

Aman Hogan-Bailey

University of Texas at Arlington

CSE-5351: Parallel Processing

## Contents

<b>Performance Analysis: MPI vs MPI+OpenMP Implementation of 3D Heat Equation .....</b>	<b>2</b>
<b>Runtime Performance .....</b>	<b>2</b>
<b>Scaling Behavior .....</b>	<b>2</b>
<b>Process Count Impact.....</b>	<b>3</b>
<b>Conclusions .....</b>	<b>3</b>
<b>Tables.....</b>	<b>4</b>

## Performance Analysis: MPI vs MPI+OpenMP Implementation of 3D Heat Equation

Since the runtime for N=400 always takes around 20-30 minutes on lower processor counts a lower N value was used. Here are the values of N respectively:

1. Pure MPI implementation with N=120
2. Hybrid MPI+OpenMP implementation with N=80

### Runtime Performance

- The hybrid implementation starts significantly faster, with 2 processes taking only 0.96s
- Pure MPI with single process takes 126.95s, indicating substantial sequential overhead

### Scaling Behavior

1. Pure MPI shows strong scaling:
  - a. 8.2x speedup from 1 to 8 processes
  - b. 11x speedup from 1 to 50 processes
2. Hybrid implementation shows performance degradation:
  - a. Runtime increases from 0.96s (2 processes) to 31.69s (50 processes)
  - b. Performance worsens by approximately 33x from 2 to 50 processes

## Process Count Impact

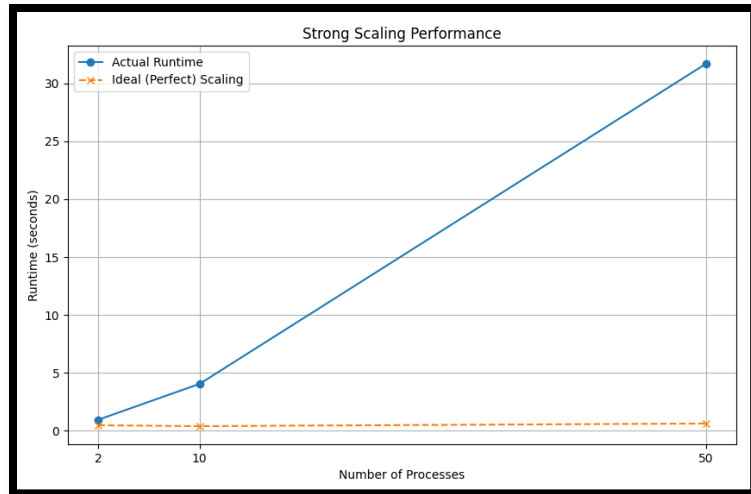
1. Pure MPI benefits from increased processes, though with diminishing returns
2. Hybrid approach shows inverse scaling, performing worse with more processes

## Conclusions

1. Implementation Preferences
  - a. For small process counts: Hybrid MPI+OpenMP is superior
  - b. For large process counts: Pure MPI shows better stability
  - c. The crossover point appears to be between 8-10 processes
2. Scaling Characteristics
  - a. Pure MPI shows expected scaling behavior with increased processes
  - b. Hybrid implementation's performance degradation suggests serious resource contention or synchronization issues
3. Problem Size Considerations
  - a. Hybrid implementation handles  $N=80$  more efficiently at low process counts
  - b. Pure MPI manages  $N=120$  with better scaling but higher absolute runtimes

## Figures

*Plots Runtime and efficiency for splitting and processor counts for the MPI+OMP combination.*



## Tables

### *Hybrid MPI+OpenMP Implementation (N=80)*

NProcs	Runtime (s)
2	0.960819
10	4.054080
50	31.688322

### *Pure MPI Implementation (N=120)*

NProcs	Runtime (s)
1	126.951661
8	15.463419
50	11.481844