

parallel-ventriloquist-duel

~~a actor using a thread fights for his future~~

A repository for me to dive into Scala's parallel programming approaches. Consisting of pure threads, actors, futures, and possibly more!

Project Proposal

Introduction

From preliminary research, it seems that a technique known as actors resides on an even higher level of abstraction than futures. It may be an even better approach to parallelism that I have yet to learn, meaning it's time to get down to it. This project will serve as an opportunity to study different parallel approaches to problems, and how the size (and possibly type) of data affects runtime.

Rough Methodology

To this end, I think a suitable problem I can test is the web crawler. Even though I have done this in a recent assignment, I'd like to implement the different parallel methods involving at least an upgraded version of my collection of futures implementation, Akka actors, pure Java threads, concurrent hashmaps (which seem to use mutexes and partitioned segments? fact check required), and possibly more if time and accessibility allows it.

Next, the runtime of each of these can be measured by crawling different parts of the web. Some factors affecting performance that I want to test include:

- latency
 - this can be tested by crawling webpages hosted on opposite sides of the earth
- depth of crawl
- "size" of webpage (word count?)

Measuring Success

I feel that this project would be successful if complete benchmarks comparing each implementation of web crawling are made, as well as having good clean code. Being optimistic, learning of each of these approaches will keep me aware of the benefits and limitations of each implementation, as well as being able to code the "ever-awesome" concept of actors on other programming problems I will encounter in the future.