**Distributed Merge Sort Final Metrics Summary**

**Project Summary**

* Successfully implemented and benchmarked multiple merge sort variants using MPI and OpenMP.
* Analyzed performance across dimensions from 4 to 32K elements.

**Key Implementations**

|  |  |  |
| --- | --- | --- |
| **Algorithm** | **Parallelism Model** | **Notes** |
| serial | Sequential | Baseline |
| task0 | MPI + Sequential Merge | Scatter/Gather with root merging |
| mergepath | MPI + Sequential Merge Path | Improved merge function after gather |
| mergepath\_omp | MPI + OpenMP Merge Path | Best shared-memory speedup |
| distributed | Fully Distributed MPI | Binary tree distributed merge |

**Speedup vs Serial**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Dimension** | **task0** | **mergepath** | **mergepath\_omp** | **distributed** |
| 512 | 4.17× | 3.11× | 3.83× | 5.78× |
| 1024 | 5.66× | 4.27× | 4.32× | 6.64× |
| 2048 | 7.63× | 5.42× | 5.26× | 3.50× |
| 4096 | 4.43× | 3.86× | 3.33× | 3.49× |
| 8192 | 6.52× | 4.98× | 4.91× | 7.72× |
| 16384 | 6.50× | 5.21× | 5.42× | 6.80× |
| 32768 | 7.52× | 5.57× | 4.94× | 7.58× |

*(selected dims for compact view — full table available)*

**Main takeaway:**

* I implemented a distributed merge sort on top of MPI, debugging buffer overflows, ownership transfers, MPI pairing logic, and finally designed a fully scalable binary tree reduction pattern with dynamic resizing.
* Shared memory (mergepath\_omp) dominates for small/medium sizes.
* Distributed MPI version is fully functional & scales better at larger problem sizes >=million.
* Distributed merge initially failed on some runs due to ranks not properly synchronized between iterations.
* Memory resizing is critical in any distributed or dynamic merging algorithm.
* Send/receive sizes must always match exactly.
* In distributed algorithms, always track how many elements each rank owns after each communication round.
* unique\_ptr helps memory safety, but only if ownership transfers are carefully handled.
* MPI works once you get the message sizes, steps, and pairing rules right.