Parallel Graph Searching Algorithm on the GPU using OpenCL

Péter Rónaszéki[a], **Gábor Kusper**[b]

[a]Eszterházy Károly College
ronaspet@gmail.com

[b]Eszterházy Károly College
gkusper@aries.ektf.hu

Abstract

The graphics processing units (GPUs) are essential part of today's high-performance computing systems. Open computing language (OpenCL) is an open and free framework to utilize the performance of these devices. The graph searching algorithms — graph traversal and pathfinding —, are exciting computational tasks to fit to the parallel architecture of the GPUs.

In our case the graph represents a problem. [1] The nodes of the graph are states of the problem, the edges are the applicable operators. We have a start node and some goal nodes, our task to find a path between the start node and one of the goal nodes. The founded path will represent the solution of the problem.

The constructed algorithm adopts the basic graph searching algorithms to GPU architecture. We encode a sequence of operators as relative small number using a number system, where the bases of the number system is the number of operators, and the digits are the operators in the sequence. Note, that this encoding takes much smaller space than a usual problem state representation. We use a buffer which contains such encoded numbers. Each GPU computing unit is fed from this buffer. A unit works as follows: it gets an encoded number from the buffer, decodes it, performs the list of operators, checks whether it is a goal state, if yes it signals that there is a solution, if not, it creates all new sequence of operators, where the last ones are applicable operators, encodes all sequence, puts the new numbers in the buffer.

Keywords: parallel computing, graph searching, GPU, OpenCL, artificial intelligence

MSC:

References

[1] Gergely Kovásznai, Gábor Kusper, Artificial Intelligence, Eger, Eszterházy Károly College (2011).