

Homework 6 - Choose a Language for a Flynetting Swarm

Raymond Lin

6 December, 2019

CS131 Programming Languages - Fall 2019

Abstract

In this assignment, we are asked to research potential technologies that can be used to implement the machine learning software that will be run on a large number of devices. We must identify three aspects of the device: the programming language of the software, the software framework that will be used by the language, and the hardware device on which the software will run.

1. Introduction

XuFly is a startup company that is developing devices that have the ability to kill flying insects and other pests. The device consists of many small robots that fly and swarm insects in its vicinity. However, such a device has many technological requirements in order to be successful.

The software that runs on each robot is a machine learning algorithm. The robot must be able to collect data on what it sees and learn what a stationary or flying insect looks like. Machine learning algorithms are computationally heavy, and so we must choose technologies that are able to run efficiently and quickly. Multithreading is a way to speed up execution. The robots must also be able to communicate with each other quickly and reliably and maintain concurrent connections. Machine learning algorithms also take up a large amount of memory, because there will be a large amount of incoming data that must be stored at a given time. Furthermore, the software must be reliable, because we are developing a product. It must not crash or fail during runtime, or users will not be happy.

The hardware for each robot also has some requirements. It must have a small form-factor, because it must be able to fly faster than insects in order to swarm and kill them. Additionally, the hardware must also be energy efficient. Since the devices will not be wired, they will run on battery and should not need to recharge every couple of minutes. Finally, the hardware must be compatible with the required software that we choose.

In short, we must select the best combination of programming language, machine learning framework, and hardware that satisfies the above requirements.

2. The Programming Language

The programming language we select must be well suited for machine learning. This means that it must run fast and preferably have the ability to be multithreaded. In addition, it must also be reliable. Since we are writing software for production, it should be bug-free and ideally should not crash during runtime. Finally, the programming language should have support for machine learning libraries.

2.1 C++

The C++ programming language has been around for a long time, and it is still widely used today. C++ is a low-level programming language, meaning that it runs closer to the hardware and has a very fast runtime. Additionally, C++ can be multithreaded, which means that it can run even faster.

Although C++ runs very quickly, the programmer must be very careful when developing C++ code, because it is more prone to runtime errors. The developer must ensure that many edge cases are handled, and even then, a C++ implementation may experience crashes that result in undefined behavior. This, along with its verbosity, makes C++ not very reliable and is prone to errors and crashes.

2.2 Go

Go is a relatively new programming language. It boasts its ability to run programs concurrently. Go's special feature is known as a 'goroutine', a fast, light-weight version of a coroutine. Go supports multithreading, as each 'goroutine' can run on a separate thread. The developer does not need to worry about race conditions

and other multithreading issues, as this is all abstracted by the programming language. Furthermore, Go is a statically typed language, and therefore, it is a reliable language. There will be fewer runtime errors. Go also runs fast, as it is compiled directly to machine code.

However, Go has its downsides too. Because it is a relatively new language, it lacks the community, libraries and support that other, more established, languages have.

2.3 Java

Java is one of the most popular programming languages in the world. Java is also statically typed, and thus should be able to write reliable programs. Additionally, Java is a portable language, as it is run inside a Java virtual machine. Finally, Java has support for machine learning libraries and multithreading.

However, because of this, Java is not the fastest language. Furthermore, its syntax is similar to that of C++, making it verbose and harder to write than other languages.

2.4 Python

Python is, hands down, the most popular language used for machine learning today. According to Liam Tung of tech news website, ZDNet, Python is now used in approximately 69% of machine learning projects [1]. Its main attraction is its community and support for many machine learning libraries. Many developers state that a program written in another language can be written in Python in much fewer lines of code. Additionally, Python is dynamically typed, making it one of the easiest languages to write programs in.

However, even Python has its drawbacks. Because Python is an interpreted language, it runs slower than most languages. Additionally, although Python's dynamically typed nature makes it easy to write, it also means that it is more prone to error. Since types aren't checked at compile time, there could potentially be more crashes during runtime. Finally, Python's global interpreter lock prevents it from being truly multithreaded, thus reducing its runtime speed.

2.5 The Verdict

Analyzing the advantages and disadvantages of each programming language, we can come to a conclusion of which one to use for XuFly. XuFly has already tried

a C++ implementation, and the main problem with C++ is in its runtime errors and crashes. Python seems to be the obvious choice, due to its large application in machine learning. However, since we are not just training a model and are actually deploying this code to a consumer product, it must be reliable and fast. Java is another potential language. While it has support for machine learning libraries and multithreading and is not too slow, Java does not seem to excel in any one area. This leaves us with Go. Go is a great language as it runs extremely fast due to its compilation nature. It also was designed for concurrent and multithreaded applications, making it even faster. Its statically typed nature allows it to be reliable enough for production. Although Go has a newer, smaller community, it is increasing quickly and machine learning libraries and being developed rapidly.

3. Supporting Software

Aside from the programming language, we must use additional software to actually implement the machine learning algorithms.

3.1 Tensorflow Lite

Tensorflow Lite is an open source deep learning framework that is designed specifically for mobile and IoT devices. It is derived from the original Tensorflow, both of which are developed by Google. It works by converting a pretrained model into a compressed flat buffer. This buffer is then deployed onto the device and optimized by converting floats to integers. Tensorflow Lite works well with image classification, object detection and natural language processing.

3.2 Nvidia TensorRT

TensorRT is a platform for high performance deep learning. It boasts its low latency and high throughput for deep learning. TensorRT is developed by Nvidia and thus works on CUDA, their parallel programming model. This makes it a powerful tool for artificial intelligence, autonomous vehicles, computer vision and graphics. It also uses float to integer optimizations.

3.3 OpenVINO

OpenVINO, developed by Intel, is a toolkit that specializes in computer vision. Its strengths include deep learning at edge devices. An edge device is a device that can collect data, such as smartphones,

drones, IoT devices etc...

3.4 The Verdict

Of all three options, the most suitable supporting software for our needs would be OpenVINO. OpenVINO specializes in computer vision, which is exactly what XuFly needs. Its swarm of robots need to be optimized for computer vision, as they need to be able to recognize flying insects. Furthermore, OpenVINO is compatible with Go. There exists a language binding from Go to OpenVINO, using the GoCV package [2].

4. The Hardware

Finally, we need to select a hardware device to run the Go code and machine learning algorithms.

4.1 Coral SoM

Google's Coral SoM (System-on-Module) is a fully-integrated system for machine learning applications. It has a small form factor and includes many useful features such as Wi-Fi, Bluetooth, Edge TPU etc... It has a quad-core processor, 1 GB of RAM and 8 GB of flash storage.

4.2 Jetson Xavier NX

Nvidia's Jetson Xavier NX is an extremely powerful system for its size. It can run modern neural networks in parallel. It has a 6-core processor, 8 GB of RAM and 16 GB of storage.

4.3 Movidius VPU

Intel's Movidius VPU (Vision Processing Unit) is a system on a chip that is used in edge media and computer vision. It claims to be extremely power efficient, more so than the Jetson Xavier NX. Furthermore, it supports the OpenVINO toolkit.

4.4 The Verdict

The Coral SoM is the smallest system of the options. It is also the only one that is available in the market as of today. While it is weaker than the two other systems, it will most likely be the cheapest option. For cost effectiveness, this would be the best option.

However, it seems that the most compatible hardware product would be Intel's Movidius VPU. Because we are using OpenVINO, the Movidius VPU has direct support for this software. Furthermore, it supposedly consumes less power than the Jetson Xavier, a trait that we need from our system. From a compatibility standpoint, the Movidius VPU would be the best choice.

5 Conclusion

In conclusion, XuFly should choose the following for its flynetting swarm project. Go should be the primary programming language for its fast execution time, its multithreading ability, its concurrency ability, and its reliability. Intel's OpenVINO should be used for the machine learning algorithm, because it specializes in computer vision and is compatible with the Go programming language. Finally, the hardware should be Intel's Movidius VPU as it is power efficient and is compatible with OpenVINO.

References

- [1] Liam Tung, *Programming Languages: Python developers now outnumber Java ones*, April 15 2019, <https://www.zdnet.com/article/programming-languages-python-developers-now-outnumber-java-ones/>
- [2] GoCV, Github Repository <https://github.com/hybridgroup/gocv>