



CSCI-GA.3033-009  
**Multicore Processors:  
Architecture & Programming**

**Suggested Projects**

Mohamed Zahran (aka Z)  
mzahran@cs.nyu.edu  
<http://www.mzahran.com>



# The project part of the course

- 50% of the total grade
- groups of 1 or 2 students
- presentations in the last two lectures of the course
- You have 3 choices:
  - pick one of the suggested projects
  - suggest a modifications to a suggested project
  - suggest a different project

# Project:

## Parallel Computation Model

- Survey of parallel computational models currently available
- show their strengths and weaknesses
- propose your own
- compare all of them

# Project:

## Heap management for multithreaded programs

- Design your own malloc/free
- Compare it to HOARD
  - <http://www.hoard.org/>
- Criteria:
  - speed
  - heap efficiency

# Project:

## Pick an application to parallelize

- Must demonstrate lessons beyond "I implemented X on the multicore, and it runs faster than a sequential version".

These lessons may include:

- Interesting generalized algorithmic or data structure contributions that can be applied to other applications (and these other applications should be made explicit)
- Interesting optimization techniques that can be applied beyond the application in the paper
- Interesting data management/representation techniques: how to best leverage the memory hierarchy, how to represent data in the most suitable way
- Rigorous comparisons with existing implementations.

# Project:

## A benchmark to compare several programming languages

- A good comparison helps understand the tradeoffs made in the design of a programming language.
- What are the criteria?
- The project should have some introductory material describing the different categories of benchmarks that were used, and why.
- A good comparison must address 3 points.
  - You must make sure the quality of the benchmark implementations are uniform.
  - You must have an analytic performance target and detailed profiling so you can assess the quality of the language implementations.
  - You must assess the suitability of the algorithms relative to the design goals of the language.

# Project:

## Parallel Data Structure Library

- What data structure do we need for parallel programs?
- How will you manage race condition?
- Is it just extension to traditional data structure?

# Project:

## Cache replacement Policy for Shared Cache

- What block to kick when set is full?
- Must be cheap in terms of hardware usage.
- Measures of success:
  - higher hit rate than competition
  - minimum effect on access latency



# Milestones

## 1. Problem Definition Report (10%)

- What is the problem?
- Why is it important?
- Motivation
  - Example, model, etc.
- Survey:
  - What is the previous work **related** to this project?
  - What are the pros and cons of this related work?
  - What can we learn from it?
  - How will your work be different?

# Milestones

## 2. Final Report (25%)

- Format is a like a paper
  - 2 columns
  - Abstract
  - Introduction
  - Motivation & Problem definition
  - Previous work
  - Proposed Solution
  - Experiment setup
  - Results and Analysis
  - Conclusions

# Milestones

## 3. Presentation (15%)

- 15 minutes: 10 mins presentation + 5 mins questions
- 7 slides
  - slide 1: Title & names
  - slide 2: What is the project about?
  - slide 3: motivation
  - slide 4: proposed solution
  - slides 5 and 6: results
  - slide 7: conclusions
- If the slides has bullets → no more than 5 bullets per slide.
- Phrases not sentences
- Font cannot be less than 24