DISTRIBUTED EXPLICIT STATE MODEL CHECKING

Geof Sawaya (Ganesh Gopalakrishnan, Robert Palmer)
School of Computing

One unsolved problem with the micro-miniaturization of architecture size and the corresponding explosion of systems size, complexity and concurrency is verification of these systems. There are several tools available, but at the moment they are unable to verify the correctness of these labyrinths of control-flow complexity due to the astronomical set of states that can exist within it – we don’t have the capability to store all of the possibilities with perfect certainty as a record of all reachable configurations (given a system of n processes and m statements each, there may be as many as mn states). Our current work is to build a distributed model checker so that we have the memory space to store all the attainable conditions a system can be in and explore all potential paths between them. The model checker can be described in terms of the following functions:

\[ f: \text{state} \rightarrow \text{state list} \]

This function finds all the states reachable in one atomic step from the input state.

\[ g: \text{state} \rightarrow \text{number} \]

This function determines which node within the cluster is responsible for processing a given state.