Distributed System for Calculating N-grams

Group Number 16: Arjit Yadav, Prateek Chaturvedi, Sashan Samarajeewa

1. Introduction

In the era of big data and natural language processing, the efficient analysis of textual data plays a pivotal role in extracting meaningful insights. This project aims to develop a distributed system for calculating n-grams in paragraphs, facilitating the parallel processing of large volumes of text across multiple nodes. The proposed system will be designed with a focus on dynamic host discovery, fault tolerance, voting mechanisms, and ordered reliable multicast to ensure scalability, reliability, and consistency in the distributed computing environment.

2. Architecture Model

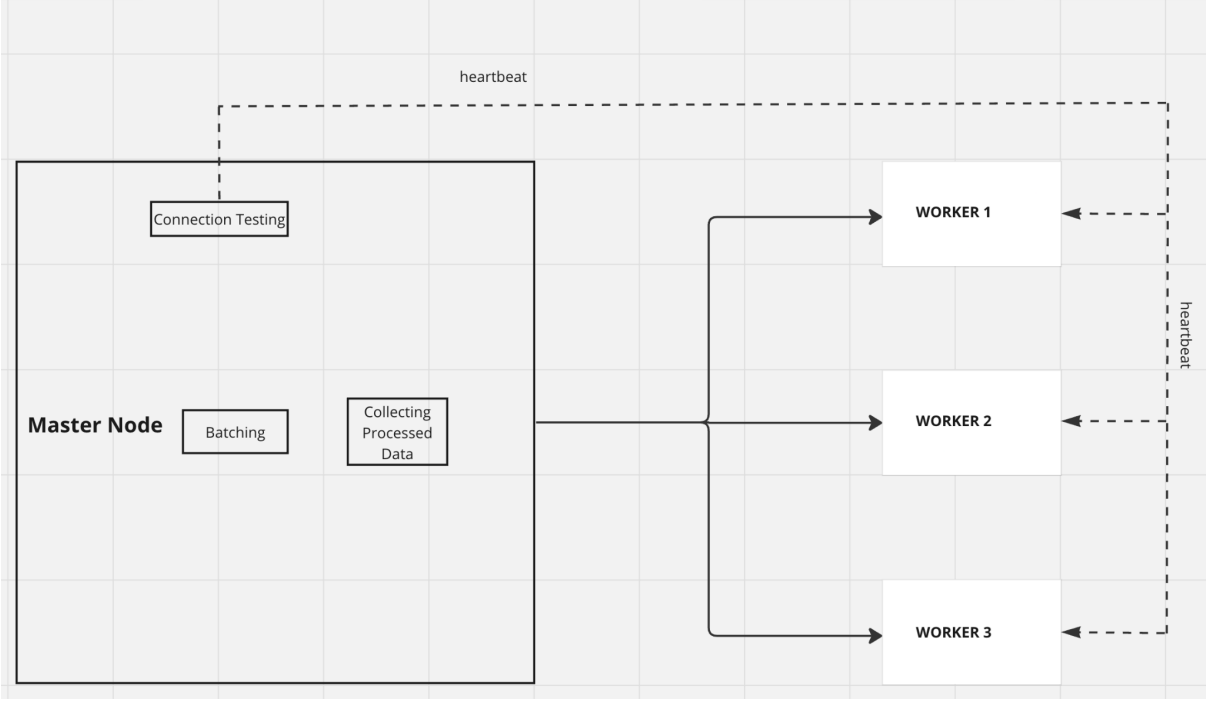

Fig. 1. System Architecture for Data Processing

The architecture of the proposed system is illustrated in Figure 1. The system consists of a central coordinator or master node and multiple worker nodes responsible for computing n-grams. The key components include:

2.1. Master Node

- Responsible for overall coordination and management of the distributed computation.
- Utilises the dynamic host discovery service to adapt to changes in the number of worker nodes.
- Implements a fault-tolerant task distribution mechanism.

2.2. Worker Nodes

- Receive tasks from the master node and compute n-grams for assigned paragraphs.
- Vote to indicate task completion and participate in the voting mechanism for decision-making.

3. Project Requirements

3.1 Dynamic Discovery of Hosts

For dynamic discovery of hosts, we will be using UDP broadcast listeners and senders. In our case, UDP will be more suitable for dynamic host discovery due to its natively built broadcasting capabilities, lower overhead, and simplicity in sending messages to multiple nodes without establishing individual connections. UDP offers lower overhead and lower latency compared to TCP since it does not involve establishing connections or ensuring reliability through acknowledgments and retransmissions.

3.2. Fault Tolerance

Use heartbeat signals to monitor the health of nodes. If a node stops sending heartbeats, it can be marked as failed, and its tasks can be reassigned to healthy nodes. This enabled us to Detect failures dynamically and redistribute tasks from failed nodes to available nodes. This dynamic redistribution prevents a single node failure from causing a bottleneck.

3.3. Voting

For our project, we have a leader node that will be tasked with dividing the data that is to be processed and sending it to worker nodes. It will also be tasked with synchronising, collecting, and compiling the results. Our leader node need not be a very high-performance machine and hence any node in the system can act as the leader node. For this reason, we have selected the Bully algorithm as it allows for a simple synchronous leader election technique.

3.4. Ordered Reliable Multicasting

Ordered Reliable Multicasting will be used to exchange coordination messages between master and worker nodes, ensuring ordered delivery for task assignment, status updates, and completion notifications. This enables delivering and synchronising final n-gram results across nodes to maintain a consistent and ordered collection of results.

4. Summary

In summary, the proposed distributed n-gram calculation system addresses the requirements of dynamic host discovery, fault tolerance, voting, and ordered reliable multicast. The system's architecture is designed to scale with varying computational resources, providing an efficient and fault-tolerant solution for processing large-scale text data.