Data Science & Artificial Intelligence

High Performance Computing

HYBRID MPI/OPENMP MANDELBROT SET COMPUTATION

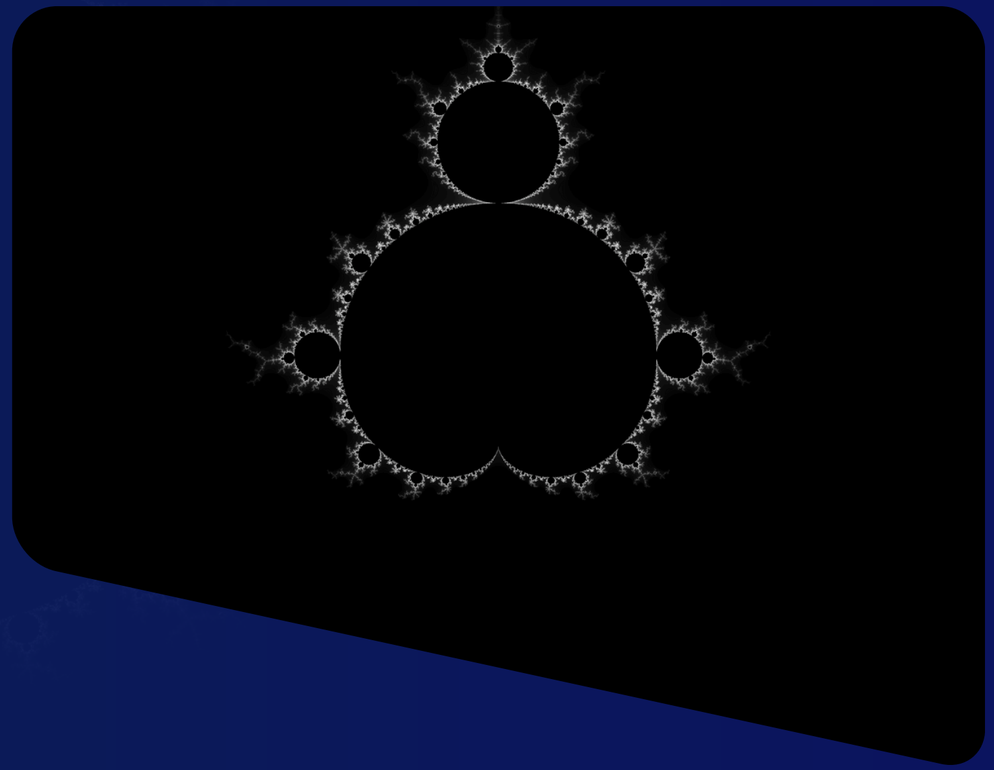
Pranay Narsipuram

SM3800006

July 2024

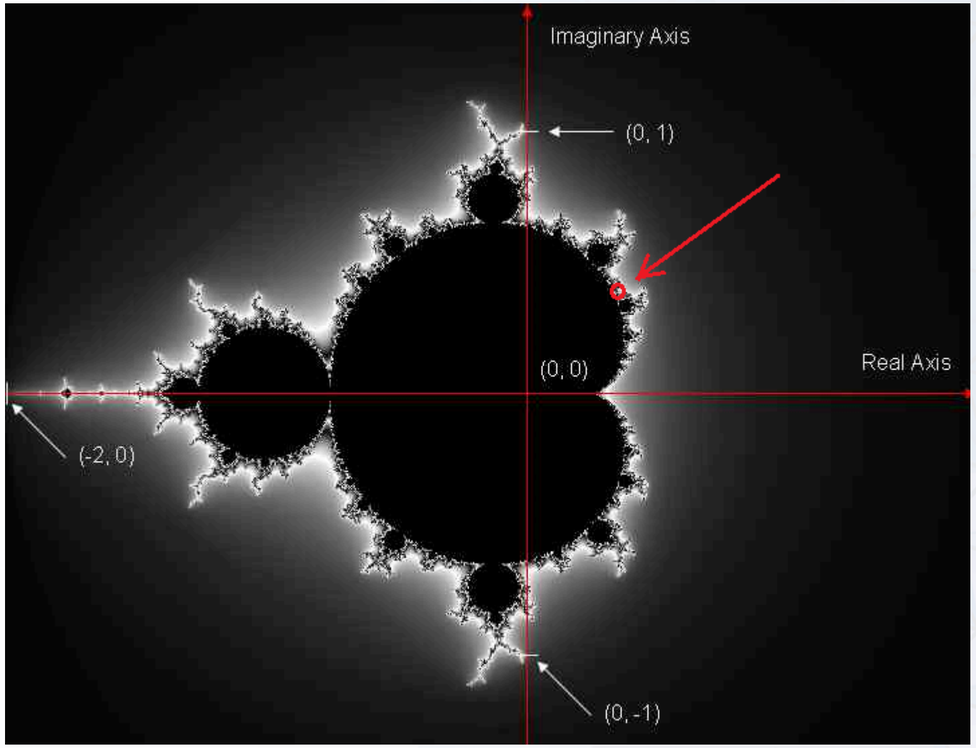
MANDELBROT SET

- A collection of complex numbers c for which the function $z_{n+1} = z_n^2 + c$ does not diverge when iterated from $z_0 = 0$.
- Each point on a 2D plane represents a complex number, where the iteration count before escaping a threshold is mapped to grayscale values.



Objective:

- Compute and visualize the Mandelbrot set.
- Utilize MPI for distributed memory parallelism.
- Utilize OpenMP for shared memory parallelism.
- Efficient computation on the ORFEO cluster.



PROBLEM ENCODING

GRID OF COMPLEX NUMBERS

- Each grid point represents a pixel.
- Iteration: $z_{n+1} = z_n^2 + c$ at each point.
- Grayscale value assignment based on iteration count.

SOURCE CODE STRUCTURE

- main.c: MPI initialization, distribution, timing.
- mandelbrot.c: Computes the Mandelbrot set with OpenMP.
- image_utils.c: Writes Mandelbrot set to a PGM image file.

MPI PARALLELIZATION

MPI Distribution:

- The 2D grid of complex numbers is divided among MPI processes.
- Minimizes communication overhead via sequential workload division.
- Gathers results and writes final image from root process.

OPENMP PARALLELIZATION

OpenMP Utilization:

- OpenMP parallelizes the loop iterating over grid points within each MPI process.
- Dynamic scheduling for balanced workload among threads, especially for regions with varying computational complexity.
- Handles varying computational complexities.

COMMAND LINE FLEXIBILITY

Arguments:

- Grid size: n_x, n_y
- Complex plane bounds: $x_{\min}, y_{\min}, x_{\max}, y_{\max}$
- Maximum iterations: $I_{\max}$
- Customization without code modification.

EXECUTION SCRIPTS

Strong Scaling:

- MPI: Fixed problem size, varying MPI tasks.
- OpenMP: Fixed problem size, varying OMP threads.

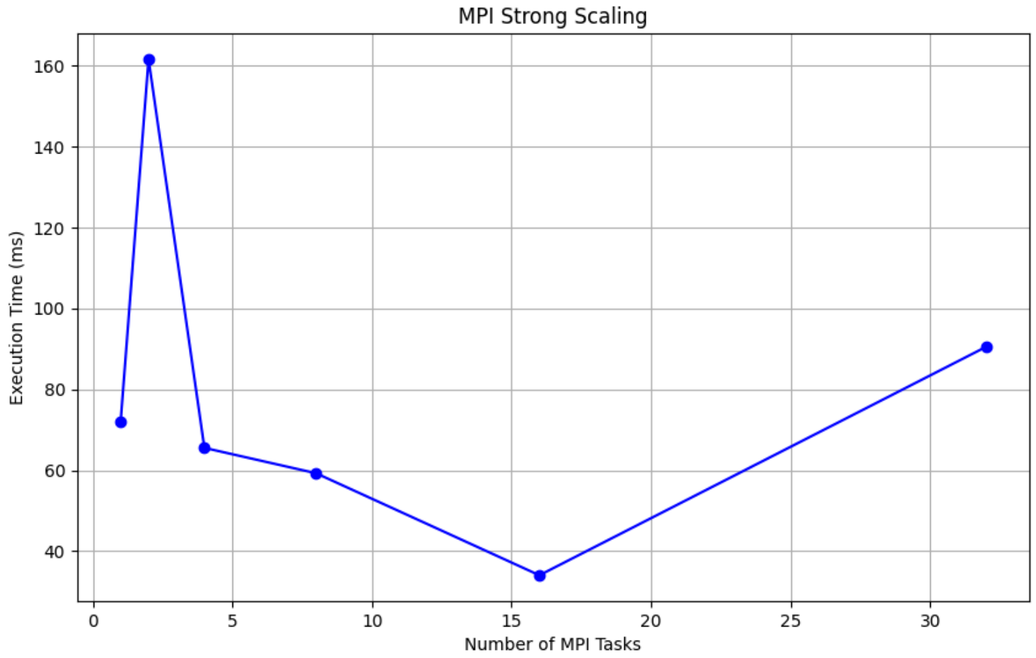
Weak Scaling:

- MPI: Increasing problem size with MPI tasks.
- OpenMP: Increasing problem size with OMP threads.

MPI STRONG SCALING

Performance Metrics

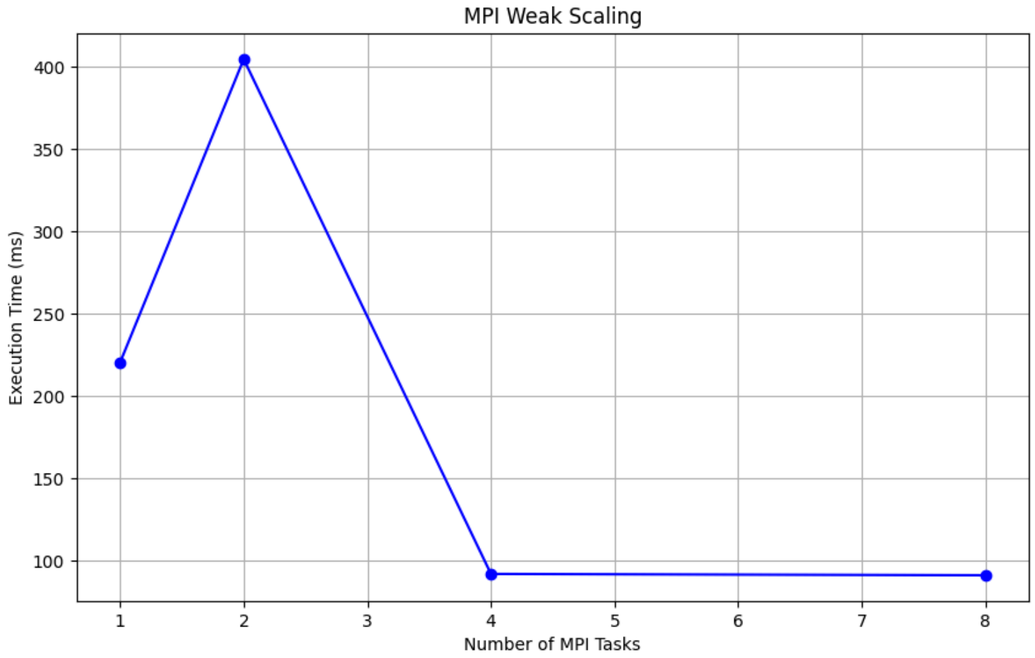
- Decrease: Execution time drops initially with more tasks.
- Increase: Slight rise in execution time after a minimum point.
- Conclusion: Good scalability initially, with communication overhead impacting at higher task counts.



MPI WEAK SCALING

Performance Metrics

- Initial Spike: Increase at 2 tasks, then stabilization.
- Stability: Execution time stabilizes as tasks increase.
- Conclusion: Maintains stable performance with larger problem sizes.

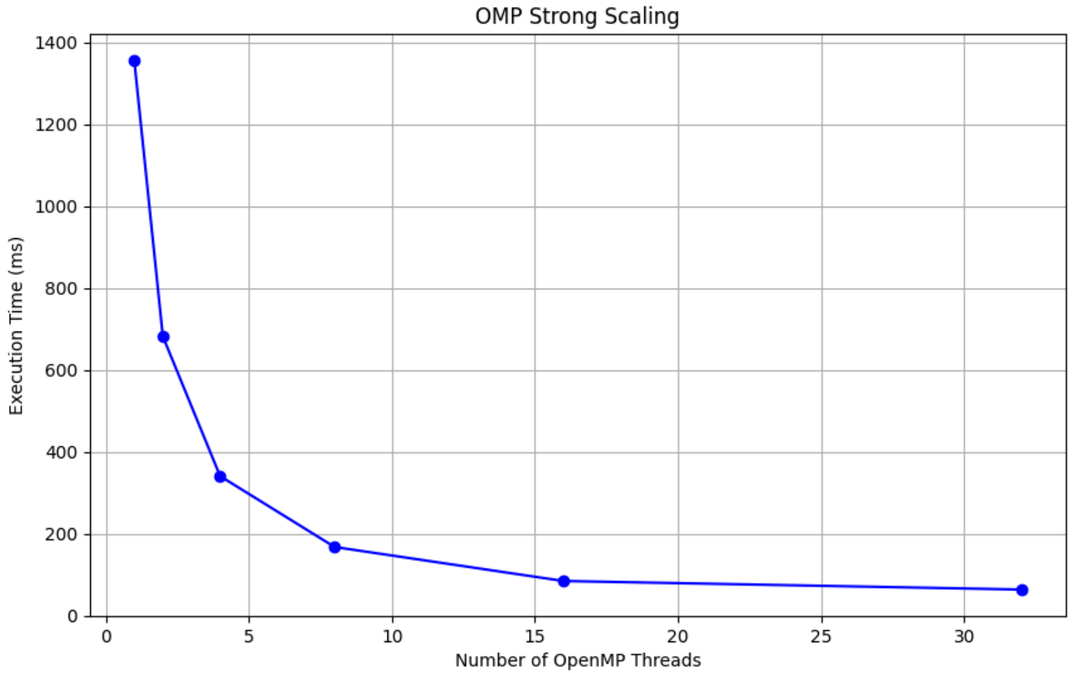


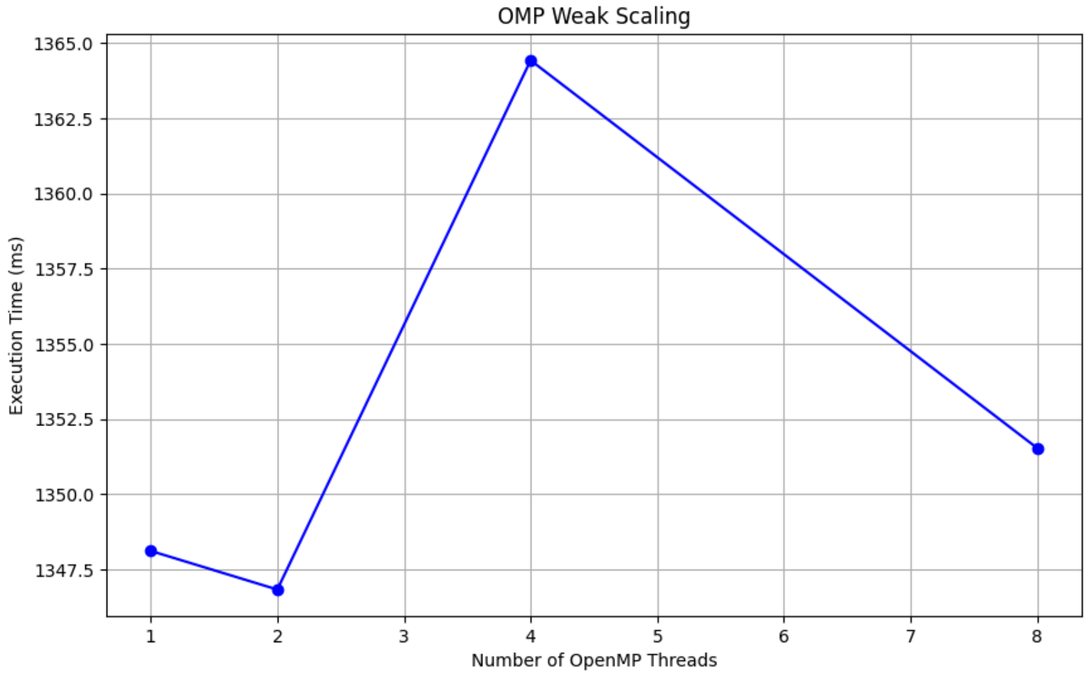
OMP STRONG SCALING

- Decrease: Significant drop in execution time with more threads.
- Plateau: Execution time stabilizes beyond a certain number of threads.
- Conclusion: Good scalability up to a certain point.

OMP WEAK SCALING

- Fluctuation: Initial increase at 2 threads, then steady decrease.
- Efficiency: Generally, execution time decreases with more threads.
- Conclusion: Handles increasing workloads effectively.





CONCLUSION

Hybrid Parallelization Effectiveness:

- MPI and OpenMP provided robust parallelization for Mandelbrot set computation.
- MPI distributed workload across nodes; OpenMP utilized multi-core processors.

Strong Scaling:

- OMP Strong Scaling:
  - Execution time decreased with more OpenMP threads.
- MPI Strong Scaling:
  - Execution time decreased with more MPI tasks; communication overhead at higher task counts.

Weak Scaling:

- OMP Weak Scaling:
  - Execution time stable with more threads; handled increased workloads.
- MPI Weak Scaling:
  - Execution time stable with more MPI tasks; good scalability with larger problem sizes.
  -

Overall Performance:

- Significant performance improvements on ORFEO cluster.
- Effective parallelization strategy with reduced execution time and efficient workload distribution.