Template for Project Proposal

CS/ECE 5~~5~~504: Computer Architecture (W. Feng)

At the end of this write-up, you will find a list of sample ideas for final projects for CS/ECE 5504: Computer Architecture. Whether you choose one of the project ideas or propose a project idea of your own, you must flesh out your project idea into a modest proposal write-up. The proposal write-up should consist of the following and *not* exceed four (4) pages; references may appear on a separate fifth page, if needed:

1. Abstract
   - This summarizes what you are proposing along with expected artifacts and/or outcomes
2. Introduction
   - This motivates and frames your project proposal. Why is it important to do your proposal from a technical research perspective?
3. Related Work
   - This presents research work that has been done and is related to yours. (Make sure to always *cite* your sources.)
4. Proposal
   - This is where the "meat" of your proposal will be. Articulate what you are proposing in sufficient detail that it can be followed and reproduced.
   - What expectations, if any, do you have with respect to result(s)?
5. Timeline
   - This will provide a proposed timeline of milestones. It is highly recommended that you come up with a ***weekly schedule of milestones***, while also making sure to leave sufficient time to transform your proposal write-up into a final project research paper.
6. References
   - This is a list of citations of related work for your proposal.

For this project proposal and for the final project, you may work in groups of two CS/ECE 5504 students (<u>with the expectation that the scope of the project will be proportional to the number of people in the group</u>). While it is recommended to work in groups of two so you can get something a bit more substantive done, working alone is also fine. Below are some guidelines and caveats:

1. If working in a group of two, *EACH PERSON* from that group of two must separately submit the same proposal document to Canvas, which must also include identical lists of the names of the team members. (If working alone, just submit as usual.)
2. If working in a group of two, choose your partner wisely. You may *not* switch groups once you have submitted this project proposal assignment.
3. It is highly recommended that a github repository be configured for your team to house any tangible artifacts or experimental results, which in turn, you will report on in your final project research paper.

Your Task

You must identify and propose a non-trivial project in parallel computer architecture. Ideally, the project would align with or relate to your research interests, whether you propose your own idea or select from the project list at the end of this write-up. You are strongly encouraged to use the *rlogin* or *glogin* resources provided by the Department of Computer Science. (If you want to use other computing resources that are better suited to your project, you must *explicitly articulate what those resources are, including their specifications.* If these other resources are CPU- or GPU-based, then you should also ensure that they can run on *rlogin* or *glogin* as well.)

Your project should result in a brief research paper at the end of the semester (e.g., a paper that could be submitted to a workshop for peer-reviewed publication) that could be eventually turned into a conference submission in the future (if work were to continue on it).

Task Outline: Guidelines

1. Provide a project title and author list for your project proposal.
2. Identify a project idea (or goal) and describe it in a paragraph or two.
3. Scope the extent of the project, e.g., start with a bulleted list of what you propose to do.
    - What do you propose to do?
    - Why is it important?
    - What is the current "state of the art" in the literature? How is what you propose similar or different? In short, do a literature search to understand the context of what you are proposing and cite your references in your project proposal. (It is ok to update, validate, and verify existing research results but for current CPU and GPU hardware technology. In addition, you must cite the research that you are updating, validating, and verifying.)
    - What will your approach be towards achieving what you propose? In short, explain your approach in further detail.
    - Articulate the resources, particularly computing infrastructure, that you will use to complete your project.
4. The project proposal should not exceed four (4) pages in length, not including references (i.e., references may spill onto additional pages beyond the 5 pages), in IEEE conference format: https://www.ieee.org/conferences_events/conferences/publishing/templates.html

What to Submit

- *Everyone* must submit their proposal to Canvas.
- If you work in a team to two,
    - Your submissions should be identical and should explicitly specify the team members.
    - You should keep a journal or blog of project progress to document your progress, preferably on the aforementioned github repository.
    - You must remember that <u>the expectation is that the scope of the project will be proportional to the number of people in your group</u>.

SAMPLE PROJECT IDEAS

POWER ISA for gem5:

a. [gem5]: Implementation of one or more system-level features in GEM5 for POWER ISA, e.g., as noted above, byte swapping, floating-point arithmetic, file I/O operations, and/or random number generation.

b. [gem5]: Implementation of an out-of-order execution CPU (i.e., O3CPU) that supports the POWER ISA
   - Debug existing issues with the O3CPU implementation in GEM5 for POWER ISA
   - Add the necessary support in the O3CPU model for POWER ISA (https://www.gem5.org/documentation/general_docs/cpu_models/O3CPU

Simulation-Based Projects for the POWER Instruction Set Architecture (ISA)

1. Systems-Based Projects (Functionally Accurate & Cycle Accurate)

   The gem5 build for the POWER ISA supports integer arithmetic, but it does *not* support floating-point arithmetic, file I/O operations, and random number generation. So, to evaluate system-based projects on GEM5 for the POWER ISA, use application workloads that do *not* require the unsupported features.

   b.c. [gem5]: Performance evaluation of a three-level cache hierarchy in GEM5
   - Add support for L3 caches in your configuration script and evaluate the performance of at least two workloads on the POWER8 and POWER9 architectures. Do the results provide insight as to why IBM changed their memory hierarchy between POWER8 and POWER9?
   - Vary the L1, L2, and L3 cache parameters and identify the best cache configuration for representative workloads of your choice, e.g., floating-point intensive vs. integer intensive.

   f.d. [gem5]: For one or more workloads of your choice, vary the configuration parameters in GEM5 (e.g., cache size, associativity, replacement policy, branch prediction policy) and identify the best configuration for the workload(s).

   g.e. [gem5]: Performance evaluation of one (or more) workloads of your choice on multiple architectures (POWER/x86/ARM) in GEM5, i.e., fix your workload but vary the ISA, to understand the efficacy a particular ISA has on a particular workload or workloads. Present a thorough quantitative and qualitative analysis of the performance.

   h.f. [gem5]: Modeling and evaluation CPU-GPU interaction. `gem5-GPU` combines GEM5 and GPGPU-Sim to model tightly integrated CPU-GPU systems (with memory access being routed through Ruby). Experiment with the simulation of both systems – CPU and GPU – with separate CPU and GPU physical address spaces or with coherent caches and a single virtual address space across the CPU and GPU. See Lowe-Power et al., “gem5-gpu: A Heterogeneous CPU-GPU Simulator."
   - The use of open standards for tight integration of CPU and GPU is preferred, e.g., OpenCAPI rather than the proprietary NVLink.

2. Systems-Based Projects (ONLY Functionally Accurate)

   a. [POWER10 functional simulator]: Compare the performance of a set of workloads with and without -O3 in P10 functional simulator. Compare the observed performance with the performance of the same implementations on real hardware.

   b. [POWER10 functional simulator]: Manual vectorization of a workload using 1) AltiVec extensions and 2) using MMA instructions. Compare the observed performance with the performance of the same implementations on real hardware.

3. Application-Oriented Projects (Functionally Accurate, Cycle Accurate, or Both)

   a. [gem5]: Select applications that can be supported by the current POWER ISA implementation in GEM5 and characterize the level of robustness of the POWER ISA to execute the applications.

      - Holy Grail: Run LINPACK on POWER ISA in gem5

   b. [POWER10 functional simulator]: Functional verification of a set of workloads in the POWER10 functional simulator (https://www.ibm.com/support/pages/node/6493433)

   c. [POWER10 functional simulator + GEM5]: A combination of 3.a and 3.b

- Application characterization study on a fixed POWER-based architecture (e.g., POWER9) but varying the workload (hmm … integrate this with part a above

FPGA Simulation/Emulation Projects for the POWER ISA

Microwatt supports simulation of bare-metal C programs for the POWER ISA. See https://github.com/antonblanchard/microwatt

1. Systems-Based Projects

   a. Look at many of the aforementioned GEM5 projects in the context of the Microwatt environment (https://github.com/antonblanchard/microwatt) but work on a smaller subset of workloads, for example, due to the additional overhead of having to configure an FPGA or an FPGA simulator.

   b. [Microwatt + GHDL] GHDL (VHDL→low-level Verilog→C) easier to run in but slower. Look at system aspects of configuration and evaluation of workloads.

   c. [Microwatt + Verilator] Verilator (Verilog→C++) faster but harder to run in. Look at system aspects of configuration and evaluation of workloads.

   d. [Microwatt] Add new instructions in Microwatt

      i. Demo from the Open Power Summit: https://youtu.be/g3slH03MCmo?t=526

      ii. Step by step instructions for adding a "wait" instruction. Great simple example. https://www.talospace.com/2019/09/a-beginners-guide-to-hacking-microwatt.html

   e. [A2I + A2O + Verilator/FPGA] Install and run the unit tests for A2O (https://git.openpower.foundation/cores/a2o/src/branch/master/dev) using Verilator (or an FPGA board). Compare and analyze the performance of the code samples with the in-order execution core A2I((https://github.com/openpower-cores/a2i/tree/master/rel)

2. Application-Oriented Projects
    a. [Microwatt + GEM5]: Simulate a workload in Microwatt (https://github.com/antonblanchard/microwatt) and the same workload in GEM5, vary the configurable parameters (e.g., cache size, associativity, replacement policy, branch prediction policy) and identify the best configuration for that workload in Microwatt and in GEM5. Are the results consistent or wildly different? What are the implications of this?
    b. [Microwatt on FPGA → This project requires access to an FPGA]: Synthesize Microwatt on an FPGA (https://github.com/antonblanchard/microwatt#synthesis-on-xilinx-fpgas-using-vivado) and evaluate the performance of a workload of your choice on the POWER ISA.
    c. [Microwatt POWER → POWER CPU]: Compare and contrast the performance of one or more workloads on Microwatt running POWER ISA versus an actual POWERxy CPU, e.g., POWER8, POWER9, or POWER10. Look at total cycles, total execution time, and if infrastructure allows, power consumption and energy efficiency with respect to performance per watt.

Propose Your Own Project

1. ***Build on your answer from the exercise on CH-I, Problem M.4 (d)***, i.e., "(d)
   Based on the GEM5-based problems assigned in this and previous homework assignments, write a detailed paragraph about extensions to this case study on the POWER memory hierarchy that could be pitched as potential (modest) project proposals."
2. ***Build on your answer from the exercise on CH-II Problem I.1 (d)***, i.e., "(d)
   Based on the GEM5-based problems in this assignment, write a detailed paragraph about extensions to this case study and/or the case study from HW #4 on the POWER memory hierarchy that could be pitched as potential (modest) project proposals."
3. Propose something else that is based on parallel computer architecture beyond SISD processing via instruction-level parallelism (ILP), e.g., SIMD via data-level parallelism (DLP) or MIMD via thread-level parallelism (TLP), which will be covered during the last month of the semester.