# Parallel Programming and Computing: Group Project

Christopher D. Carothers
Department of Computer Science
Rensselaer Polytechnic Institute
110 8th Street
Troy, New York U.S.A. 12180-3590

March 28, 2013

## DUE DATE: Midnight, Tuesday, May 7th

## 1 Description

The central key requirement of your group project is that it **must involve** the design, development in software and experimentation of a massively parallel system. **A key factor is that your parallel system must involved a high degree of interaction and not be considered "embarrassingly parallel"**. This means that projects involving "wide-area", "campus-area" or other asynchronous, master-slave approaches are not a viable project for this course.

*The end goal is that you are making progress towards a result that is of high enough quality to be published*. Beyond that, you are free to explore any particular "non-embarrassingly" parallel application that is relevant to your research. More specifically, use the format of papers we have read as part of this course as a template.

Examples of a project might be:

- Parallelization of components in an existing Computational Fluid Dynamics (CFD) code using threads or any other HPC/scientific computing application.
- Construct a new PARALLEL discrete-event model using a parallel simulator (e.g., ROSS). For example, create a human "agent" model of a large-scale city that is being evacuated due to a weather diaster. (Note, ROSS will be discussed in class on Tuesday, April 2nd.)
- Parallelize an existing sequential data algorithm using threads, MPI or CUDA parallel programming frameworks or a hybrid combination of 2 or 3 of them. CUDA will be the next topic.
- Final note: if your project does not involve MPI, threads or CUDA at some level you are probably on the wrong track for this project and should speak with the Prof. Carothers.

The key components of your project submission paper:

1. First and foremost, your project MUST be about High-Performance Computing and performance. So, this excludes any sorts of Cloud Computing projects or embarrassingly parallel "over the web" projects. While those are valid types of computation they are not what this course is about. We are concerned about high-performance computing on supercomputing class systems.
2. **It is critical you pick something you and your team can complete.**
3. List your team members. Teams can have up to 3 members. For each team member describe their contribution to the overall project. That is, what did you do besides attend meetings?.
4. Describe your parallel implementation both code and algorithms used.
5. We have read a number of papers this semester. So, provide a review of related published articles. Do a **Google Scholar** search.
6. Performance results. This will include your sequential and parallel results and indicate your overall speedup. You should perform a "strong scaling" study (i.e., problem size remains the same and processor count increases).
7. Analysis of performance results. Provide additional information on why your performance turned out they way it did. In particular, you should understand how much communication overhead your program incurred versus doing real computational work. This should be measured and quantified. Use the cycle counter to measure these aspects.
8. Summary and future work. Provide a summary of what you did and directions of where you think this project could go in the future. That is what problems did you not have time to solve?
9. You can expect your paper/project write-up to be a similar length as a regular conference paper – 6 to 8 pages double column, single spacing, 10 point font with performance graphs and references. Hand-in a PRINTED COPY by the above deadline.
10. Place your code on *kratos.cs.rpi.edu* and note in the write-up which team member's account has the code. Also, provide some sort of README on how to run the code.
11. EMAIL Prof. Carothers early if have questions or want any sort of guidance about your project. Waiting until a few days before the deadline to ask how or what you should is not acceptable.