**Weekly Assignment-2**

Saisrikar Paruchuri

44751575

S4475157

INFS7410

**Weekly Assignment-2**

**Major challenges of web crawler**

**Phases are performed sequentially:** The web crawler procedure is done sequentially and to combine the steps it takes approximately one hour by leaving the remaining 3 hours of the 4 hours to perform the actual crawl which as 25% loss in utilization of the data.

**Reduced parallelism towards the end of the crawl:** Depends on the importance of the domains we try to schedule approximately more links from this domains per web crawling when compare to unimportant domains. The top web crawler google uses page rank algorithm and it gives importance to top websites and provides more links then other websites. Web crawler will spend more time on crawling the data from important websites and quickly crawl that is spend less time for crawling on unimportant websites. So, by the end of the web crawling cycle we are facing with long tail problems. Throughthe statistics we found that we are losing approximately 10% of our utilization by the long tail problem.

**Removing invalid links from the web crawler:** After the web crawling it must show the valid links to the user by removing invalid links from the crawler and being polite as much as possible. We check the robots.txt file in all domain websites that are scheduled to be crawled before we start crawling and remove any banned links from the schedule. The main problem with the process of removing banned links is the scheduler has limited number of slots. So, the scheduler will end up with the links that are already banned and it will take some time for this during the crawl process because sometimes it can be done offline.

**Addressing the corresponding issues**

After rectifying the above problems for the web crawler. Then the web crawler will be simple which allow us to use the machine more aggressively and achieve even faster crawl results. Below are the ways that we can improve our performance of the web crawler

**Scheduler will cluster invalid links and remove them offline:** The improvement is to move the removal of invalid links to the scheduler. Mainly for most of the domains robots.txt file is updated infrequently. If the robots.txt is updated the changes rarely affect us. We see other crawlers that are banned and very occasionally we see admins trying to help the web crawlers by stopping them from getting useless content.

This is necessary to scan the robots.txt files periodically but it is not important to scan them for every four hours. We can rectify this problem by scanning the robots.txt file as part of scheduling that is if you have changed the robots.txt file it will not reflect changes in the crawler for up to 24 hours. However, the processing time on large important domains will be faster. To compensate this delay, we will add the ability to manually force a quick re-check if there are complaints which are modified by the crawler within 30 minutes of the complaint.

**Changing to service oriented architecture:** Existing crawlers are controlled by wrapper that does book keeping and re starts the crawler, but still it is a very simple application means it is complex inside but externally very simple. We need a web crawling service that can be dynamically sized then the crawler service will crawl the web continuously because the service consists of as many or few machines as we needed and it can be dynamically sized and the index increases.

**Long tail problem:** Service oriented architecture helps us to solve our long tail problems. The existing crawler uses 10 independent schedules, each assigned to 2 web crawlers. But our web crawler that is new system uses 72,000 schedules. Where each crawler will have some portion of 2,500 schedules locally and be processing 2000 at any time. When the schedule is completed it will move to process and a new schedule is retrieved in the background. We have 500 buffer spare schedules on the crawler, we can continue crawling while other side we are uploading the result and download a replacement schedule for the completed schedule by this we can continuously saturate crawling parallelism.

**Background upload and download:** While the upload/download is happening in the background the main part of the web crawling process is still running. The background process means it is not necessary to stall the crawl process to retrieve the next part of the schedule.

**Increased simplicity:** In the new web crawler, we have wrapped this heart in a new service-based design which will use smaller and more independent schedules. Mainly the schedules are designed to be simpler. The complexity of the crawler has been reduced by this the CPU cycles are free for other activities. By the above improvements we can approximately improve 15% on page processing and because of simpler design we can easily increase parallelism in processing of web pages by increasing the speed.

The above improvements can significantly increase the speed and efficiency of the web crawler approximately we can increase our crawling speed to 12 billion pages every 6 days and we can only predict the accurate results after implementing the web crawler and by seeing the real-world results.