Searching the Web
What is this Page Known for?

Luis De Alba
ldealbar@cc.hut.fi

Searching the Web

Arasu, Cho, Garcia-Molina, Paepcke, Raghavan

August, 2001.

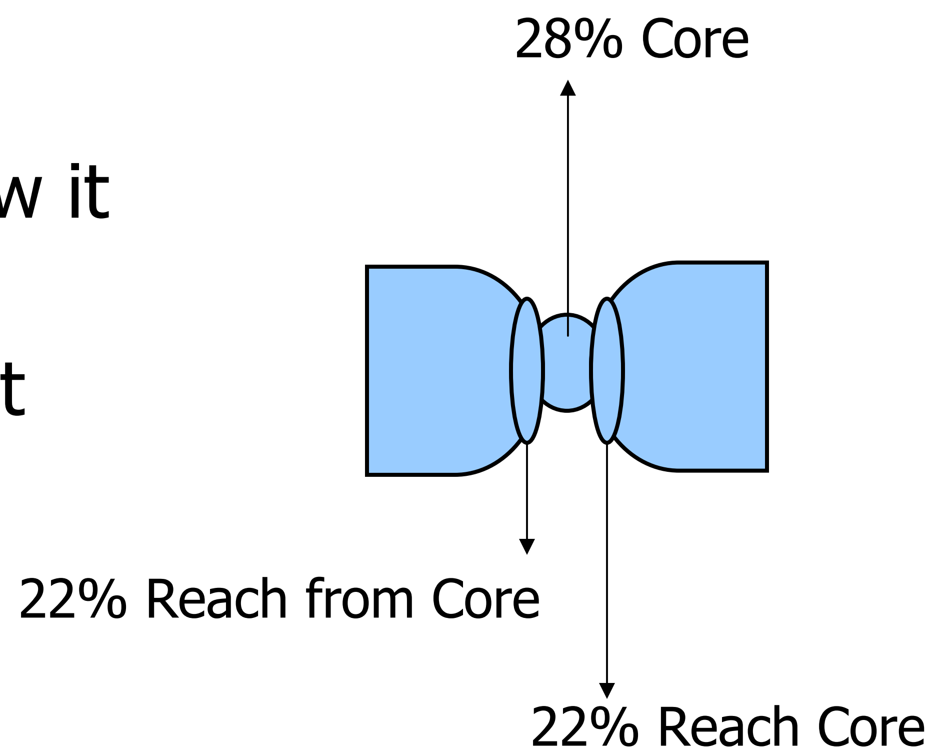
Stanford University

Introduction

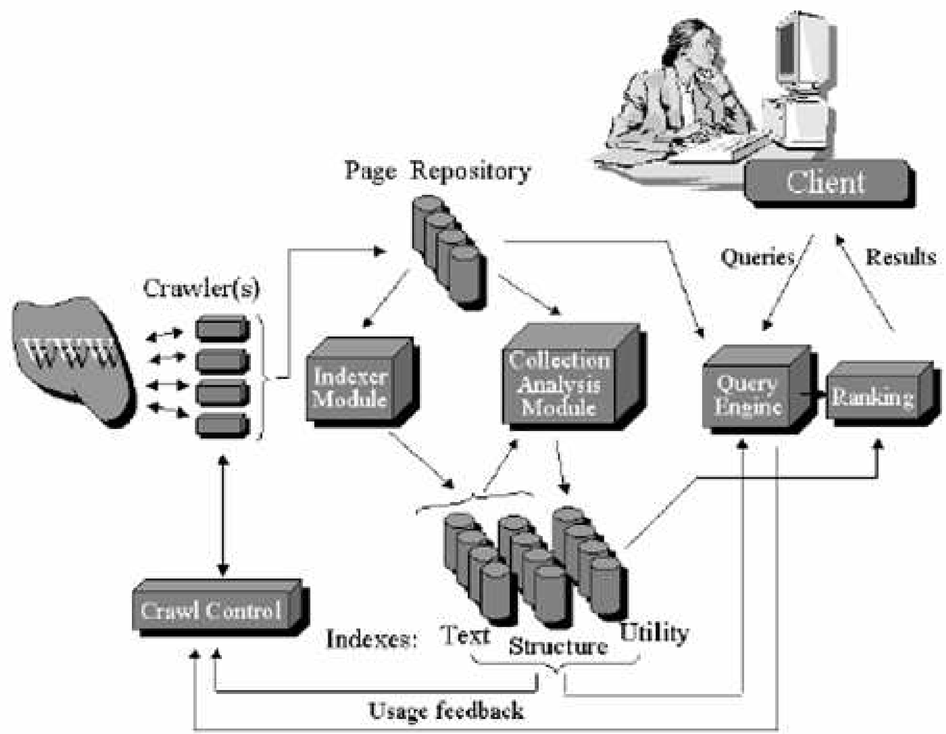
- People browse the Web using entry Points or using a Search Engine (many)
- The Web is Massive, No Coherent, Changes rapidly and it its geographically distributed.
- Over 8 billion pages.
- In .com domain 40% pages expected to change daily.

Introduction

- Studies aim to Web's linkage structure and how it can be modeled.
- Web is somewhat like a "bow tie".

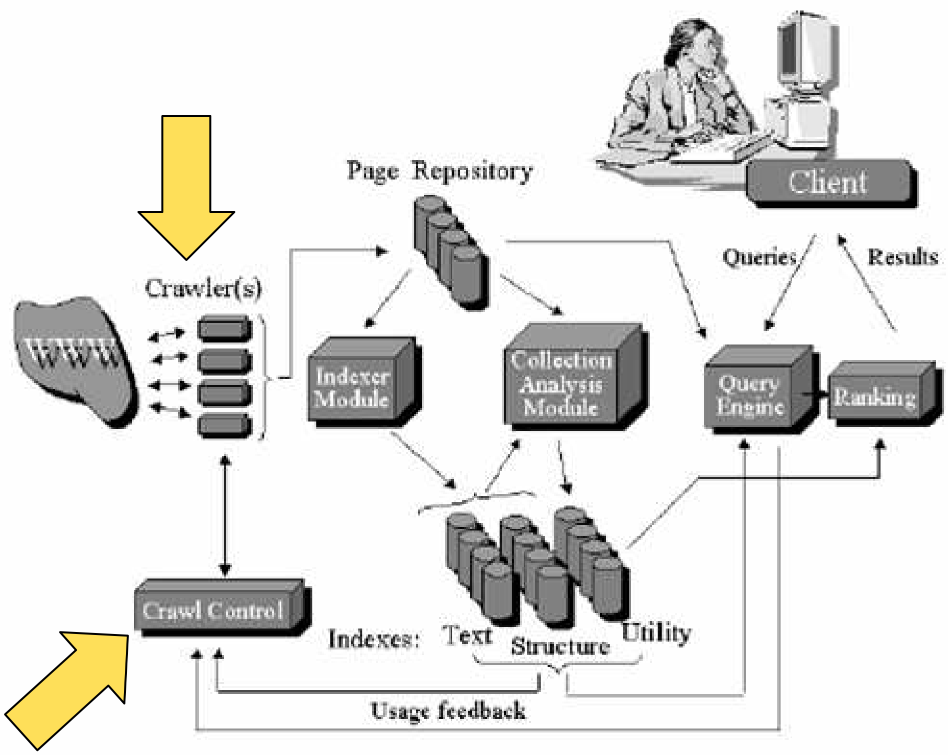


Search Engine



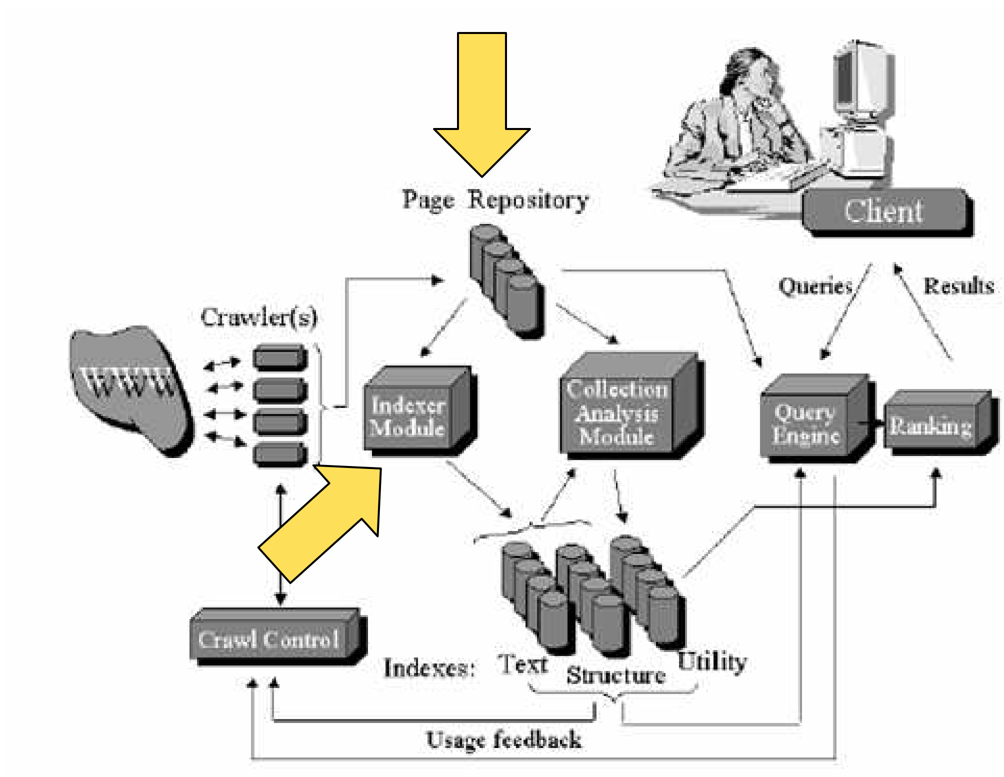
- How a Web search engine is typically put together.

Crawler



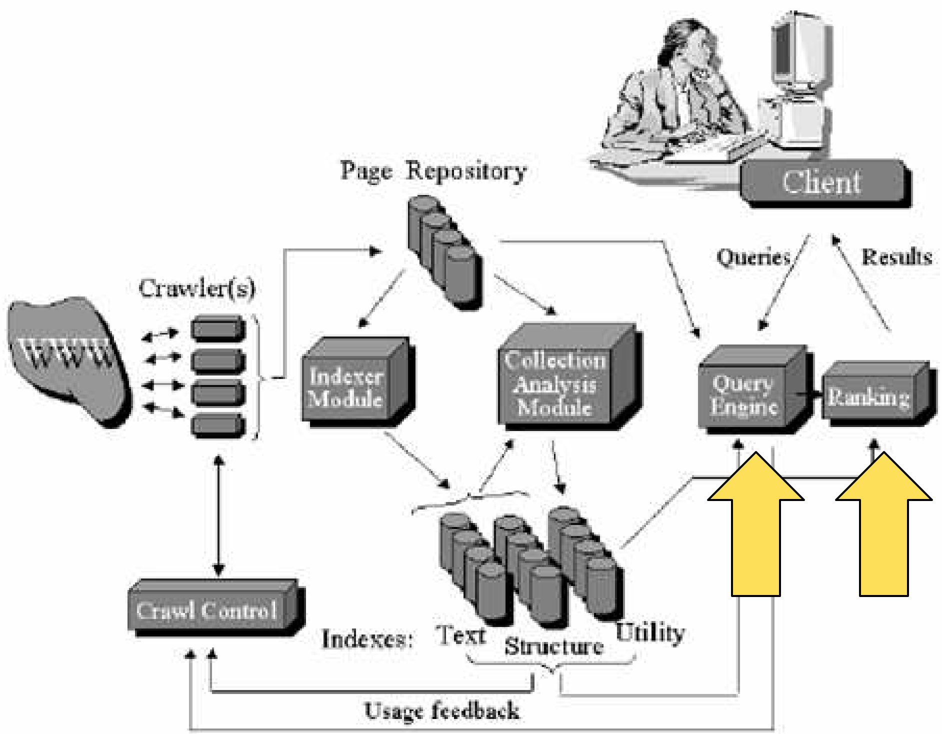
- Crawlers are programs that browse the Web on the search engine's behalf.
- Crawl Control module: to keep crawlers working and in which way.

Indexer & Repository



- Indexer: Extracts words from each page and records URLs.
- Repository: Collection (temporary) of retrieved pages.

Query Engine & Ranking



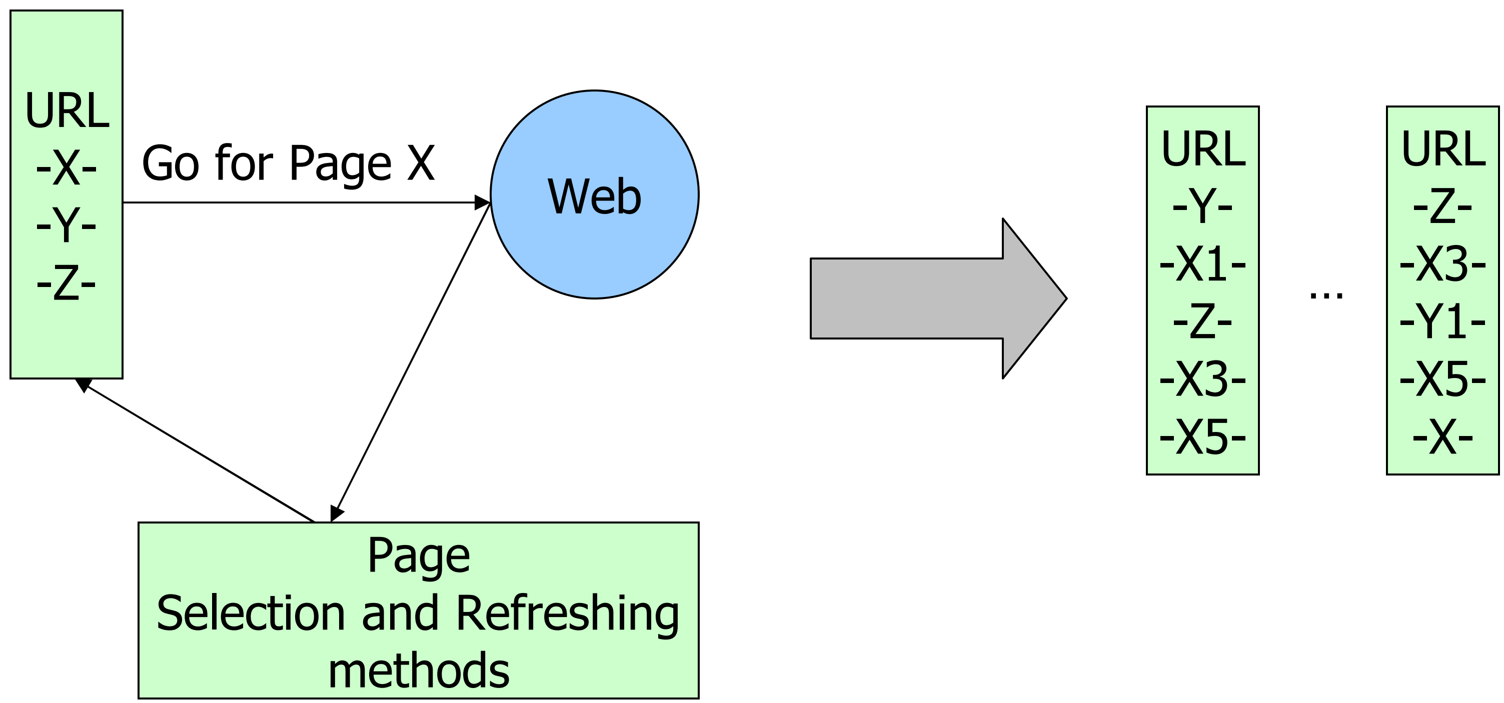
- Query E: Receives and fills Search Request from Users.
- Ranking: Due to Web size results are very large, hence the ranking will sort them.

Modules

- Crawling
- Storage
- Indexing
- Ranking

Crawling

- Start with an initial Set of URL's



Crawling

- What pages to Download?
  - Not all, only “important” ones, prioritizing the Queue.
- Refreshing pages.
  - Download pages then “revisit” to update if changed. Impact on “freshness”.
- Load on the visited Web sites.
  - Consuming resources belonging to others.

Crawling – Page Selection

- Importance Metrics: good pages to visit.
  - Interest Driven: Similar words in Page and Query. Relationship between how many times the Word appear in the Web and in the Page. (Web size).
  - Popularity Driven: Links that point to Page *P* from any other Page *P′* (Web size).
  - Location Driven: URL, fewer slashes, .com

Crawling – Page Selection

- Crawler Models: visiting mainly high-importance pages.
  - Crawl & Stop: Start at Page *P* and stop after *K* Pages. Some may be of high Importance.
  - Crawl & Stop + Threshold: *T* is Importance target. Only accept above/equal *T*.
- Ordering Metrics: order URLs in queue due to importance.

Crawling – Refresh

- Pages are maintained up-to-date
- Freshness Metric:
  - Local page vs. real world counterpart.
  - Collection of Pages calculations:
    - Freshness: how fresh the collection is.

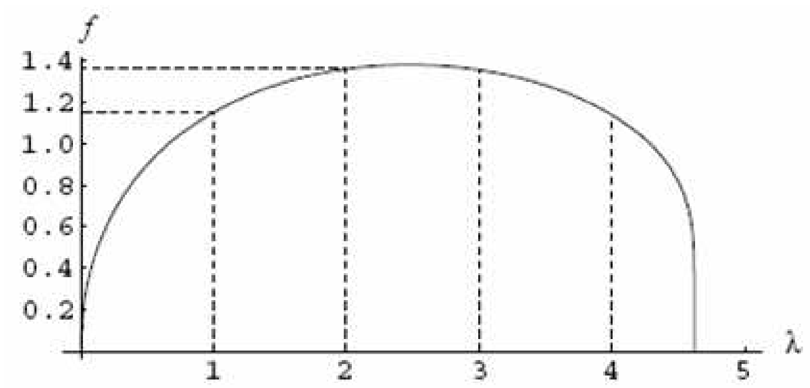
$$F(e_i;t) = \begin{cases} 1 & \text{if } e_i \text{ is up-to-date at time } t \\ 0 & \text{otherwise.} \end{cases} \qquad F(S;t) = \frac{1}{N}\sum_{i=1}^{N} F(e_i;t).$$

    - Age: how old the collection is.

$$A(e_i;t) = \begin{cases} 0 \\ t - \text{modification time of } e_i \end{cases} \qquad A(S;t) = \frac{1}{N}\sum_{i=1}^{N} A(e_i;t).$$

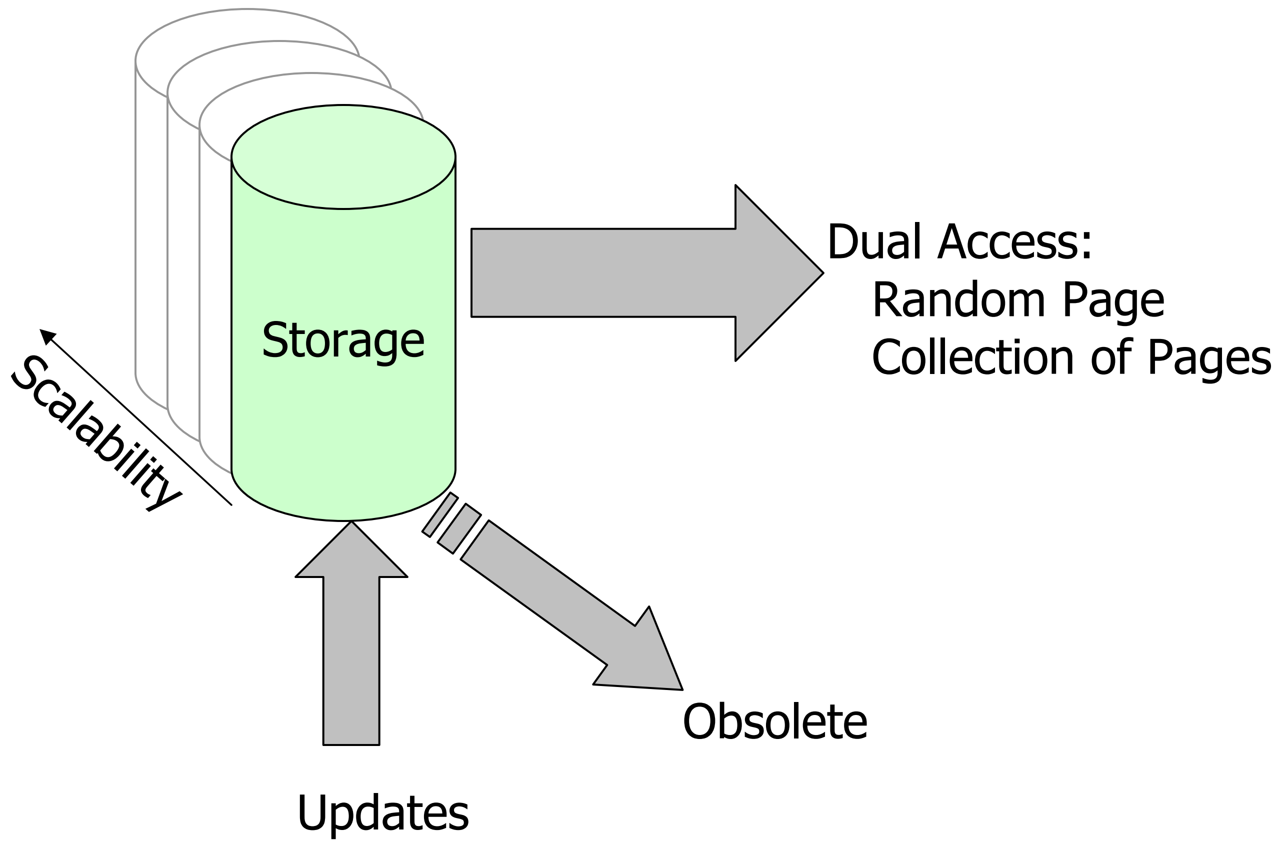
Crawling – Refresh

- Refresh Strategy
  - Uniform or Proportional refresh policy.
  - Available resources.
  - What page to refresh? Poisson process.



Storage

- Key challenges



Storage – Design

- Page distribution: to which node to assign.
  - Uniform Distribution: all nodes are treated identically, page can go to any node.
  - Hash Distribution: page allocation depends on page identifiers.

Storage – Design

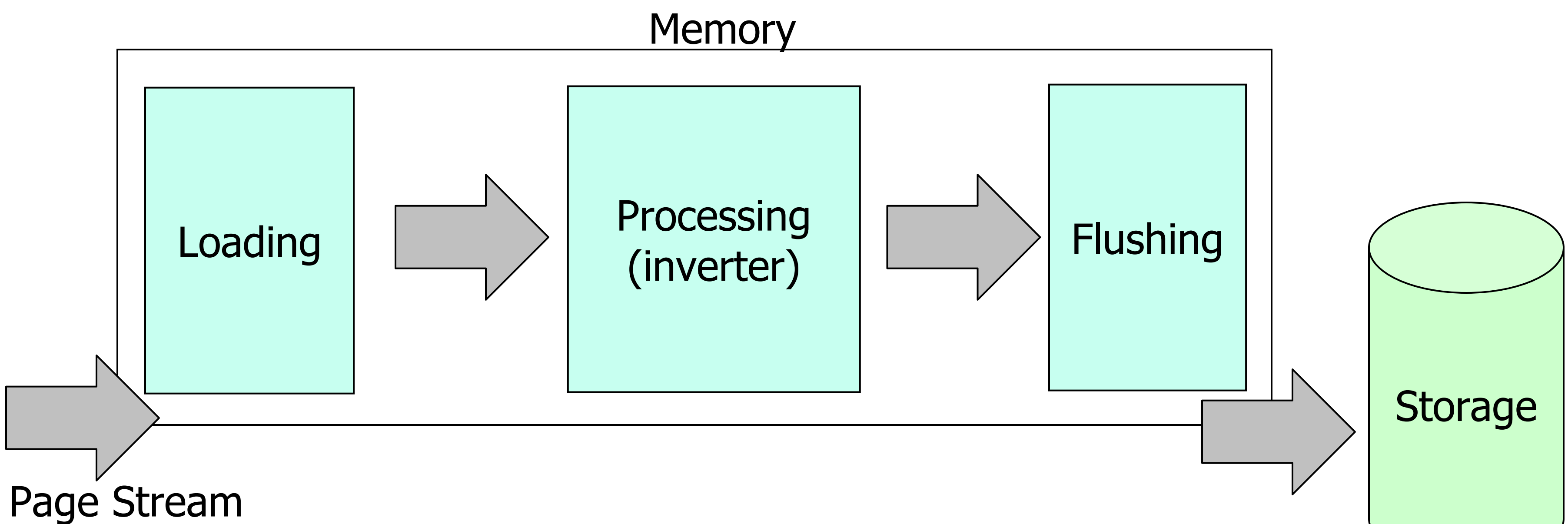
- Physical Page Organization: operations to be executed, addition / streaming / random page access.
  - Hashed organization based on identifiers.
  - Log structures with B-tree index of locations
  - Hash-Log

Storage – Design

- Update Strategies: dependant to crawler characteristics.
  - Batch Mode Crawler, “some day some time”.
  - Steady Crawler, runs with no pause.
  - Partial/Complete Crawls, specific set of pages or sites.
- Shadowing: cache and then update

Indexing

- Process:

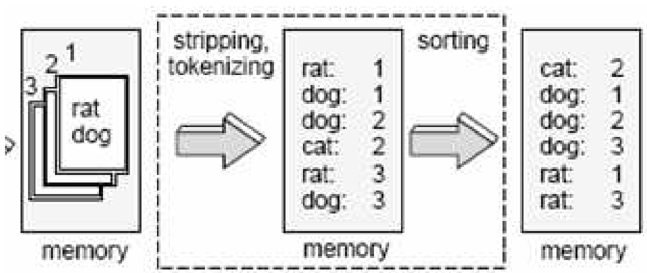


Indexing

- Indexer module builds two indexes:
  - Link Index: portion of the Web is modeled as a graph. Edge *A* to *B* represent a hyperlink. Given page *P* get incoming and outward links. (Web size)
  - Text (content) Index: Primary method to identify pages relevant to a query.
    - Inverted indexes, index structure choice of the Web.

Indexing – Inverted Index

- Inverted list for a term is a sorted list of locations where the term appears.
- Location: Page Identifier & Position in the Page.

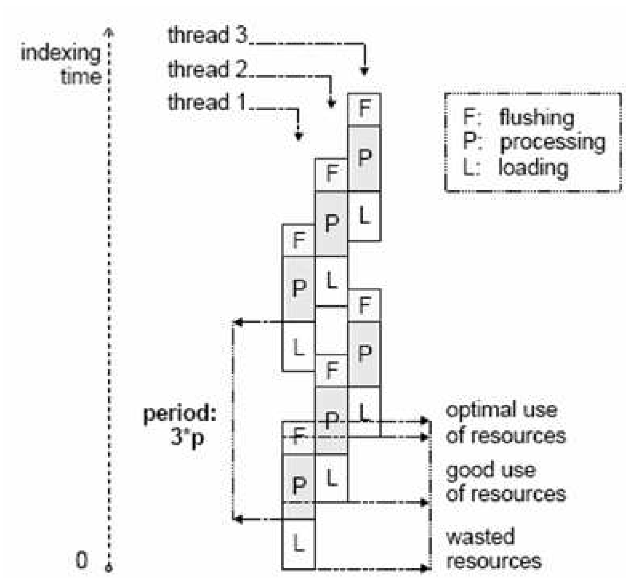


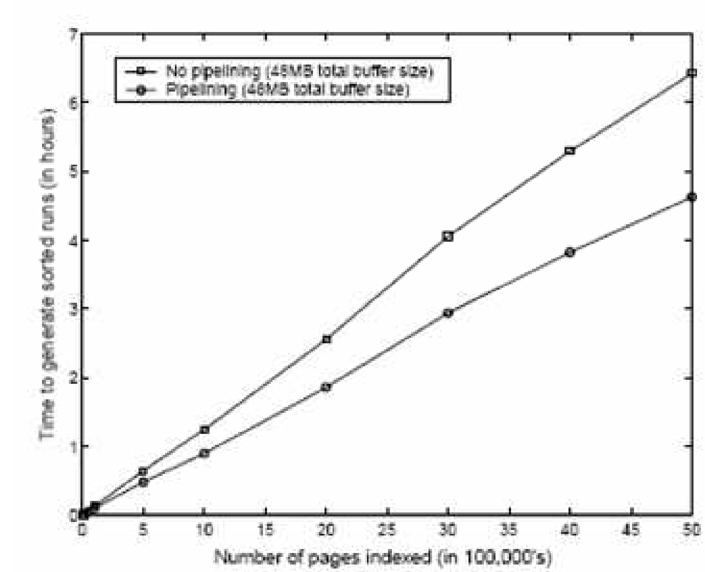
Indexing – Partitioning

- How to add the inverted list?
  - Local Inverted File, different nodes with different subsets of pages. Queries are broadcasted to all nodes.
  - Global Inverted File, each server stores only a subset of terms.
    - [a-m] => Node 1
    - [n-z] => Node 2

Indexing – Threads

- Experiments showed that sequential index-builder is 30%-40% slower than pipelined one.





Indexing – Statistics

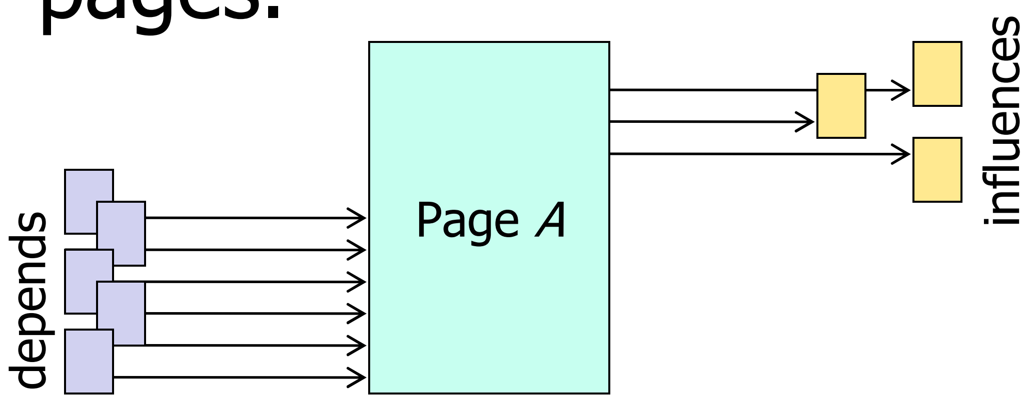
- Statistics are often used to rank search results.
- Statistics can be computed by the indexing system.
  - IDF inverse document frequency
    - $log(N/df_w)$
    - N pages in collection
    - df_w pages where w occurs

Ranking

- Pages that contain the search terms may be of poor quality or not relevant.
- Web pages are not sufficiently self-descriptive. Can be manipulated.
- Link Structure:
  - If *A* links to *B* then author of *A* recommends *B*.
  - At Global Level it is robust against spamming.

Ranking – Page Rank

- "Importance" of a page.
- Importance of pages that point to *A* and Importance of pages that *A* points to.
- Recursive, Depends and Influences other pages.



Ranking – Page Rank

- Simple Page Rank:
  - Assume that Web pages form a strongly connected graph.
  - *N(i)* denotes number of outgoing links *i*
  - *B(i)* denotes the set of pages that point to *i*
  - *r(i)* denotes Page Rank of page *i*

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

Ranking – Page Rank

- Practical Page Rank
  - Web is far from strongly connected.
    - Rank Sink, no links point outwards.
    - Rank Leak, page with no links.
    - Random Surfer will get stuck or lost.
  - Remove the Leak nodes and add a decay factor *d*.
    - Leak nodes will point back.
    - Random Surfer jumping randomly (decay factor)

Ranking – Page Rank

- Computational Issues.
  - Important value is the Page Order given by the Page Rank no the Values of the Page Rank.
  - Is not necessary to “finish” the iterations.
  - Algorithm can be stopped when values start to converge.

Ranking – HITS

- HITS, Hypertext Induced Topic Search
- Instead of global rank it is Query-Dependant.
- Produces two scores, Authority and Hubs.
  - *Authority* pages are most likely to be relevant.
  - *Hub pages* point to several authority pages.

Ranking – HITS

- Algorithm
  - Using the Query String
  - Identify a small subgraph of the Web and search for Authorities and Hubs.
    - Form a root set *R* and expand it to the pages in the neighborhood.
    - Link Analysis,
      - Authority Value = number of Hubs pointing to it.
      - Hubs Value = number of Links pointing to Authorities.

Ranking – HITS

- Algorithm
  - Resulting set shall be rich in Authorities and Hubs.
  - Authorities usually do not point to Authorities
    - Toyota -> Honda

Conclusion

- Searching the Web is the basis for many tasks.
- Search Engines are being relied in extracting the required information with one or two input keywords.
- Audio, Video, Images, new challenges for search engines.

What is this Page Known for?

Rafiei, Mendelzon
2000.
University of Toronto

Introduction

- Objective: Given a Page/Site on what topics is this page **considered** an authority by the **Web community**?
- Page classification.
  - What is a Page/Site about?
  - How is a Page/Site perceived?
  - What is a Person known for?

Related Work

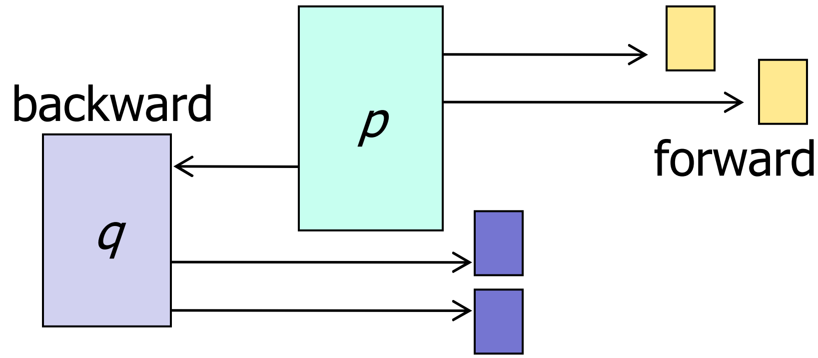
- Methods:
  - Page Rank.
  - HITS, Authority and Hubs.
  - Random Surfer.
- Difference
  - Ranking respect to a topic instead of computing a universal rank.

Random Walks

- One-Level Influence Propagation:
  - Jumps of the Random Surfer are *forward*.
  - Pages with relatively high reputations on a topic are more likely to be visited by the RS searching for that topic.
  - The number of visits of the RS depends on the pages on the same topic pointing to this one page and the reputation of those pages.

Random Walks

- Two-Level Influence Propagation:
  - The Surfer has two choices in page *p*
    - Transition out of page *p*
    - or, randomly pick any page *q* that has a link to page *p* and make a transition out of page *q*
  - Surfer can go Forward or Backward



Reputation of Pages

- Is not enough to use the “terms” and “phrases” that appear in a page.
  - Some terms may not be explicitly on the page.
- How to:
  - Start in page *p*
  - Collect all “terms” that appear in it.
  - Look at incoming links and collect “terms”.
  - Stop when incoming links have small effect.

Experiments

- Known Authoritative Pages
  - java.sun.com <Search> <Microsoft>

URL : java.sun.com   *500 links examined (out of 128653 available)*

Highly weighted terms: Developers, JavaSoft, Applets, JDK, Java applets, Sun Microsystems, API, Programming, Solaris, tutorial

Frequent terms: Java, Software, Computer, Programming, Sun, Development, Microsoft, Search

Experiments

- Personal Home Pages
  - Don Knuth <Dilbert>

URL : www-cs-faculty.stanford.edu/~ knuth   *500 links examined (out of 1733 available)*

Highly weighted terms: Don Knuth, Donald E Knuth, TeX, Dilbert Zone, Latex, ACM

Experiments

- Computer Science Departments
  - www.cs.helsinki.fi <Linux> <Linus>

URL : www.cs.helsinki.fi   *500 links examined (out of 9664 available)*

Highly weighted terms: Linux Applications, Linux Gazette, Linux Software, Knowledge Discovery, Linus Torvalds, Data Mining

  - www.cs.toronto.edu <Russia><Hockey>

Conclusion

- Algorithms are working as expected but still work to do improving their “TOPIC” prototype.