Visualizing MPICH

Abhishek Gupta (18111001)

Pranjal Jain (18111050)

TOC

Overview

MPICH is a standard software used for message-passing for distributed-memory applications used in parallel computing. MPICH is Free and open source software.

MPICH provides many MPI calls like Reduce , Broadcast , Send , Gather , etc to provide different functionality in parallel implementation of a code.

This project is to show a visual representation of underlying message passing that happens among the nodes for various different function calls and on different topologies.

Understanding the problems

01 There are different physical topologies that exist in parallel programming.

02 It is difficult to visualize how MPI functions internally on different topologies and different calls

03 Each call takes different amount of time on different topologies as number of hops that a message takes in each step differs

Project objective

- Goal is to visualize 3 of the topologies namely

  1. Fat Tree 2. 2D Torus 3. Dragonfly

- Calculate number of hops for each step in the MPI function implementation
-
- Give a visual representation of different MPI calls on these topologies
-
- Calculate number of hops taken by a message in each step

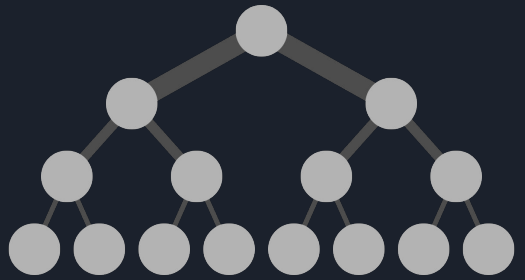
Target audience

- **Teachers**
-
- **Learners**
-
- **Researchers**

Topologies

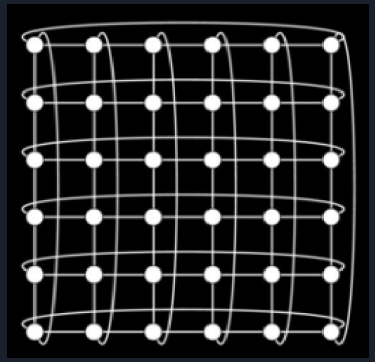
Fat Tree

Nodes are arranged as nodes of a Binary Tree , with more links in upper levels than lower



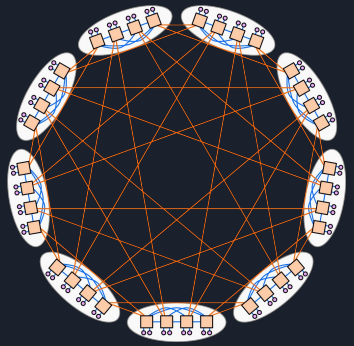
Torus

Torus is a mesh connection with extra links connecting the boundary nodes



Dragonfly

Nodes are grouped and arranged in a circular manner



Implemented MPI Functions

MPI_Send ( ) :

MPI_Bcast ( ) :

- Naive Algorithm
- Binomial Tree

MPI_Reduce ( ) :

- Recursive doubling
- Rabenseifner Algorithm

MPI_Allreduce ( ) :

- Rabenseifner Algorithm

MPI_Gather ( ) :

- Recursive Doubling

MPI_Allgather ( ) :

- Naive Algorithm
- Recursive Doubling

Project Modules

User Interface

Users gives the configuration of the topology and MPI function

Topologies

Graphical representation of the topologies in XY plane

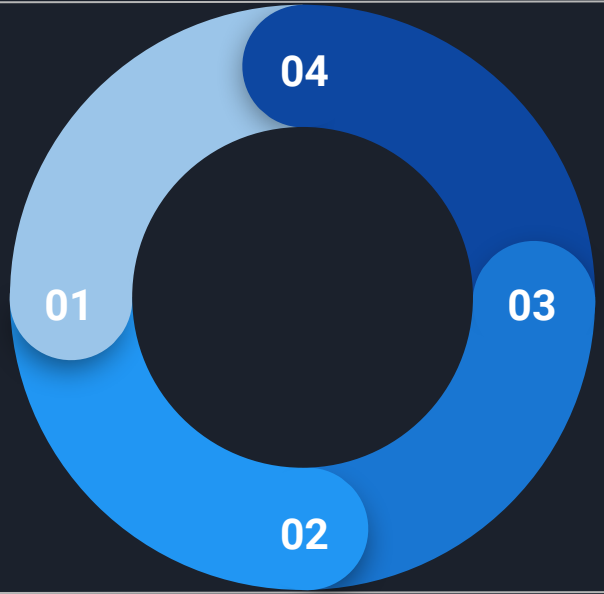


Image Generation

Snapshot of each step in the MPI call implementation

MPI Functions

Algorithms for different MPI calls like reduce , broadcast , gather , ets

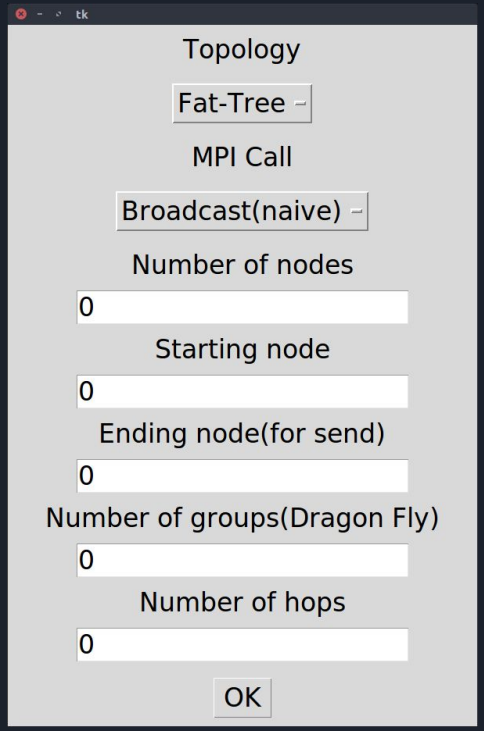
Technology Used

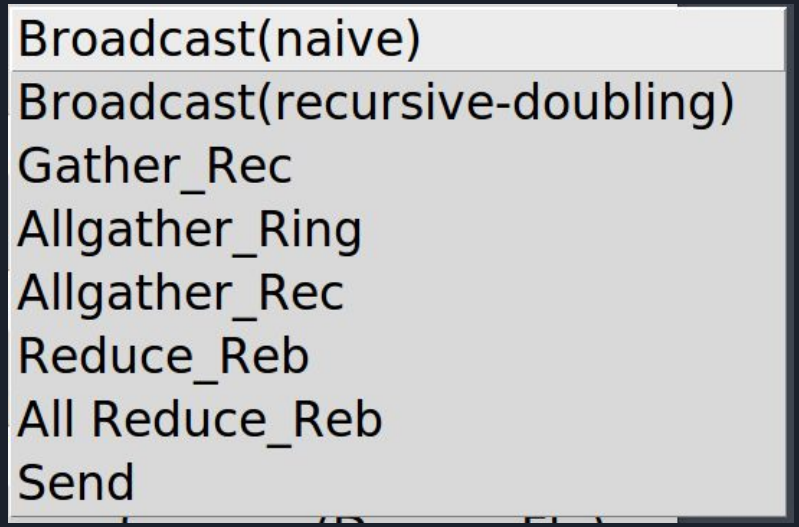
Python

Whole application is built upon python using different packages and 1 KLOC

- *Tkinter : For UI*
- *Matplotlib : To generate images*
- *OS : To execute some system commands*

Screenshots (1 of 3):



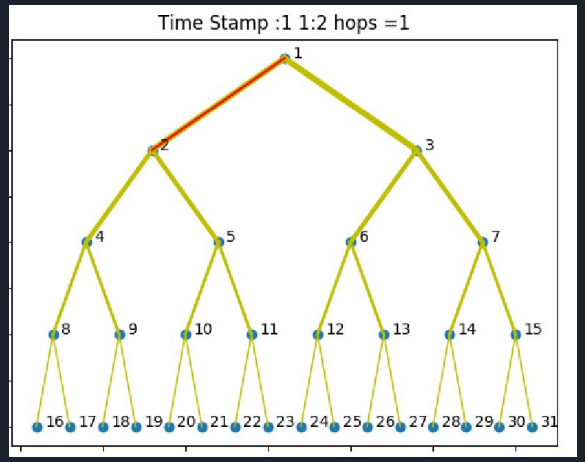


User Interface and MPI Function Calls

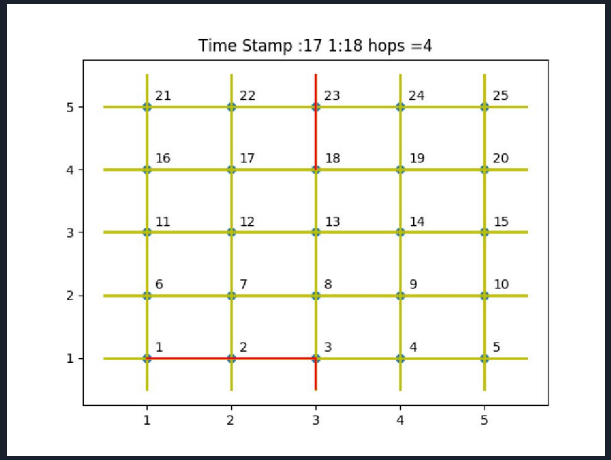
Screenshots (2 of 3):

Generated Images with Communcation , Timestamp and Hops Information

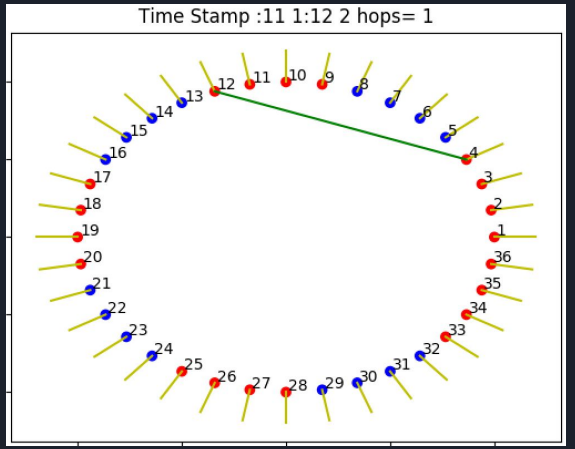
Screenshots (3 of 3):



Fat Tree Representation



Torus Representation



Dragonfly Representation

Contribution

Abhishek Gupta

- 2D Torus and Dragonfly Topology representation
- Send function for 2D Torus and Dragonfly
- Algorithms for Naive Bcast , Reduce , All Reduce , Gather , All Gather
- Reports and Presentation

Pranjal Jain

- Fat Tree topology representation
- Send Function for Fat Tree
- Algorithm for Recursive Doubling Bcast
- Gallery and image generation
- User Interface

Thank you!