COL380: Introduction to Parallel and Distributed Computing

MapReduce Implementation of Google's PageRank Algorithm

using MPI

Vasu Jain: 2017CS10387 , Shreya Sharma: 2017CS50493

March 15, 2020

Contents

Design Philosophy, Objectives and Workflow

We approached this assignment with the following objectives and design philosophy: The time measurements were done by using the `std::chrono::high_resolution_clock` These objectives were achieved to a large extent by continuous evolution of the code-base. The design philosophy caused changes across the objectives in tandem. However, the general design cycle was

Optimize Serial ⟶ Parallelize algorithm ⟶ Refactor Code ⟶ Optimize Serial ⟶ ...

Our code and scripts can be found in the repository at `https://github.com/jainvasu631/MPI-MapReduce-PageRa`

Data Structures and Serial Optimization

Data Structure

Initialization

Time Complexity Analysis

MPI - Message Parsing Interface

Message Passing Interface (MPI) is a communication protocol for parallel programming. MPI is specifically used to allow applications to run in parallel across a number of separate computers connected by a network. Message passing programs generally run the same code on multiple processors, which then communicate with one another via library calls which fall into a few general categories:

- Calls to initialize, manage, and terminate communications
- Calls to communicate between two individual processes (point-to-point)
- Calls to communicate among a group of processes (collective)

Types of Communications

Point-to-point Communication

- Blocking P2P Communication - A blocking call suspends execution of the process until the message buffer being sent/received is safe to use `(MPI_Send, MPI_Recv)`
- Non-blocking P2P Communication - A non-blocking call just initiates communication `(MPI_Isend, MPI_Irecv)`; the status of data transfer and the success of the communication must be verified later by the programmer (`MPI_Wait` or `MPI_Test`).

Collective Communication

Collective calls involve ALL processes within a communicator. There are 3 basic types of collective communications -

- Synchronization `(MPI_Barrier)`
- Data movement `(MPI_Bcast/Scatter/Gather/Allgather/Alltoall)`
- Collective computation (`MPI_Reduce/Allreduce/Scan`)

Parallelization of Algorithm

Embarrassingly Parallel For Loop

Matrix Multiplication

Checking Correctness

Differences among the 3 Versions of Parallel Algorithm

In terms of Principle

In terms of Implementation

Observations and Conclusions

Execution Time, Speedup and Efficiency

$$\text{Speed Up} = \frac{\text{Serial Execution Time}}{\text{Parallel Execution Time}}$$

$$\text{Efficiency} = \frac{\text{SpeedUp}}{\text{Number of Threads}}$$

Time Complexity Analysis

Observations and Explanations of MPI Graph Trends