Programming Assignment 3 (DRAM Attacks and Mitigation Techniques)
CS 665: Secure Memory Systems, Fall 2018
Computer Science and Engineering
Indian Institute of Technology Kanpur
Due Date: 27th November, 2018, 11:55 PM

In this assignment, you will implement a Denial-of-service (DOS) attack at the DRAM level, and implement a mitigation technique to prevent the same.

For mounting a DOS attack, you have to write an attacker code that can deny service to other applications in a multi-core systems. To start with, you could use a simple matrix multiplication with large enough memory footprint as your victim.

This section will provide the task list which needs to be accomplished by you for this assignment. The task list for this assignment is as follows:

A. **[You must be kidding: 7 points]** You have to report the execution time of the victim when it runs alone on your laptop/system compared to when your attacker perform a DOS attack on the victim application. Pin your attacker code to one core and the victim application to another core to nullify the private cache effects.

   Report the slowdown (Execution-time-with-DOS/execution-time-when-alone). If the ratio is small, improve your attacker code so that the ratio will be big and may be bigger.

B. **[Time for the mitigation: 13 points]** Mimic part A with ChampSim by creating traces for your victim and the attacker. Run the 2-core setup with ChampSim and report slowdown again (IPC-when-alone/IPC-when-DOS). For finding out IPC-alone, run a 1-core trace of your victim with ChampSim. To bridge the gap between IPCs, change DRAM policies like: Address mapping, command scheduling policy, Row open/close policy, bank partitioning, spatial/temporal partitioning……. You are free to use anyone or all. The goal is to get an IPC of the victim/attacker that is unaffected by the attacker/victim. Once you reach it, (Voila, cheers). For this task, you will be changing only one file:

   https://github.com/ChampSim/ChampSim/blob/master/src/dram_controller.cc

[Note: All the tasks have to be be done on Linux Operating System. We will accept your submission via Canvas only, and other submission mode is strictly prohibited such as submitting via email or piazza. While using Piazza/email, use PA3 in the header]

Deliverables:

1. Source code for A and B. Document your code properly. Share your results and experiences in a pdf: why something worked/did not work. Any insights, any experiments that you did to validate your assumption/confusion. Make sure to document your code well. Also, In the pdf, provide division of labor: Who did what, Say Biswa1 did A, Biswa2 did B and 25% of C. Submissions will be graded based on the maximum slowdown in Task A and maximum degree of isolation in Task B.
2. Feel free to discuss about the assignment among your friends and with Biswa. However, at the end of the day, we want you to do everything on your own.