

Paul Holmes, pwh2125
GPU Computing, COMS W4995
Term Project Proposal

I have a strong personal interest in Artificial Intelligence; that is, the study of and search for machine-based intelligence, not the dryly mathematical analysis of Big Data that the field has become in recent years. So I propose a project that is related to an AI-related project I did for a previous class: in my Programming Languages and Translators (PLT) class last year, I designed a language whose purpose, at least in theory, was the modeling of behavior. In a nutshell, the idea of the language was that one would model an initial condition using objects called Entities and the Rules that operate on them, and then “wind it up and let it go”. Each iteration, the system evaluates all the Rules, which are really just logical expressions coupled with an action to perform if the expression evaluates to true. The states of the Entities evolve according to these Rules, and then at some point you observe the final state of the system to see what it did.

The problem was that depending on how the system was modeled, the interactions could become exponentially complex and the system could become computationally intractable, so I was limited to modeling only the most trivial systems.

So I would like to try to take advantage of CUDA’s parallel processing capabilities to perform the logical evaluations in such a system more quickly.

I recognize that resolving a large set of “any arbitrary logical expression” is probably not a problem that would be evaluated efficiently even on a GPU because the logic in each thread would be very complex and their execution would become too divergent. Therefore I would, at least at first, limit the scope to resolving a large set of “some simple, canonical logical expression”, such as a decision list or conjunction normal form (CNF) formula. I would also, at least at first, try to implement this as a standalone solution rather than trying to integrate it directly with my programming language.

Furthermore, I don’t necessarily intend to try to implement the computationally most efficient algorithm for evaluating the logical data structure I ultimately select. The intention here is to make the best possible use of the GPU and to demonstrate my knowledge of the GPU programming techniques taught in this class. The optimal algorithm for evaluating, say, decision lists may or may not fit well into that framework.

I intend to work on this project by myself. I would like to implement it in Java, both because my language is implemented in Java and because it is the language in which I am most proficient. Therefore researching CUDA’s Java bindings will be my first priority. If it proves impractical to do this project in Java, I’ll switch to C/C++ right away. My next priority would be to select the specific type of logical expression to evaluate. Then at the end, if time permits I’d try to integrate the project with my language (which is called Geppetto, incidentally).