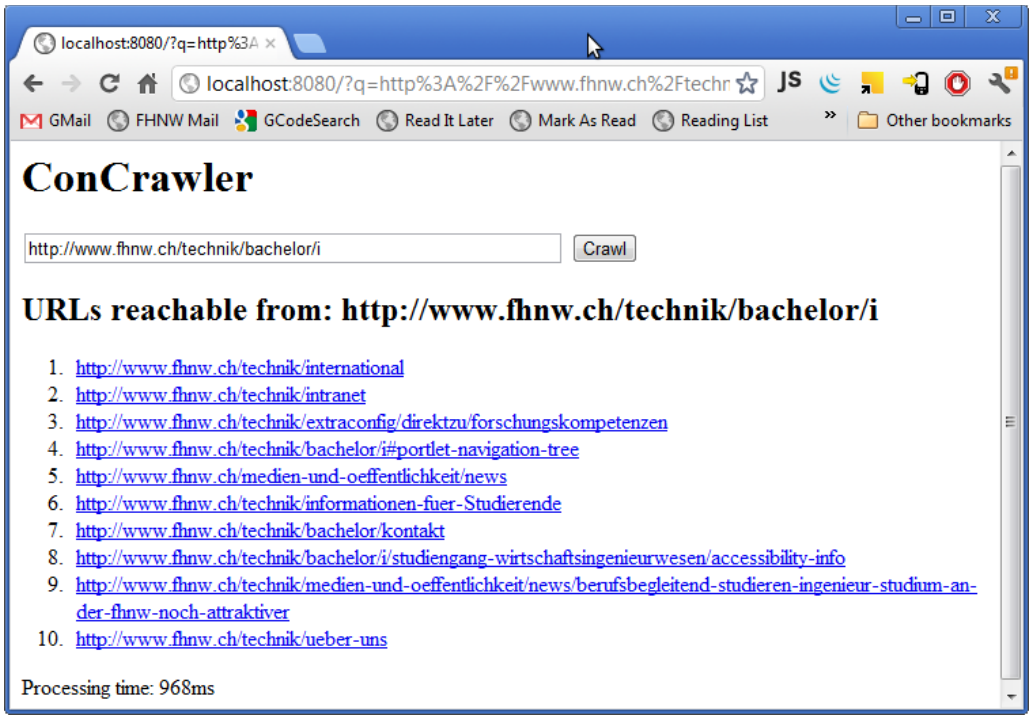
ConCrawler – Multithreaded Webcrawler

Gegeben ist ein minimaler Webserver, der Webcrawler-Funktionalität bereitstellt. Wenn Sie die Applikation (*concrawler.ConCrawler*) starten, ist die Seite über HTTP auf Port 8080 erreichbar. Die Idee ist, dass Sie im Textfeld eine URL eingeben und als Resultat eine Liste der über diese URL erreichbaren Seiten bekommen.



Der Crawler wird dabei nicht nur den Inhalt der Start URL analysieren, sondern folgt rekursiv auch allen Links, die darin zu finden sind. Da das Internet doch ziemlich stark vernetzt ist, muss eine vernünftige Abbruchbedingung eingebaut werden. Aktuell ist die Anzahl der Links im Resultat auf 20 begrenzt[1].

Es folgt eine Skizze des verwendeten Algorithmus (Breitensuche).

1. Die Start URL in eine Queue einfügen
2. URL aus der Queue entfernen
3. Inhalt der URL herunterladen und nach Links durchsuchen
4. Besuchte URL als Resultat merken
5. Alle gefundenen Links in die Queue einfügen
6. GOTO 2.

Aufgaben:

1. Der Webserver kann momentan nur eine Anfrage gleichzeitig bewirten. Bauen Sie das Connection Handling so um, dass mehrere Anfragen gleichzeitig verarbeitet werden können.
   Verschaffen Sie sich einen Überblick über die Factory-Methoden der Klasse j.u.c.Executors und wählen Sie eine geeignete aus.
2. (Optional) Setzen Sie auf den Worker-Threads mittels der ThreadFactory eigene Namen (z.B. ConCrawler-ConnectionHandler-1..n).
3. Die Crawler Funktionalität ist in der Klasse *SeqCrawler* umgesetzt. Ihre Aufgabe ist die Implementierung eines parallelen Crawlers, der mit mehreren Threads den Links folgt und diese verarbeitet. Vergleichen Sie den Zeitgewinn zwischen dem SeqCrawler und Ihrer Lösung.

[1] Im Grunde müsste ein Crawler sich zuerst die robot.txt Datei herunterladen und sich dann entsprechend den Regeln verhalten: http://www.robotstxt.org/