Performance Analysis Module

The crawler crawls in an average speed of a thousand page in three minutes, while the indexer indexes a thousand pages in four minutes it also depends on the number of rows in the indexer tables in database. As the database is full the indexer requires more time in indexing pages.

We tried to impose links of different more popular and trending fields in the crawler as seeds, so it can crawl thousands of pages in different topics, in order to handle as much search queries as possible.

But to be honest, in testing we were concerned more about the specific fields we enforced in the seeds, and the search engine showed high and acceptable efficiency.

As the number of crawled web pages increases, it makes the result set larger, more precise and relevant. On the other hand, the inverted database (indexer database) has more rows which make the results take longer time.

The search query takes more time in case of large size of index table, but results are more precise.

The bottle neck of the ranking process is calculating popularity term for each web page and it is done after crawling it requires great memory allocation and time complexity related to matrix multiplication as the time and memory complexity is square the number of the crawled pages. While other ranking procedures performed on the result set take less time complexity equivalent to result set size added to the complexity of searching in index table which is typically higher. As analysis of search process bottle necks, we deduced the following:

	Web links search (sec)	Image search (sec)
Average overall search time	12.9437	5.205135677
Query processor portion	0.0807	0.02605365
Ranker portion	9.2991	3.2611706
Network portion	3.5640	1.917911427
Analysis chart