Capstone Project Proposal

Project Number: S16-031
Project Title: Distributed Machine Learning Using Raspberry Pis
Project Term: Spring 2016
Students: Nikhil Shenoy
Revan Sopher
Professor: Dr. Anand D. Sarwate

February 5, 2016

Description:

Distributed computing refers to the use of multiple networked computers to perform data analysis at scale, the cornerstone of the big data trend. Traditional efforts focus on using powerful consumer-grade workstations housed in a data center, but we propose the construction of a distributed computing cluster from multiple Raspberry Pis – inexpensive, low power devices suitable for embedded use.

This novel arrangement, generally defined as a sensor network, creates a new field of challenges, notably in working with the constraints of considerably weaker hardware, but allows for exciting real world scenarios – for example, distributed fault detection by installing the devices on highway infrastructure. Such an application would be ruled out by a traditional datacenter cluster – the workstations are too large and power-hungry to be deployed in the field, but power costs of maintaining a constant uplink from the sensors back to the data center are also prohibitive.

Several instances of sensor networks have been designed and implemented, with remarkable results. The company ShotSpotter has exploited the advantages of ad hoc sensor networks in order to monitor gun violence in urban areas; by deploying a series of microphones on the rooftops of buildings, the company can monitor entire neighborhoods for suspicious sounds. The system then collects the audio data recorded by each microphone and decides whether a gun was fired. We aim to develop a similar system using Raspberry Pis, with the intention that those with relatively little knowledge of distributed computing and sensor networks can obtain a cheap, portable, and efficient system with which they can analyze data.

Preliminary work suggests that deployment infrastructure to update the devices will be the first challenge. We expect to use Docker to containerize the application, and resin.io to push updates to the fleet. To abstract away the inter-device networking, we will base our data analysis on Apache Spark.