

CS442/ECE432: Project 3

April 24, 2008

General

- The project proposal is due **due April 29, 2008**.
- This project is **due May 8, 2008**.
- Submit via e-mail to riesen@cs.unm.edu (Mail it before class on the 8th.) Your subject line must say: "Project 3 Submission"
- If you choose to use more than one source file, also provide a Makefile that uses the appropriate gcc compiler (mpicc, mpiCC, etc.) and produces the executable ex_01.
- Your program must compile without warnings (we use the `-Wall` flag for gcc)
- Name your description file ex_01.pdf
- Put your files into a tar or zip file called Proj3.tgz (or Proj3.zip)

1 Implement a Parallel Algorithm of your Choice

1.1 Project Proposal

Find a research paper, a section in a textbook, or an Internet resource that describes an algorithm. The description may be for a parallel or serial version of the algorithm. You will get more points, if you decide to start with a serial algorithm.

Talk to me or email me your selection for approval. The algorithm you choose should be a little bit more difficult than the projects we have done so far. Graduate students are expected to chose a project that is more difficult than a simple dot-matrix multiply.

By **April 29, 2008**, you must give me a URL or a copy of the description you found of an algorithm that you intend to implement. Your proposal must include a description of how you intend to implement the algorithm; i.e. how will you decompose the problem, how do you intend to handle communication, how will you map the tasks to processes, etc.

1.2 Implementation

For the second part of the project, you must implement your chosen algorithm as an MPI program that works on Hammer. The usual rules apply: your program must compile without warnings, and it must run without segmentation faults and other serious errors before I will grade it.

1.3 Documentation

You must provide a written report as a PDF file along with the source code of your program. Your report must reference the source of your algorithm (web page, journal paper, etc.), it must detail how you designed your program from the algorithm description. Your report must also include instructions on how to run your program; e.g., command line parameters etc. If your program requires input data; e.g., an input matrix, provide one or more examples to test your program with as files in the tar archive.

Finally, conduct a brief scaling study and show how well (or poorly) your program scales.

1.4 Notes

It is OK to implement an algorithm for which there already exists an implementation. However, you should start with the algorithm description and come up with your own design. **Do not copy** an existing program.

If your algorithm requires a core function of some sort; e.g., the method to calculate the Mandelbrot number for a single point, or an implementation of a mathematical function that all (or some) nodes will execute, then it is permissible to use existing code and incorporate it in your program. In your report you must cite where you got code that is not your own and clearly indicate which portions of your program use software other than what you have written for this project.

Talk to me if you are not sure.

Some ideas for algorithms are: some (simple) two-player game, a sorting method, operations on a dense or sparse matrix (I'd prefer if it was not a matrix multiply), a graph algorithm, string matching, etc.