



INSTITUTE FOR APPLIED
COMPUTATIONAL SCIENCE
AT HARVARD UNIVERSITY



HARVARD
School of Engineering
and Applied Sciences

Guide: Performance Optimization on AWS

Ignacio M. Llorente, David Sondak, Simon Warchol

v3.0 - February 16, 2021

Abstract

This is a guideline document to show the necessary actions to set up and use `gcc` to evaluate its performance optimization support on Ubuntu (18.04).

Acknowledgments

The author is grateful for constructive comments and suggestions from David Sondak, Charles Liu, Matthew Holman, Keshavamurthy Indireskumar, Kar Tong Tan, Zudi Lin and Nick Stern.



1. Spin up EC2 instance

1. Using Lab 1 as a reference, please spin up a **t2.2xlarge** EC2 instance running Ubuntu 18.04, with your CS205-key
2. Connect to this instance w/ `ssh` (or `putty` on Windows), again using your CS205-key.

2. Install gcc

3. Install `gcc` via the toolchain PPA

```
$ sudo apt-get install software-properties-common
$ sudo add-apt-repository ppa:ubuntu-toolchain-r/test
$ sudo apt-get update
$ sudo apt-get install gcc
```

4. To check the `gcc` installation is successful run following command in the terminal

```
$ gcc -v
```

3. Evaluate Performance Flags

This section includes a simple optimization session aimed at verifying the correct installation of the `gcc` compiler.

- Upload to the VM the [seq_mm.c](#) code and compile with several optimization flags (you also need [timing.c](#) and [timing.h](#)). This simple code performs a 1,500 by 1,500 matrix multiplication. See that by default the matrices are created in the stack of the process (8MB), you should use `ulimit -s 64000` to increase the stack to < 64MB, which is the hard limit for the stack size.

```
$ gcc -DUSE_CLOCK seq_mm.c timing.c -o seq_mm -lm
$ gcc -O3 -DUSE_CLOCK seq_mm.c timing.c -o seq_mm_O3 -lm
$ time ./seq_mm > output
```

```
real 0m28.533s
user 0m28.380s
sys 0m0.052s
```

```
$ time ./seq_mm_O3 > output
```

```
real 0m3.964s
user 0m3.836s
sys 0m0.032s
```

Stop your instances when are done for the day to avoid incurring charges



Terminate them when you are sure you are done with your instance