EC7207 - High Performance Computing HPC - Project Proposal

EG/2020/3990

Jayasooriya L.P.M.

Project Title: High-Performance Parallel Search Engine

Problem Statement

Traditional search engines face performance limitations when processing large-scale datasets on single-threaded systems. To scale efficiently, this project proposes parallel implementations for the search engine pipeline using HPC techniques.

Project Overview

Develop a scalable and efficient search engine capable of processing large document collections using advanced parallel computing techniques. The project involves implementing core search engine components, a **crawler, a tokenizer, an inverted index builder, and a query processor, with sophisticated features such as TF-IDF and BM25 ranking, stop-word removal, stemming, and synonym support**. The implementation will be realized in three parallel computing paradigms.

1. **OpenMP** – for shared-memory parallelism on multicore systems
2. **MPI** – for distributed-memory parallelism across multiple nodes
3. **Hybrid (OpenMP + MPI)** – combining intra-node multithreading and inter-node distribution

Project Components

- Web Crawler (fetches real-world web content or local dataset)
- Parser & Tokenizer (handles raw text, HTML parsing, etc.)
- Stop-word Removal, Stemming, Synonym Expansion
- Inverted Index Construction
- Query Processor using TF-IDF and BM25 Ranking and stop-word removal, stemming, and synonym support
- Output: Ranked list of top-k relevant documents

Approach & Methodology

1. OpenMP Version

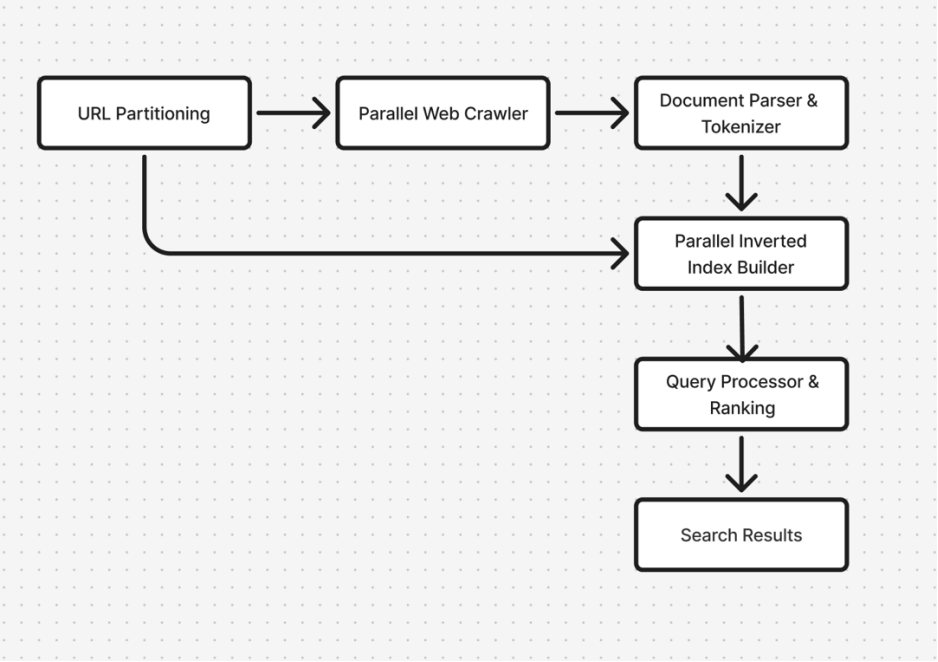
- Use **multithreading** for:
  - Parallel document reading & parsing
  - Tokenization and preprocessing
  - Building an inverted index using thread-safe structures
  - Parallel query processing & scoring
- Implement on multicore systems with shared memory

2. MPI Version

- Use a **distributed-memory model**
  - Distribute documents among MPI processes
  - Each process builds a **partial inverted index**
  - MPI collective communication (MPI_Gather, MPI_Bcast, etc.) for index merging and query distribution
- Benefits: Scales across clusters or multi-node systems

3. Hybrid (MPI + OpenMP)

- **MPI** handles inter-node communication and workload division
- **OpenMP** parallelizes within each node (across cores)
- Optimizations:
  - Overlap communication & computation
  - Minimize synchronization overhead
- Best performance and scalability for heterogeneous HPC systems



Evaluation Metrics

Metric	Description
Indexing Time	Time to parse, process, and build index
Query Latency	Time to return ranked results
Speedup	Compared to serial version
Scalability	Performance trend with increasing dataset/cores
Memory Usage	RAM required for each version