

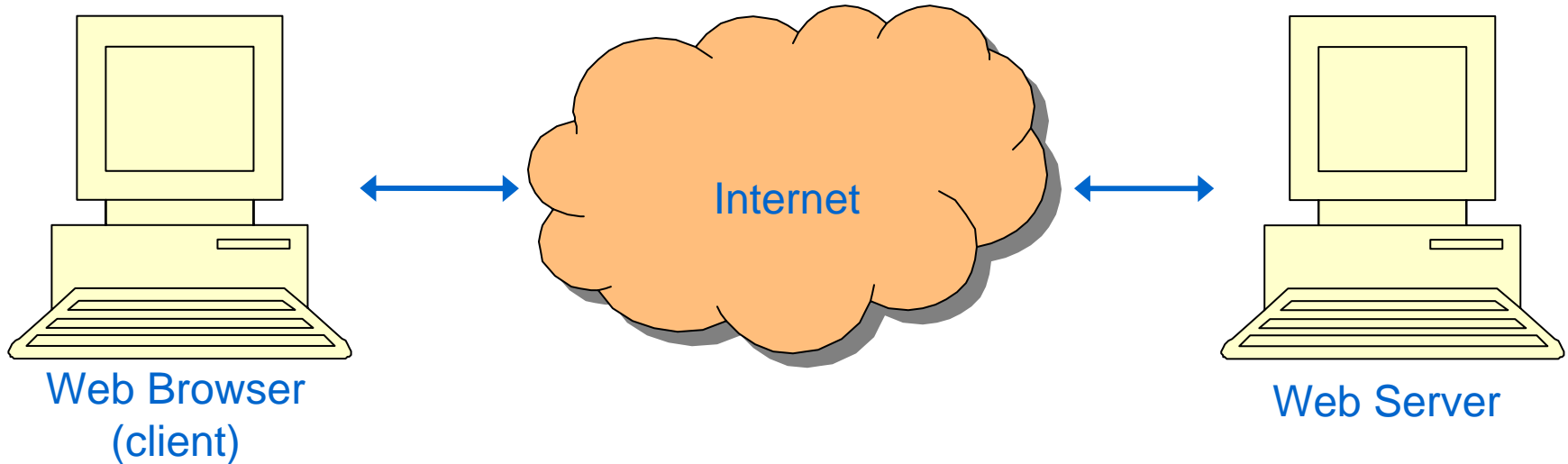
Searching the Internet

Internet Technologies and Applications

Contents

- Refresher on Web Technologies
 - Web Technologies
 - Statistics on Web Content and Servers
- Search Engine Architecture
 - Crawler
 - Indexer
 - Relevance Ranging
 - Retrieval Engine
- Search Engine Optimisation
- Improving Search Engines

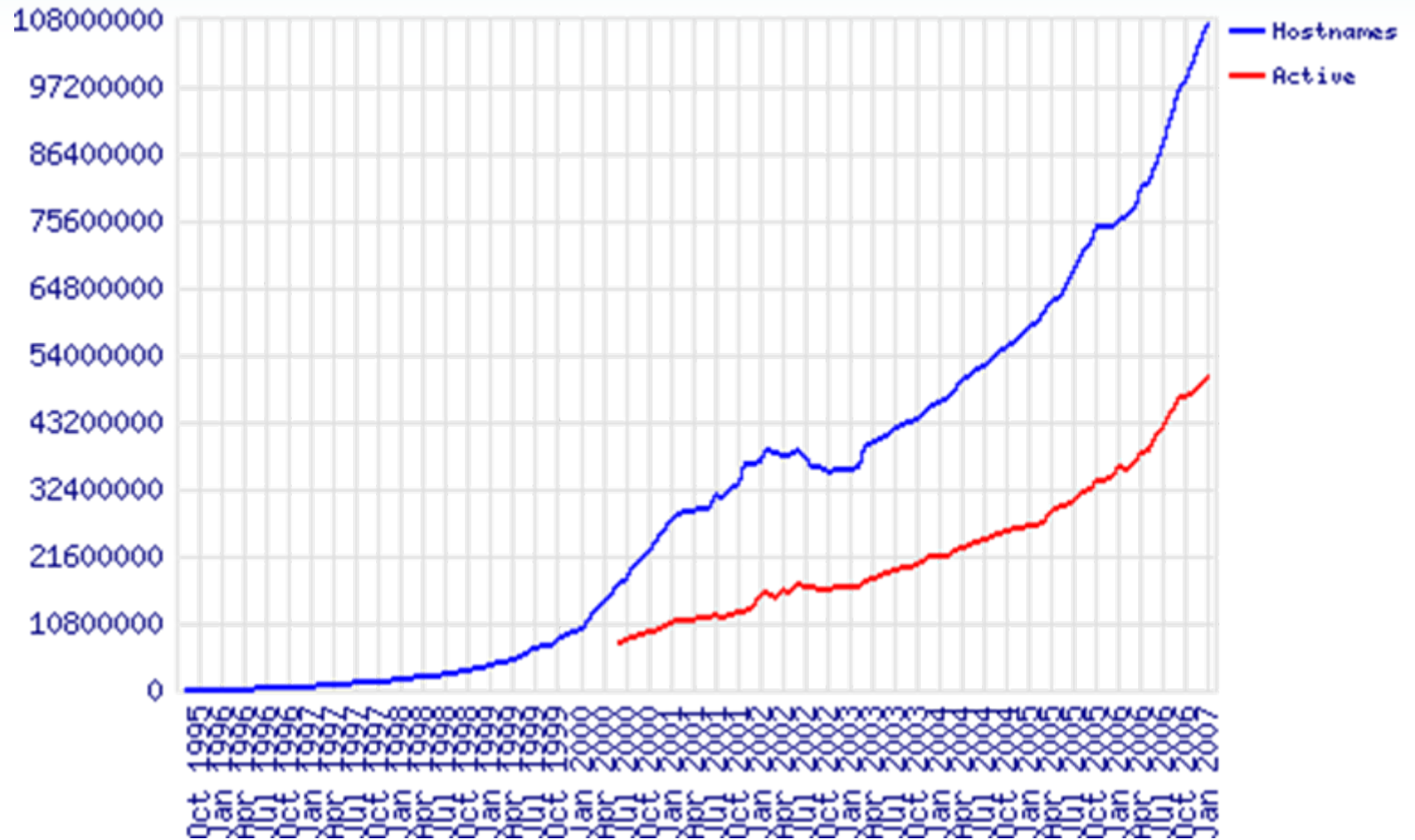
Refresher on Web Site Structure



- User enters URL (or clicks on link)
- Browser sends HTTP GET request to that IP address for URL
- Upon receiving HTTP REPLY, browser renders the HTML code to display the content to user

- Server passively listens for connections
- Pages (content) are stored in directories
- Upon receiving HTTP GET request, server retrieves the page and sends to client

Number of Web Servers



Source: January 2007 Netcraft Web Server Survey: www.netcraft.com

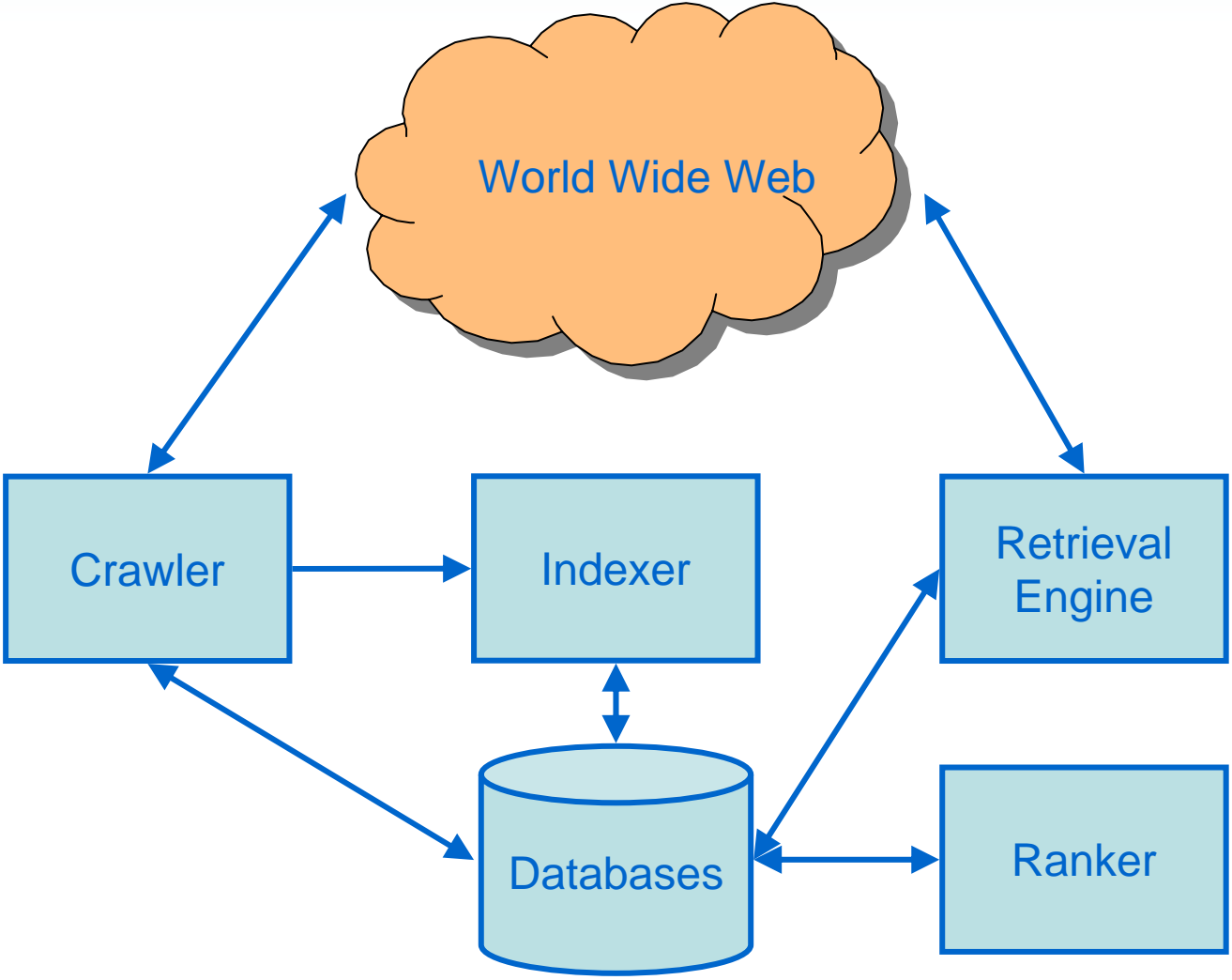
Number of Web Pages

- Indexed by search engines
 - 3-10 billion pages each
 - Hard to find accurate statistics
 - “Indexing” a page may mean different things in different search engines
- Some estimates say search engines only gather about 30% of all pages
 - And many search engines overlap each other

Search Engines and Directories

- Internet Directories
 - Manual (or semi-manual) construction
 - Pages are categorized and reviewed
 - Examples: Yahoo, DMOZ/Open Directory Project, WWW Virtual Library, ...
- Search Engines
 - Automatic indexing of pages
 - Crawler visits web pages
 - Examples: Google, Windows Live Search, Yahoo! Search, ...
- In practice – a combination of directories and search
 - Most directories include a search feature
 - Most search engines promote directories

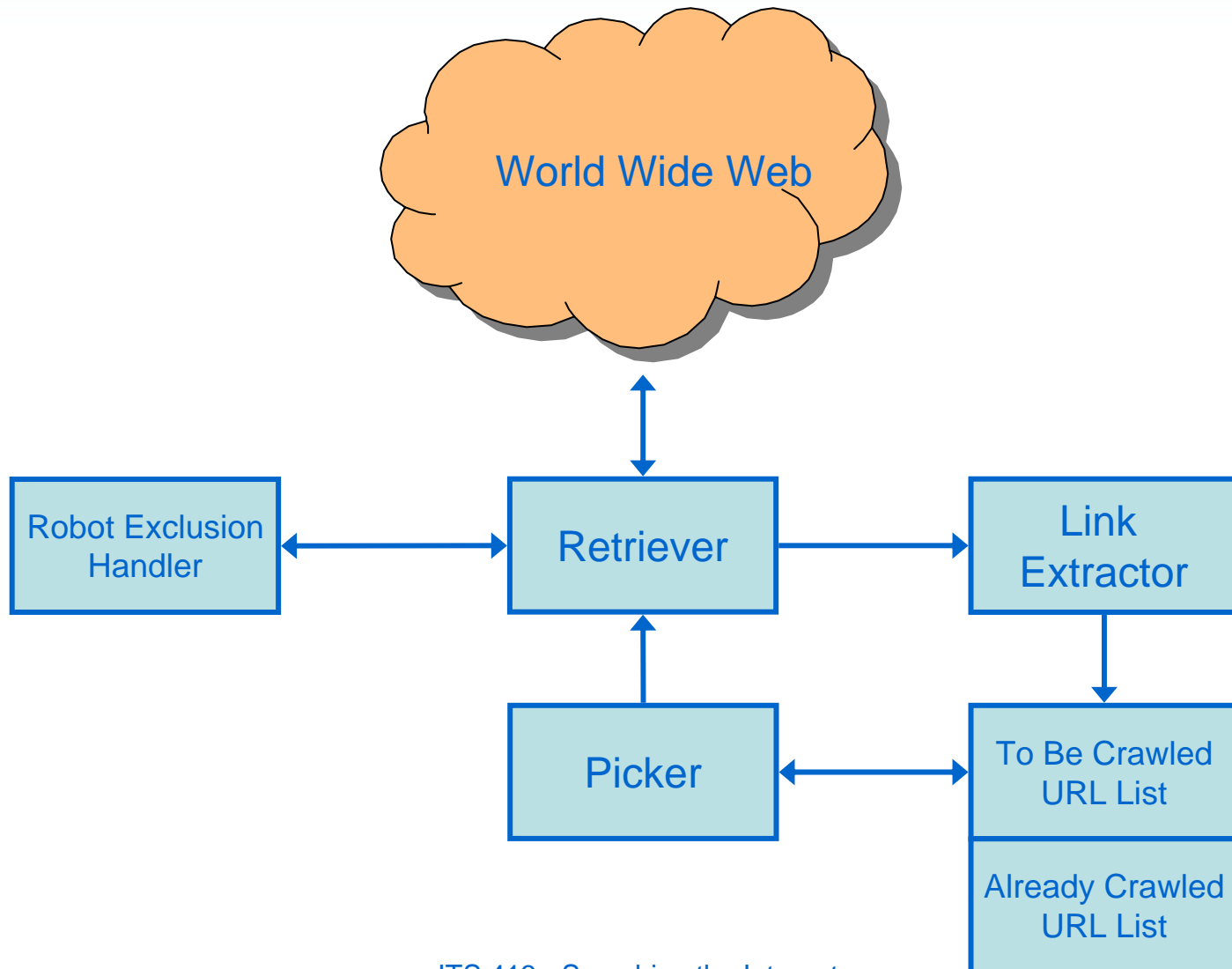
Search Engine Architecture



Crawler

- Also known as:
 - Robots, spiders, worms, walkers, wanderers, ...
 - Implementations: Googlebot, Yahoo Slurp, MSNBot,
- Discovers information on the web
 - Starts with a list of seed URLs
 - Visits a page at a seed URL
 - Follows links found in each page it visits
- Aim of a good crawler:
 - Find as many pages as possible in a given time
- Design issues:
 - A problem for crawlers:
 - Number of pages in the Internet is too large to visit them all
 - The pages are always changing (modified, created, deleted)
 - Therefore a crawler cannot visit all pages
 - The solution:
 - Aim to visit the most useful pages within a given time

Crawler Architecture



Crawler: Lists of URLs

- A crawler keeps two lists of URLs:
 - Pages it has yet to visit
 - Initialised with the seed URLs
 - Seed URLs should be popular pages with many outgoing links
 - Pages that have already been visited
- Performance of URL lists
 - Crawler may have billions of URLs in lists, representing gigabytes of memory
 - Store URL lists on disk and cache commonly used URLs in memory

Crawler: Picker

- Must pick the next URL to be visited from “To be Crawled” list
- Why is picking the next URL is so important?
 - A practical crawler cannot visit all URLs; need to select the most useful URLs
- Picking the next URL:
 - Breadth First Search (BFS)
 - Pick a URL from a site that has not been visited yet
 - Provides coverage of a small portion of a large number of sites
 - Depth First Search (DFS)
 - Pick a URL from the current site
 - Provides coverage of a small number of sites, but in great detail
 - Reputation-based
 - Pick a URL with the highest “reputation”
 - Reputation: can be based on links between pages (e.g. more links to a page, the higher reputation the page has) – covered later in ranking
 - Requires crawler to continuously re-calculate the reputation as it discovers new page
 - Increased computing power needed

Crawler: Retriever

- Once picker selects a URL, retriever request page from server
- Multiple retrievers work in parallel
 - Maximise the efficiency of using the network
 - BUT must consider impact on web server
 - Too many concurrent requests on a web server can overload it
 - Good crawlers will have concurrent requests to *different* web servers
 - A single web server will only have one request at a time
 - There will be reasonable delays between subsequent requests to the same server

Crawler: Link Extractor

- Once the page is retrieved, must extract links from that page
 - Must first identify the type of document: HTML, Word, PDF, ...
 - A parser for specific document type extracts link from the document
 - Parser can be simple enough to just handle links (for example, does not have to parse <h1> tags)
 - But a good parser can handle incomplete or erroneous document types (e.g. an incorrectly structured HTML page)

Crawlers and Robot Exclusion

- Webmasters may not want all of their website indexed by a crawler
 - Contains private data; crawlers accessing many files may cause performance problems
- Robot Exclusion Protocol (www.robotstxt.org)
 - A file robots.txt in a web directory is read by a crawler
 - A “good” crawler will follow instructions of the file
 - E.g. do not crawl these directories
 - Similar functionality can be obtained using a META tag in the actual web page
 - Robots.txt file has very simple format
 - Does not allow for complex combinations of files to be specified (e.g. wildcards)
 - Relies on crawlers to follow the standard (that is, to be “good”):
 - Nothing to stop a crawler from not reading or following the robots.txt file
- More secure protection of content is achieved using password protection

Robot Exclusion Example

- Using file: `http://www.domain.com/robots.txt`

- Disallow all crawlers from accessing a directory:

```
User-agent: *  
Disallow: /private-directory/
```

- Disallow specific crawlers from specific files:

```
User-agent: GoogleBot  
Disallow: /~joe/private.html  
Disallow: /~joe/foo.html  
Disallow: /~joe/bar.html
```

- Using META tag in the HTML file:

```
<html>  
<head>  
<meta name="robots" content="noindex,nofollow">  
<meta name="description" content="This page ....">  
<title>...</title>  
</head>  
<body>  
...
```

- “index” = allowed to index this page; “follow” = allowed to follow links on this page (and opposite for “noindex”, “nofollow”)

Indexer

- Index contains the content extracted from the crawled web pages
- The Indexer must process the web page (document) and extract the content
 - Indexing can be performed in parallel with crawling
 - Indexing is very computation and memory intensive
 - Must process billions of pages, and store content extracted from those pages
- Parsing (or pre-processing) the document
 - Must determine which parts (words) of the document best determine its content
 - Limit the number of indexed words per document (to save space)
 - Omit common or non-descriptive words (e.g. “the”, “and”, “I”)
 - Using stemming of words: “reformulation”, “reformative” both indexed as “reform”
 - Output of parsing: list of words for document, and for each word, a description of where it is found (e.g. in title, heading, link)
- The content indexed varies across search engines

Ranking

- With index, a user can submit a query and find all documents with given words
 - Millions of documents returned therefore, not practical; a user wants the documents to be ranked in order of relevance
- Relevance ranking can be:
 - Connectivity-based: number of links to/from the page; number of times page is accessed
 - Content-based: number, frequency and location of terms (words) in document and index
 - Term Frequency: Documents with more occurrences of search word relative to document length receive higher weight
 - Term Location: Terms in title, headings, links (href), figures, metatags may receive higher weight than terms in text
 - Proximity: Terms of a phrase that are close together in document gives higher weight
- Ranking formula:
 - Differs among search engines
 - Kept secret by search engines
 - So webmasters cannot unfairly get their sites higher rankings

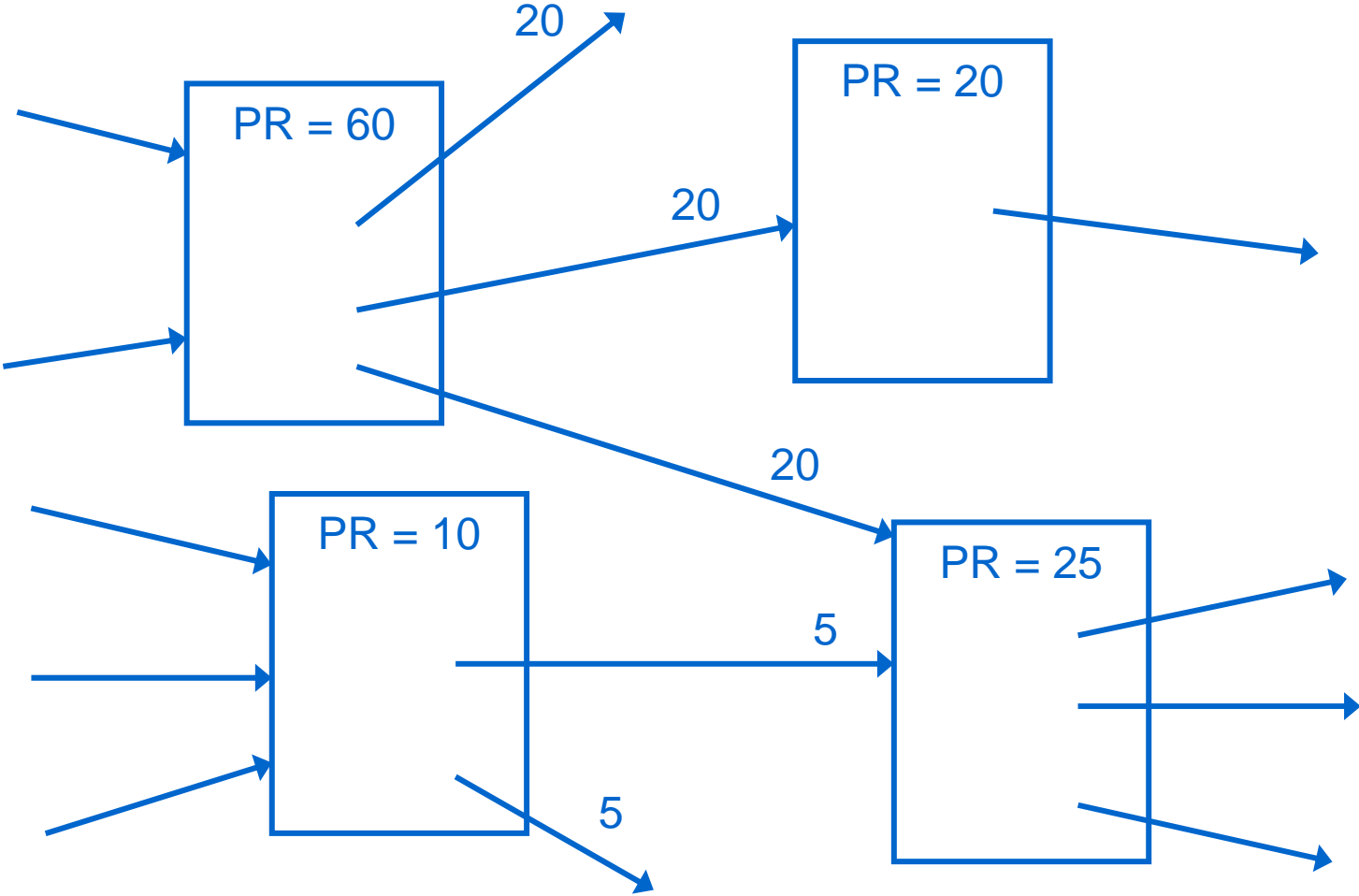
Ranking Techniques

- Query Independent: algorithms used to rank all pages independent of search terms
 - Use counts of links to and from pages; number of accesses
 - Can be pre-calculated (e.g. before a query is run)
- Query Dependent: calculate rank based on content to see how close the documents relate to search terms
 - On-the-fly computation: must be calculated for each search query; resource consuming
- Hubs and Authorities
 - Authority pages are considered a primary source of information (on some topic)
 - E.g. Thammasat University website is an authority on Thammasat
 - Hubs are a collection of links (e.g. Yahoo directory)
 - Ranking is based on:
 - Good authority pages will be linked to from many good hubs
 - Good hubs will link to many good authorities
 - Ranking calculated by analysing links for collection of pages relevant for a query
 - On-the-fly computation
 - Works well; but computation intensive and not efficient enough for large search engines
- Search engines use a combination of approaches

Google PageRank

- Google uses more than 140 criteria to rank pages
 - Exact details and algorithms of criteria is secret
 - PageRank is a key concept used by Google
- PageRank
 - Example of Query-independent (global) ranking
 - Main idea of PageRank:
 - A “web surfer” follows links in pages and calculates rank
 - Number of links to pages will contribute to rank
 - Links from higher ranked pages contribute more than links from lower ranked pages
 - Process is iterative – cycles may occur where the ranks of pages will keep increasing
 - To avoid cycles, sometimes surfer randomly move to another unlinked site
 - Problem with PageRank
 - Newly created pages will not have many links to them – low PageRank
 - Low PageRank for a page means will not be found in search results (and hence hard to get links to the page, i.e. cannot increase PageRank)

Simple PageRank Example



Retrieval Engine

- Combines Index and Ranking to present results to user:
 - Parse the user's search query
 - Remove any common words (e.g. "and", "the")
 - Find relevant documents from Index
 - Rank documents
 - Present results to user

Search Engine Optimisation

- Search engines are the main technique for finding information on the Internet
 - Financial advantages of getting your website ranked highest
 - More people use your products and services
 - You can demand more from advertisers on your site
- Search Engine Optimisation (SEO): Techniques to get your website a higher ranking
 - Good: improve your website and content to suit search engines algorithms
 - Bad: use artificial techniques to get higher ranking than your content deserves!
 - (Distinction between good and bad techniques is more ethical and business oriented)
 - Follow search engine guidelines, do not try to deceive

SEO Techniques

- Many companies offer ways to increase your chances of higher rank
 - Some techniques are not allowed by search engine guidelines
- Basic approach for good ranking
 - Include relevant keywords in:
 - Page title and headings
 - Links from other sites to yours
 - Keywords in your domain/URLs?
 - Page text
 - Have links from other popular sites to yours
- Link exchange
 - Informal exchange between sites (“I will link to yours if you link to mine”)
 - Pay companies and organisations to link to your site

SEO Techniques

- Malicious Techniques
 - Keywords stuffing: Include many (hundreds) keywords in title, metatags, page (including to topics that aren't on your site) (easy to detect)
 - Crawler traps: software/pages that try to keep crawler on site (so crawler thinks its important site (easy to detect)
 - Ghost sites: create many simple one-page sites that link to your site
 - Blogs/Wikis: include many links in blogs/wikis to your website
- Banning Sites
 - Search engines have banned sites (removed from index) if guidelines are not followed
 - BMW and Ricoh were briefly banned in 2006

Improving Search Engines

- New Features and Capabilities:
 - User interface: graphical visualisation of results
 - Metadata: Include more descriptive information about a document; follow W3C standards on metadata
 - Metasearch: search multiple engines at once
 - Problems with determining ranking of results, conflicting ranks of same document, and speed depends on slowest engine
- Research:
 - Including dynamic content in index: database content
 - Peer to peer search: use peers to do crawling and indexing
 - Can be very slow in getting responses to queries
 - Semantic Web: better ways for describing and searching based on the meaning of content