

Final Report

Project Title: Dealing With Memory in the Parallel Computing Era

Participants:

- Adviser: Prof. Mohamed Zahran mzahran@cs.nyu.edu
- Student: Katelyn S. Mulgrew ksm348@nyu.edu
- Student: Sanjna Verma sanjnaverma@nyu.edu

Institution: Computer Science Department, New York University

General Project Description:

The proposed project involves two memory systems related issues.

- The first is to design a library for data structures to be used in parallel programming. The traditional data structures, such as queues, stacks, trees, lists, etc, are very well known to programmers and used extensively in the vast majority of programs. But what if two threads, in a parallel program, try to access the stack for example? Race condition will lead to non-deterministic behavior. The straightforward solution is to use locks. But there will be a big performance hit due to synchronization overhead. That part of the project will build a library of data structures that is highly optimized for performance, yet easy to use by programmers.
- Any data structure will grow/shrink at runtime, which makes dynamic allocation a crucial step. This is the second part of the project. In that part, students will explore how to build a memory allocator for multithreaded programs. Dynamic memory allocation for a single-threaded process has been extensively studied in literature. But for a multithreaded parallel program, when several threads share the same heap, things become more challenging due to contention, fairness to all threads (You do not want some threads to wait for long time before getting their requested memory slice!), and so on. Because we are talking about threads, then the whole project will be targeting shared memory architecture, which is the architecture of all multicore processors currently available in the market. Therefore, the library for parallel data structure and the library for dynamic memory allocation will have very wide applicability and usage.

The plan was that the two students will get trained to write parallel programs in OpenMP. We picked OpenMP because it is the best one that has a good compromise of learning curve (students know C) and performance.

- We spent summer of 2015 reviewing C programming. Even though students know C, but I wanted from them to be experts.
- Starting with Fall 2015, students started learning parallel programming on shared-memory machines.
- Near the end of Fall 2015, we started branching:
 - Katelyn started reviewing the literature and working on parallel data structure.
 - Sanjna started reviewing the literature and working on memory allocators. She studied HOARD memory allocator in details.

During the whole period, and through Spring 2016 we kept in touch as follows:

- Weekly meeting for one-hour where we brainstorm and each student plays the devil's advocate for the other student's work.
- We have shared documents on Google docs where they write the survey, their ideas, and their questions. I answer their questions and comment on their work in that same document. We had 3 documents: one for C and OpenMP, one for the parallel data structure part, and one for the parallel memory allocator part.
- To the outside world, students keep the blog: <http://creucrow.blogspot.com/>

However

Some unfortunate events brought this project to a halt by Feb 2016.

- Katelyn emailed me that she won't be able to continue working on the project due to other commitments and that she decided to graduate that same semester so she had to concentrate on courses only.
- Few weeks later, Sanjna got hospitalized and diagnosed with dehydration, fatigue, and pneumonia two weeks ago. She fainted and her roommate had to call 911. Therefore, she had to drop all activities and some courses.

Despite that, I can say that students completed about 60% of the project though.