15-418/618, Fall 2019
Final Project Proposal and Information

Assigned: Wednesday, October 23rd

Your 15-418/15-618 final project gives you the opportunity to dive deeply into a parallel systems problem of your choosing for the final month of the course. What you attempt for your project is completely up to you. There are only two requirements: (1) We want your project to be challenging (you should learn something relevant to the themes of this class) and (2) we want your project to be fun (you should be pumped to work on it)!

Key Deadlines

Description	**Due When**
Discuss Idea with Instructors	Monday, October 28th, 9:00-10:30am or Tuesday, October 29th, 9:00-10:30am
Project Proposal Due	Wednesday, October 30th, 11:59pm
Checkpoint Report Due	Monday, November 18th
Final Report Due	Monday, December 9th
Poster Session in GHC 6115	Tuesday, December 10th, 1:00-4:00pm

Choosing a Project

One common way to choose a project is to design a parallel solution to a problem in an application area that is interesting to you. Projects you have attempted in other classes are a good source of ideas. For example, projects in machine learning, AI, graphics, computational photography, and computer vision often stand to benefit greatly from parallelization. If you can convince the course staff that a parallel programming problem in one of these application domains is sufficiently challenging (that is, the solution to get good speedup is not obvious to you from the start), it's likely it will make a good project.

Other project ideas focus on system design or workload evaluation. For example, a project might compare the performance of CPU and GPU implementations of a parallel algorithm, and describe which platform is better suited for the job. Alternatively you could choose to evaluate different versions of an algorithm for different architectures. You could simulate the behavior of code on machines with different SIMD widths, add a feature to the ISPC compiler (its implementation is open source), or develop a parallel debugging tool that helps visualize bottlenecks and performance in parallel programs.

You may implement your project on any parallel platform. The machines in the GHC labs, the latedays cluster, GPUs, Blacklight, Amazon EC2, iPhone/iPad/Android SoC, FPGAs, simulators are all possible and welcome platforms for projects. If you need specific resources, please let the staff know immediately and we'll see what we can provide. For example, many professors have high core count machines (32-64 cores) for the research groups and it is likely possible to get you access for your projects.

Coming up with a project is up to you. You can find a list of previous projects Spring 2012, Spring 2013, Spring 2014, Spring 2015, Fall 2016, Fall 2017, and Fall 2018.

Meet with Course Instructors (Monday (10/28) or Tuesday (10/29))

Rather than having a lecture during our usual lecture slot on Monday, October 29th, we will use that time instead to have individual meetings between project groups and one of the course instructors. In addition to this Monday meeting slot, we will have a second meeting slot the following morning. Each group should sign up for just a single slot on one of the two days (not on both days). These meetings will be held in the instructors' offices (not in the lecture hall).

There is a two step process for signing up for a meeting time:

1. Please go to the following SignUpGenious page to sign up for a single meeting slot (remember, just one slot per group): https://www.signupgenius.com/go/8050448a5a62ba13-15418618
2. Once you have your meeting slot, please use the following Google form to enter information about your group and your project ideas: https://forms.gle/emVHAYsdoABrSRvY7

Please also send email to whichever instructor you are scheduled to meet with (by 5pm the day before) describing your current thoughts on what you would like to do. This will help to make the meetings more productive.

Project Proposal (due Wednesday, October 30th)

The purpose of the proposal is two-fold:

1. Writing your ideas down forces you to organize your thoughts about your project.
2. It gives 15-418/618 course staff the ability to verify your plans are of the right scope given our expectations (it also gives us the ability to offer suggestions and help).

Please create a **web page** for your project. Your project web page should contain the following sections and content:

TITLE: Please provide the title of your project, followed by the names of all team members. Teams must be two students. There are no exceptions to this rule.

URL: Please provide the URL for your project web page.

SUMMARY: Summarize your project in no more than 2-3 sentences. Describe what you plan to do and what parallel systems you will be working with. Example one-liners include (you should add a bit more detail):

- We are going to implement an optimized Smoothed Particle Hydrodynamics fluid solver on the NVIDIA GPUs in the lab.
- We are going port the Go runtime to Blacklight.
- We are going to create optimized implementations of sparse-matrix multiplication on both GPU and multi-core CPU platforms, and perform a detailed analysis of both systems' performance characteristics.

- We are going to back-engineer the unpublished machine specifications of the GPU in the tablet my partner just purchased.
- We are going to implement two possible algorithms for a real-time computer vision application on a mobile device and measure their energy consumption in the lab.

BACKGROUND: If your project involves accelerating a compute-intensive application, describe the application or piece of the application you are going to implement in more detail. This description need only be a few paragraphs. It might be helpful to include a block diagram or pseudocode of the basic idea. An important detail is what aspects of the problem might benefit from parallelism and why?

THE CHALLENGE: Describe why the problem is challenging. What aspects of the problem might make it difficult to parallelize? In other words, what to you hope to learn by doing the project?

- Describe the workload: what are the dependencies, what are its memory access characteristics? (is there locality? is there a high communication to computation ratio?), is there divergent execution?
- Describe constraints: What are the properties of the system that make mapping the workload to it challenging?

RESOURCES: Describe the resources (type of computers, starter code, etc.) you will use. What code base will you start from? Are you starting from scratch or using an existing piece of code? Is there a book or paper that you are using as a reference (if so, provide a citation)? Are there any other resources you need, but haven't figured out how to obtain yet? Could you benefit from access to any special machines?

GOALS AND DELIVERABLES: Describe the deliverables or goals of your project. *This is by far the most important section of the proposal!*

- Separate your goals into what you PLAN TO ACHIEVE (what you believe you must get done to have a successful project and get the grade you expect) and an extra goal or two that you HOPE TO ACHIEVE if the project goes really well and you get ahead of schedule, as well as goals in case the work goes more slowly. It may not be possible to state precise performance goals at this time, but we encourage you be as precise as possible. If you do state a goal, give some justification of why you think you can achieve it. (e.g., I hope to speed up my starter code 10x, because if I did it would run in real-time.)
- If applicable, describe the demo you plan to show at the poster session (Will it be an interactive demo? Will you show an output of the program that is really neat? Will you show speedup graphs?). Specifically, what will you show us that will demonstrate you did a good job?
- If your project is an analysis project, what are you hoping to learn about the workload or system being studied? What question(s) do you plan to answer in your analysis?
- Systems project proposals should describe what the system will be capable of and what performance is hoped to be achieved.

PLATFORM CHOICE: Describe why the platform (computer and/or language) you have chosen is a good one for your needs. Why does it make sense to use this parallel system for the workload you have chosen?

SCHEDULE: Produce a schedule for your project. **Your schedule should have at least one item to do per week.** List what you plan to get done each week from now until the parallelism competition in order to meet your project goals. Keep in mind that due to other classes, you'll have more time to work some weeks than others (work that into the schedule). You will need to re-evaluate your progress at the end of each week and update this schedule accordingly. Note the intermediate checkpoint deadline is November 19th. In your schedule we encourage you to be precise as precise as possible. It's often helpful to work backward in time from your deliverables and goals, writing down all the little things you'll need to do (establish the dependencies!).

Submitting your proposal writeup: Please post your writeup on your project web page. (Note that you will also be posting updates and results to this web page, so your proposal should be a link on your web page, not the entire web page itself.) In addition, please submit your writeup to Gradescope (as a group submission). As noted earlier, that writeup should include a URL for your project web page. (Note: the Gradescope submission should be a PDF of the full proposal; not just a link to the proposal on your project web site.)

Project Checkpoint (due Monday, November 18th)

The checkpoint exists is to give you a deadline approximately halfway through the project. The following are suggestions for information to include in your checkpoint write-up. Your goal in the writeup is to assure the course staff (and yourself) that your project is proceeding as you said it would in your proposal. If it is not, your checkpoint writeup should emphasize what has been causing you problems, and provide an adjusted schedule and adjusted goals. As projects differ, not all items in the list below are relevant to all projects.

- Make sure your project schedule on your main project page is up to date with work completed so far, and well as with a revised plan of work for the coming weeks. As by this time you should have a good understanding of what is required to complete your project, I want to see a very detailed schedule for the coming weeks. I suggest breaking time down into half-week increments. Each increment should have at least one task, and for each task put a person's name on it.
- One to two paragraphs, summarize the work that you have completed so far. (This should be easy if you have been maintaining this information on your project page.)
- Describe how you are doing with respect to the goals and deliverables stated in your proposal. Do you still believe you will be able to produce all your deliverables? If not, why? What about the "nice to haves"? In your checkpoint writeup we want a new list of goals that you plan to hit for the poster session.
- What do you plan to show at the poster session? Will it be a demo? Will it be a graph?
- Do you have preliminary results at this time? If so, it would be great to included them in your checkpoint write-up.
- List the issues that concern you the most. Are there any remaining unknowns (things you simply don't know how to solve, or resource you don't know how to get) or is it just a matter of coding and doing the work? If you do not wish to put this information on a public web site you are welcome to email the staff directly.

- Meet with the course staff to discuss your progress.
- **Submit your checkpoint report** by (i) posting it to your project web page, and (ii) uploading it (a full PDF, not just a link) into Gradescope.

Final Project Report (due Monday, December 9th, 11:59pm)

You should write up your final project report in the style of a research paper. Please go into detail regarding the analysis that you did throughout your project, including any initial approaches that did not work well (and how you diagnosed and fixed their performance problems), etc. To go into this amount of detail, a typical report might be roughly 10 pages long (double-spaced, including figures), although that is just a ballpark estimate (we will not be counting pages). While you have flexibility in how you structure you report, we suggest that you include the following sections. (You are also encouraged to provide more detail if you wish.) Note that some of the information in your final writeup can be pulled directly from your proposal if it is still accurate.

SUMMARY: A short (no more than a paragraph) project summary. If applicable, the summary should list your project deliverables (including what you plan to show at the parallelism competition) and what machines they ran on. Here are some examples:

- We implemented smooth particle hydrodynamics in CUDA on the GPU and in ISPC on the CPU and compared the performance of the two implementations.
- We parallelized a chess bot. Our 64 core implementation on Blacklight achieves a 40x speedup and won several games on an internet chess server.
- We accelerated image processing operations using the GPU. Given the speed of our implementation, we demonstrate that a brute-force approach to breaking CAPTCHAS is effective.

BACKGROUND: Describe the algorithm, application, or system you parallelized in computer science terms. Figures would be really useful here.

- What are the key data structures?
- What are the key operations on these data structures?
- What are the algorithm's inputs and outputs?
- What is the part that computationally expensive and could benefit from parallelization?
- Break down the workload. Where are the dependencies in the program? How much parallelism is there? Is it data-parallel? Where is the locality? Is it amenable to SIMD execution?

APPROACH: Tell us how your implementation works. Your description should be sufficiently detailed to provide the course staff a basic understanding of your approach. Again, it might be very useful to include a figure here illustrating components of the system and/or their mapping to parallel hardware.

- Describe the technologies used. What language/APIs? What machines did you target?

- Describe how you mapped the problem to your target parallel machine(s). IMPORTANT: How do the data structures and operations you described in part 2 map to machine concepts like cores and threads. (or warps, thread blocks, gangs, etc.)
- Did you change the original serial algorithm to enable better mapping to a parallel machine?
- If your project involved many iterations of optimization, please describe this process as well. What did you try that did not work? How did you arrive at your solution? The notes you've been writing throughout your project should be helpful here. Convince us you worked hard to arrive at a good solution.
- If you started with an existing piece of code, please mention it (and where it came from) here.

RESULTS: How successful were you at achieving your goals? We expect results sections to differ from project to project, but we expect your evaluation to be very thorough (your project evaluation is a great way to demonstrate you understood topics from this course). Here are a few ideas:

- If your project was optimizing an algorithm, please define how you measured performance. Is it wall-clock time? Speedup? An application specific rate? (e.g., moves per second, images/sec)
- Please also describe your experimental setup. What were the size of the inputs? How were requests generated?
- Provide graphs of speedup or execute time. Please precisely define the configurations being compared. Is your baseline single-threaded CPU code? It is an optimized parallel implementation for a single CPU?
- Recall the importance of problem size. Is it important to report results for different problem sizes for your project? Do different workloads exhibit different execution behavior?
- **IMPORTANT:** What limited your speedup? Is it a lack of parallelism? (dependencies) Communication or synchronization overhead? Data transfer (memory-bound or bus transfer bound). Poor SIMD utilization due to divergence? As you try and answer these questions, we strongly prefer that you provide data and measurements to support your conclusions. If you are merely speculating, please state this explicitly. Performing a solid analysis of your implementation is a good way to pick up credit even if your optimization efforts did not yield the performance you were hoping for.
- **Deeper analysis:** Can you break execution time of your algorithm into a number of distinct components. What percentage of time is spent in each region? Where is there room to improve?
- Was your choice of machine target sound? (If you chose a GPU, would a CPU have been a better choice? Or vice versa.)

REFERENCES: Please provide a list of references used in the project.

LIST OF WORK BY EACH STUDENT, AND DISTRIBUTION OF TOTAL CREDIT: Please list the work performed by each partner. Given that you worked in a group, how should the total credit for the project be distributed amongst the participants? (e.g., 50%-50%, 60%-40%, etc.) If you do not feel comfortable placing this information on a public web page, it is okay to include it only on the version that you submit via Gradescope.

Submitting your report: Your project report should be (i) submitted via Gradescope, and also (ii) posted to your project web page. In addition, before the start of the poster session, please also submit a tar file with your source code via Autolab, and a copy of your poster (details described below) via Gradescope.

Project Poster Session (Tuesday, December 10th, 1:00-4:00pm, Location TBD)

Rather than having oral presentations to present your projects to the class, we will instead have a poster session on the last day of class. We will provide the easels, posterboard, and tape.

You should bring up to **eight 8.5x11" pages** to tape onto the posterboard. When you first arrive, you will tape your pages to the posterboard, put them on your easel, and then you are ready to go. (You can keep the posterboards at the end.) Note: if you prefer to print out your poster on poster-sized paper, you are free to do that, but that is optional. (See https://computing.cs.cmu.edu/desktop/printing-posters.html in case you do want to print in poster format.)

The advantage of a poster session is that it is interactive: you can ask other people questions one-on-one, and you can prioritize your time based upon which projects interest you the most. One member of your group should stand near your poster while the other group member circulates to ask people about their posters. You will take turns doing this, so everyone will spend part of the time presenting their poster, and part of the time asking others about their posters. The poster session is open to the public. Please feel free to invite others to attend it.