

A summary on
Chapter 15: Web Search
Chapter 16: XML Retrieval

By Akshat Kukreti

Web Search

- Web pages are gathered by a Crawler and stored by the search engine.
- The snapshot of the web is refreshed on a regular basis.
- Web graph: Sites contain many web pages that link to web pages in other sites.
- Indexable web: Those pages that should be considered for inclusion in a general purpose web search engine.

Web Search

- Navigational queries, Informational queries, Transactional queries.
- Ranking
 - Static rank during indexing process
 - Dynamic rank = Static rank + query-dependent features (term proximity, frequency)

Web Search

- PageRank of a page depends on
 - PageRank of pages that have outgoing links to it
 - PageRank of sinks (pages with no outgoing links).
- On applying the PageRank formula we get a system of linear equations with number of variables = number of pages.
- Fixed point iteration
 - Guess the value of variables, apply to right hand side.
 - Iterate till the change in values from iteration to iteration drops below a certain value.

Web Search

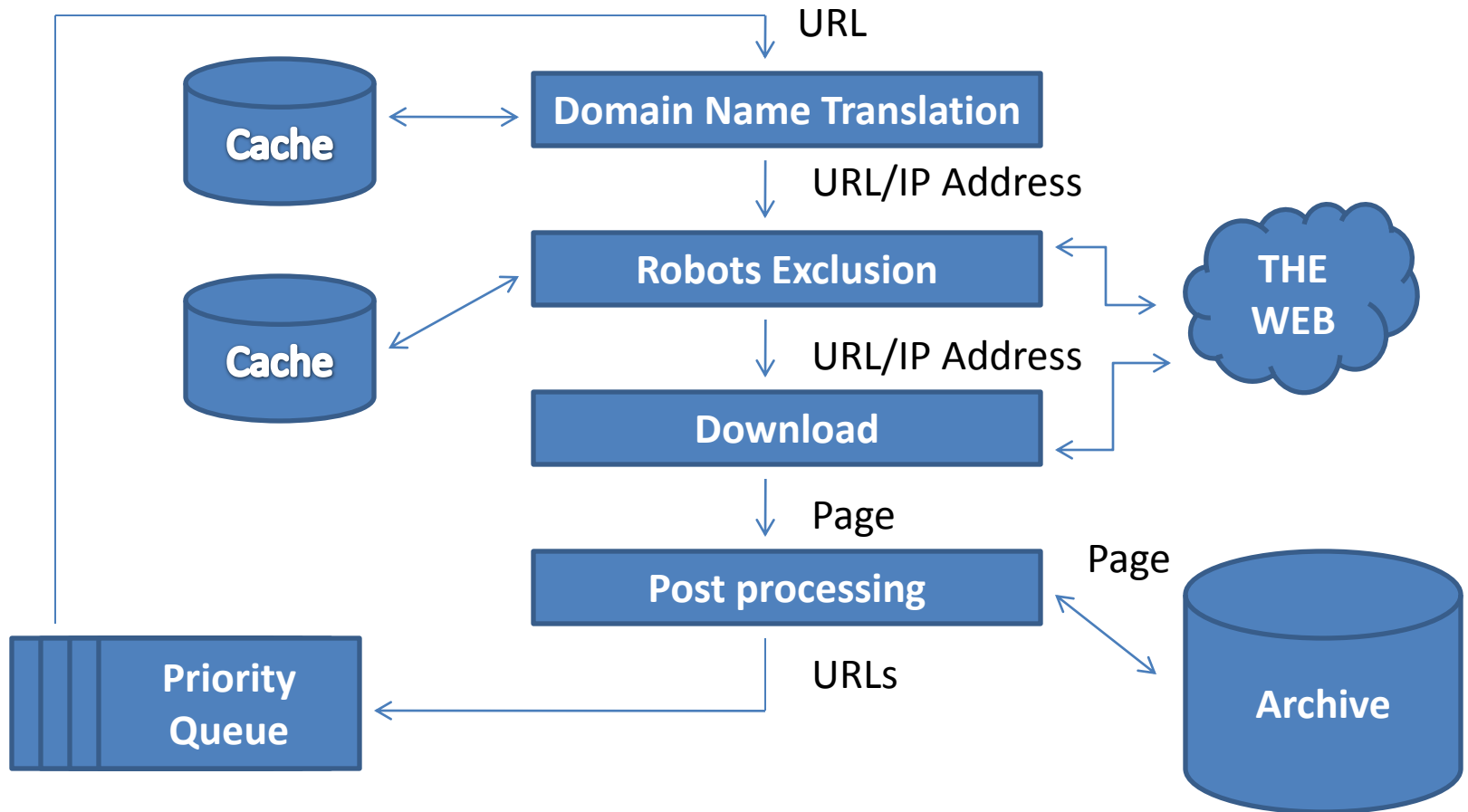
- Extended PageRank
 - Uses Jump vector for random jumps
 - Uses follow matrix $F[i,j]$ = probability of reaching page j from page i when following a link.
- PageRank variants
 - Personalized PageRank: High jump velocities to pages of personal interest (Bookmarks etc.).
 - Topic-oriented PageRank/Focused PageRank: High jump probabilities to pages that are related to a topic.

Web Search

- Web Crawlers
- Download web pages, extract links from them and do the same with the extracted links.
- Issue: Scale and speed of the crawl process.
- Crawler must download pages concurrently from multiple sites.
- Avoid re-downloading pages and retry if download fails
- Respect the wishes of the site it visits. Robots.txt
- Should not interfere with the normal functioning of the website it visits.
- Maintain a Priority Queue of Indexes for crawling and re-crawling (Updating the index).

Web Search

- Components of a Crawler

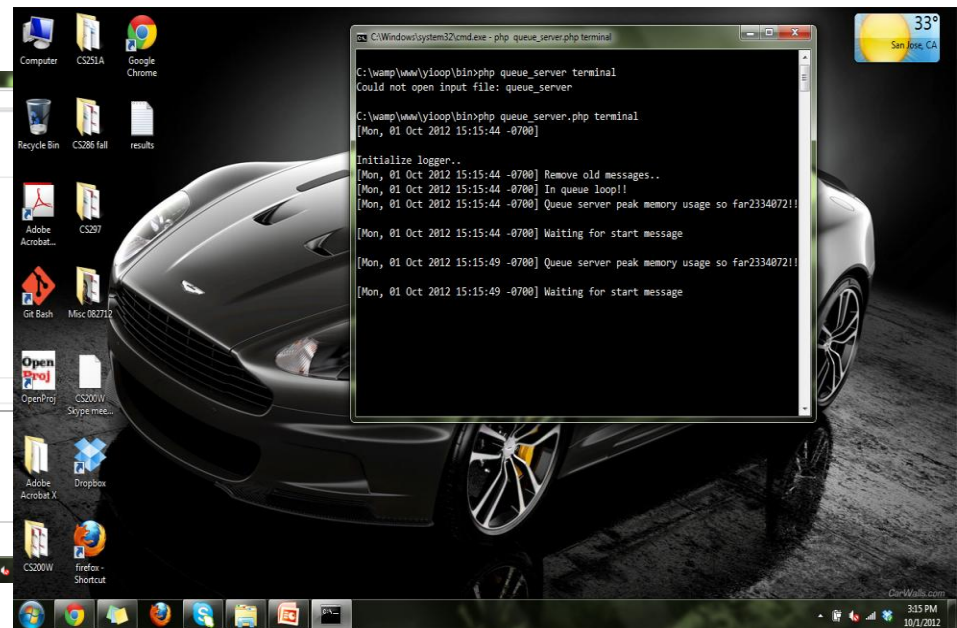
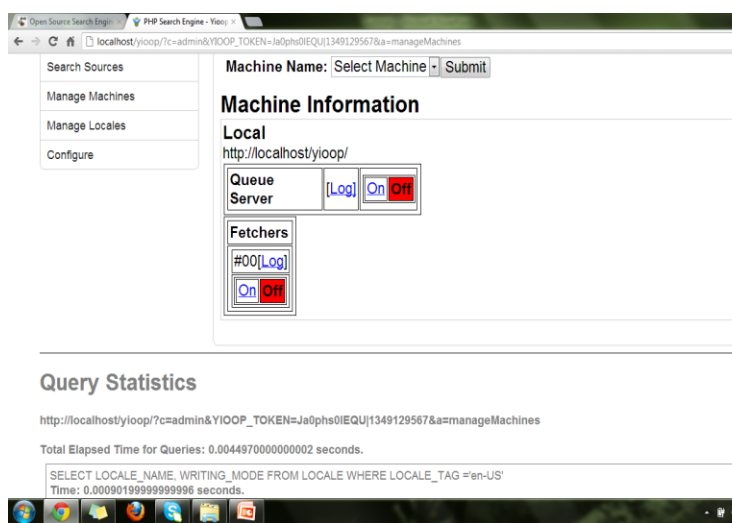


Yioop! Search Engine software

- An open source search engine software developed in PHP.
- Uses it's own web archiving format for scalability.
- Provides users with a web-based and a function API.
- Uses the multi-curl library (written in C language) for multithreading.

Yioop! Search Engine Software

- Managing fetcher and queue server through Yioop!'s web interface and command line interface

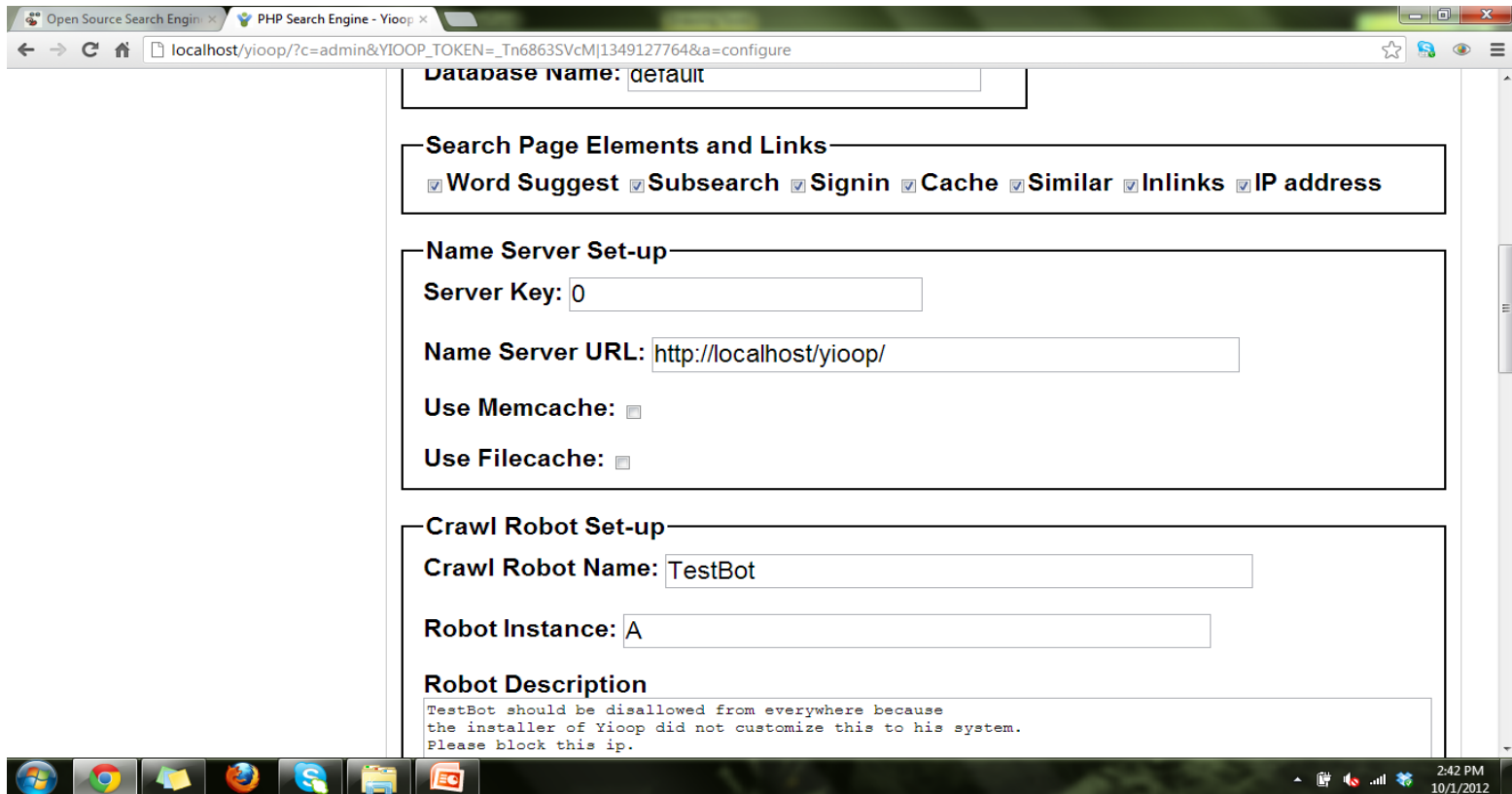


Yioop! Search Engine software

- **Yioop!'s model**
- In a distributed setup, each node has a web server running.
- **Name Server:** The node that is the coordinator for crawls.
- **Queue Servers:** Processes that manage indexing and scheduling jobs
- **Fetchers:** Processes responsible for downloading of web pages.

Yioop! Search Engine software

- Name server setup



The screenshot shows a web browser window with the URL `localhost/yioop/?c=admin&YIOOP_TOKEN=_Tn6863SVcM|1349127764&a=configure`. The page displays the configuration interface for the Yioop! search engine. The "Name Server Set-up" section is highlighted, showing the following fields and options:

- Database Name:** default
- Search Page Elements and Links:** Word Suggest, Subsearch, Signin, Cache, Similar, Inlinks, IP address (all checked)
- Name Server Set-up:**
 - Server Key: 0
 - Name Server URL: `http://localhost/yioop/`
 - Use Memcache:
 - Use Filecache:
- Crawl Robot Set-up:**
 - Crawl Robot Name: TestBot
 - Robot Instance: A
 - Robot Description: TestBot should be disallowed from everywhere because the installer of Yioop did not customize this to his system. Please block this ip.

Yioop! Search Engine Software

- **Yioop!'s model contd...**
- Fetchers ping name server to get the current crawl and the list of queue servers.
- Fetchers then request queue servers for messages and schedules in a round-robin manner.
- Schedule: Data to process
- Messages: Control information

Yioop! Search Engine software

- **Yioop!'s model contd..**
- Queue servers generates schedules
- Fetchers process the schedule and POST the results(summaries of web pages) to the queue server's web server
- The data is written to a set of received files.
- The queue server then merges the results with the index.
- Queue servers maintain the Priority queue of URLs

Yioop! Search Engine software

- Detects if a website is congested. If yes, slows down the crawling of the site.
- Supports domain name caching
- Allows users to dynamically change crawl settings. For example, adding new seed sites to an active crawl.

XML Retrieval

- XML header:
- `<?xml version =“1.0” encoding=“UTF-8”?>`
- `<article journal=“IEEE TKDE” volume=“9” number=“2” year=“1997”>`

Element name

Value

Attribute

Attribute, Value pair

- `<name/>` Empty element tag

XML Retrieval

- As XML elements nest, they can be viewed as a tree.
- An XML retrieval is framed as a tree-matching or path-matching problem.
- Xpath, NEXI and Xquery

XPath

- Provides a standard notion for specifying sets of elements and other nodes in XML documents.
- Path expressions specify sets of document elements.
- `/article/body/section` (top level sections)
- `/article/body/section/section` (level two sections)
- Predicates
 - `/article/body/section[2]` (second section in document order)
 - `//section/title` (all section titles)

NEXI

- XPath does not support ranked retrieval.
 - Exact match semantics like Boolean algebra and Region algebra
 - Uses contains
 - `//article//section[contains(.,dogs)]`
- NEXI (Narrow Extended XPath I)
 - Narrows path expressions to a subset of those included in Xpath
 - Provides extension for ranked retrieval
 - Replaces contains with about
 - `//article//section[about(.,dogs)]`
 - Allows path elements to be combined using Boolean operators “and” and “or”.
 - Supports wildcards in path expressions.

XQuery

- Extends Xpath with facilities for manipulating and transforming XML documents.
- Allows dynamic construction of XML documents. The output itself is an XML
- XQuery operations operate on ordered sequences of nodes and values.
- FLWOR – for, let, where, order by, return.

Indexing and Query Processing

- XML Indexing and query processing problems not handled by region algebra
- Recursive nesting of elements
 - //section/section
- Direct containment relationships
 - /article/body/section/p
- Extended Inverted List ADT
 - (start, end, depth)

Indexing and Query Processing

- Methods
 - **first(term)**
 - Returns first tuple
 - **last(term)**
 - Returns last tuple
 - **next(term, current)**
 - Returns first tuple starting after current position
 - **prev(term, current)**
 - Returns last tuple starting before current position

Ranked retrieval

- Queries represented as term vectors
 - <“text”, “compression”>
- Queries represented as path expressions
 - /article/body/section[about(./title, “memory requirements”)]
- Strict interpretation
 - Return only top level sections.
- Loose interpretation
 - Return sections, subsections and so on...

Ranked Retrieval

- Content-only task
 - Ranking elements according to term vectors
- Content-and-structure task
 - Ranking elements according to path
- Method
 - Ignore about function and rank based on term vectors