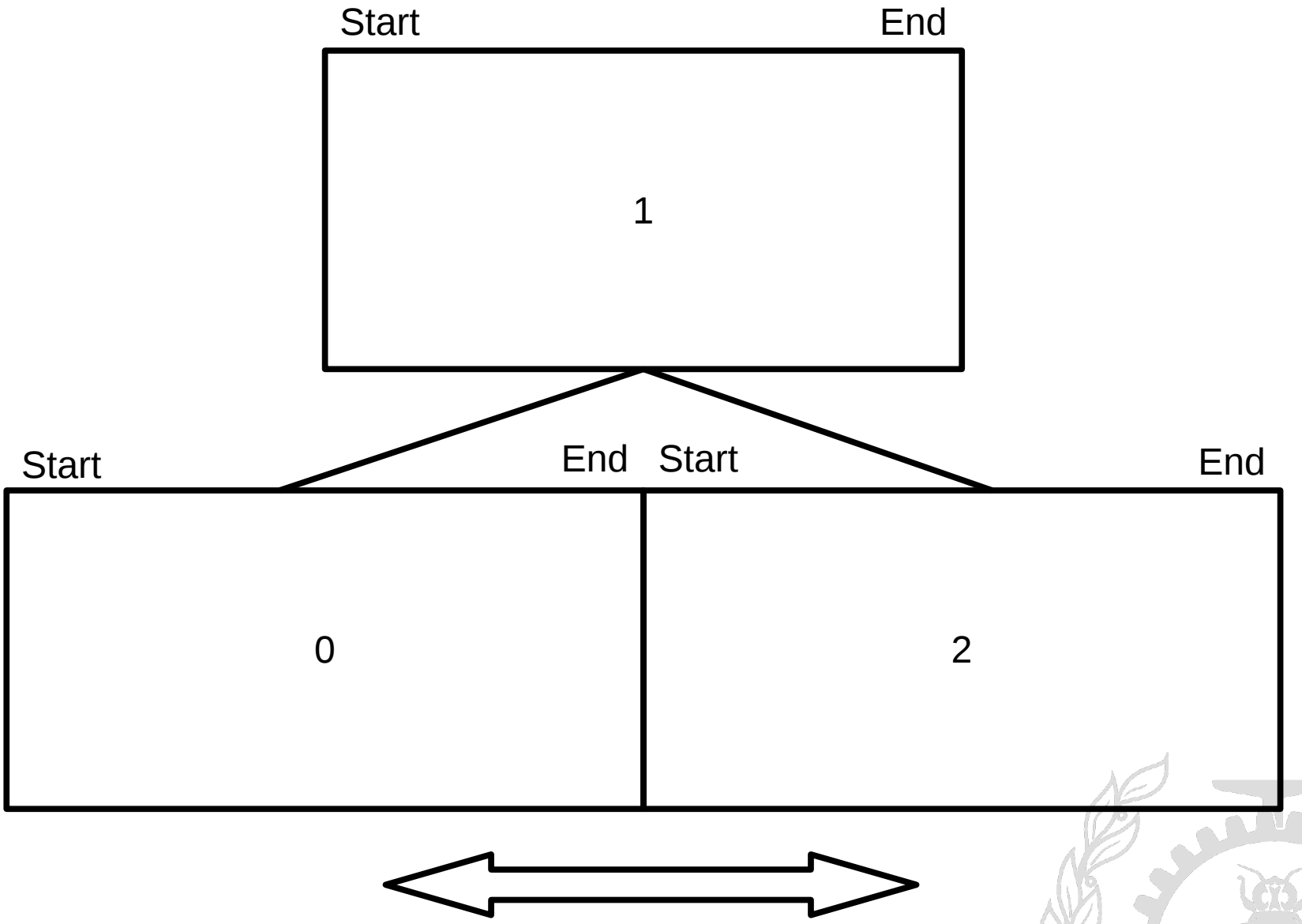
Node #2776, Core Node, 1 children, 0 intervals (0% used), [14242719969238176 44 - 14242720009356649 41]
Node #2777, Core Node, 2 children, 831 intervals (33% used), [14242719969238176 44 - 14242720009356649 41]
Node #2742, Core Node, 4 children, 2579 intervals (99% used), [14242719969238176 44 - 14242720009356649 41]
Node #2748, Core Node, 2 children, 2479 intervals (99% used), [14242719969238176 44 - 14242720009356649 41]
Node #2751, Core Node, 2 children, 2469 intervals (99% used), [14242719969238176 44 - 14242720009356649 41]
Node #2743, Core Node, 4 children, 2482 intervals (99% used), [14242719969238176 44 - 14242720009356649 41]
Node #2754, Core Node, 3 children, 1149 intervals (48% used), [14242719982171528 70 - 14242720009356649 41]
Node #2641, Core Node, 22 children, 1920 intervals (99% used), [14242719993270732 30 - 14242720009217916 89]
Node #2593, Core Node, 23 children, 1916 intervals (99% used), [14242719992584530 73 - 14242720008515401 09]
Node #2571, Core Node, 21 children, 1910 intervals (99% used), [14242719992239640 73 - 14242720008138055 65]
Node #2095, Core Node, 2 children, 1859 intervals (99% used), [14242719986075233 22 - 14242720002173219 22]
Node #166, Core Node, 17 children, 2020 intervals (99% used), [14242719969238176 44 - 14242720006864382 61]
Node #2121, Core Node, 24 children, 333 intervals (14% used), [14242720001542206 90 - 14242720002173219 22]
Node #2096, Core Node, 24 children, 1921 intervals (99% used), [14242719986075233 22 - 14242720001771637 07]
Node #167, Core Node, 21 children, 2568 intervals (99% used), [14242719969238176 44 - 14242719979207781 68]
Node #2468, Core Node, 2 children, 2513 intervals (99% used), [14242719969238176 44 - 14242720009356649 41]
Node #2221, Core Node, 11 children, 17 intervals (0% used), [14242719987759077 24 - 14242720006864382 61]
Node #2461, Core Node, 6 children, 482 intervals (25% used), [14242719991259240 51 - 14242720006864382 61]

Traces with many threads : SHT Structure

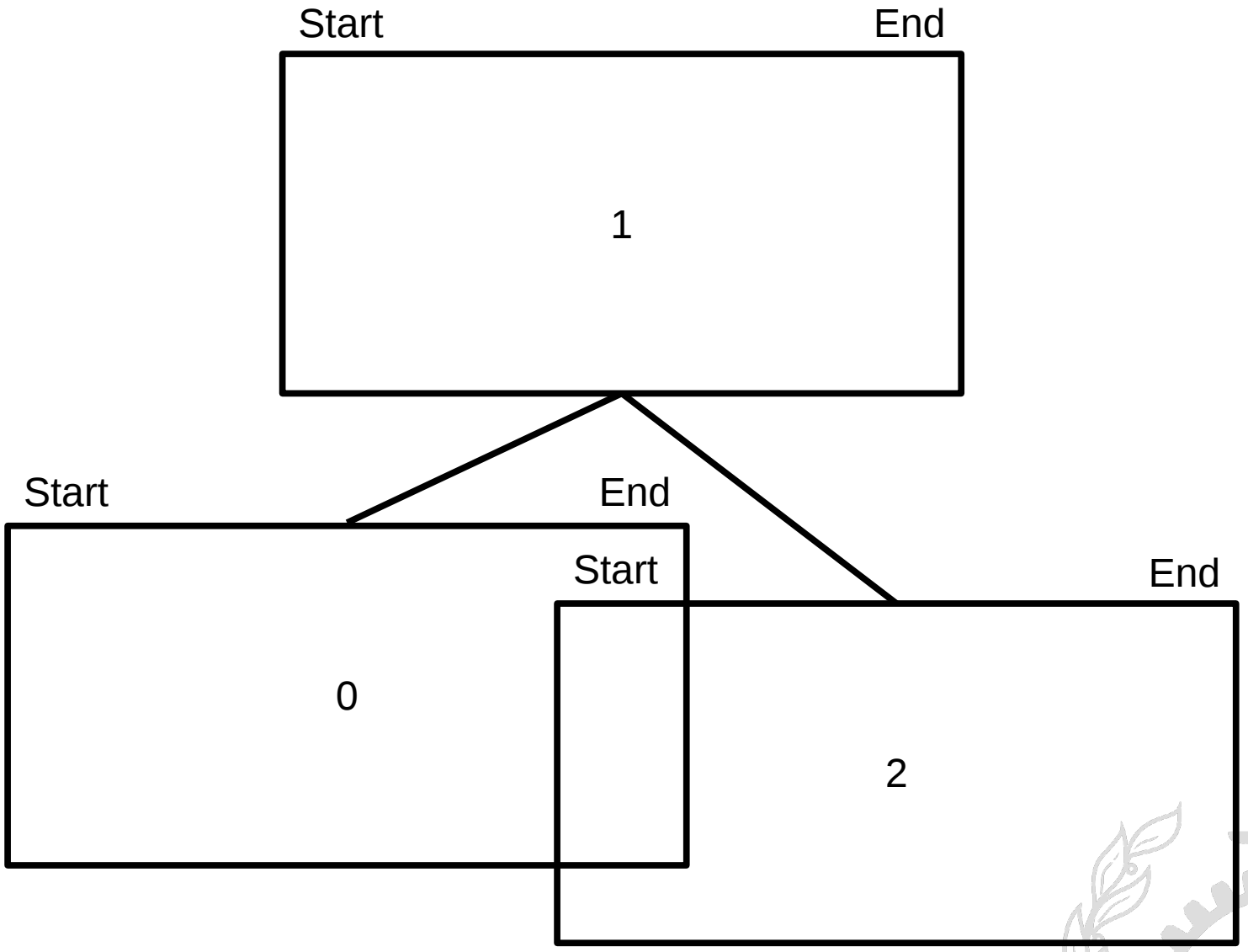
Tree depth is proportional
to the number of threads!



Traces with many threads : Explanation

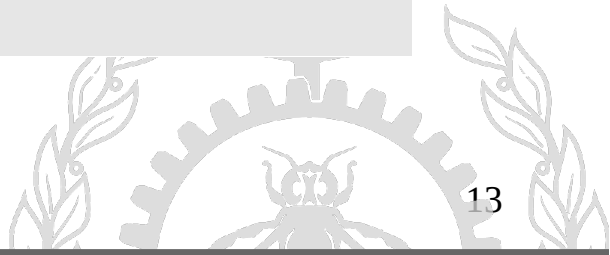


Traces with many threads : Modification



Traces with many threads : Fix

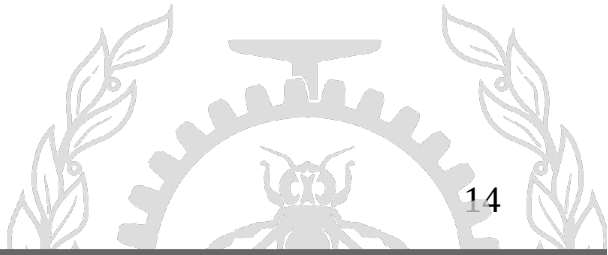
	Baseline 1.1 (Before)	Overlapping Nodes (After)	Gain
Build Time (ms)	**28 845**	**12 222**	**57.6%**
Nodes	**23 716**	**2 788**	**88.2%**
SHT Size (MiB)	1 486	178	**88.0%**
Average Fill	**9%**	**95%**	**x10.5**
Depth	206	7	96.6%
Single Query	**84 Nodes Searched**	**41 Nodes Searched**	**51.2%**



Overlapping Nodes - Benefits

- Denser State History Tree
- Faster Build Times
- Faster Queries
- Multithreaded Queries (future work)
- Shallower Tree → Shorter Branches:

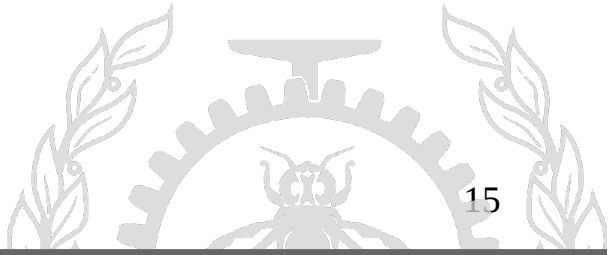
  Bug 430050 - [LTTng] OutOfMemoryError when opening a trace with a lot of streams



Alternative : Splitting Intervals (Future Work)

Reasoning: long intervals go higher up in tree : depth penalty

- When to split them?
- Which intervals to split?
- How to distinguish split intervals?
- Advantages/Disadvantages

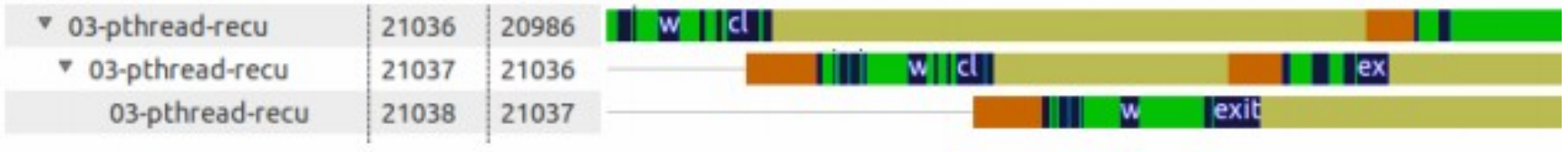


Future Work

- Removal:

	Intervals	Size	Depth
Constant Values	4%	50%	30%
Null Values	16%	60%	45%

- Queries:



Source: https://github.com/giraldeau/inf8601-scratchpad/tree/master/03-pthread-recursive

- Quark Locality
- Attribute Tree

Questions?
Use Cases?

loic.prieur-drevon@polymtl.ca