James Pane-Joyce

4/13/16

Distributed Computing over the Internet

CSCI 6500

Distributed Reasoning Implementation in MATR

MATR is a higher-order automated theorem prover that has been under development at Rensselaer AI and Reasoning (RAIR) LAB[1]. MATR is already written to be highly parallelization and extensible. A high level description is that of series of small “codelets” each of which is designed to represent a single logical operation, such as Conjunction Introduction or Modus Ponens. These codelets can be run asynchronously and independently from each other. They report to a “codelet manager” which is responsible for: answering queries about what information is currently known and what statements desirable; accepting sorting, and implementing suggestions from codelets; and finally selecting and starting codelets to execute. The codelet manager then communicates with a “knowledge-base” which actually contains the currently known information.

This application is a perfect candidate for a distributed implementation for two reasons. The first is the asynchronous and independent nature of codelets. Codelets are designed to iteratively improve the proof with little regard for what is going on outside their logical preview.

The second advantage is the structure of MATR's proofs. Logical formula in MATR are designed to be solid and efficient for simple first order and predicate logic, but are easily extensible to higher-order and meta logic. These statements are connected by inferences which store and communicate the logical antecedents and consequents generated by codelets. The proof is further broken up into “boxes” which contain suppositions and goals, and are laid out in a hierarchical child-parent format. It is these boxes which could easily be partitioned off into different theaters as the majority of codelets only interact with a single box at a time.

For my project I would develop a distributed implementation of the codelet manager and knowledge base that intelligently portions sets of boxes to various machines and balances codelet execution across them. MATR is written in Java which makes it simple to extend via Salsa. In addition, many boxes have axiom sets that are very commonly used across multiple runs, such as the empty axiom set which would benefit from permanent presences online presences which can be used over and over or even simultaneously by multiple proofs. Having written the majority of the code for MATR I feel that this is a fairly reasonable next step for the application.

There are many papers on distributed automated reasoners, although most of these were done decades ago due to the fall of logic based AI and most were limited to first-order predicate logic. One example is Multistage Negotiation in Distributed Planning[2], which details how distributed agents can reason about the impacts of local reasoning to the overall plan. Another more fleshed out example is the DARES, which was a distributed predicate logic reasoner from 1990.[3]

1 http://rair.cogsci.rpi.edu/projects/automated-reasoners/matr/
2 http://dl.acm.org/citation.cfm?id=60232
3 http://www.aaai.org/Library/AAAI/1990/aaai90-012.php