EC527 – Project (1/27/2015)

Broad Objectives
The purpose of the project is to give you in-depth experience with high performance programming, especially, by using the methods you're learning during the semester.
Part of the project is a presentation by each group. Collectively the presentations should comprise a practical workshop on which methods work well on which problems and with how much effort.

Deliverables
→ A program that is an improvement over the serial reference version. It should use at least some single core and some multicore methods. Ideally it would map to a GPU as well, but this is not required.
→ A 10-15 minute presentation depending on number of groups -- see below
→ A write-up – see below

Timeline
To do soon → Form a group of size <= 3. If you'd prefer to work alone, that's fine too. For groups size = 3, more planning is required to partition tasks.

By Wednesday February 18th → Choose an application. Try to find on that is both easy to explain (like matrix multiply) but that gives plenty of room for optimization over multiple architectures. That is, the topic/problem should be easy to present and visualize.

By Tuesday February 24th → Create a serial reference code for your project. This is Assignment 4. To keep the project manageable, you should (probably) write your baseline code. Part of the deliverable (more later) is to debug and validate for correctness. Ideally you will have found a library version, either from MatLab, Intel MKL, or other reputable library, and created a few test cases. You can use these for validation. You can also use them for performance comparison: MatLab should be very easy to beat, MKL very hard.

By Friday April 10th → An update of your Assignment 4 document. Should include architectures you plan on supporting and plans for mapping. For example, for multicore and GPU you should describe the way you plan on partitioning the problem.

During Week of April 13th-17th → Group meetings to check progress/problems.

By Wednesday April 22nd → A full presentation (10-15 minutes) describing the bulk of the work. There are tentatively three presentation days, 4/22, 4/27, and 4/29. You should be prepared to go on any of the days.

By Friday May 1st @17:00 → Final write-up (put together the pieces and incorporate feed-back)

Presentation guidelines
The talk should be high quality and well-prepared. It's OK if you haven't completely finished, but you should be substantially done. Giving a 10-15 minute presentation on a problem that most of the audience is not familiar with is challenging and will take real work to make coherent. In any case, your talk should include the following:

- Description of the problem
- What the serial code/algorithm looks like
- How you modified the algorithm to run in parallel (or which parallel algorithm you selected if there is a choice)
- Overview of your optimized codes

- Experiments and results
- A couple minutes for some brief Q&A

Writeup guidelines
Basically the same as the presentation guidelines, except this time I'll be your audience and you will have reached some definite conclusions.

Project suggestions
The first set is reasonably well-defined, almost certainly appropriate, and I will be able to answer questions at some depth, or at least provide references. The second set is less well-formed: there are definitely good projects in these areas, but they will need at least a little bit of work to narrow down.

Well-defined
1. Neural nets for character recognition (neuron specification given)
2. Molecular Dynamics w/ cell lists and range-limited forces
3. Astronomical Tree Code such as Barnes-Hut
4. Dense Matrix-Matrix Multiply
5. Advanced MMM - o(N3) algorithm
6. Sparse Matrix-Matrix Multiply
7. Gaussian Elimination / LU Decomposition
8. Biological sequence alignment such as Smith-Waterman
9. FFT
10. Graph Traversal -- Breadth-First Search
11. Branch and Bound w/ application to optimization
12. Sorting
13. QR Factorization with Givens reflections
14. Structured Grid -- Fixed Mesh Refinement (SOR)
15. Structured Grid -- Adaptive Mesh Refinement
16. Structured Grid -- Multigrid
17. Unstructured Grid
18. Stencil (not iterated)
19. Structured Dynamic Programming -- Knapsack
20. Unstructured Dynamics Programming -- The technology mapping problem

Good potential, but need to be specified. That is, these are domains rather than algorithms. For example in computer vision there are many standard functions, e.g., segmentation, and many of these have hundreds of published algorithms. You probably only want to do one (or a small number).
- Other Graph algorithms – Ex: Shortest path? Clustering? ???
- FSM-based algorithms -- Ex: String algorithms like text matching w/ errors?
- Image processing -- Reconstruction? Classifiers?
- Computer vision -- feature extraction? Segmentation?
- Other solutions to classic "hard" problems and their approximations -- TSP?, ILP?
- Basic graphics functions
- Linear optimization
- Serious arithmetic optimization: for example from cryptography like AES block code?? CRC or other combinational app that requires bit-level operations
- Mandelbrot sets w/ load balancing (could be too easy)