

CSC 548: Project

PRASUN RATN (pratn@ncsu.edu)

<http://www4.ncsu.edu/~pratn/548>

Implementing computation timing deltas on the MPI Trace Compression Framework replay engine

Problem Statement

An MPI trace compression framework has been developed to capture MPI call traces and store them in constant or sub-linear space. This trace is later decompressed and replayed so as to provide a means to analyze the program execution.

The above mentioned framework captures the MPI calls along with various parameters of interest, such as node information, data size etc. However, it currently does not capture the relative timing of such MPI events. Thus, the information obtained from the MPI traces does not tell us anything about the temporal pattern of the communication and hence is inaccurate to that extent.

This project will attempt to fill this gap by capturing timing information and using that information to replay the trace with time gaps between communications to accurately model the program execution. This would provide us with a more accurate representation of the actual program execution.

As we can see there are two parts to this ---

- i. timing information capture (done by the record engine),
- ii. replaying this timing information (done by the replay engine)

This project implements part (ii) above, while the part (i) is developed separately in another project.

Approach

The record engine captures compresses and stores the MPI call traces. This is later used by the replay engine. A similar approach would be used for capturing timing information too.

The record engine will be modified in a separate project to generate timing traces too. Similar to the MPI traces, the timing traces would also need to be constant or sub linear. The timing trace would be used by the replay engine to insert time gaps between communications. These time gaps would correspond to the computation time deltas in the actual execution of the program.

Implementation

Implementing timing deltas on the replay engine has an obvious dependency on the implementation of timing deltas on the record engine. We plan to tackle this by agreeing on a trace format for the timing delta. Since the replay engine need not be aware of the way in which the traces are obtained, we can go ahead with the implementation of timing deltas on the replay engine by using hand-generated timing traces.

Plan of Work

We plan to follow these steps for the implementation:

- i. Understanding the previous work
 - a. Experimenting with small test programs
 - b. Reading source code, papers etc.
- ii. Collaborate on establishing the timing trace format
- iii. Implement timing trace decompression
- iv. Integrate timing delta information with the replay engine
- v. Finally, test replay engine and record engine in conjunction i.e., use traces generated by the record engine