Updates on Scalability, MPI, and ROCm in Trace Compass

Arnaud Fiorini

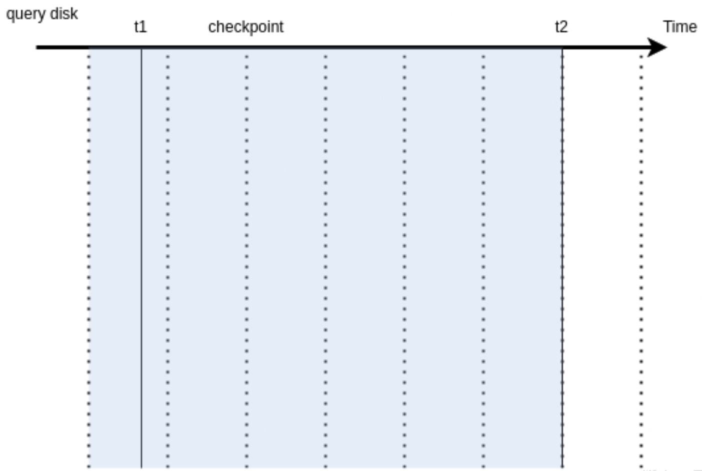
Polytechnique Montréal
Laboratoire DORSAL

Agenda

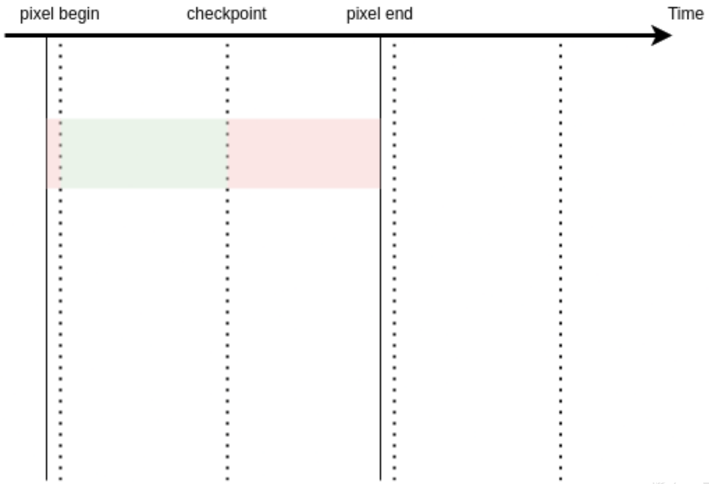
Partial State Implementation

- Full state system: Recording the interval on disk when we receive the end time.
- Partial state system: Recording the interval on disk if the interval intersects a checkpoint.

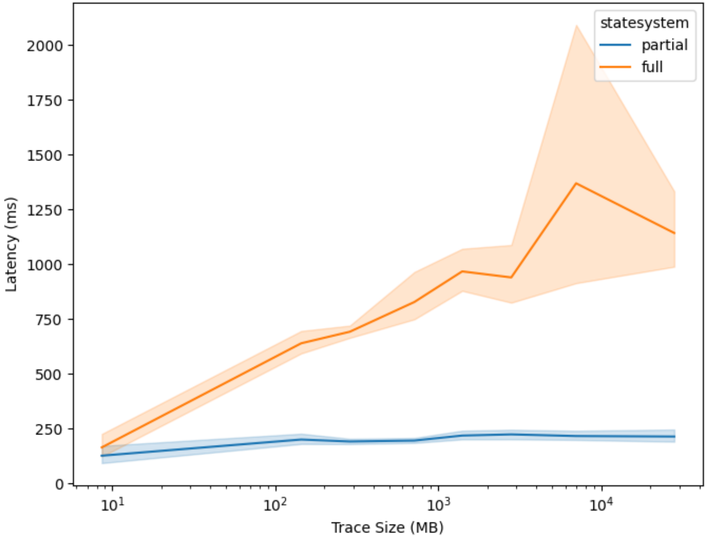
Partial State Implementation



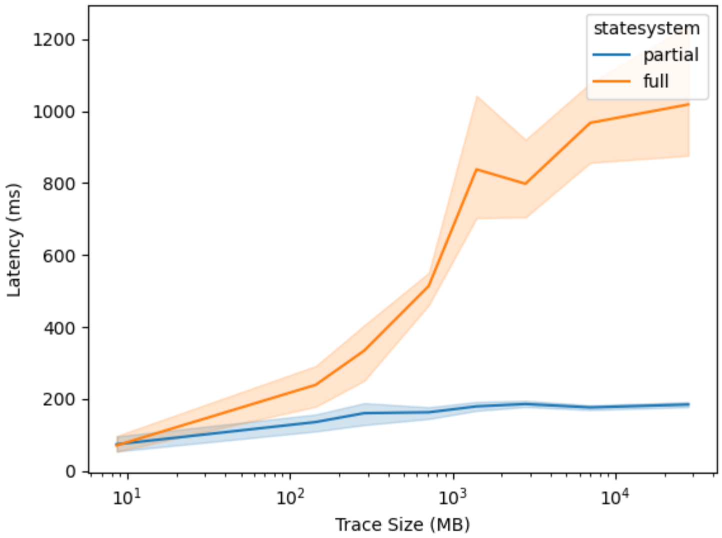
Partial State Implementation



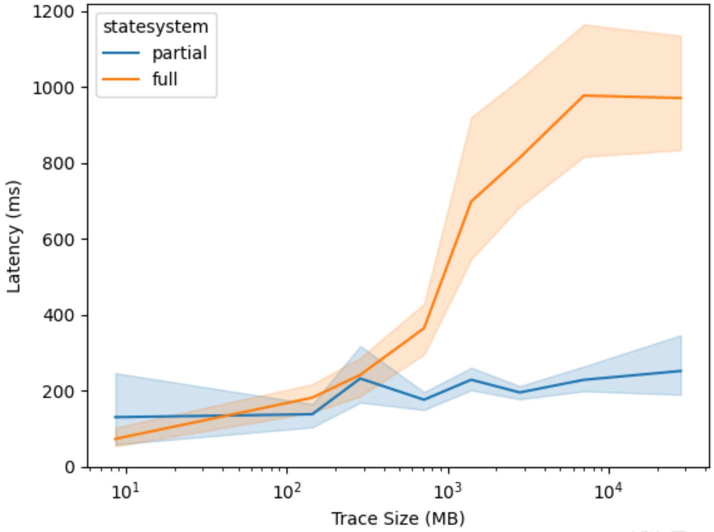
Partial State Results - Requesting 100%



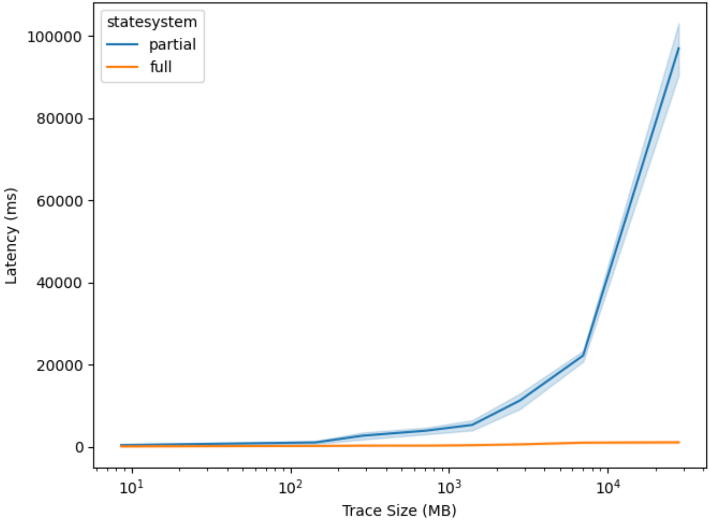
Partial State Results - Requesting 10%



Partial State Results - Requesting 5%



Partial State Results - Requesting 2.5%



Partial State Results

- These results were obtained using hard disk drives
- Analysis time is still a bottleneck
- Other optimisations remain to be explored

GPU Analysis

- Previous analysis specific to one version of ROCm traces
- Change analysis logic necessary for each change
- Support for multiple sources needs duplicated code

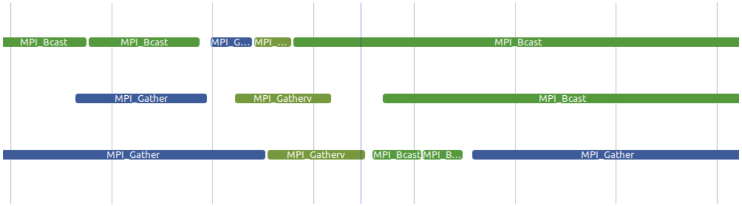
GPU Analysis

- One generic analysis for all GPU traces
- Necessary fields are implemented with an interface
- Works well for GPU API traces (Thapi, ROCm)
- Compute kernel launches are harder to correlate generically

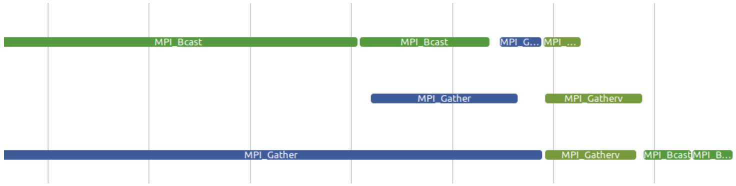
MPI Integration

- Exatracer provides MPI calls events
- Integration is done in Trace Compass to show a timeline view
- Development is still ongoing

MPI Use Case - Normal loop



MPI Use Case - Dead Lock



Conclusion

- Partial State shows good scalability but will require changes
- ROCm integration is done with the goal of being usable with other GPU tracing tools
- MPI Trace Compass plugin development is in progress to integrate dependencies

References

- https://github.com/argonne-lcf/THAPI