## Demonstrieren Sie, dass sich Ihr Programm automatisiert sauber bauen lässt. Beschreiben Sie wie's geht.

Anbei befindet sich ein Makefile, dass unseren Crawler automatisch baut. Im Prinzip lässt sich das Programm mit einem Befehl kompilieren**: g++ -g -Wall -Iinclude/ -pthread -o $@ src/main.o src/ThreadPool.o src/HTTP.o src/HTMLParser.o src/Crawler.o -lboost_thread-mt -lboost_regex-mt -lcurl –lhtmlcxx** Hier gibt's nicht viel zu erklären…

## Optionale Funktionen die wir eingebaut haben

- Ihr Programm soll eine Seite nur einmal untersuchen.
- Ihr Programm kann mit Redirects umgehen?
- Erweitern Sie Ihr Programm so, dass innerhalb einer domain gesucht wird.
- Unser Programm kann mit SSL umgehen
- Linux/Max und Windows Kompatibilität (auf allen Systemen kompiliert) Unser Team entwickelte dieses Programm auf allen drei Plattformen

## Führen Sie an, welche Bibliotheken Sie verwenden und unter welcher Lizenz diese stehen.

- **HTMLCXX → LGPL**
- **BOOST → Boost Software License - Version 1.0:**
  Boost Software License - Version 1.0 - August 17th, 2003

  Permission is hereby granted, free of charge, to any person or organization
  obtaining a copy of the software and accompanying documentation covered by
  this license (the "Software") to use, reproduce, display, distribute,
  execute, and transmit the Software, and to prepare derivative works of the
  Software, and to permit third-parties to whom the Software is furnished to
  do so, all subject to the following:

  The copyright notices in the Software and this entire statement, including
  the above license grant, this restriction and the following disclaimer,
  must be included in all copies of the Software, in whole or in part, and
  all derivative works of the Software, unless such copies or derivative
  works are solely in the form of machine-executable object code generated by
  a source language processor.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR ANYONE DISTRIBUTING THE SOFTWARE BE LIABLE FOR ANY DAMAGES OR OTHER LIABILITY, WHETHER IN CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

- **Curl → MIT/X derivate (curl license ):**
  COPYRIGHT AND PERMISSION NOTICE

  Copyright (c) 1996 - 2010, Daniel Stenberg, <daniel@haxx.se>.

  All rights reserved.

  Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

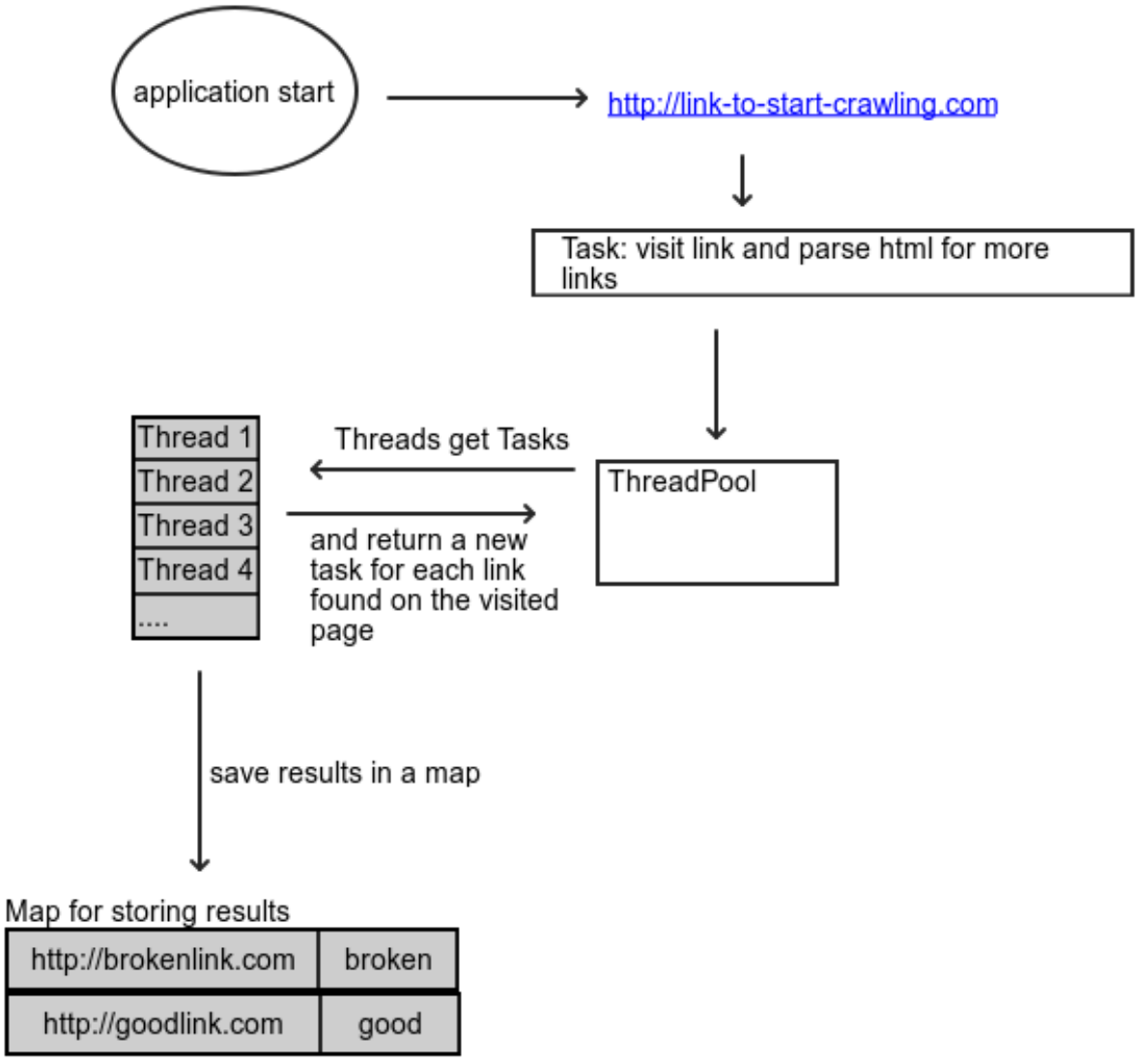
  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

  Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

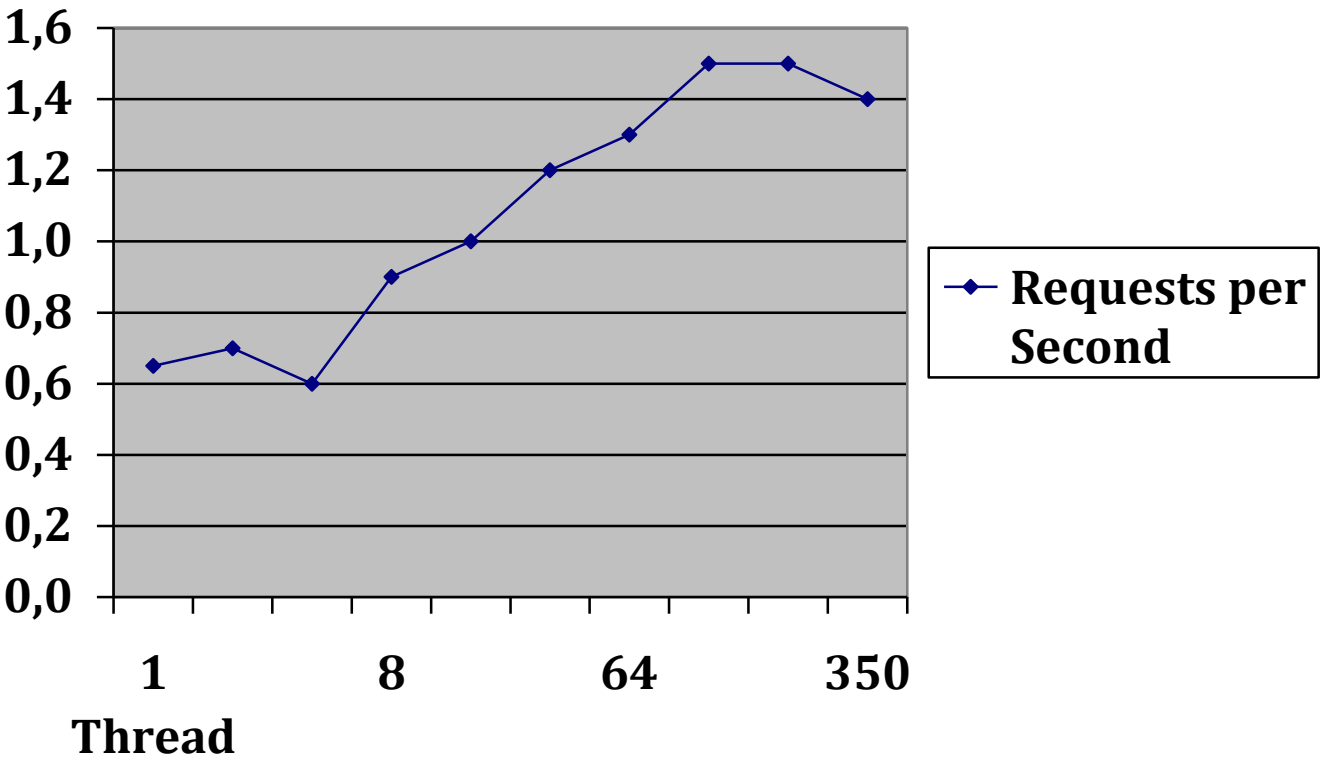
## Beschreiben Sie, welche Teile ihres Programms als Thread ablaufen und wie diese zusammenarbeiten. Diagramm!

Wir haben einen Threadpool, der Tasks entgegennimmt. Sobald ein Thread einen Task abgearbeitet hat, holt er sich einen neuen aus einer Taskqueue. Ein Task besteht aus folgenden Tätigkeiten:

1. Aufrufen einer Url und herausfinden ob diese erreichbar ist.
2. Die Url der besuchten Seite wird zusammen mit der Information ob Broken oder nicht in eine Map gespeichert
3. Wenn erreichbar wird der HTML Quellcode der Seite nach Links durchsucht
4. Für jeden dieser Links wird ein neuer Task erstellt und in die Task Queue gesteckt.

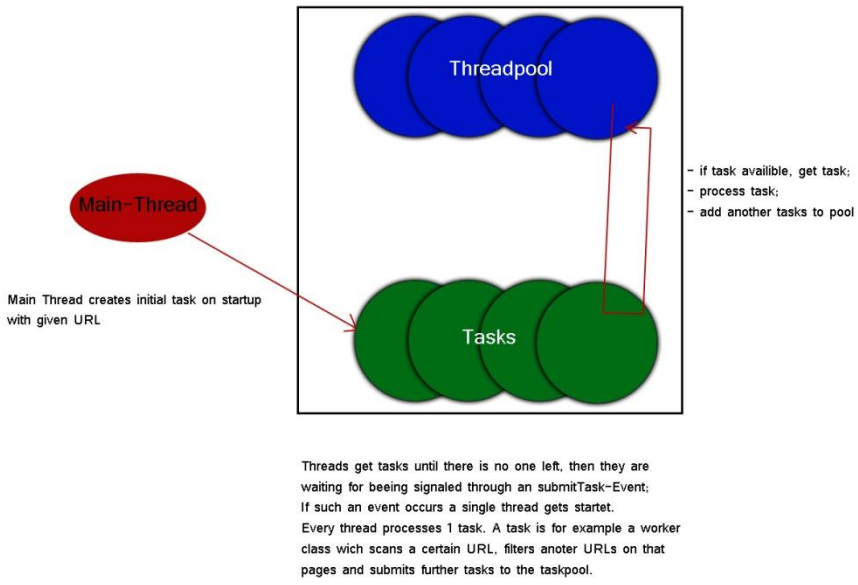


## Untersuchen Sie, ob Ihr Programm durch Multithreading schneller arbeitet (d.h. mehr Webseiten in gegebener Zeit durchsuchen kann). Grafik!




Die Rate steigt mit der Zahl der Threads stark an. Dies liegt daran, dass ein Request gerade bei meinem DSL 2000 sehr lange dauert. Diese Arbeit lässt sich also sehr gut synchronisieren. Ab einer gewissen Zahl von Threads (in meinem Fall ca. ab 150 Threads) scheint meine Internetverbindung ausgelastet zu sein und es wird eher langsamer wenn man dann noch mehr Threads benutzt.

Architecture:




Logfile:

Can be found after crawling in executable folder.

Found broken URL: http://de.wikipedia.org/wiki/web_2.0

ParentUrl: http://webreload.de

Found broken URL: http://jigsaw.w3.org/css-validator/check/referer

ParentUrl: http://webreload.de/angebote.html

Found broken URL: http://www.bistro-b20.de

ParentUrl: http://webreload.de/referenzen.html

Found broken URL: http://www.bullrobin.de

ParentUrl: http://webreload.de/referenzen.html

Found broken URL:
https://www.cleverbridge.com/342/cookie?affiliate=3676&product=29945&redirectto=http%3a%2f%2fwww.malwarebytes.org%2f

ParentUrl: http://webreload.de/pc_service.html

Searched URL "http://webreload.de" for broken links...

scanned 22 links

Found

5 broken Links (22.7273%)

17 good links (77.2727%)

scanning successfully...

domi

hubsi

robz say: THX!