CUDA Creatures

Andrew Hershberger
Vadim Ogievetsky
July 17th, 2010

http://code.google.com/p/cuda-creatures/

Motivation

In 1981, Axelrod and Hamilton examined the iterated Prisoner's Dilemma. It was shown by computer simulation that the best strategy was one known as TIT FOR TAT, in which a creature cooperates in its first encounter with another creature and on every subsequent turn plays the same move as its opponent.

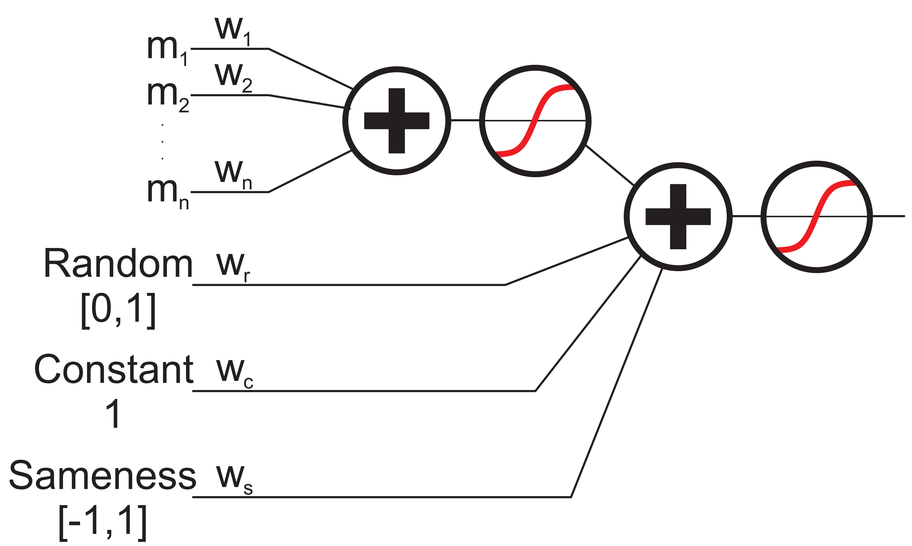
Such computer simulations quickly become impractical due to the round-robbin (number of creatures squared) nature of the simulation. However, increased performance, and therefore increased ability to explore variants of the problem, is possible by taking advantage of the single instruction, multiple data (SIMD) parallelism of the interactions between agents.

CUDA Creatures is a high-performance platform for exploring the space of strategies available to creatures engaged in variants of the iterated Prisoner's Dilemma.

Simulation Design

Each creature is encoded as a vector of weights that parameterize the calculation of a simple neural network, the result of which determines the creature's behavior. As shown below, the inputs to the neural network are a finite memory of the creature's interactions with its opponent and several other optional inputs to allow variations on the information available to the creature as it makes its decision.

CUDA Creatures uses a genetic algorithm to evolve the creatures by allowing mutated versions of the fittest creatures to replace the sickliest.



Parallel Strategy

On the CPU each creature-creature interaction must be performed sequentially, but on the GPU, many interactions can be calculated in parallel. In an initial naïve implementation, CUDA Creatures used the same implementation on the host and device by using atomics and many accesses to global memory. Even in this impoverished implementation the speedup was significant due to the parallel execution.

In a second iteration on the design of the CUDA kernel, atomic operations were eliminated by storing intermediate per-thread results and then using a reduction.

One final optimization was to pack the interaction history for each pair of creatures into a single register, allowing the shared memory to be used to cache the creature weights – eliminating the bulk of the global memory accesses.

880x Speedup

CUDA Creatures has achieved a massive speedup by exploiting the data parallelism of the problem. In one case it achieved an 880x speedup and regularly delivered speedups in the range of 600x to 700x over the CPU implementation.

References

Axelrod, Robert; Hamilton, William D. (27 March 1981), "The Evolution of Cooperation", Science 211: 1390–96, doi:10.1126/science.7466396

Experimental Results

In accordance with Axelrod's findings, our simulation reviled that TIT FOR TAT and similar strategies produced the best results. In particular the fittest creature has to be “nice” (i.e. it cooperates on the first round).

Baseline (naïve)	100x
Replacing atomic add with reduction	2x
Smart bit vector packing and heavy use of shared memory	3x