Project Guidelines and Ideas
Portland State University - Advanced Computer Architecture II - Winter 2022

Key Project Dates

Project team forming request: email instructor project team partner(s) request by Jan 10
Project Proposal due: Jan 24
Project Progress Report due: Feb 14
Project Demo/Presentations due: Mar 7 – Mar 10
Final Project Report and Source Code due: Mar 11

Project Guidelines

The project is a significant part of your grade, and you should use it to gain experience in developing and evaluating ideas in architecture, multiprocessor architecture and parallel computing. The other alternative is to write a survey that covers a certain area in multiprocessor architecture.

You are encouraged to come up with your own project ideas.

You should form project teams, 2-4 students each. Some of the project ideas below includes an estimate of how many people will be needed. You should attempt to have people with diverse skill sets on your team. Please send me names of people on each team by end of day on Jan 10.

The project will include both a written report and a project demo/presentation. The projects will be graded on: i) problem definition and motivation, ii) survey of related work, iii) experimental methodology, iv) presentation of results.

Project Proposal: (Due Jan 24) You should submit a project proposal, no longer than two pages, containing the following information: Project title, project team members' names and email addresses, a description of your topic, a statement explaining why this topic is important, a description of the methods you'll use to evaluate your ideas, and references to at least two papers related to your topic. You should submit the project proposal in pdf format via course D2L.

Project Progress Report: (Due Feb 14) This should be an updated version of the project proposal with an additional 2-3 pages describing the progress you've made so far in your project. You should focus on the tasks that you have completed and not on tasks that you've just started. You should provide at least four references for papers related to your project.

Final Project Report: (Due Fri Mar 12) Your report should mimic a conference paper with a title, author names, abstract, introduction, body, references, and optional appendices. Your report should be no longer than 15 pages at double-spaced 10-pt font. Any extra material beyond 15 pages should be put in appendices. The abstract should summarize the contributions of the report in less than 300 words. The class D2L site has links to writing resources for people with limited experience of writing in English. A sample format of an exemplified final project report (from ECE-587) is posted on course D2L under Project directory. If you are doing programming project, you also need to submit all source code and any command line instruction and arguments on how to run your program plus your project presentation

slides (if any) by Mar 12. If you are doing a survey project, please submit the final project report and project presentation slides by Mar 12.

Project Demo/Presentations: (Mar 8 – Mar 10) Each team should give a presentation/demo to TA between Mar 8 or Mar 11 (arrange demo time with TA beforehand). Each presentation should be 10-15 minutes long (depending on project and number of group members). Each member in the team is required to present a part of the presentation. The presentation should highlight important findings from the project, explain the project idea and results. If the project team is doing programming project, please send project report, presentation slides and source code plus any command line instruction and arguments to TA prior to your scheduled project demo so that TA can run the program to check it. If you are doing a survey project, please send the project report and presentation slides prior to your scheduled project presentation to TA. Oral presentation resource can be found on course D2L under Useful Links folder.

SimpleScalar Simulator

For some projects, you will be expected to use SimpleScalar (SS) to implement your project ideas and evaluate its performance. Details on downloading, using and modifying SS is provided off the Fall 2021 ECE 587/687 course D2L and also on Winter 2022 ECE 588/688 course Canvas under the Project/SimpleScalar directory.

Some Potential Project Topics and Ideas

Parallel Programming #1: Parallel Equation Solver. Implement both a serial and a parallel program that takes as an input a 5000x5000 matrix that includes coefficients for a system of linear equations, and produces a solution for that system. You need to consider how to generate an input matrix that would provide a valid solution. You should do research to figure out the best available parallel equation solving algorithms. Your serial version should implement the best possible serial algorithm. You can use either pthreads or MPI to implement your project. You will be graded based on your algorithm selection and the speedup you obtain on 16 processors vs. one processor. Expected number of participants: 2-3.

Parallel Programming #2: Database Join. Implement both a serial and a parallel version of a database join operation that prints out all the record pairs in two tables "Employee" and "Trips" that share the same ID. The first "Employee" table is a 20,000 element array where each element contains a number (ID) and a 30-character string (name). The second "Trips" table is a 100,000 element array where each element contains a timestamp, an ID, and a 20-character string (destination). The join operations finds all record matches in one table that have the same ID as any of the records in the other table. Your program should either randomly generate the tables or be able to read values from a file (but IDs in the "Trips" table have to correspond to IDs in the "Employee" table). You should research the best possible serial and parallel implementations for the join operation. You should use either pthreads or MPI to implement your parallel version. You will be graded based on your algorithm selection and the speedup you obtain on 16 processors (vs. one processor). Expected number of participants: 2-3.

Face Feature Recognition: Implement both a serial and a parallel version of a program that would take a photo of a human face as an input, and attempts to extract coordinates for the major features (e.g., eyes, ear, nose, mouth, etc.). It's recommended that you implement this using a GPU programming framework such as OpenCL or CUDA. You will be graded based on how much you improve performance

using integrated or discrete graphics cards and your algorithm selection. Expected number of participants: 2-3.

Simple Cache Coherence Simulator: Implement a simple multi-processor cache simulator that estimates the performance of snooping cache coherence protocols, and use it to compare the performance of MI, MSI, and MESI protocols. Your simulator should use, as input, a list of requests by different processors (e.g., processor number, block address, access type (read or write), time), and produce as output the average, maximum and minimum time needed to handle requests. You need to profile some parallel micro-benchmarks (e.g., homework 2 problems) to produce the necessary input. You also need to come up with a reasonable multiprocessor and network configuration to use in your simulator. Expected number of participants: 3-4.

Lock and Barrier Alternatives: You should implement a simple parallel program that implements simple debit and credit transactions to a bank account using pthreads. You should try to use as many locks, semaphores, and barrier alternatives to guarantee correct synchronization between different transactions. Your program should randomly issue 10 million transactions of debit or credit transactions. You should report back the performance improvement or degradation when using different lock and barrier alternatives. You will be graded based on the research you do to explore and implement different lock and barrier alternatives. Expected number of participants: 2-3.

Multithreading Support in SimpleScalar: You should try to extend simplescalar to implement 2-way simultaneous multithreading. Your simulator should be able to run two different threads on two different logical processors. You should run simulations of multi-program workloads (e.g., two SPEC benchmarks running at the same time). Your report should compare speedups and overheads of multithreading over single-threaded execution, especially the cache sharing effects. Expected number of participants: 3-4.

Survey on a Research Area in Multiprocessor Architecture: Write a paper that surveys an area within multiprocessor architecture. Topics are not limited to areas covered in this course. The paper should summarize all relevant work, provide extensive references, and provide opinions of different authors for or against particular proposals. The survey should conclude with your opinion about the strengths and weaknesses of different approaches, and your recommendations of when it is appropriate to implement a specific solution. You will be graded on the survey completeness, relevancy of references, accuracy of your summary, and the quality of your opinion and presentation. Expected number of participants: 1-2.

Roles and Responsibility of Each Member in the Project

Each project team member needs to contribute to the project. In the final project material submission, in addition to submitting the final project report and all source code, each team needs to also submit a Project Roles & Responsibility document. In this Roles & Responsibility document, you need to list who did what for the project for every member of the project team. All team members shall work together to define and agree on the roles and responsibility of each team member for the project and the amount of work distributed to each team member shall be fairly even, thus ensuring every team member contribute to a significant portion for the project. It is the student's job to actively and regularly participate in team meetings, work with team members collaboratively and actively contributes to the project. At the final project presentation, TA and/or instructor will review and ask questions to every student on the project team on the individual student's contribution to the project.