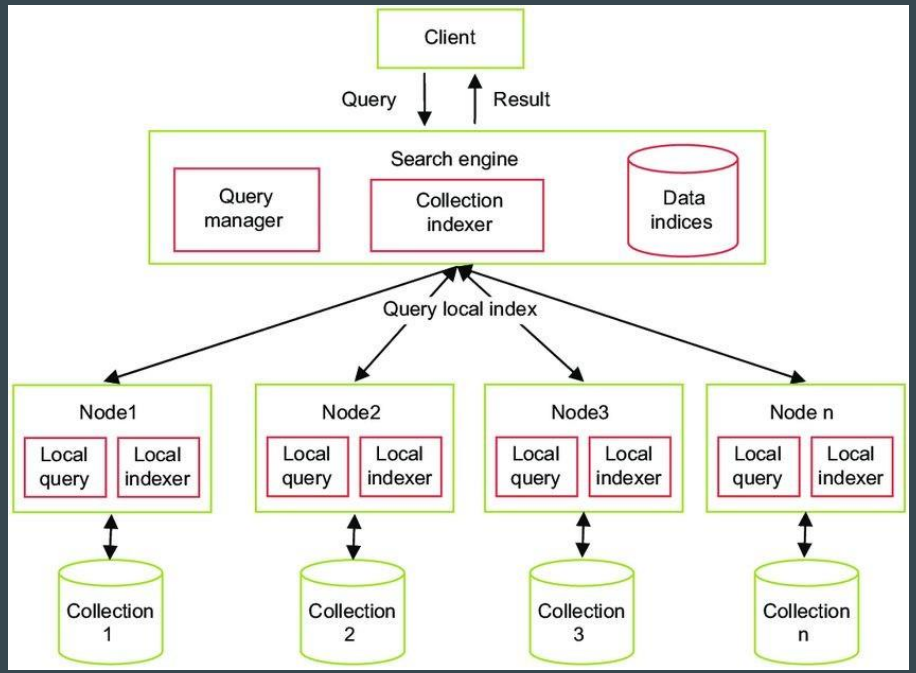
Query Processing in Distributed IR

...

What is Distributed IR?

- Allows organizations to efficiently search and retrieve information from multiple sources for information in distributed, heterogeneous, and possibly decentralized environments, including databases, web pages, and multimedia repositories.
- Improve scalability and fault tolerance of information retrieval systems by distributing the processing load across multiple machines and minimizing the impact of failures or outages in any one machine.
- Enable efficient data sharing, collaboration, and knowledge discovery across multiple organizations, which is critical for data-driven decision-making and innovation.



Query Processing in Centralized IR

Steps:

- Tokenization
- Stopword Removal
- Stemming
- Inverted Indexing
- Term Weighting

Benefits:

- Faster Query Response Times
- Simpler Architecture
- Easier Maintenance
- Better Resource Utilization
- Term Weighting

Then why need Distributed IR?

- Scalability: Distributed IR system can be designed to handle load by distributing the data and processing across multiple nodes.
- Fault Tolerance: A distributed system can continue to function even if some nodes fail
- Geographic Distribution: A distributed IR system can be more easily distributed geographically, which can reduce latency and improve response times for users in different regions.
- Heterogeneous Data Sources: Allows organizations to index and search across a wide variety of data sources.
- Privacy and Security: A distributed IR system can provide better privacy and security by allowing data to be stored locally on nodes and limiting access to certain nodes or users. Eg. Government of India restricting the storage of UPI transaction data in data centres located within Indian territory.

Query Processing: Distributed Indexing

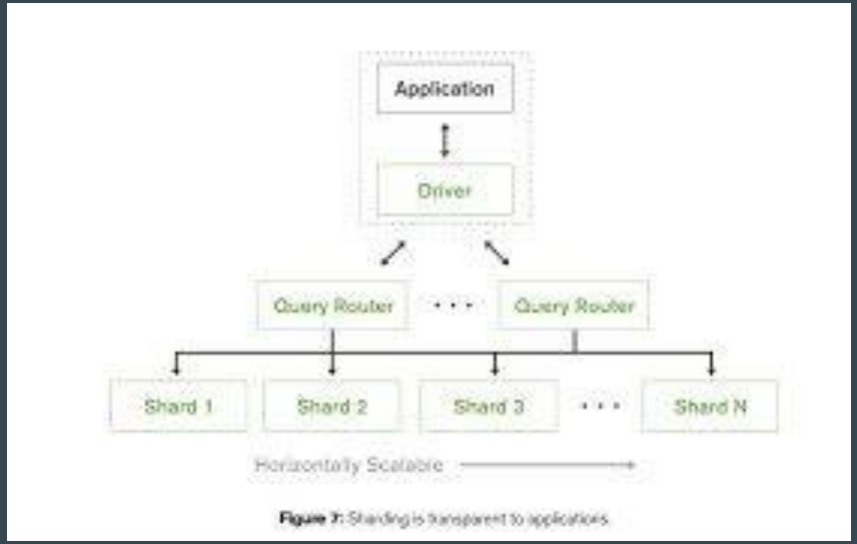
Distributed indexing distributes the indexing process across multiple machines or servers, allowing for greater scalability and faster indexing times. Each machine or server is responsible for indexing a subset of the documents in the collection, which are then combined into a single index.

The process of distributed indexing involves several steps:

- Partitioning: The document collection is partitioned into smaller subsets, or shards, which are distributed across multiple machines or servers.
- Indexing: Each machine or server indexes its assigned subset of documents.
- Merging: The indexes from each machine or server are merged into a single index.
- Querying: Queries are distributed across the machines or servers, which search their portion of the index and return results to the user.

Query Processing: Distributed Query Routing

- A distributed search engine may use Distributed Query Routing to route a user's search query to only those nodes that are likely to contain relevant documents.
- This can be achieved by analyzing the query to identify the keywords and then routing the query to only those nodes that contain documents that match the keywords.
- By routing queries only to relevant nodes, the system can reduce the amount of network traffic and reduce the overall response time for users.



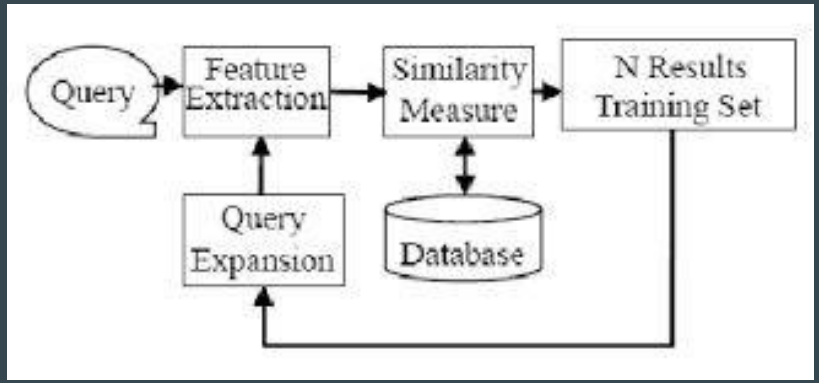
Query Processing: Query Expansion

Query expansion is a technique used in information retrieval to improve the effectiveness of a query by adding additional terms or concepts.

There are two main approaches to query expansion:

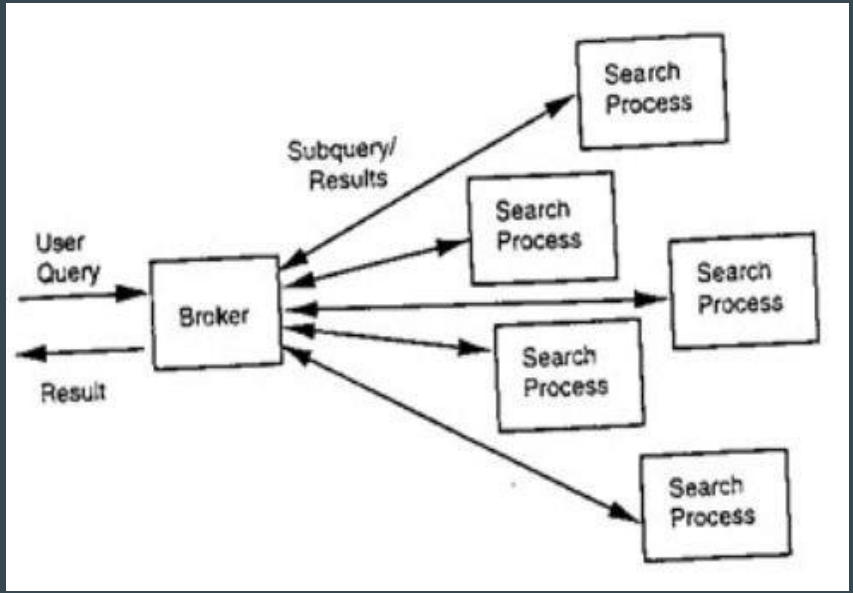
1. Local expansion: The technique of adding additional terms to the query based on the context of the original query. The added terms are usually synonyms, related terms or phrases, and can be used to improve the recall of the query.

2. Global expansion: The technique of using information from the entire document collection to add terms to the query. Involves identifying important concepts and terms in the collection and adding them to the query to improve the recall of the search.



Query Processing: Evaluation and Metrics

- Query processing performance can be evaluated by measuring the time taken to process a query and the amount of data transferred between computers in a network.
- The arrangement of data transmissions and local data processing is known as a distribution strategy for a query.
- Two cost measures: response time and total time are used to judge the quality of a distribution strategy.



Evaluation and Metrics: Hypothetical Example

Question:

We have two tables, Table A and Table B. Table A has 1000 tuples and Table B has 100 tuples.
The query is to find all the tuples in Table A that match a certain condition. The condition is that the value of attribute X in Table A is equal to the value of attribute Y in Table B.
The size of attribute X is 4 bytes and the size of attribute Y is 4 bytes. The size of each tuple is 100 bytes. The size of the network message is 100 bytes. The network bandwidth is 1Gbps.
The network latency is 1ms. The processing time for each tuple is 1ms. The processing time for each message is 1ms. The processing time for each semi-join operation is 1ms.

Solution:

The amount of data transferred to execute the query using a semi-join operation is 1000 * 4 bytes = 4000 bytes.
The time taken to transfer the data is 4000 bytes /(1 Gbps /8) = 32 microseconds.
The time taken to process the data is 1000 * 1 ms = 1 second.
The time taken to process the message is 1 ms.
The time taken to perform the semi-join operation is 1ms. The total time taken to execute the query is 1second + 1 ms + 1ms + 32 microseconds = 1.032032 seconds.