COMP 6231 Assignment 3

Task 2: MPI Performance Analysis

Qiang Xue (40300671) Yicheng Cai (26396283) Yikai Chen (40302669)
Yifan Wu (40153584) Alexander Sutherland (40321783)

November 5, 2025

1 Overview

The MPI cluster consists of 11 containers (1 master + 10 workers) distributed across four physical computers. Four analysis scripts (q1_t3.py through q4_t3.py) were executed with varying container counts (4-10) using the Books_rating.csv dataset. Execution times were measured for each configuration to evaluate scalability.

2 Execution Time Results

Worker Containers	Q1 (s)	Q2 (s)	Q3 (s)	Q4 (s)
4	5.697	6.530	8.673	6.553
5	9.661	9.304	13.038	10.614
6	12.017	12.348	12.451	12.431
7	10.961	10.769	10.697	11.279
8	15.865	13.819	14.295	13.319
9	14.445	12.109	12.676	13.091
10	15.424	13.615	13.621	12.144

Table 1: Execution times for all four analysis scripts

3 Performance Analysis

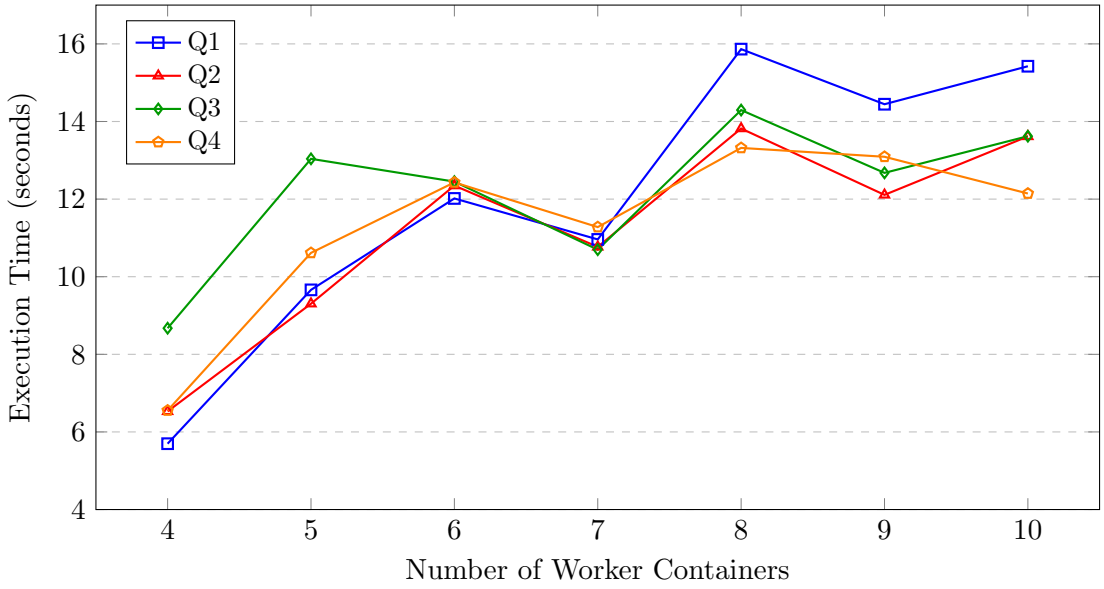

Figure 1: Execution time vs. number of containers

3.1 Key Observations

- **Optimal Performance at 4 Containers:** All four scripts achieved best performance with 4 containers (5.7-8.7 seconds). This configuration places exactly one worker container on each of the four physical computers, eliminating resource contention within individual machines and maximizing network bandwidth utilization per node.
- **Performance Degradation Beyond 4 Containers:** Increasing containers beyond 4 forces multiple containers to share resources on the same physical machines, causing CPU, memory, and network bandwidth contention. At 8 containers, execution times increased by 65-178% compared to the 4-container baseline due to intra-machine resource competition and increased inter-container communication overhead.
- **Resource Contention:** With more than 4 containers, multiple workers on the same physical machine compete for shared resources (CPU cores, memory bandwidth, network interface). This contention, combined with increased MPI communication overhead (scatter, gather, synchronization), severely degrades performance.
- **Performance Recovery at 7 Containers:** Interestingly, 7 containers show better performance (10.7-11.3 seconds) compared to 6 and 8 containers, despite having more workers. This suggests that the specific container-to-machine distribution at this configuration achieves better load balancing across physical machines, reducing some resource contention effects.

4 Conclusions and Lessons Learned

4.1 Key Findings

- **Optimal Configuration:** 4 containers (one per physical computer) provide the best performance by eliminating intra-machine resource contention while maintaining efficient inter-machine

parallelization.

- **Physical Distribution Impact:** Performance degrades when multiple containers share the same physical machine due to CPU, memory, and network bandwidth competition. The 4-container configuration avoids this contention entirely.
- **Scalability Limitation:** Beyond 4 containers, the system suffers from both intra-machine resource contention and increased MPI communication overhead, causing performance to degrade rather than improve.
- **Communication vs. Resource Bottleneck:** The primary bottleneck is not just MPI communication overhead, but the combination of resource contention on physical machines hosting multiple containers and the resulting network congestion.

4.2 Lessons Learned

- **Container Distribution:** Always distribute containers to maximize one-container-per-physical-machine deployment. For this 4-machine cluster, use exactly 4 containers for optimal performance.
- **Scaling Strategy:** To achieve better scalability beyond 4 containers, add more physical machines rather than deploying multiple containers on existing machines. Each additional physical computer can host one more container without resource contention.
- **Resource Isolation:** If multiple containers per machine are necessary, ensure adequate CPU cores, memory, and dedicated network bandwidth to minimize contention. Consider using Docker resource limits and CPU pinning.