

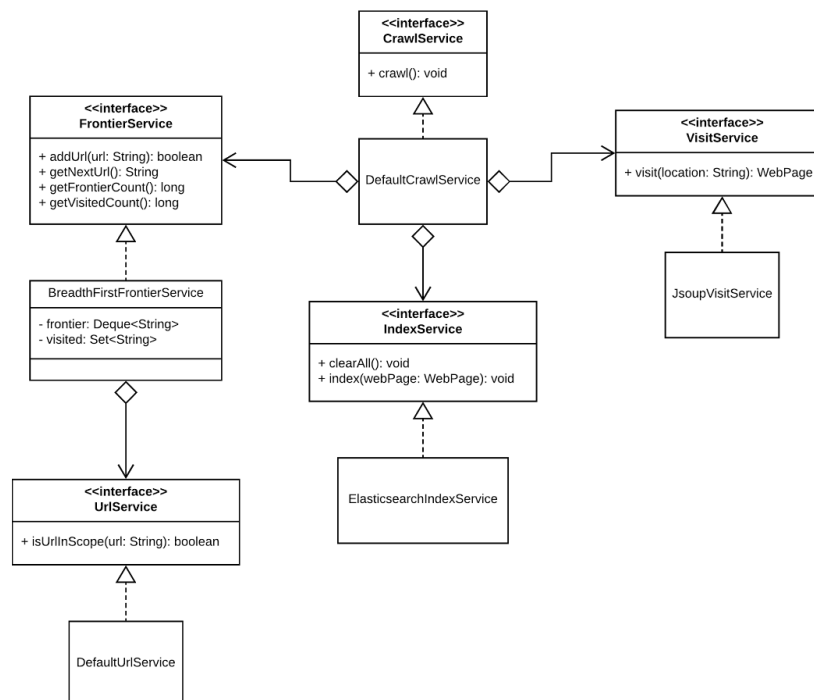
McKoon Spider: A Simple Web Crawler

Introduction

In this assignment, I designed and implemented my own web crawler, called “McKoon Spider”. As an experiment, I ran McKoon Spider against the cc.gatech.edu domain for almost eleven hours, indexed all visited pages, and collected runtime metrics. In this paper I will discuss the design of McKoon Spider, its strengths and weaknesses, and the metrics gathered from its eleven-hour crawl of the cc.gatech.edu domain.

Design

The McKoon Spider design consists of four primary components: FrontierService, VisitService, IndexService, and CrawlService.



- FrontierService
 - The FrontierService manages the frontier queue (URLs to visit) and the set of previously visited URLs. Its primary methods are for adding URLs to the frontier and getting the next URL to visit.
- VisitService

- The VisitService has a visit method that takes in a URL to visit and returns a WebPage object containing the extracted text, links, and other information about the page.
- IndexService
 - The IndexService has an index method that takes in a WebPage object. This service adds the WebPage data to a search index.
- CrawlService
 - The CrawlService has a crawl method that iteratively gets a URL from the FrontierService, visits the URL with the VisitService, indexes the web page with the IndexService, and then adds newly found URLs to the frontier with the FrontierService.

Implementation

McKoon Spider is a Java console application. The application uses Gradle to build an executable jar file containing all dependencies. The application uses Guice as its dependency injection framework (<https://github.com/google/guice>). For logging, the application uses SLF4J, via the Logback implementation (<https://logback.qos.ch>). Typesafe Config is used for configuration (<https://github.com/lightbend/config>).

For loading and parsing HTML web pages from URLs, the Jsoup library is used (<https://jsoup.org>). For gathering runtime metrics, the Dropwizard Metrics library is used (<http://metrics.dropwizard.io>), via metrics-guice annotations (<https://github.com/palominolabs/metrics-guice>).

McKoon Spider connects to an Elasticsearch cluster for indexing visited pages (<https://www.elastic.co/products/elasticsearch>). The application uses Jackson to convert WebPage objects into JSON, which it sends to Elasticsearch for indexing (<https://github.com/FasterXML/jackson>).

Front-end

In order to make querying and viewing the search index more enjoyable, I developed a simple front-end to query the Elasticsearch REST API. This front-end “spider-client” project is a simple AngularJS (<https://angularjs.org>) application using AngularJS Material (<https://material.angularjs.org>) for its UI elements. The spider-client uses elasticsearch-browser (<https://www.npmjs.com/package/elasticsearch-browser>) to send HTTP requests to the Elasticsearch cluster. To quickly build and serve the spider-client, the project uses Gulp (<https://gulpjs.com>).

McKoon Spider Search Client

Page search

Search

RESETSEARCH

Search results

No results found.


McKoon Spider Search Client


Page search


Search
Liu


RESETSEARCH

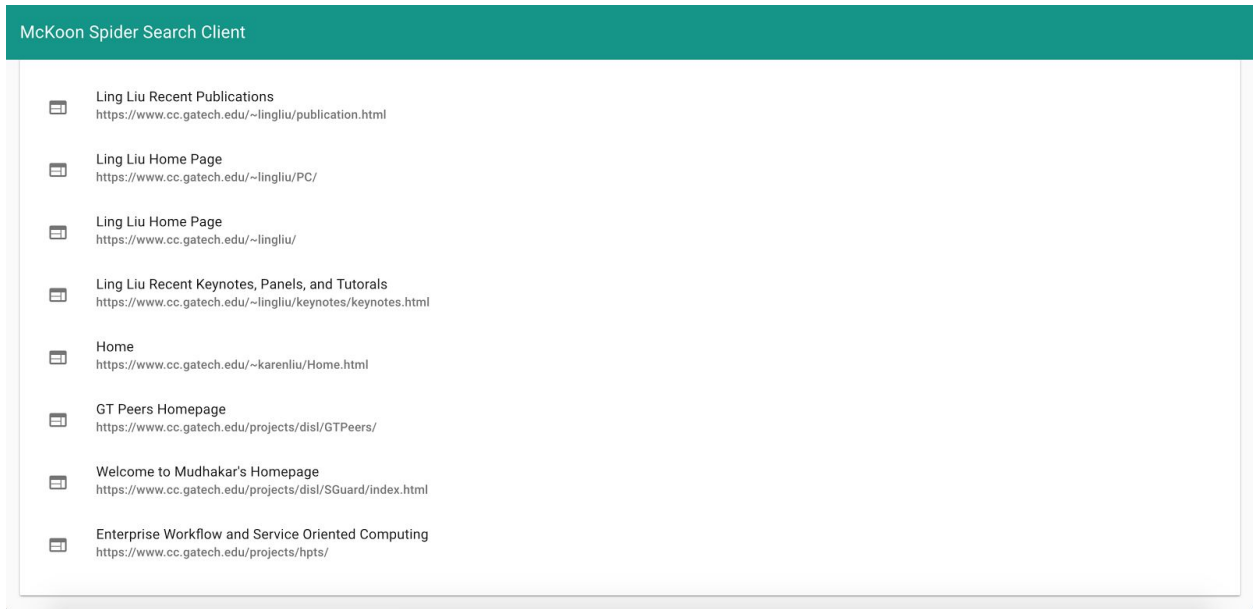
Search results

Ling Liu Home Page
<https://www.cc.gatech.edu/home/lingliu/>

Ling Liu Home Page
<https://www.cc.gatech.edu/home/lingliu/index.html#ResearchProjects>

Ling Liu Recent Publications
<https://www.cc.gatech.edu/~lingliu/publication.html>

Ling Liu Home Page



Environment

Hardware

The experiments were ran on a MacBook Pro (Retina, 15-inch, Mid 2014) with a 2.5 GHz Intel Core i7 processor, 16 GB 1600 MHz DDR3 memory, running macOS Sierra (10.12.6).

Virtual Machine Setup

A virtual machine was created to be the Elasticsearch cluster that the McKoon Spider uses for indexing. First, VirtualBox version 5.2.6 r120293 (Qt5.6.3) was installed on the host machine. Then a new virtual machine was created and allocated with 8192 MB of RAM and 20 GB of storage. Ubuntu 17.10 was downloaded and installed.

The virtual machine was then configured by installing gcc, make, and perl via the following commands, as these are dependencies of VirtualBox Guest Additions:

```
sudo apt-get update  
sudo apt-get install gcc make perl  
sudo apt-get install virtualbox-guest-dkms
```

Then vim was installed, and the window manager was swapped from Wayland to XOrg due to flickering issues.

```
sudo apt-get update
```

```
sudo apt-get install vim

sudo vim /etc/gdm3/custom.conf
(Uncommented "#WaylandEnable=false")
```

Then OpenJDK 8 was installed and configured, and git and ssh were installed:

```
sudo apt-get install default-jdk

sudo vim /etc/environment
Edited to contain: JAVA_HOME="/usr/lib/jvm/java-8-openjdk-amd64"

sudo ~/.bashrc
Edited to contain: source /etc/environment

sudo apt-get install git
sudo apt-get install ssh
```

Install Elasticsearch

To install Elasticsearch, I followed steps similar to those described at https://www.elastic.co/guide/en/elasticsearch/reference/current/_installation.html.

First, I installed curl and used it to download Elasticsearch. I then unpackaged it and moved the files to a permanent location under /usr/local. Then I added an environment variable for this folder to /etc/environment:

```
sudo apt-get install curl

curl -L -O https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.1.2.tar.gz

tar -xvf elasticsearch-6.1.2.tar.gz

sudo mv ./elasticsearch-6.1.2 /usr/local/elasticsearch-6.1.2

sudo vim /etc/environment
Edited to contain: ELASTICSEARCH_HOME="/usr/local/elasticsearch-6.1.2"
```

Elasticsearch Network Configuration

In order to be able to connect to the Elasticsearch cluster from outside the virtual machine, I completed the following steps.

```
sudo vim $ELASTICSEARCH_HOME/config/elasticsearch.yml
```

Edited to have the following line so it can be reached on the network:

```
network.host: 0.0.0.0
```

Also added these lines for XHR access from my custom search client:

```
http.cors:  
enabled: true  
allow-origin: "*"
```

Exposing Elasticsearch on a non-local IP address enables its production bootstrap checks. Several settings must be changed to pass these. Below are the configuration changes I made to satisfy these checks:

From <https://underyx.me/2015/05/18/raising-the-maximum-number-of-file-descriptors>

```
sudo vim /etc/security/limits.conf
```

Add these lines:

```
* soft nofile 65536  
* hard nofile 131072  
root soft nofile 65536  
root hard nofile 131072
```

```
sudo vim /etc/pam.d/common-session
```

```
session required pam_limits.so
```

```
sudo vim /etc/pam.d/common-session-noninteractive
```

```
session required pam_limits.so
```

```
sudo vim /etc/pam.d/su
```

```
Uncomment: # session required pam_limits.so
```

From

<https://superuser.com/questions/1200539/cannot-increase-open-file-limit-past-4096-ubuntu>

Modify `/etc/systemd/user.conf` and `/etc/systemd/system.conf` with the following line (this takes care of graphical login):

`DefaultLimitNOFILE=65536`

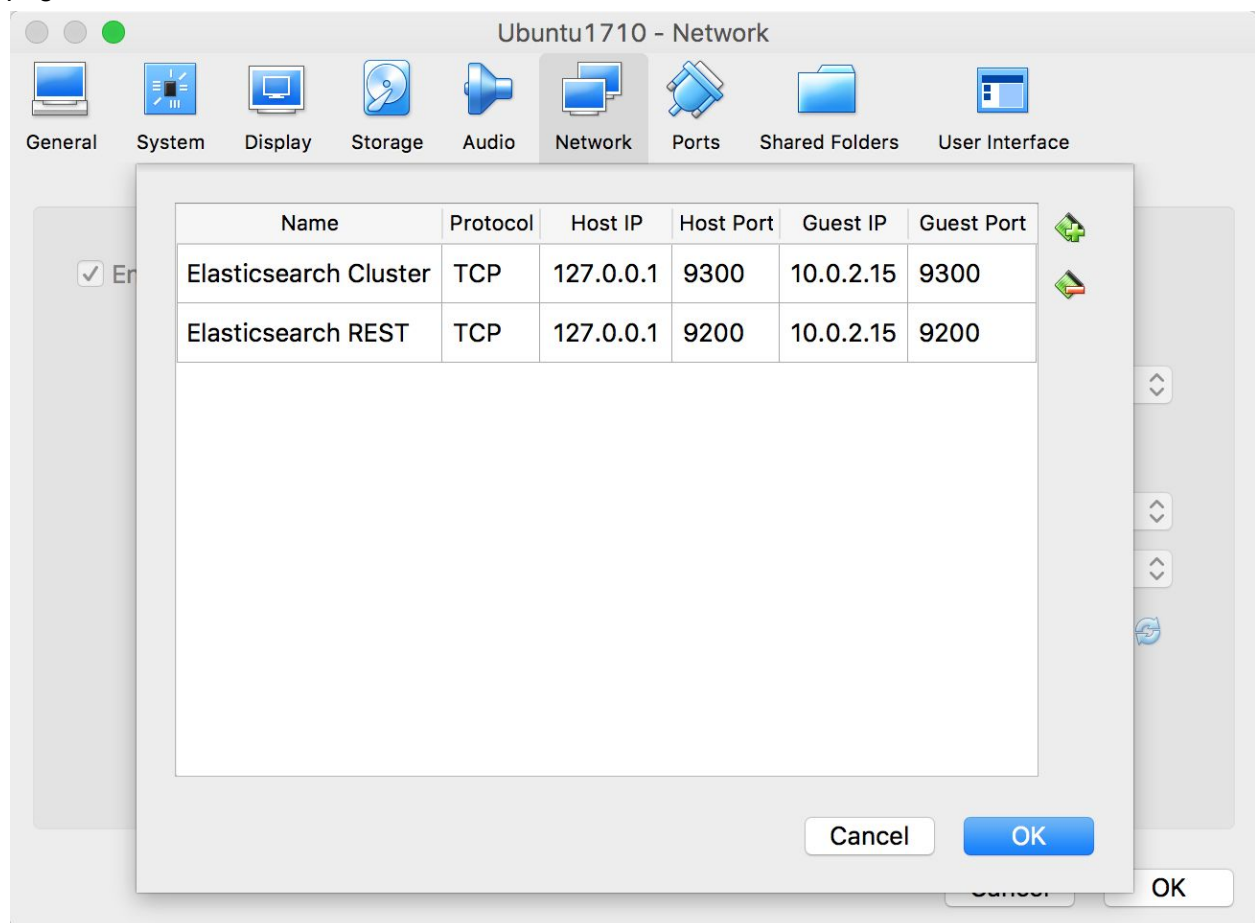
“To pass the maximum map count check, you must configure `vm.max_map_count` via `sysctl` to be at least 262144”

`sudo vim /etc/sysctl.conf`

Add this line:

`vm.max_map_count = 262144`

With all of that completed, I installed `net-tools` to get the IP address of the Ubuntu virtual machine, and then I edited the VirtualBox machine settings to enable port-forwarding. Port 9200 needs to be exposed for external access to the Elasticsearch REST API. Port 9300 needs to be exposed so that the McKoon Spider can connect to the Elasticsearch cluster to index web pages.



Now Elasticsearch will start and be accessible outside the virtual machine. The cluster health may show as “yellow” instead of “green” due to a lack of redundancy because there is only one node in the cluster.

```
jay@mckoonubuntu1710:~$ $ELASTICSEARCH_HOME/bin/elasticsearch
[2018-02-01T19:52:01,316][INFO ][o.e.n.Node               ] [] initializing ...
[2018-02-01T19:52:01,501][INFO ][o.e.e.NodeEnvironment   ] [59xZasY] using [1] data paths, mounts [/ (/dev/sda1)], net usable_space [12.8gb], net total_space [19.5gb], types [ext4]
[2018-02-01T19:52:01,501][INFO ][o.e.e.NodeEnvironment   ] [59xZasY] heap size [1015.6mb], compressed ordinary object pointers [true]
[2018-02-01T19:52:01,549][INFO ][o.e.n.Node               ] node name [59xZasY] derived from node ID [59xZasYQRLqMZu7vcCaa2Q]; set [node.name] to override
[2018-02-01T19:52:01,554][INFO ][o.e.n.Node               ] version[6.1.2], pid[1445], build[5b1fea5/2018-01-10T02:35:59.208Z], OS[Linux/4.13.0-32-generic/amd64], JVM[Oracle Corporation/OpenJDK 64-Bit Server VM/1.8.0_151/25.151-b12]
[2018-02-01T19:52:01,554][INFO ][o.e.n.Node               ] JVM arguments [-Xms1g, -Xmx1g, -XX:+UseConcMarkSweepGC, -XX:CMSInitiatingOccupancyFraction=75, -XX:+UseCMSInitiatingOccupancyOnly, -XX:+AlwaysPreTouch, -Xss1m, -Djava.awt.headless=true, -Dfile.encoding=UTF-8, -Djna.nosys=true, -XX:-OmitStackTraceInFastThrow, -Dio.netty.noUnsafe=true, -Dio.netty.noKeySetOptimization=true, -Dio.netty.recycler.maxCapacityPerThread=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.disable.jmx=true, -XX:+HeapDumpOnOutOfMemoryError, -Des.path.home=/usr/local/elasticsearch-6.1.2, -Des.path.conf=/usr/local/elasticsearch-6.1.2/config]
[2018-02-01T19:52:03,165][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [aggs-matrix-stats]
[2018-02-01T19:52:03,166][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [analysis-common]
[2018-02-01T19:52:03,166][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [ingest-common]
[2018-02-01T19:52:03,166][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [lang-expression]
[2018-02-01T19:52:03,167][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [lang-mustache]
[2018-02-01T19:52:03,167][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [lang-painless]
[2018-02-01T19:52:03,167][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [mapper-extras]
[2018-02-01T19:52:03,167][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [parent-join]
[2018-02-01T19:52:03,168][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [percolator]
[2018-02-01T19:52:03,168][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [reindex]
[2018-02-01T19:52:03,171][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [repository-url]
[2018-02-01T19:52:03,172][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [transport-netty4]
[2018-02-01T19:52:03,172][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [tribe]
[2018-02-01T19:52:03,172][INFO ][o.e.p.PluginsService     ] [59xZasY] no plugins loaded
[2018-02-01T19:52:06,175][INFO ][o.e.d.DiscoveryModule    ] [59xZasY] using discovery type [zen]
[2018-02-01T19:52:07,142][INFO ][o.e.n.Node               ] initialized
[2018-02-01T19:52:07,144][INFO ][o.e.n.Node               ] [59xZasY] starting ...
[2018-02-01T19:52:07,412][INFO ][o.e.t.TransportService   ] [59xZasY] publish_address {10.0.2.15:9300}, bound_addresses {[::]:9300}
[2018-02-01T19:52:07,431][INFO ][o.e.b.BootstrapChecks    ] [59xZasY] bound or publishing to a non-loopback address, enforcing bootstrap checks
[2018-02-01T19:52:10,555][INFO ][o.e.c.s.MasterService    ] [59xZasY] zen-disco-elected-as-master ([0] nodes joined), reason: new_master {59xZasY}[59xZasYQRLqMZu7vcCaa2Q]{sht-cIVmSF6kgF3gBQfpuA}{10.0.2.15}{10.0.2.15:9300}
PerThread=0, -Dlog4j.shutdownHookEnabled=false, -Dlog4j2.disable.jmx=true, -XX:+HeapDumpOnOutOfMemoryError, -Des.path.home=/usr/local/elasticsearch-6.1.2, -Des.path.conf=/usr/local/elasticsearch-6.1.2/config]
[2018-02-01T19:52:03,165][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [aggs-matrix-stats]
[2018-02-01T19:52:03,166][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [analysis-common]
[2018-02-01T19:52:03,166][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [ingest-common]
[2018-02-01T19:52:03,166][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [lang-expression]
[2018-02-01T19:52:03,167][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [lang-mustache]
[2018-02-01T19:52:03,167][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [lang-painless]
[2018-02-01T19:52:03,167][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [mapper-extras]
[2018-02-01T19:52:03,167][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [parent-join]
[2018-02-01T19:52:03,168][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [percolator]
[2018-02-01T19:52:03,168][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [reindex]
[2018-02-01T19:52:03,171][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [repository-url]
[2018-02-01T19:52:03,172][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [transport-netty4]
[2018-02-01T19:52:03,172][INFO ][o.e.p.PluginsService     ] [59xZasY] loaded module [tribe]
[2018-02-01T19:52:03,172][INFO ][o.e.p.PluginsService     ] [59xZasY] no plugins loaded
[2018-02-01T19:52:06,175][INFO ][o.e.d.DiscoveryModule    ] [59xZasY] using discovery type [zen]
[2018-02-01T19:52:07,142][INFO ][o.e.n.Node               ] initialized
[2018-02-01T19:52:07,144][INFO ][o.e.n.Node               ] [59xZasY] starting ...
[2018-02-01T19:52:07,412][INFO ][o.e.t.TransportService   ] [59xZasY] publish_address {10.0.2.15:9300}, bound_addresses {[::]:9300}
[2018-02-01T19:52:07,431][INFO ][o.e.b.BootstrapChecks    ] [59xZasY] bound or publishing to a non-loopback address, enforcing bootstrap checks
[2018-02-01T19:52:10,555][INFO ][o.e.c.s.MasterService    ] [59xZasY] zen-disco-elected-as-master ([0] nodes joined), reason: new_master {59xZasY}[59xZasYQRLqMZu7vcCaa2Q]{sht-cIVmSF6kgF3gBQfpuA}{10.0.2.15}{10.0.2.15:9300}
[2018-02-01T19:52:10,568][INFO ][o.e.c.s.ClusterApplierService] [59xZasY] new_master {59xZasY}[59xZasYQRLqMZu7vcCaa2Q]{sht-cIVmSF6kgF3gBQfpuA}{10.0.2.15}{10.0.2.15:9300}, reason: apply cluster state (from master [master {59xZasY}[59xZasYQRLqMZu7vcCaa2Q]{sht-cIVmSF6kgF3gBQfpuA}{10.0.2.15}{10.0.2.15:9300} committed version [1] source [zen-disco-elected-as-master ([0] nodes joined)])
[2018-02-01T19:52:10,648][INFO ][o.e.h.n.Netty4HttpServerTransport] [59xZasY] publish_address {10.0.2.15:9200}, bound_addresses {[::]:9200}
[2018-02-01T19:52:10,650][INFO ][o.e.n.Node               ] [59xZasY] started
[2018-02-01T19:52:10,940][INFO ][o.e.g.GatewayService     ] [59xZasY] recovered [1] indices into cluster_state
[2018-02-01T19:52:12,015][INFO ][o.e.c.r.a.AllocationService] [59xZasY] cluster health status changed from [RED] to [YELLOW] (reason: [shards started [[web][1]] ...]).
```

Experiment

McKoon Spider reads its configuration with Typesafe Config. The default configuration is located in reference.conf within the source code. This configuration sets the seed URL for the

run to “<https://www.cc.gatech.edu/>”. The configuration also specifies that all URLs in the frontier must have a host name that contains “cc.gatech.edu”.

McKoon Spider uses Logback to write logs, and Dropwizard Metrics for runtime statistics. By default, the project is set up to dump all metrics to a metrics.log every thirty seconds, and to write other log messages to spider.log at the debug level in real time. Gauges are configured on the size of the frontier (to-be-visited URLs) and the size of the visited URLs, so these sizes are logged regularly. Also every non-private method in the application records statistics for its runtimes and number of times invoked.

To begin the experiment, first the Ubuntu virtual machine running Elasticsearch was booted. Then the Elasticsearch server was started with the following command:

```
$ELASTICSEARCH_HOME/bin/elasticsearch
```

On the host machine, McKoon Spider was compiled and packaged as a self-contained jar file by running the following command:

```
./gradlew clean shadowJar
```

On January 28, 2018 at 6:22:17 PM, McKoon Spider execution was started by running the following command on the host machine:

```
java -jar ./build/libs/spider.jar
```

The application was allowed to run without interruption or any intervention until I terminated the program on January 29, 2018 at 5:27:47 AM. This resulted in a total runtime of about 11 hours. This run generated a 983 MB spider.log, and a 15 MB metrics.log.

Metrics

Final Collection Sizes

Frontier (discovered, unvisited URLs)	23,591
Visited URLs	17,355

Key Method Statistics

Method	Call Count	Min Runtime	Max Runtime	Mean Runtime
ElasticsearchIndexService.index	14637	2007 ms	2055 ms	2020 ms
JsoupVisitService.visit	17355	34 ms	18107 ms	604 ms
BreadthFirstFrontierService.getNextUrl	17355	0.068 ms	0.3552 ms	0.11558 ms
BreadthFirstFrontierService.addUrl	1220649	0.057 ms	0.6577 ms	0.0975 ms

The experiment ran for 665 minutes and visited 17,355 pages. This means that McKoon Spider visited 26 pages per minute. The ratio of visited pages (17,355) to discovered not visited pages (23,591) is 0.7357.

Reviewing the mean runtime of methods from the experiment, it becomes apparent that the bottleneck is indexing web pages with Elasticsearch, as the mean runtime for this is just over two seconds. Improving this task could be the target of future work.

Source Code

I have open-sourced the McKoon Spider source code and licensed it under the MIT license. The code is available on GitHub.

The source code for the McKoon Spider console application is available on GitHub at <https://github.com/mckoon/spider>.

The source code for the spider-client front-end for searching Elasticsearch is available on GitHub at <https://github.com/mckoon/spider-client>.

Future Work

As the metrics show, the biggest bottleneck is indexing the web pages with Elasticsearch. One possible improvement could be to add more server nodes to the Elasticsearch cluster and add multithreading to the McKoon Spider. This would allow multiple pages to be indexed simultaneously, while the crawler continues to visit additional pages.

One current limitation of the McKoon Spider is that it does not support stopping and resuming. Currently the frontier and visited collections are not persisted to disk, so when the McKoon Spider console application is stopped that data is lost. To ensure that the search index is in the same state as the console application, on start-up the console application deletes the search

index in Elasticsearch and starts fresh. Future work could be to persist the frontier queue and the visited set to disk to allow for the application to stop and resume.

Another issue with the McKoon Spider is that it treats differently formatted URLs to the same HTML page as different pages. For example, McKoon Spider will treat "<https://www.cc.gatech.edu/women-computing>", "<https://www.cc.gatech.edu/women-computing/>", and "<https://www.cc.gatech.edu/women-computing#main>" all as different pages, visiting and indexing each. One simple improvement would be to normalize the URLs via a call to the Java `URI.normalize()` method. This would solve logically duplicate URLs due to slashes and dots in the path, but not query parameters or fragments after the hash. Query parameters and fragments after the hash (e.g. `#main`) are problematic, because they can be the same HTML page, or completely different content as with many single page applications. Additional work could be done in this area to determine the best way to handle these URLs.

McKoon Spider also only handles HTML content, so no other type of files will work. Within the HTML files that are crawled, only links in the href attributes of anchor tags will be followed.