

## Задача 2 — Поисковый робот

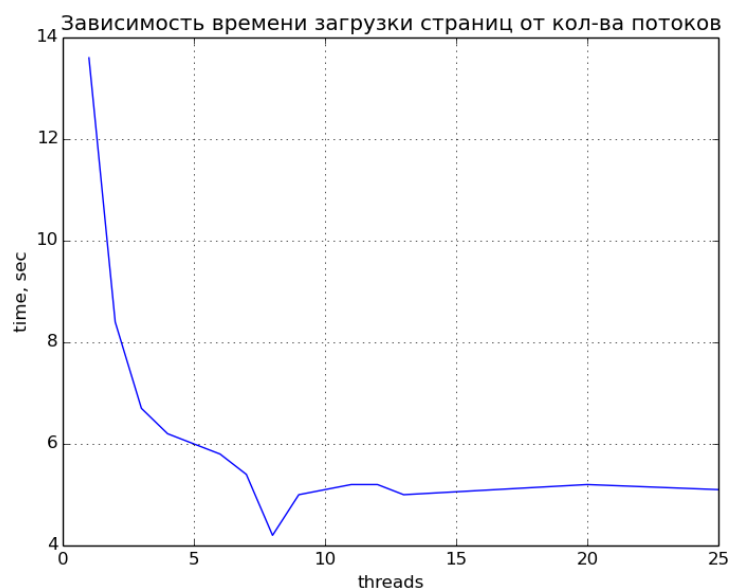
Для реализации поискового робота необходимо создать потокобезопасную очередь (`concurrent_queue`) для реализации очереди задач и потокобезопасное множество (`concurrent_set`) для сохранения и проверки посещенных страниц, оба представлены в файле «`concurrent_structs.h`». Также для удобной работы с потоками был создан пул потоков, представленный в файле «`thread_pool.h`». Метод `loop` выполняется в каждом потоке пула пока в очереди есть задачи (если задач нет, то поток ждет их появления на условной переменной) и не взведен флаг `stop`. Программа работает следующим образом: создается пул с необходимым количеством потоков, в него добавляется первая задача (загрузка начальной страницы). Поток загружает страницу, парсит ее, находит вложенные ссылки, и каждую из них добавляет в пул как новую задачу. Каждый поток возвращает результат работы в виде объекта `future` и кладет их в очередь выполненных задач. Главный поток дожидается получения результата от всех объектов `future`, выводит результат выполнения каждого из потоков (страница успешно загружена и сохранена или сообщение об ошибке), выставляет флаг `stop` у пула потоков и вызывает метод `join` для каждого из потоков и завершает выполнение.

Результат работы программы: `time ./crawler http://www.yandex.ru P D ./pages/ N 10`

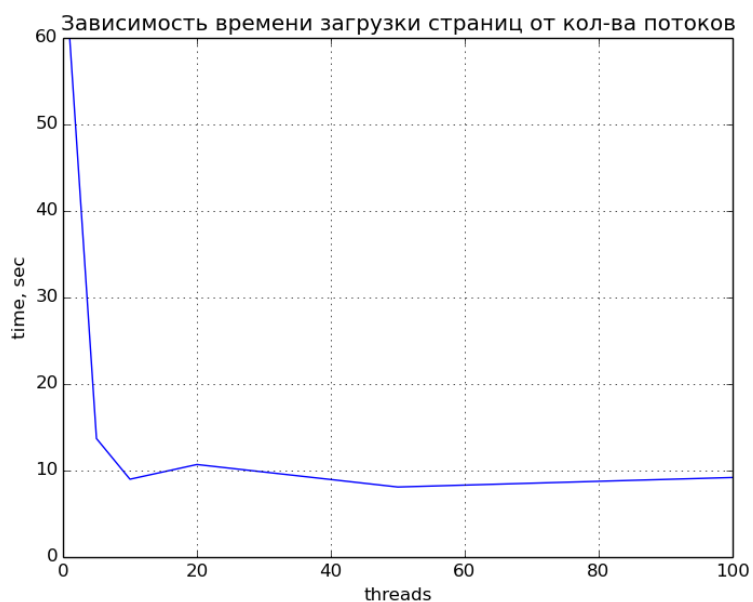
P = 10 страниц	
N, кол-во потоков	время, сек
1	2.1
2	1.2
3	1.0
4	1.0
5	0.8
6	0.8
7	0.9
8	0.6
9	0.6
10	0.6



P = 25 страниц	
N, кол-во потоков	время, сек
1	13.6
2	8.4
3	6.7
4	6.2
5	6.0
6	5.8
7	5.4
8	4.2
9	5.0
10	5.1
11	5.2
12	5.2
13	5.0
20	5.2
25	5.1



P = 100 страниц	
N, кол-во потоков	время, сек
1	60.0
5	13.7
10	9.0
20	10.7
50	8.1
100	9.2



Можно заметить, что после определенного кол-ва потоков в пуле суммарное время выполнения всех задач не уменьшается, а остается постоянной или даже возрастает. Скорее всего это связано с тем, что при возрастании кол-ва потоков возрастает время на синхронизацию, а также задачи (страницы на загрузку) не успевают генерироваться с нужной скоростью, и большинство потоков просто находятся в состоянии ожидания, не выполняя полезной работы.

**NB:** работает корректно при компиляции gcc 4.9 (в 4.8 не полностью реализованы regex).