Umeå University
Department of Computing Science
18th April 2008

5DV050
VT08
Assignment 2

Optimizing SGEMM for Cell BE

Design and Analysis of Algorithms
for Parallel Computer Systems
Assignment 4

1. You should complete this assignment in **groups of at most 2 students**.
2. Write the **name** and **email** of every group member on the front page.
3. Include the **path to the source** in your report and append a printout.

Good Luck!

1 Introduction

Getting the most performance from a SIMD architecture requires expertise knowledge. In this assignment, you will practice the most fundamental loop optimizations for improving the performance of `SGEMM` on the Cell BE processor. You are given skeleton code (download the `assgn4.tgz` tarball from the course webpage) that computes the specific variant of `SGEMM` called the T,N variant (A is transposed, B is not). To be precise, the code computes

$$C \leftarrow C + A^T B,$$

where all matrices are of size 64×64 and stored in column major order in contiguous storage.

2 Part 1: `SGEMM` on a single SPE

Unpack the skeleton code, compile it, and make sure it runs as expected (check the course notes for performance figures). In the tarball you will find three files:

```
Makefile   # To build the executable ppe
ppe.c      # The PPE-side code
spe.c      # The SPE-side code (containing SGEMM)
```

Modify the `sgemm()` function (and only that function) found in `spe.c` using any optimization techniques you want. You are encouraged to make use of the `spu_timing` tool (`make spe.s`) to inspect the low-level interactions between your code and the pipelines in the SPU. To pass this part of the assignment, your `sgemm()` function must reach at least 5.00 Gflops/s. If you get stuck you may of course ask the assistants for hints on how to proceed.

3 Part 2: Sketch a parallel algorithm

You should be somewhat familiar with how the SPEs and the PPE can communicate via mailboxes and asynchronous DMA transfers. Consider a matrix multiplication $C \leftarrow C + A^T B$ where A, B, and C are matrices of size 512×512 partitioned in blocks of size 64×64. Each block is stored contiguously in memory. Assume you have a good routine for performing a matrix multiplication on the block level (i.e., the `SGEMM` you have just constructed in Part 1). Sketch a parallel algorithm for performing the large matrix multiplication using all 8 SPEs for computations and the PPE for control. You should provide enough details to make the reader understand that your algorithm is possible to implement, given enough time to do it. Besides that, there is only one requirement: your algorithm should take advantage of the high-bandwidth SPE to SPE communication available on the Cell BE.

4 Part 3 (Optional): Implementation

Implement, test, and evaluate your algorithm from Part 2.