Joseph Myles 49961121 t8t8

James Deng 13340112 w5z7

Reed Mullanix 57312134 t1d9

Background Report: Why to use Rust-lang for a BitTorrent protocol implementation

I. Introduction

Our project is an implementation of BitTorrent in Rust-lang for a type 3 project. Starting at the most basic level, none of us knew Rust before we started working on the project. Once giving the project, we considered Rust based on interest in working with a fast and reasonably new language with plentiful possible future industry applications. Throughout the planning process, we compared Rust with other similar languages, like Go. Rust piqued our interest most due to our projection that Rust would be more widely applicable and widely accepted in the future. We feel that Go, by comparison, does not contain the same possibilities as Rust and that Rust, therefore, will be more widely accepted in industry. We therefore thought that learning Rust would be more beneficial and insightful for future work.

Once we set our main goal to working in Rust, our next step was to decide upon a program worth implementing. We considered a few options, including some server-connection protocols. We ultimately decided upon implementing a BitTorrent protocol based on Rust’s speed and ease of thread use. Again, we considered other language options to see if a BitTorrent protocol would be easier to implement in another language. For this purpose, we considered both languages’ speeds and threading usability. We also considered Rust’s lifetime management that leverages Rust’s guaranteed memory safety when comparing it to other options and applying it to a BitTorrent protocol. We concluded that Rust fitted our needs well, and better than comparable languages, based on the fact that threads are a reasonably easy-to-implement part of Rust and based on Rust’s memory safety. Comparisons of other languages with Rust for our specific implementation and other applications have been a frequent point of discussion with peers since our decision.

Next, we set out to put a cohesive plan together for the implementation of the BitTorrent protocol. Naturally, we started by reading BitTorrent and Rust-lang specifications in order to consider how we would implement the BitTorrent protocol in Rust. These works gave us insight into the methods for implementing a BitTorrent protocol. Specifically, the BitTorrent specification discusses the necessary components of implementation and gave us insight into which aspects of the BitTorrent protocol need to be implemented before others. These works also gave us insight into Rust-lang. Beyond learning the syntax of Rust, language specifications and coding examples taught us about use cases and strengths of Rust. Details of these and other works will be discussed in greater depth below.