Identifying Similarities in TMBL Programs with Alignment to Quicken Their Compilation for GPUs

Tony E Lewis
George D Magoulas
Two Major Approaches to GPU Acceleration of GP

Data parallel
Compile new GPU code for each new batch

Population parallel
Write one GPU interpreter to process all batches
The Aim of the Work:
To Minimise the Weakness of Data-parallel

Data parallel
Evaluation: very fast
Compilation: long

Population parallel
Evaluation: fast
Compilation: none
The Problem: Compilation Stops
Small Datasets Getting Top Speed
Two Strategies to Ease Load for Compiler; This Talk is about the Second

1. PTX
   Write the individuals in a lower level language

2. Alignment
   Exploit similarities between individuals
The Motivation is Code Duplication

```java
if (prog == 0) {
    ...
    slot1 += slot3;
    slot2 = 3.1096370;
    ...
}
else if (prog == 1) {
    ...
    slot1 += slot3;
    slot2 = 3.1096370;
    ...
}
else if (prog == 2) {
    ...
    slot1 += slot3;
    slot2 *= slot1;
    slot2 = 3.1096370;
    ...
}
```
The Motivation is Code Duplication

if (prog == 0) {
    ...
    slot1 += slot3;
    slot2 = 3.1096370;
    ...
}
else if (prog == 1) {
    ...
    slot1 += slot3;
    slot2 = 3.1096370;
    ...
}
else if (prog == 2) {
    ...
    slot1 += slot3;
    if (prog == 2) {
        slot2 *= slot1;
    }
    slot2 = 3.1096370;
    ...
}
Need an Algorithm to Link Equivalent Code
This is a General Problem: Alignment

Instructions

Shapes

Amino acids
Needleman-Wunsh (NW)

Standard in bioinformatics
Optimal
Reasonable time complexity
NW is Not Suitable Here

Protein Data

Items in protein two

Items in protein one

TMBL Data

Items in TMBL program two

Items in TMBL program one
A Quick But Rough Alignment
A Quick But Rough Alignment
Glue Pairs Together
Look For Extra Links
Beware of Dangers of Extra Links
1: Don't Join Items Within a List
Beware of Dangers of Extra Links
2: Don't Form Crosses
Beware of Dangers of Extra Links
3: Fear the Complex Crosses
The Benefits of Extra Links
The Benefits of Extra Links
The Benefits of Extra Links
Results:
Alignment time is small (and negative!)

![Graph showing the time per individual to align and generate source (in seconds) vs. the number of TMBL instructions. The graph compares aligned and not aligned cases with a linear increase in time as the number of instructions increases.]
Results:
Evaluation speed is only slightly lower (~3.6%)
Results:
Compile time is considerably reduced (~4.8x)
Conclusions

Effort

Effectiveness

Requirements

Applicability
Can Align PTX Too
Can Align PTX Too

Evaluation Speed
+29.889%

Compilation Time
-98.265%
(ie 57.625x faster)
Thanks

EPSRC

Reviewers

You