Search Engines for the Web

An Overview

- Arasu et al: *Searching the Web*. ACM Trans. Internet Technology, 1, 2001.
- Brin and Page: *The Anatomy of a Large-Scale Hypertextual Web Search Engine*. 7th Int. WWW conference, 1998

Information Retrieval

- Process data, build index
- Query the index:
  - Find all documents relevant to query
  - Rank documents, show most relevant first

Classic Information Retrieval (IR):

Methods developed for small to medium sized homogeneous collections of text documents.

Examples: Scientific document collections, news collections, libraries.

IR on the Web

Difficulties:

- Documents not local.
- Documents very heterogeneous.
- Documents constantly changing in contents and number.
- Very large document collection (billions of documents, total size measured in Terabytes).
  - Storage and performance are important issues. Distribution and parallelism necessary.
  - Many (e.g. 100.000) relevant documents for most queries. Good ranking methods are essential.

Advantages:

- Extra structure on document collection: links.

Further Challenges of the Web

- Many near-duplicate documents (30%)
- Users heterogeneous and impatient. Advanced search interfaces not viable.
- How to search and index non-text documents.
  - Multimedia contents.
  - Database interfaces.

  This course: only consider text documents.

Basic Tasks of Search Engines

Gather data:

- Web crawling (traversal of the web graph). Repeat: Follow link, store doc, parse doc, extract new links.

Index data:

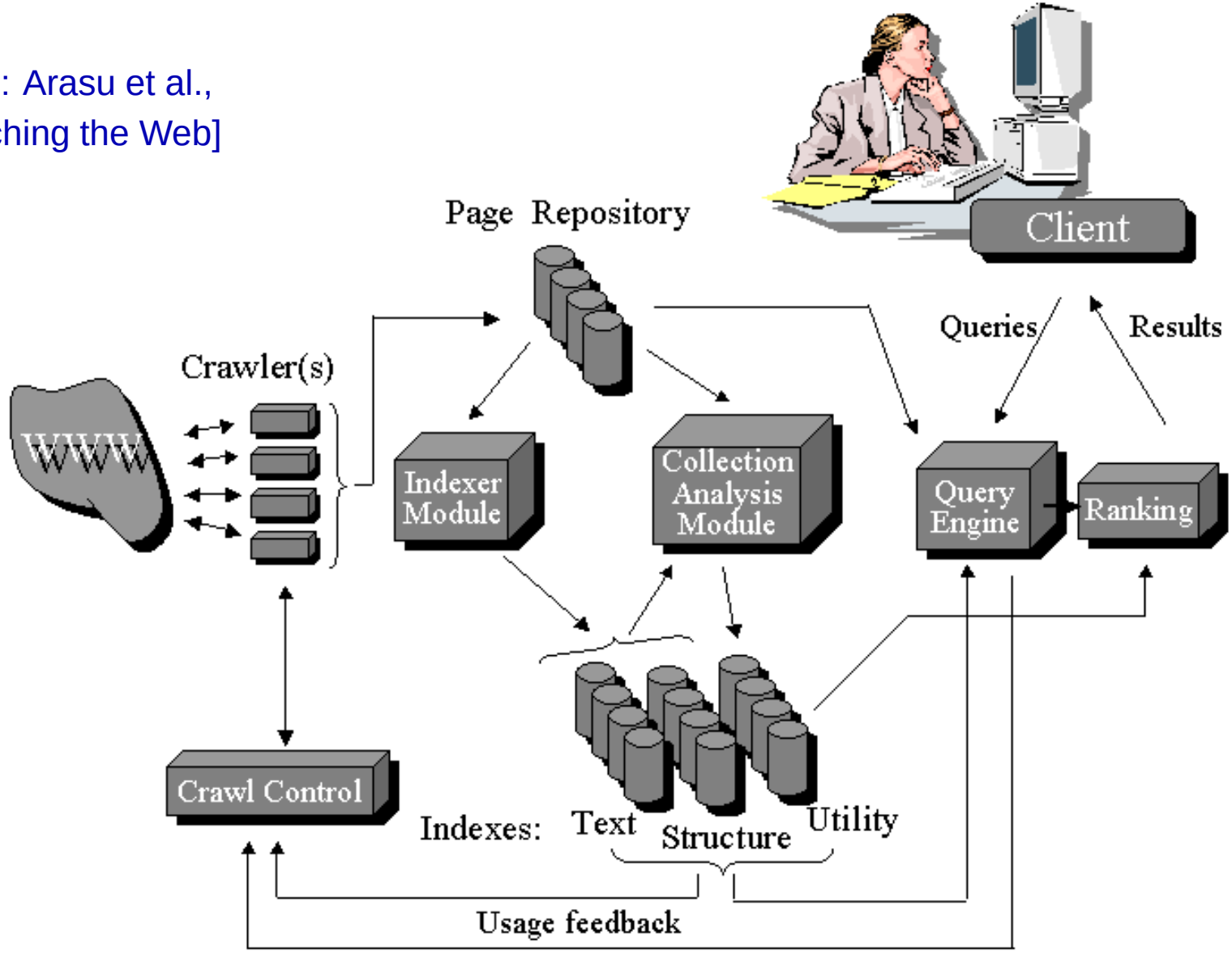
- Store documents.
- Inverted file: for all words, list docs where they occur.
- Lexicon: index over words in inverted file.

Search data:

- Retrieve docs with query words.
- Rank retrieved docs.

General Structure

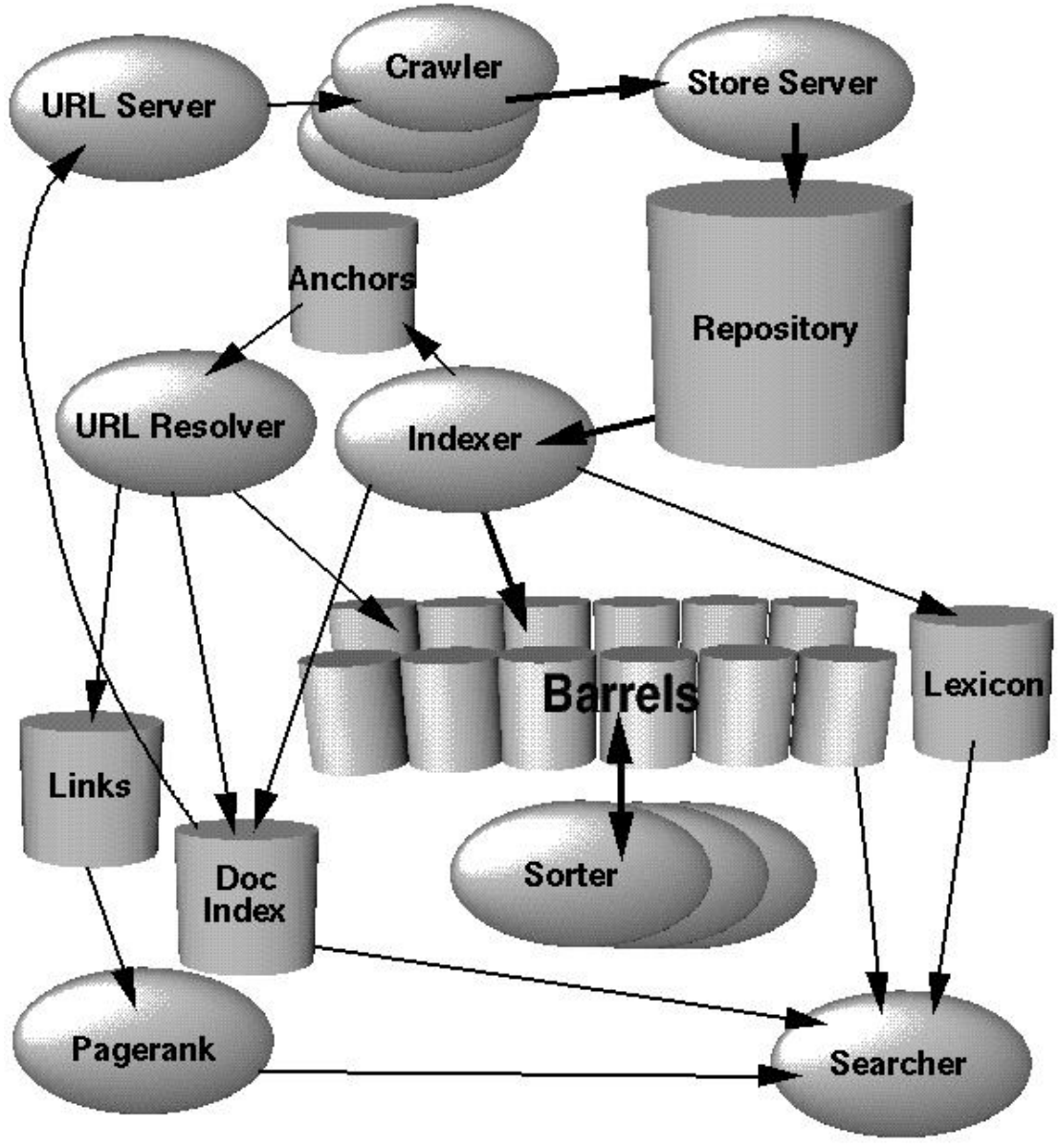
[From: Arasu et al., Searching the Web]



Specific Example

Google:
(1998)

[From:
Brin and Page,
Anatomy of. . .]



(Simple) PageRank

Idea 1: Link to page = recommandation.

Idea 2 : Recommandations by important pages should have larger weight.

Recursive def:

$$r(i) = \sum_{j \in B(i)} r(j)/N(j)$$

$B(i)$ = pages pointing to page i,
$N(j)$ = outdegree of page j.

$$\vec{r} = A\vec{r}$$

A = normalized adjacency matrix for web graph.