IR UNIT – 6 (Parallel Information Retrieval, Web Search) – END-SEM PYQ Answers

► **NOV/DEC 2022**

Q7) a) Describe Parallel Query Processing with suitable examples. [8]

Parallel Query Processing allows an IR system to process queries using multiple processors, machines, or servers simultaneously.
Its main goal is to reduce query response time and scale to large datasets (e.g., web search).

1. Why Parallel Query Processing?

- Large-scale search engines store billions of documents
- Query load is extremely high
- Need sub-second response time
  Parallelism increases throughput and decreases latency.

2. Techniques for Parallel Query Processing

i) ***Document Partitioning:*** Documents are divided into multiple servers. Each server holds a portion of the collection.

Process:

1. Query sent to all servers
2. Each server searches its partition
3. Partial results merged to produce final ranking

Example: Google stores documents across thousands of servers.
Query → multiple servers → combined ranked results.

ii) ***Term-Based Partitioning:*** Index is split based on terms. Each server stores postings lists for different terms.

Process:

1. Query terms routed to servers that store those terms
2. Results combined

Example: Server A stores postings for “India”, server B stores postings for “Cricket”.
Query: “India cricket records” → both servers process parts → merged.

iii) ***Hybrid Partitioning:*** Combination of both document and term partitioning to balance load.

iv) ***Pipeline Parallelism:*** Different stages of processing (parsing → scoring → ranking) run in parallel on different nodes.

3. Example of Parallel Processing

Query: “COVID vaccination rate India”

- Partition 1 → health documents
- Partition 2 → news articles
- Partition 3 → scientific papers
  All process simultaneously → results merged → final top-k list.

4. Benefits

- Faster query response
- Can handle millions of queries per day
- Load distributed across many servers
- Improves scalability and fault tolerance

b) Write a short note on: [9]
i) The structure of the web ii) Queries and Users iii) Static ranking

i) The Structure of the Web (3 marks)

The web is a huge graph-like structure consisting of:

- Nodes: Webpages
- Edges: Hyperlinks connecting pages

Characteristics

- Highly connected
- Follows power-law distribution
- Small-world properties (short path between any two pages)
- Contains communities or clusters of related pages

Components

- Surface Web: Indexed by search engines
- Deep Web: Behind forms/logins
- Dark Web: Encrypted, not publicly accessible

Understanding structure helps in:

- Web crawling
- Ranking (PageRank)
- Detecting spam and communities

ii) Queries and Users (3 marks)

Users interact with search engines through queries, typically very short (2–3 words).

Query Characteristics

- Short, ambiguous
- Often misspelled
- Highly diverse (navigational, informational, transactional)

User Behavior

- Users expect results within milliseconds
- Rarely go beyond first 5–10 results
- Click behavior influences ranking (implicit feedback)
- Personalization improves relevance for each user

Understanding queries and users helps in improving search quality and UX.

iii) Static Ranking (3 marks)

Static ranking scores documents independent of the query.

Examples

- PageRank: Based on link structure
- Document Popularity: Traffic, clicks
- Authority/Trust Score: Signals from credible sites
- Freshness: Newer pages may score higher

Static ranking is combined with dynamic (query-dependent) ranking to produce final results in search engines.

Q8) a) Describe MapReduce with suitable examples. [9]

MapReduce is a programming and processing model developed by Google for processing large datasets across distributed clusters.

1. Components

i) ***Map Phase****:* Takes input data and transforms it into (key, value) pairs.

ii) ***Shuffle Phase:*** Intermediate key-value pairs grouped by key.

iii) ***Reduce Phase:*** Aggregates results by key and produces the final output.

2. Example: Word Count (Classic Example)

Map Phase

Input: Text
Mapper outputs:

("the", 1)

("cat", 1)

("the", 1)

Shuffle Phase

Groups by words:

the → [1,1]

cat → [1]

Reduce Phase

Adds counts:

the → 2

cat → 1

3. Real-World Examples

i) ***Log analysis***: Mapping log entries → reducing by user or timestamp.

ii) ***Index building for search engine****:*
Map: Extract terms from documents
Reduce: Merge posting lists per term

iii) ***Sorting & Statistical Analysis****:* Large datasets processed in parallel across distributed nodes.

4. Advantages

- High scalability
- Fault tolerance
- Parallelism across commodity hardware
- Works on massive datasets (terabytes/petabytes)

b) Write a short note on: [8]
i) Evaluation web search **ii) Web Crawlers**
iii) Web crawler libraries **iv) Dynamic ranking**

i) Evaluating Web Search (2 marks)
Evaluation focuses on:

- User satisfaction
- Click-through rates
- Time to first click
- Precision/NDCG of top results
- Query logs used as implicit feedback
  Search quality measured through online A/B tests.

ii) Web Crawlers (2 marks)
Software that automatically downloads web pages.
Steps include:

1. Fetch URL
2. Parse page
3. Extract links
4. Add new links to queue
5. Continue crawling

Used for:

- Building search engine indexes
- Monitoring changes
- Extracting large datasets

iii) Web Crawler Libraries (2 marks)
Popular libraries/tools for crawling:

- Scrapy (Python) – high-level crawling framework
- BeautifulSoup (Python) – HTML parsing
- Heritrix (Java) – official Internet Archive crawler
- Nutch (Java) – scalable open-source crawler

These tools help implement large-scale automated crawlers easily.

iv) Dynamic Ranking (2 marks)
Dynamic ranking assigns query-dependent scores.

Factors

- TF-IDF
- BM25 score
- Query-document similarity

- Freshness, recency
- User behavior signals (clicks, dwell time)

Final ranking = Static ranking + Dynamic ranking.
Used by search engines to show real-time relevant results.

► **MAY/JUNE 2023**

Q7) a) Describe Parallel Query Processing with suitable examples. [8] → DONE

b) Write a short note on: [9]
i) The structure of the web → DONE ii) Queries and Users → DONE iii) Static ranking → DONE

Q8) a) Describe Map Reduce with suitable examples. [9] → DONE

b) Write a short note on: [8]
i) Evaluation web search → DONE ii) Web Crawlers → DONE
iii) Web crawler libraries → DONE iv) Dynamic ranking → DONE

► **NOV/DEC 2023**

Q7) a) Explain in detail Parallel Query Processing with suitable examples. [9]

b) Write a short note on: [8]
i) Static ranking ii) Dynamic ranking

i) Static Ranking [4 Marks]

Static ranking assigns a query-independent score to each document.
Score does not change across different queries.

Factors Used

- PageRank: Based on incoming links
- Authority: Credibility of domain
- Popularity: Click counts, traffic
- Freshness: Newer pages may rank higher
- URL quality: Shorter or structured URLs preferred

Role in Search Engines

Static ranking helps in:

- Ordering large document sets before dynamic scoring
- Boosting trusted or authoritative pages

- Reducing search latency by pre-filtering high-quality documents

Example: Wikipedia pages have high PageRank → appear at top even without many query matches.

ii) Dynamic Ranking [4 Marks]

Dynamic ranking assigns a query-dependent score based on relevance to the specific query.

Factors Used

- TF-IDF or BM25 scores
- Query term frequency
- Document-query similarity
- Freshness for time-sensitive queries
- User behavior (clicks, dwell time)

Role in Search Engines

- Provides personalized and context-sensitive results
- Adapts to query intent
- Changes rankings for each query

Example: Query: "Best laptops 2024"
Dynamic ranking will prioritize recent review pages even if static score is low.

Q8) a) Describe MapReduce with suitable examples. [9]

MapReduce is a distributed big-data processing framework developed by Google.
It processes large datasets across clusters of machines using two main functions: Map and Reduce.

1. Map Phase: Breaks input into key–value pairs.
Example (Word Count):
Input text: "the sun rises"
Mapper outputs:
(the,1), (sun,1), (rises,1)

2. Shuffle Phase: Groups all values by key.
the → [1,1,1]
sun → [1,1]
rises → [1]

3. Reduce Phase: Aggregates grouped values to produce final output.

the → 3
sun → 2
rises → 1

4. Real Examples of MapReduce

i) ***Building an Inverted Index***

- Map: For each term → (term, docID)
- Reduce: Merge postings per term

ii) ***Log Analysis***

- Map: Extract user IP
- Reduce: Count visits per IP

iii) ***Sorting Large Datasets***

- Map: Key = value to sort
- Reduce: Final sorted result

5. Advantages

- Highly scalable
- Fault-tolerant
- Can process terabytes of data
- Works on commodity hardware

b) Write a short note on: [8]
i) Evaluation web search **ii) Web Crawlers**

i) Evaluation of Web Search

Evaluating a web search engine means measuring how accurately, efficiently, and satisfactorily it retrieves information for users.
It uses both user-centered and system-centered methods.

1. ***Effectiveness Measures***

These measure the *quality* of retrieval.

- Precision & Recall – how relevant the returned results are
- NDCG (Normalized Discounted Cumulative Gain) – rewards relevant results at top ranks
- MAP (Mean Average Precision) – average relevance across queries
- MRR (Mean Reciprocal Rank) – measures how quickly first relevant answer appears

2. ***User Behavior Signals***

Modern web search uses actual user behavior:

- Click-through rate (CTR)
- Dwell time (how long user stays on result page)
- Bounce rate
- Query reformulations (indicates dissatisfaction)

*3. **Online Evaluation – A/B Testing***

Two ranking algorithms (A and B) shown to different users → measure:

- Clicks
- Engagement
- User satisfaction

Used by Google, Bing.

*4. **Query Logs***

Large-scale logs help analyze:

- Frequent queries
- User intent
- Popular pages

Conclusion: Web search evaluation combines traditional metrics, behavioral signals, and live experiments to continuously improve search quality.

ii) Web Crawlers – Short Note

A Web Crawler (also called spider or bot) is a software program that automatically browses the web and collects web pages for indexing.

*1. **How a Web Crawler Works***

1. Start with seed URLs
2. Fetch the page using HTTP request
3. Parse HTML to extract text and links
4. Store page content in search engine index
5. Add new links to crawl queue
6. Repeat for next pages

*2. **Key Features***

- Politeness: Respect robots.txt rules
- Scheduling: Decide which URL to fetch next

- URL normalization & deduplication
- Freshness: Revisit pages that change frequently
- Scalability: Distributed crawling across many servers

3. ***Types of Crawlers***

- Focused Crawlers: Collect pages about a topic (e.g., medical content)
- Incremental Crawlers: Keep index updated by detecting changed pages
- Distributed Crawlers: Run across clusters (used by Google)

4. ***Uses***

- Building search engine indexes
- Data mining and analytics
- Web archiving (e.g., Wayback Machine)
- Price comparison and monitoring

Conclusion: Web crawlers are essential for acquiring and maintaining up-to-date web content, enabling fast and relevant search results.

► **MAY/JUNE 2024**

Q7) a) Describe Parallel Information Retrieval in detail. [8]

b) Write a short note on: [9]
i) The structure of the web → DONE ii) Queries and Users → DONE iii) Static ranking → DONE

Q8) a) Describe MapReduce with suitable examples. [9]

b) Write a short note on: [8]
i) Beautiful Soup **ii) Python Scrapy**
iii) Web crawler libraries **iv) Dynamic ranking**

i) Beautiful Soup (3 marks)

Beautiful Soup is a Python library used for web scraping. It helps extract data from HTML and XML documents.

Key Features

- Parses HTML easily, even if the code is broken or messy
- Provides simple functions like .find(), .find_all(), .select()
- Supports navigation of page structure (tags, attributes, text)
- Works with parsers like html.parser, lxml, html5lib

Uses

- Extracting headlines, links, tables, product info
- Cleaning and formatting scraped content

Example: Extracting all links from a webpage.

ii) Python Scrapy (3 marks)
Scrapy is a powerful and fast web crawling and scraping framework in Python, used for large-scale scraping tasks.

Key Features

- Built-in support for crawling, parsing, item pipelines
- Handles asynchronous downloading → very fast
- Provides spider classes to define crawling logic
- Automatic handling of request scheduling and throttling
- Can export data to JSON, CSV, databases

Uses

- Building crawlers for e-commerce sites, news websites, job portals
- Data mining and large-scale scraping projects

Strength: Much faster and more scalable than BeautifulSoup for big crawling tasks.

iii) Web Crawler Libraries (3 marks)
Web crawler libraries are tools used to automatically fetch, parse, and store web content.

Common Libraries

- **Scrapy (Python)**: Full crawling framework for large sites
- **BeautifulSoup (Python)**: HTML parsing (not a crawler but used with requests)
- **Requests (Python)**: HTTP fetching library
- **Selenium**: Automates browsers for dynamic pages (JavaScript-heavy)
- **Heritrix (Java)**: Official Internet Archive crawler
- **Nutch (Java)**: Apache scalable web crawler

Purpose

- Download webpages
- Extract links and data
- Maintain URL queues & avoid duplicates

- Build search engine indexes

Enables structured and automated web data collection.

iv) Dynamic Ranking (3 marks)

Dynamic ranking assigns query-dependent scores to documents at search time. It computes how relevant each document is for a specific query.

Factors Used

- TF-IDF score / BM25 relevance
- Query term frequency in the document
- Term positions and matches (proximity)
- Recency / freshness for time-sensitive queries
- User behavior (clicks, dwell time)

Characteristics

- Ranking changes with each query
- Personalized to user intent
- Combined with static ranking for final order

Example: Query: *"best smartphones 2024"*
Dynamic ranking promotes recent reviews and comparisons.

► **NOV/DEC 2024**

Q7) a) Describe Map reduce with suitable examples. [6]

MapReduce is a distributed programming model developed by Google for processing large datasets across multiple machines. It consists of two main phases: Map and Reduce.

1. Map Phase

- Input is split into chunks.
- Each mapper processes a chunk and outputs (key, value) pairs.

Example: Word count
Mapper outputs:

(the,1), (sun,1), (the,1)

2. Shuffle Phase

- Framework groups all values with the same key.
  Example:

the → [1,1]

sun → [1]

3. Reduce Phase

- Reducer aggregates values for each key.
  Final output:

the → 2

sun → 1

Real-World Examples

- **Building inverted index**: Map → (term, docID), Reduce → merge list of docIDs
- **Log analysis**: Count user visits per IP
- **Sorting large files**: Keys used for sorting in parallel

Advantages

- Highly scalable
- Fault-tolerant
- Works on commodity hardware
- Ideal for big data (web, logs, social media)

b) Write a short note on: [6]
i) The structure of the web **ii) Python Scrapy**

i) The Structure of the Web (3 Marks)

The Web is a graph-based structure where:

- Nodes = web pages
- Edges = hyperlinks connecting pages

Characteristics

- Follows power-law distribution: few pages have many links
- Has clusters/communities of related pages
- Shows small-world properties (short path between pages)

Layers of the Web

- Surface Web: Indexed by search engines
- Deep Web: Behind forms/logins

- Dark Web: Encrypted, non-indexed

The graph structure supports crawling, indexing, PageRank, and link analysis.

ii) Python Scrapy (3 Marks)

Scrapy is a fast, high-level Python framework for building large-scale web crawlers.

Features

- Asynchronous downloads → extremely fast
- Built-in Spiders to define crawling logic
- Handles request scheduling, throttling, pipelines
- Exports data to JSON, CSV, XML
- Supports automatic link following and depth control

Uses

- Crawling e-commerce websites
- Extracting news articles
- Large web scraping projects
- Data mining and analytics

c) Describe web crawler with its components. [5]

A web crawler (spider/bot) is software that automatically downloads and processes web pages for indexing.

Components of a Web Crawler

1) **URL Frontier**

- Queue of URLs to visit
- Scheduling based on priority, domain limits, politeness rules

2) **Fetcher (Downloader):** Sends HTTP requests and retrieves page content

3) **Parser**

- Extracts text, metadata, and hyperlinks from HTML

4) **Link Extractor:** Identifies new URLs to add to frontier

5) **Duplicate Eliminator**

- Avoids re-crawling the same page
- Uses hash tables or Bloom filters

6) **Storage / Indexer**

- Stores downloaded pages
- Sends cleaned content to indexing module

Applications

- Search engines (Google, Bing)
- Price tracking websites
- Web archiving

Q8) a) Describe Parallel Query Processing with suitable examples. [6]

Parallel Query Processing executes query operations simultaneously across multiple machines or processors to reduce latency and handle huge datasets.

1. Document Partitioning: Documents split across servers.

Example: Query *"COVID effects"*

- Server A processes health docs
- Server B processes news
- Server C processes research papers
  Results merged → final list.

2. Term Partitioning: Index split by terms.

Example:

- Server 1 stores postings for "India"
- Server 2 stores "Cricket"
  Query "India cricket wins" processed in parallel.

3. Pipeline Parallelism: Different query stages run on different nodes (parsing → scoring → ranking).

Advantages

- Faster response
- High throughput
- Scalable to billions of documents
- Fault tolerant

b) Explain the following term: [6]
i) Static ranking **ii) Dynamic ranking**

i) Static Ranking (3 Marks)

Static ranking assigns a query-independent score to each document.

Examples

- PageRank based on web link structure
- Authority score based on credibility and domain
- Popularity based on traffic
- Freshness for newly updated pages

Used as a base rank before applying dynamic query scoring.

ii) Dynamic Ranking (3 Marks)

Dynamic ranking assigns query-dependent scores based on relevance.

Factors

- TF-IDF, BM25
- Term proximity
- Query-document similarity
- Recency for time-sensitive queries

Example: Query *"latest smartphones 2024"* → recent review pages rank higher dynamically.

c) Write a short note on Evaluation web search. [5]

Evaluating web search measures how effectively a search engine satisfies user needs.

1) Effectiveness Metrics

- Precision, Recall – relevance
- MAP, NDCG – reward relevant results at top ranks
- MRR – measures how quickly first relevant result appears

2) User-Centered Evaluation

- Click-through rate (CTR)
- Dwell time
- Scroll depth
- Query reformulation

3) A/B Testing

Two ranking algorithms tested on real users → compare engagement and click patterns.

4) Query Logs

Large logs analyzed for:

- Popular queries
- Seasonal trends
- User intent patterns

Conclusion: Evaluation ensures search engines improve relevance, speed, and user satisfaction continuously.

NOTE: Please verify all answers before referring.