# Code-related text search engine

Report

Yaveyn Anna
Simiyutin Boris

## 1 DESCRIPTION

Our project is a developer-oriented search engine. The main goal is to help people with technical problems, in a way similar to StackOverflow.

## 2 ARCHITECTURE AND DESIGN

We have selected Kotlin as our main programming language. Although, we keep our system modular and use other languages when it is convenient. For parallel programming we use Akka actor system.

### 2.1 Data acquisition

*2.1.1 General info.* Data acquisition in our system is performed by crawler, which collects data from the Internet, starting from popular programmer-oriented sites, such as www.linux.org, stackoverflow.com, cppreference.com and docs.oracle.com.

Our program meets politeness requirements for a web-crawler. We use open source RobotsTxt library for parsing robots.txt files and follow request rate conventions by 20 sec timeouts.

During development of crawler we faced performance issues, which were solved by making it multithreaded. We also had some troubles with RobotsTxt stability and intense memory usage.

*2.1.2 Concurrency.* We decided to perform data acquisition in parallel and we chose an actor-based concurrency model to do so. The main advantage of using actors is that you can write highly concurrent, distributed and fault-tolerant code without thinking about synchronization between different threads. One of the most popular toolkit for actors in JVM is Akka - a cross-platform open-source framework written in Scala. We use it because it is a well-known and powerful tool with comprehensive documentation and large community.

The crawler consists of one manager actor and ten worker actors. Worker actors perform the actual crawling and the manager actor handles the communication between them. Each worker is responsible for it's own range of host names and holds it's own queue of Urls. When worker comes across a host name which it is not responsible for worker sends request to manager and manager redirects the Url to corresponding worker. The manager is also responsible for assigning workers for new host names.

*2.1.3 Summary.* The resulting program runs pretty fast and uses network channel rather efficiently. However, we have some problems with rate limits at multi-domain resources e.g. stackexchange.org which we are going to fix on the next stage of the development.