Combinatorial Set Matching Using GPUs

Anand Madhavan and Etan Frachtenberg
(anand, etan)@powerset.com

The Problem

Given a number of $E_{\text{max}}$ tuples, we are interested in finding the ones for which there exists at least one $E_{\text{min}}$ tuple, as a subset

$E_{\text{min}}$ tuples (where entity having and entity eating cake are the same)

$E_{\text{max}}$ tuples (facts where cake being had and cake being eaten are the same)

Match?

no

no

yes

Early exit on Match

Results and Conclusions

- Speedups are measured against the CPU (Reference) implementation.
- Speedups are sensitive to the fraction of all tuples that ‘match’. The Sequential algorithm takes advantage of ‘early exit’ at higher fractions.
- Comparison of GPU-Min-Pre and GPU-Mins illustrates that it might be worthwhile doing ‘pre-check’ (device global memory access) for every $E_{\text{min}}$ tuple to facilitate early exit.
- Although low ‘match’ ratios are not typical, we are able to see speedups of 5x despite the low arithmetic intensity.

Challenges

Low arithmetic intensity

- $t_1(i) = t_2(i) \ldots \ldots t_1(j) = t_2(j)$

mostly boolean operations and comparisons

Typical size does not fit in constant memory

100’s of Kb of tuples

Sequential version very effective when early matches occur

Early exit on Match

Motivating example

“Who had a cake and ate it too?”

Mary and John were participants in a cake-making competition. Making a cake was a piece of cake to Mary and John. As the competition got started, Mary started making a chocolate cake. She put it in the oven, whilst John was busy making his cheesecake. Towards the end of the competition, many of them had some delicious cakes displayed on their tables. Mary gave a piece of her cake to John for tasting. John took the piece of cake. He had it in his hand, while eating his own cheesecake. He ended up eating a piece of the cheesecake instead.

Mary, on the other hand, had her chocolate cake in her hand and was waiting for John to try it. While waiting she ate some of her chocolate cake.

Who ate cakes?

Who ate the exact same cake they had?

The problem can be described as combinatorial set generation and matching.

The $E_{\text{min}}$ tuples are generated as part of a linguistic constraint. The $E_{\text{max}}$ tuples are generated as part of another linguistic constraint and the combinatorial set matching is required as part of a combined constraint.

Although the number of cakes or entities in a document is bounded, the number of set combinations can grow exponentially.