IoTSSC LAB #7

LAB EXERCISE: PERFORMANCE MEASURING AND OPTIMIZATION ACROSS THE IOT STACK

OVERVIEW

In this lab you will focus on different methods for measuring performance optimization on the embedded device, the android application and the cloud. Specifically:

1) Investigating memory usage and stack and heap profiling.

2) Investigating raw CPU usage of your embedded device.

3) Profiling android application performance.

4) Profiling Google cloud services

REQUIREMENTS

For the embedded system component you will need the **FRDM-K64F** board.

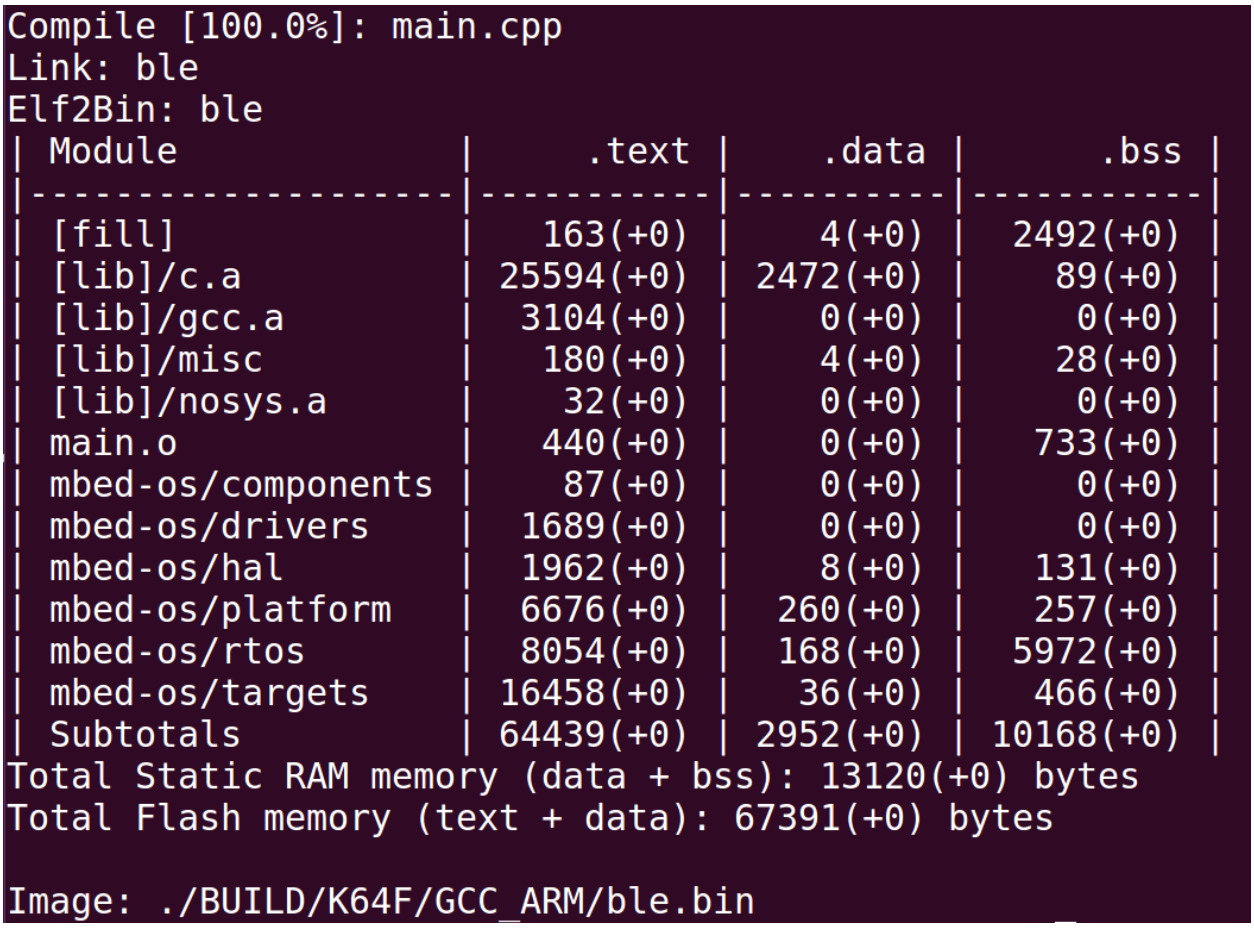
For the android app profiling component you will need an android application (preferably the working Bluetooth application introduced in Lab 5).

For the cloud profiling component you will need a working Google Cloud service that integrates with the Stackdriver logging tool. This includes Pub/Sub, App Engine or Google Cloud Functions.

EMBEDDED PROFILING

One simple way of measuring embedded performance is through the memory requirements on the code itself.

Whenever the command **mbed compile** is issued the memory requirements of the compiled program are displayed as followed:



We strongly recommend documenting these stats as you optimize your embedded application. Smaller memory requirements are **better** when comparing application that are functionally identical.

Digging deeper into the actual CPU usage of your embedded application, a dedicated tool is required:

https://os.mbed.com/users/dextorslabs/code/CPU_Usage/

The above library allows you to easily measure the CPU usage percentage in arbitrary locations in your code.

Experiment measuring CPU usage throughout your code base to find out which parts of your application are most heavily utilizing the CPU. Look for possible code optimizations in these areas.

ANDROID PROFILING

There are a number of ways to profile your app's performance built into Android Studio:

https://developer.android.com/studio/profile

Of particular interest is the ability to look at memory usage with the memory profiler:

https://developer.android.com/studio/profile/memory-profiler?hl=en

Run the memory profiler on your android application. Look for objects which take up large amounts of memory on the heap. Are there any ways you can reduce the memory required for these objects?

Take a look at the energy profiler. Do you see any ways to reduce the energy used by the application?

CLOUD PROFILING

Below we have outlined profiling methods for different Google Cloud services. Feel free to only investigate the service that most closely fit your cloud implementation.

WITH GOOGLE COMPUTE ENGINE, KUBERNETES OR APP ENGINE

Google has a built-in profiling service for the above applications.:

https://cloud.google.com/profiler/docs/how-to

Try to incorporate calls to the profiler from within your cloud application. Can you make any changes to your code to improve efficiency?

WITH PUB/SUB

Pub and Sub exposes a set of metrics to Google's Stackdriver logging tool:

https://cloud.google.com/pubsub/docs/monitoring

Investigate visualising the metrics from your pub/sub application. Are there any metrics which expose a possible efficiency improvement in your message queue?

WITH GOOGLE CLOUD FUNCTIONS

Though there is no built-in support for profiling GCF it is possible to log performance metrics (such as execution times) to Stackdriver. The following two medium posts cover how to do this:

https://medium.com/@duhroach/profiling-gcf-functions-a3d325dc29e2

https://medium.com/@duhroach/getting-google-cloud-functions-times-in-stackdriver-61ec1b33a92

Try to implement execution time logging for your cloud application. See if you can make any efficiency improvements to your cloud code that can be visualised by reduced execution time.