# Trace Compass Scalability Update

Arnaud Fiorini

Polytechnique Montréal
Laboratoire DORSAL

# Agenda

1. Background on Trace Compass backend
2. Alternative implementations
3. Performance benchmarks
4. Conclusion

# Background

- Interval Tree on disk
- Does not scale well with trace size
- Particulary with a lot of small intervals
- Create big intermediate files

# Alternative structure - Using tiles

- Allows to get only a few tiles to answer a request
- Slower for single interval queries but faster to answer 2d queries
- Use a minimal amount of extra data
- Can summarize states to ensure that we get an overview
- Two different approaches: constant size vs. constant duration

# Alternative structure - Using tiles

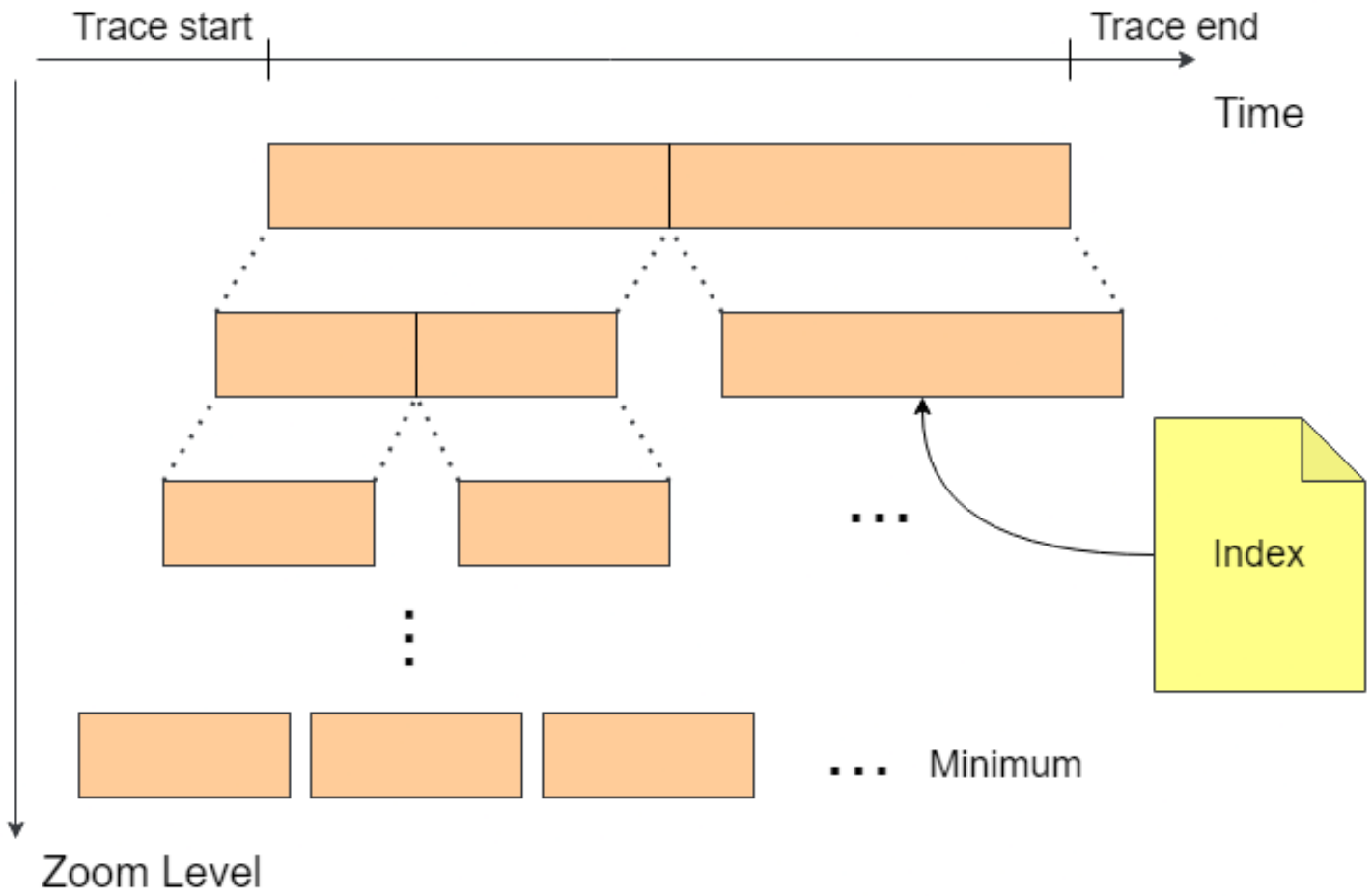


Figure: *History Tile Layers*

# Constant Size Tiles

- Ensure that all tiles are the same size on disk
- Requires a binary search to get the correct tiles
- Fewer tiles need to be stored
- Less extra data stored

# Constant Duration Tiles

- Ensure that all tiles are the same duration
- Search is constant time complexity
- Requires to store a potentially large index

# Experiment

- We compare three different versions:
  - Current implementation (full)
  - Constant size tiles (constant-size)
  - Constant duration tiles (constant-duration)
- We run a state system analysis on 7 traces of different sizes (8MB - 7GB) using each back-end
- The result of the analysis is queried for 4 different zoom levels (showing 95%, 50%, 2.5% and 0.5%) and multiple duration windows
- The queries were made between two separated nodes in a SSH tunnel

# Results

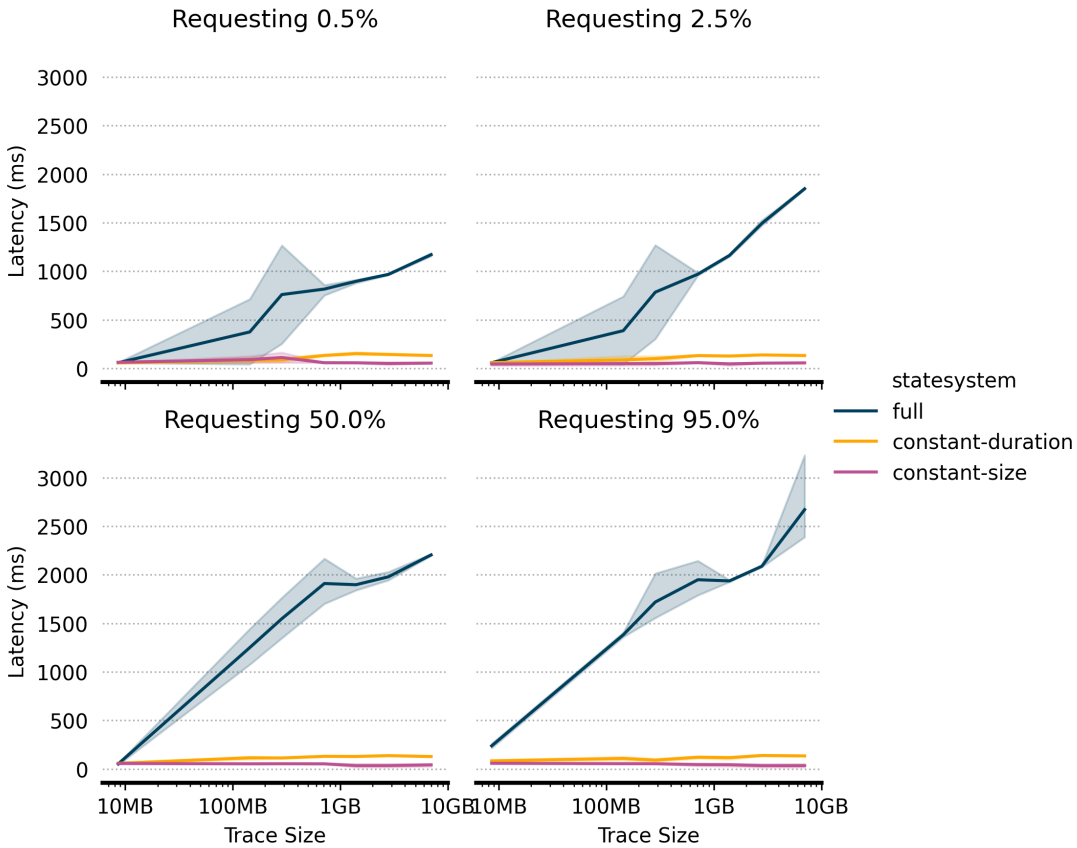


Figure: Request latencies per back-end type

# Results

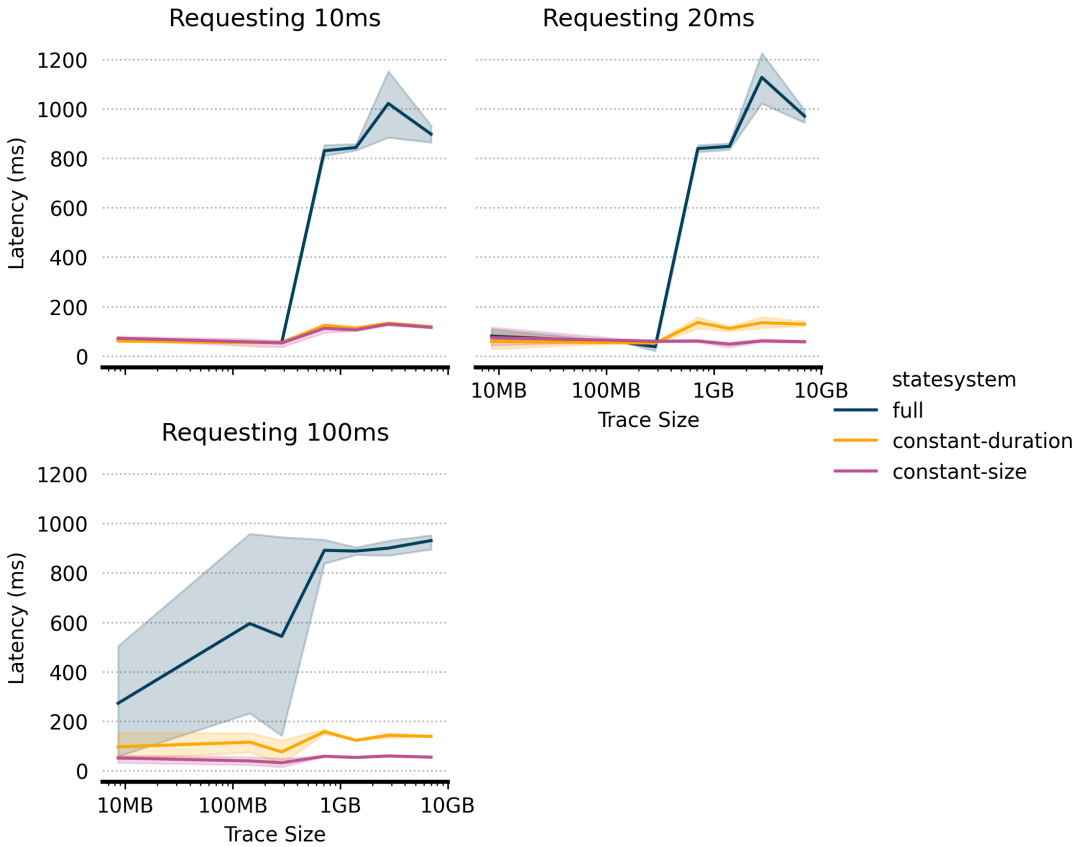


Figure: Request latencies per back-end type

# Results

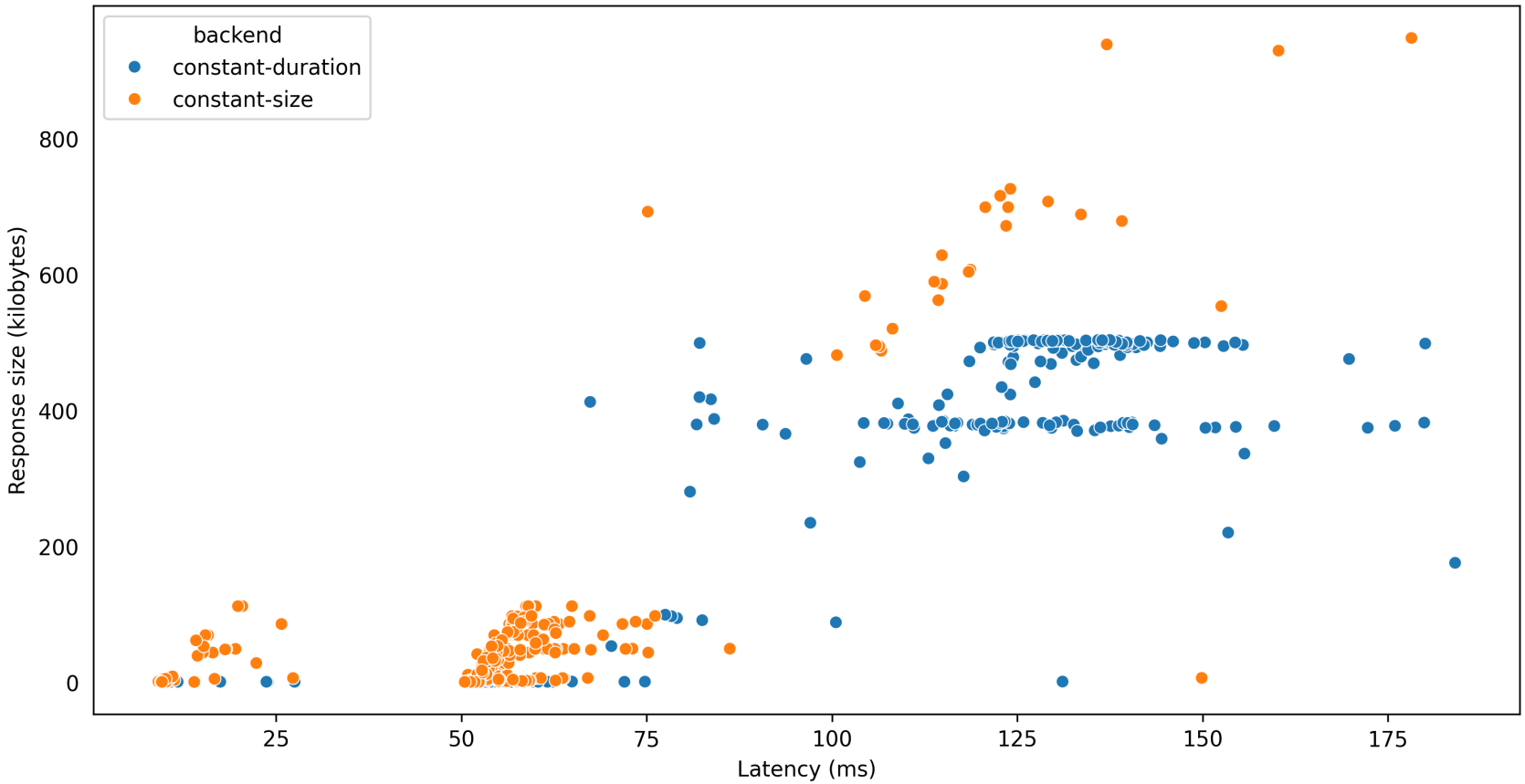


Figure: Request latencies vs. Response size

# Results

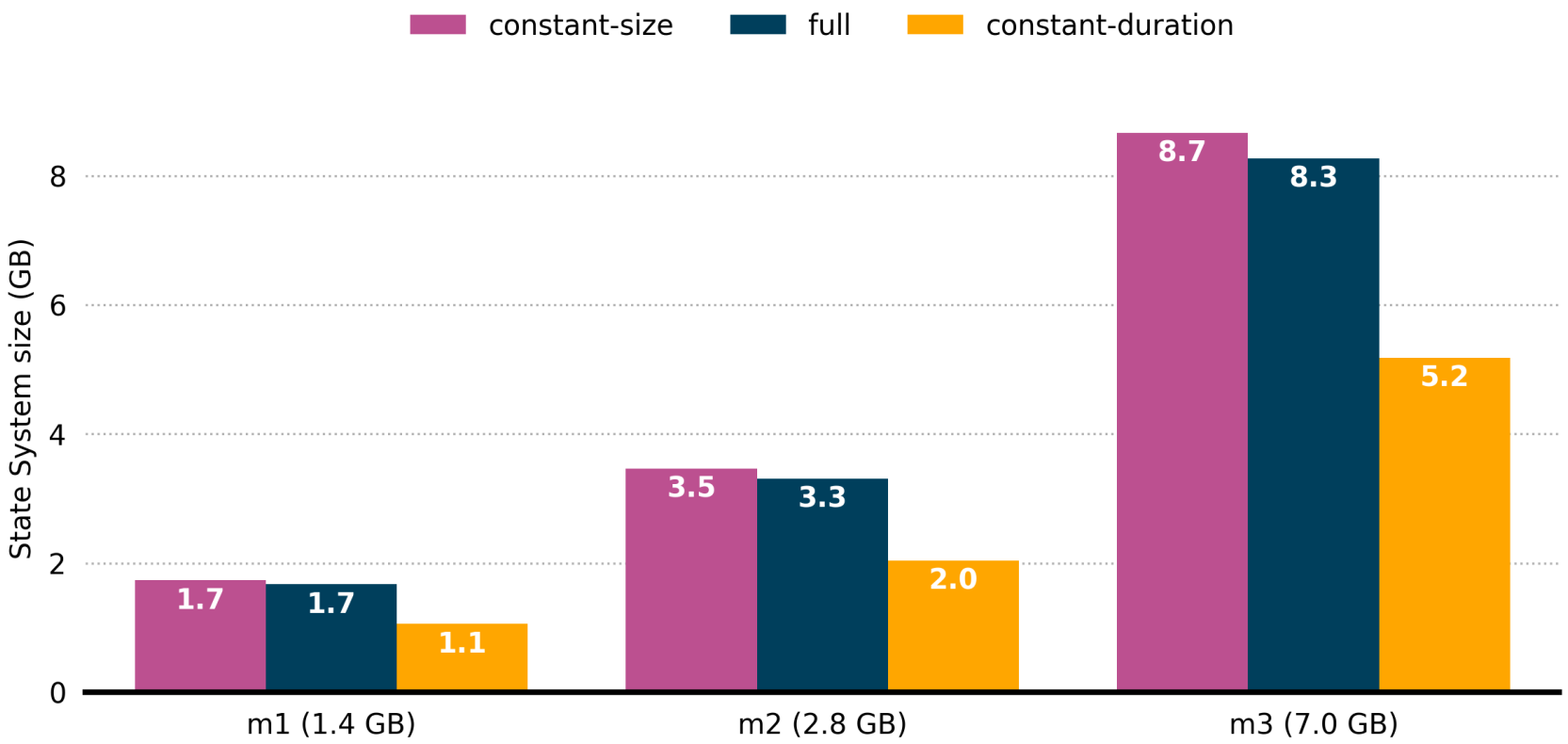


Figure: State system file sizes per trace analyzed

# Results

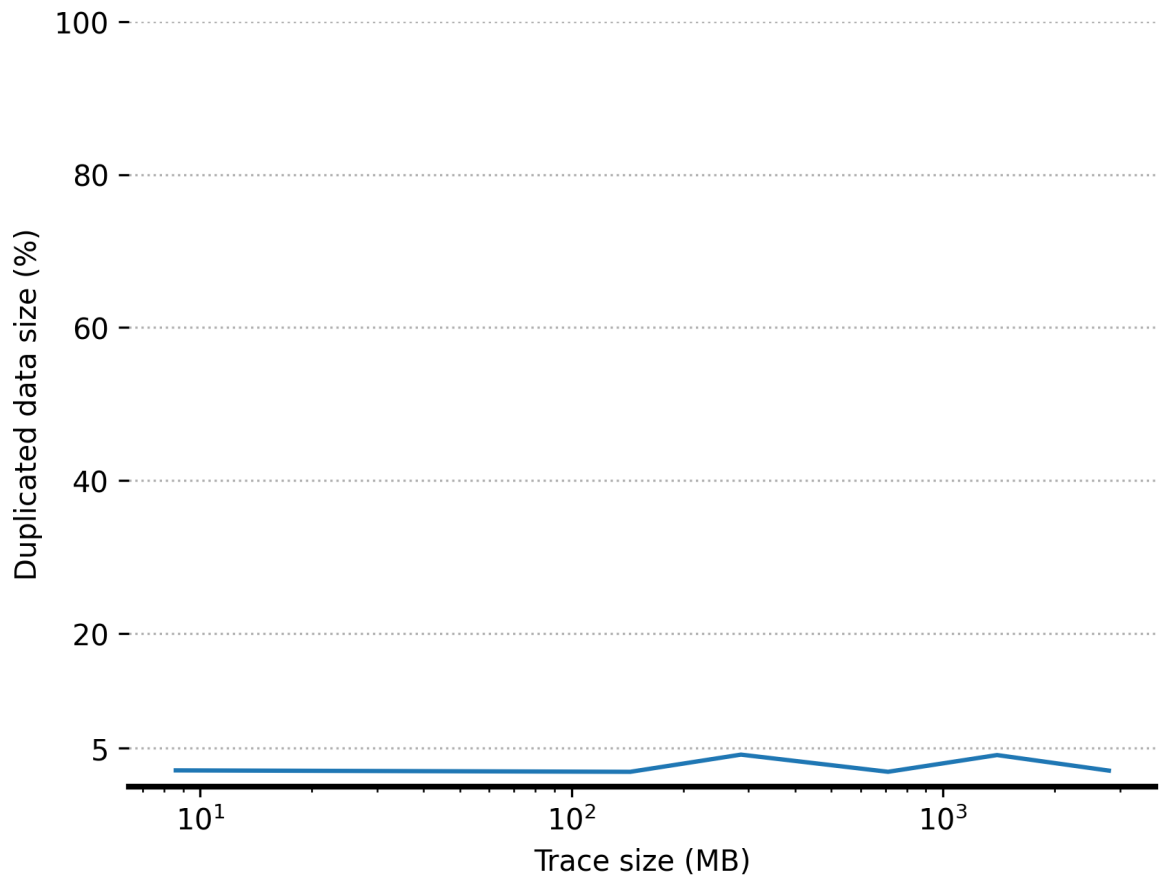


Figure: Duplicated data for the constant-duration backend

# Results

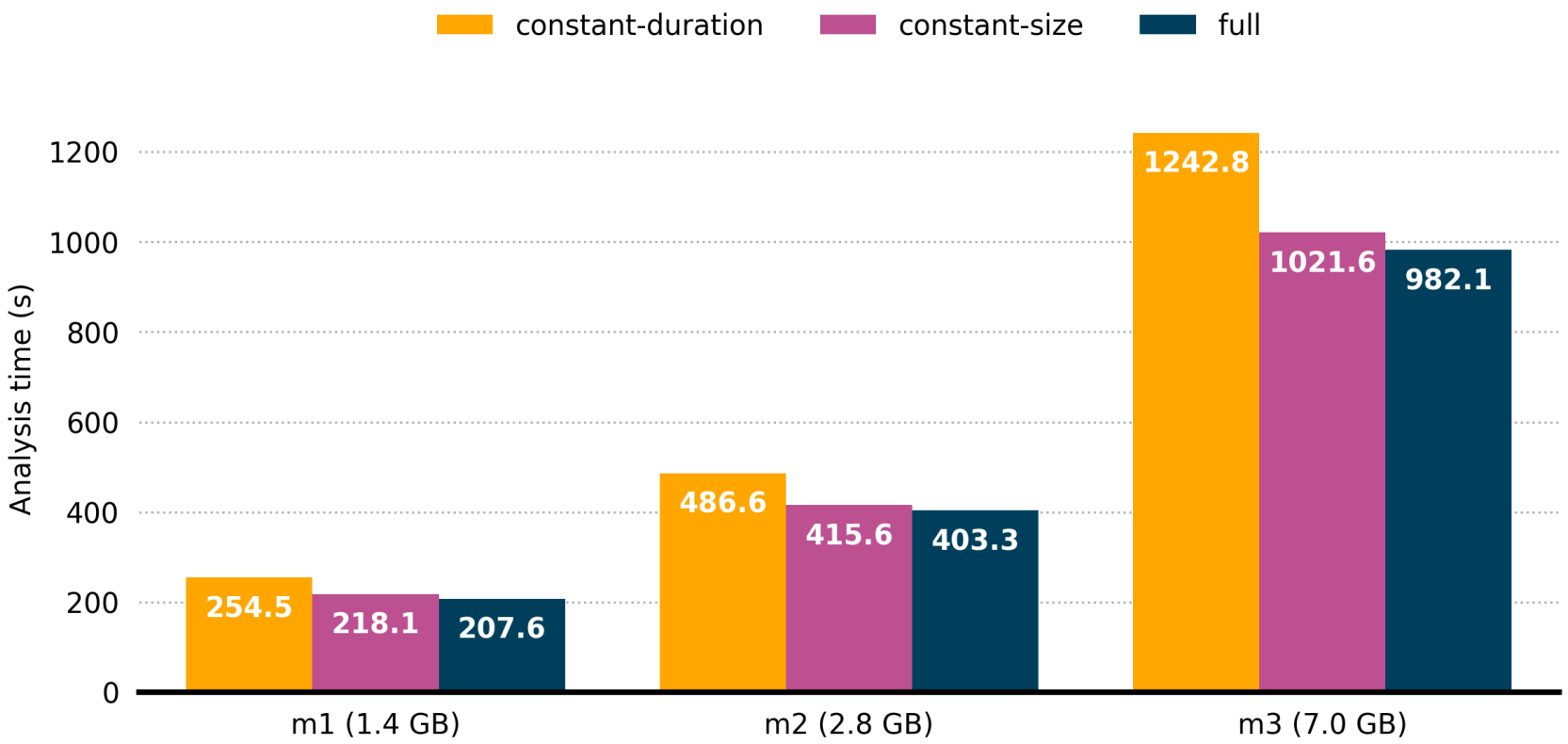


Figure: Analysis time per trace

# Results

| Level | Minimum | Maximum | Average | Standard Deviation | Count | Total |
|---|---|---|---|---|---|---|
| HistoryTileBackend:initQuery2D | 13.12 ms | 19.79 ms | 15.545 ms | 2.228 ms | 35 | 544.079 ms |
| HistoryTileBackend:writeTileToDisk | 114.688 µs | 30.765 ms | 11.029 ms | 7.459 ms | 66 | 727.934 ms |
| HistoryTileConfig:readTile | 761.344 µs | 8.007 ms | 2.577 ms | 2.15 ms | 209 | 538.56 ms |

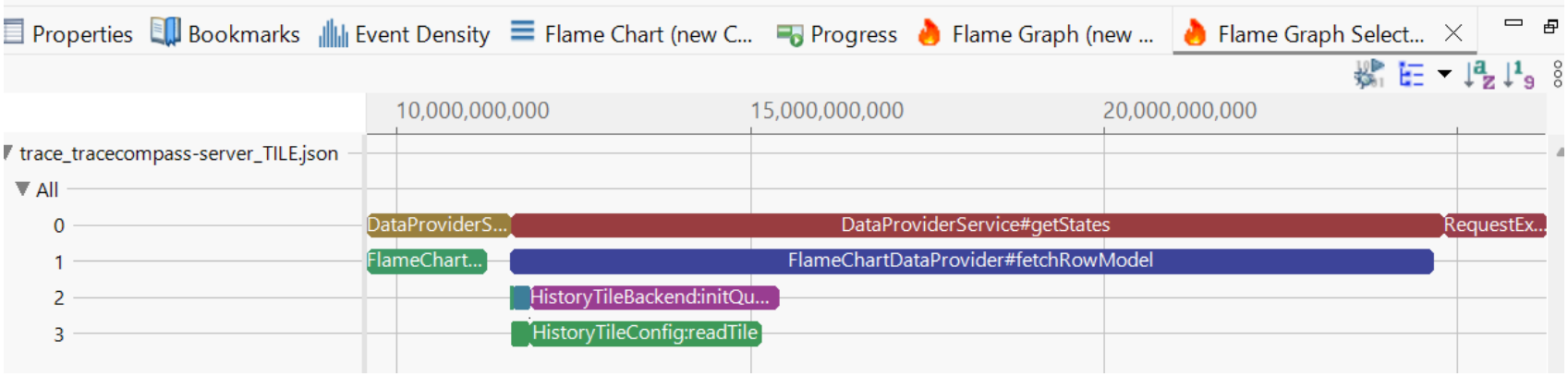


Figure: Flamegraph of the Query2D (constant duration)

# Results

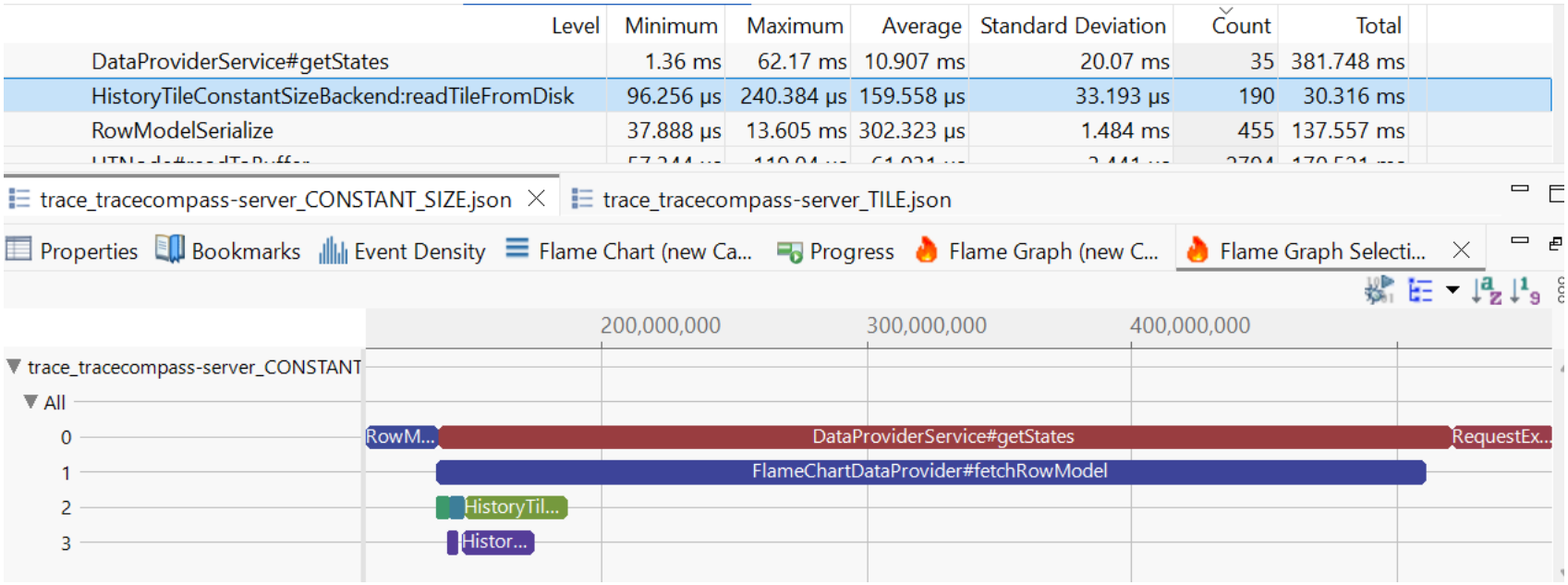


Figure: Flamegraph of the Query2D (constant size)

# Conclusion & Future work

- Constant size tile backend scales the best but the intermediate files remain large
- Implementation and merging with Trace Compass code is ongoing
- Generating a summary for larger trace very quickly is necessary

# Appendix

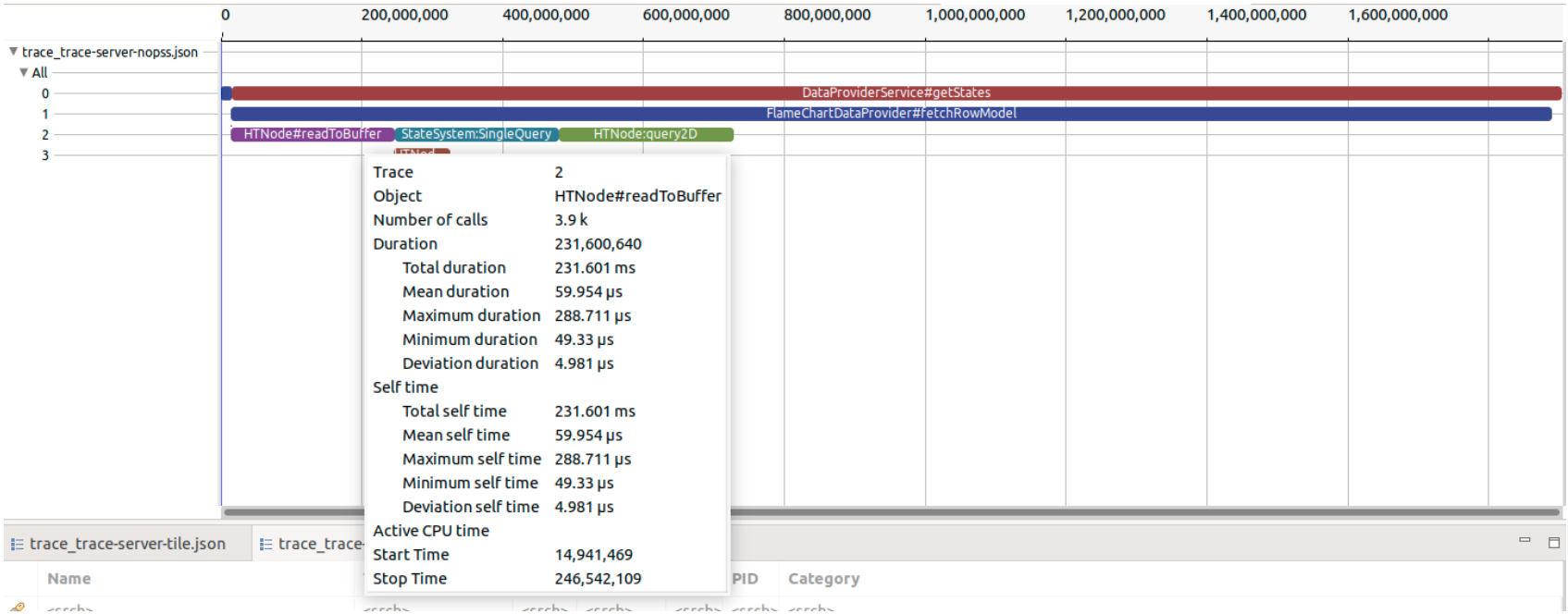


Figure: Flamegraph showing experiment requests before (full)

# Appendix

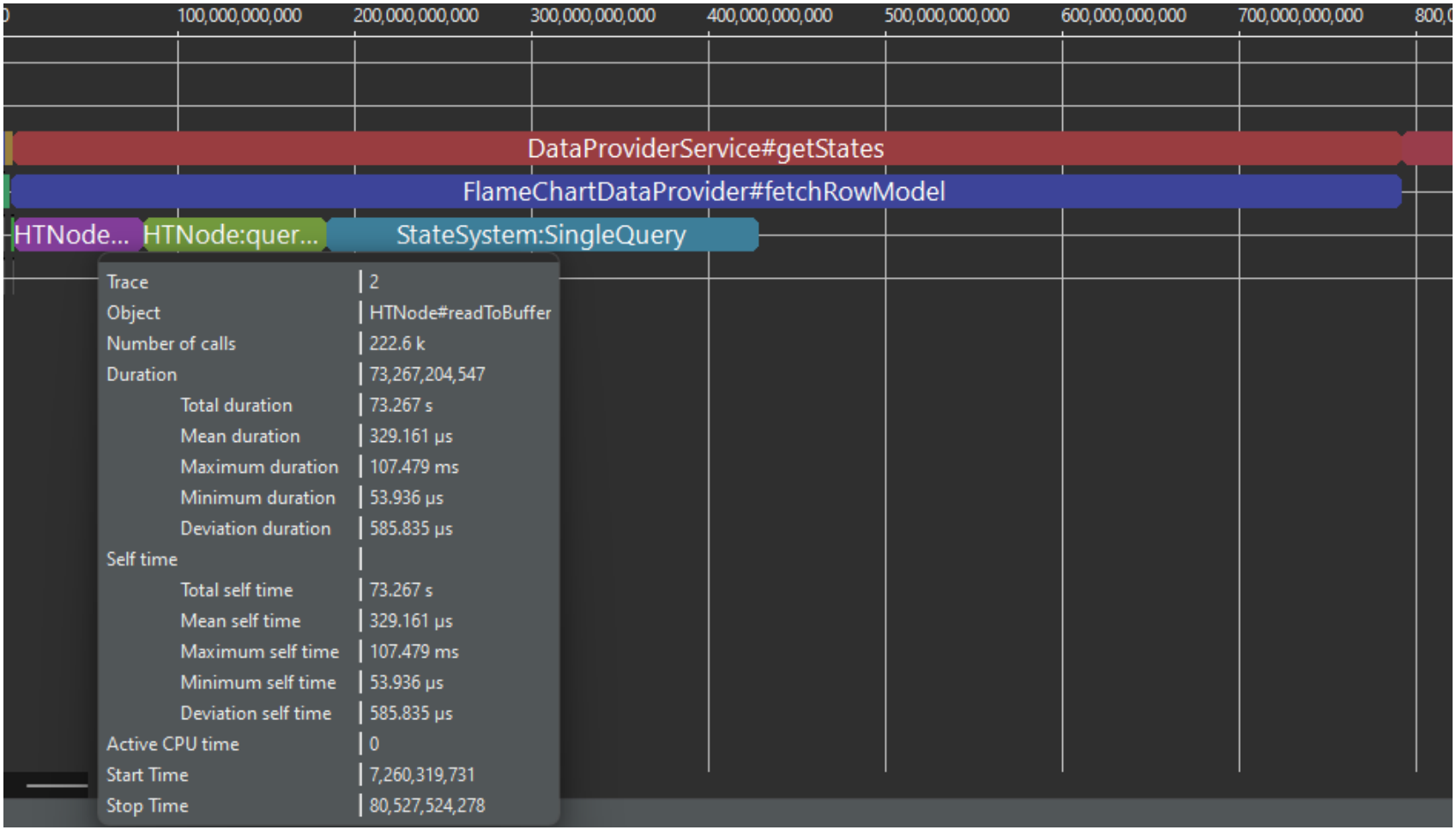


Figure: Flamegraph showing experiment requests after (full)

# Appendix

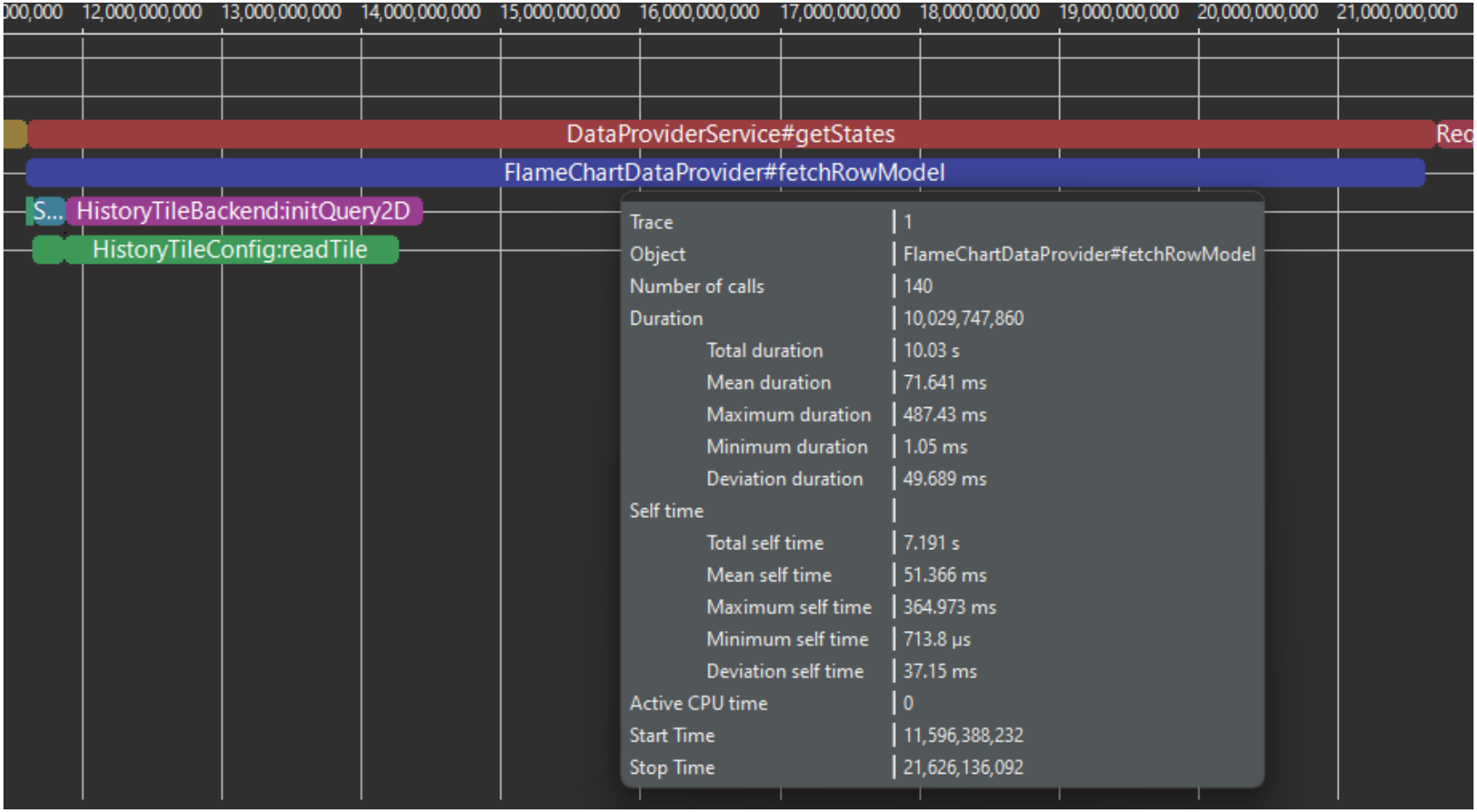


Figure: Flamegraph showing experiment requests with statistics (tile)

# Appendix

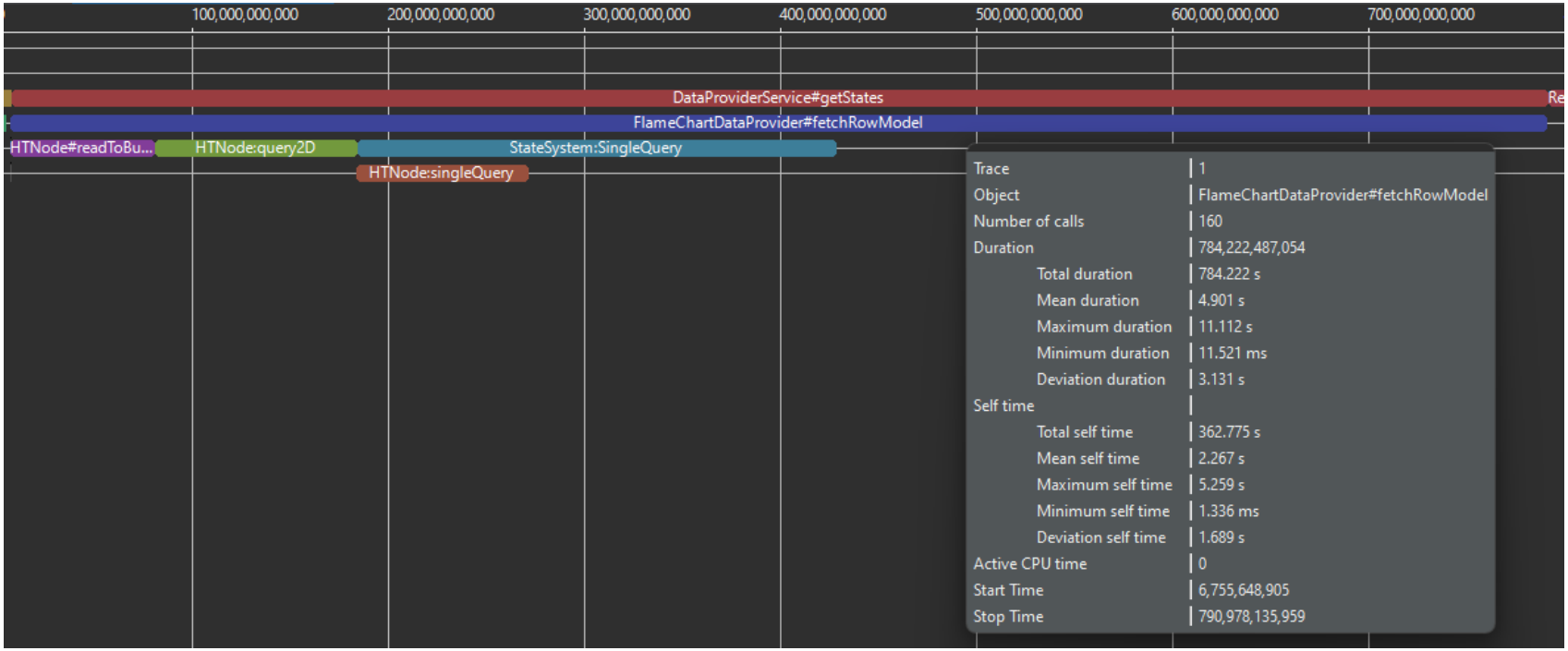


Figure: Flamegraph showing experiment requests with statistics (full)