# High Performance Computing
# Exercise 2

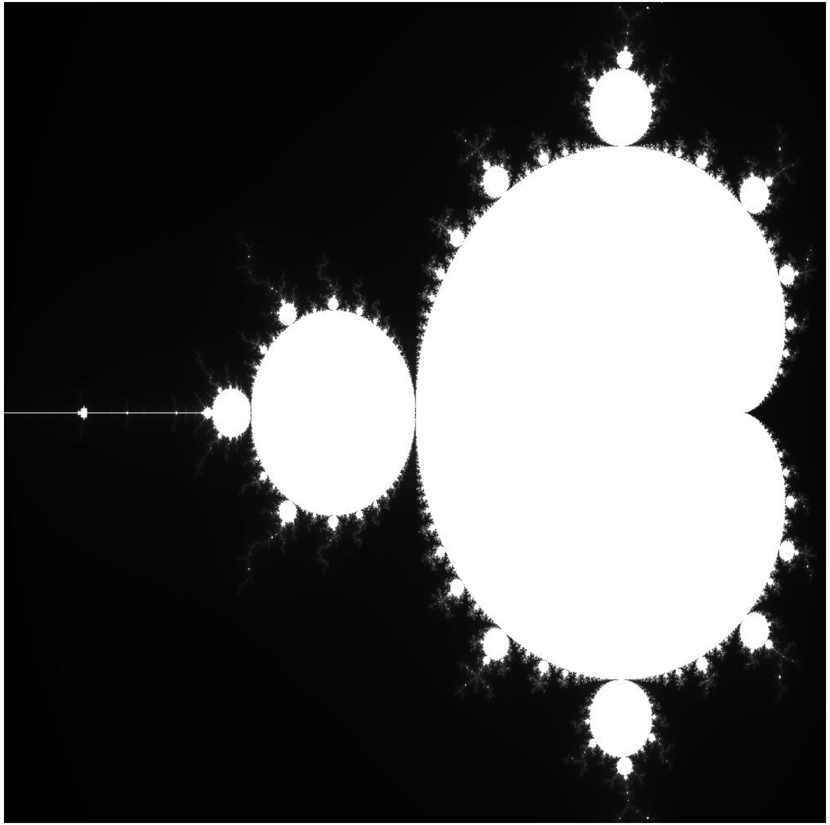
Luis Fernando Palacios Flores

# Introduction

Goals

Compute the Mandelbrot set M, in a region of the complex plane, in parallel using MPI for distributed memory and OpenMP for multi-threading.

Test speed-up for effective resources estimation.



$$\mathcal{M} = \{c \in \mathbb{C} | z_{n+1} = z_n^2 + c, \ \ |z_n| \leq 2, \ \ n \in \{0, 1, ..., N\}, \ \ z_0 = 0\}$$

# Methodology

MPI parallelism: static partitioning, as many blocks as processes (see Figure). **Load imbalance is expected**.

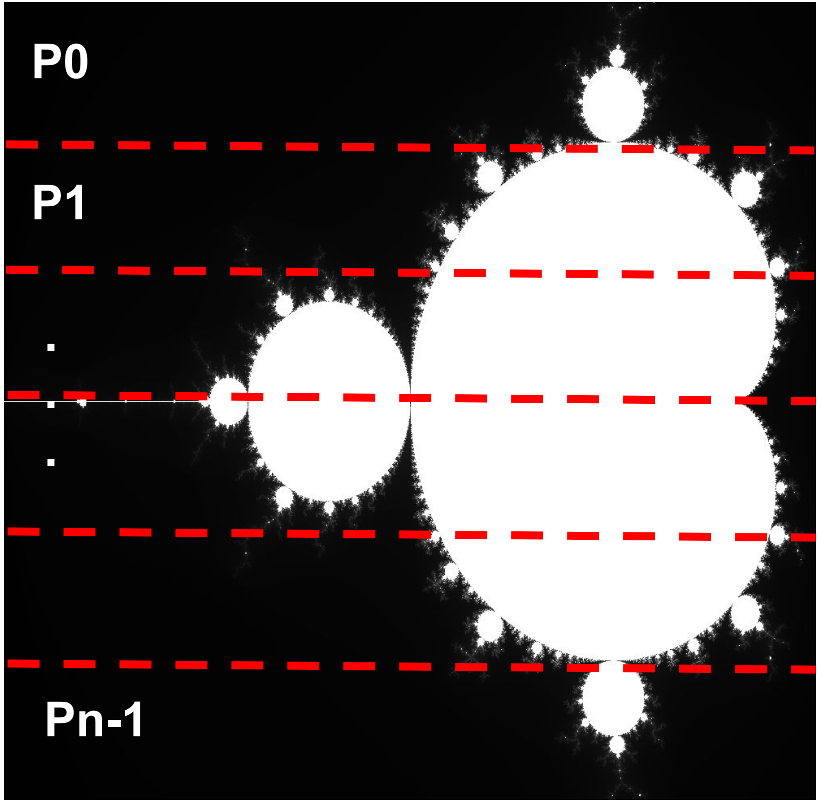
OpenMP parallelism: determine if the elements in R belong to M.

Strong scaling: 1000x1000

$$S(P) = \frac{t(1)}{t(P)}$$

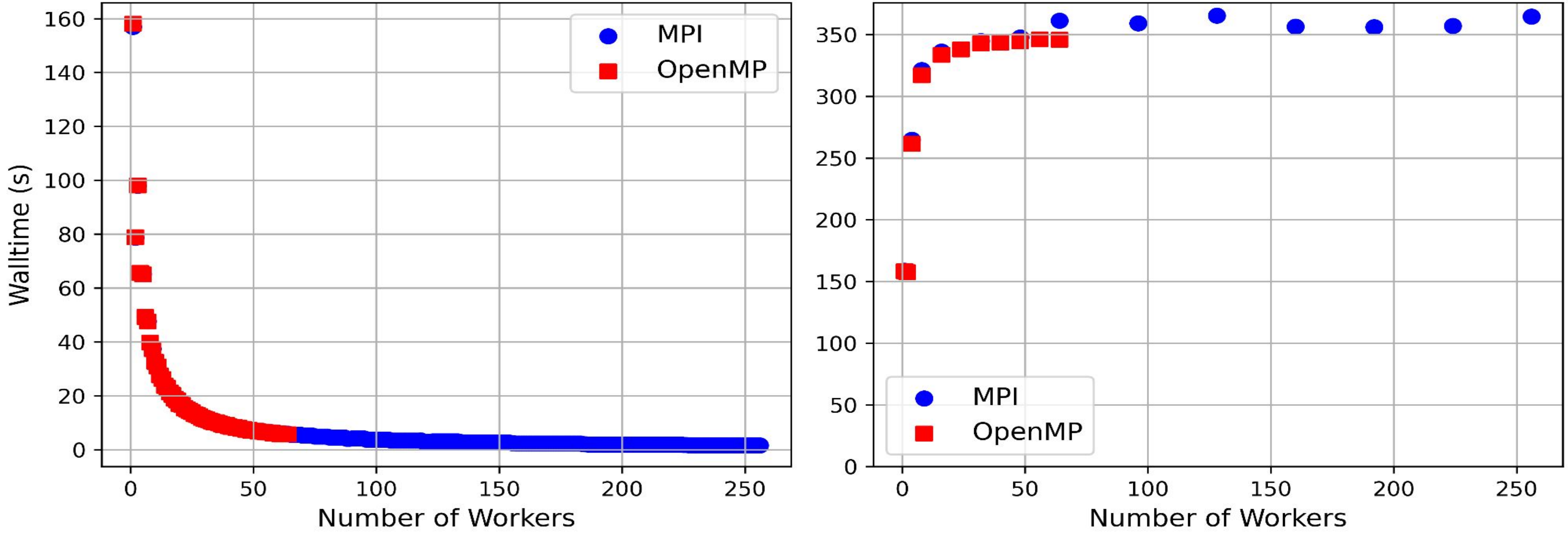
Weak scaling:  2MB per process/thread constant in square matrices, i.e., n=r=s

R = [-2, 0.5] x [-1, 1]




(rxs) pixels

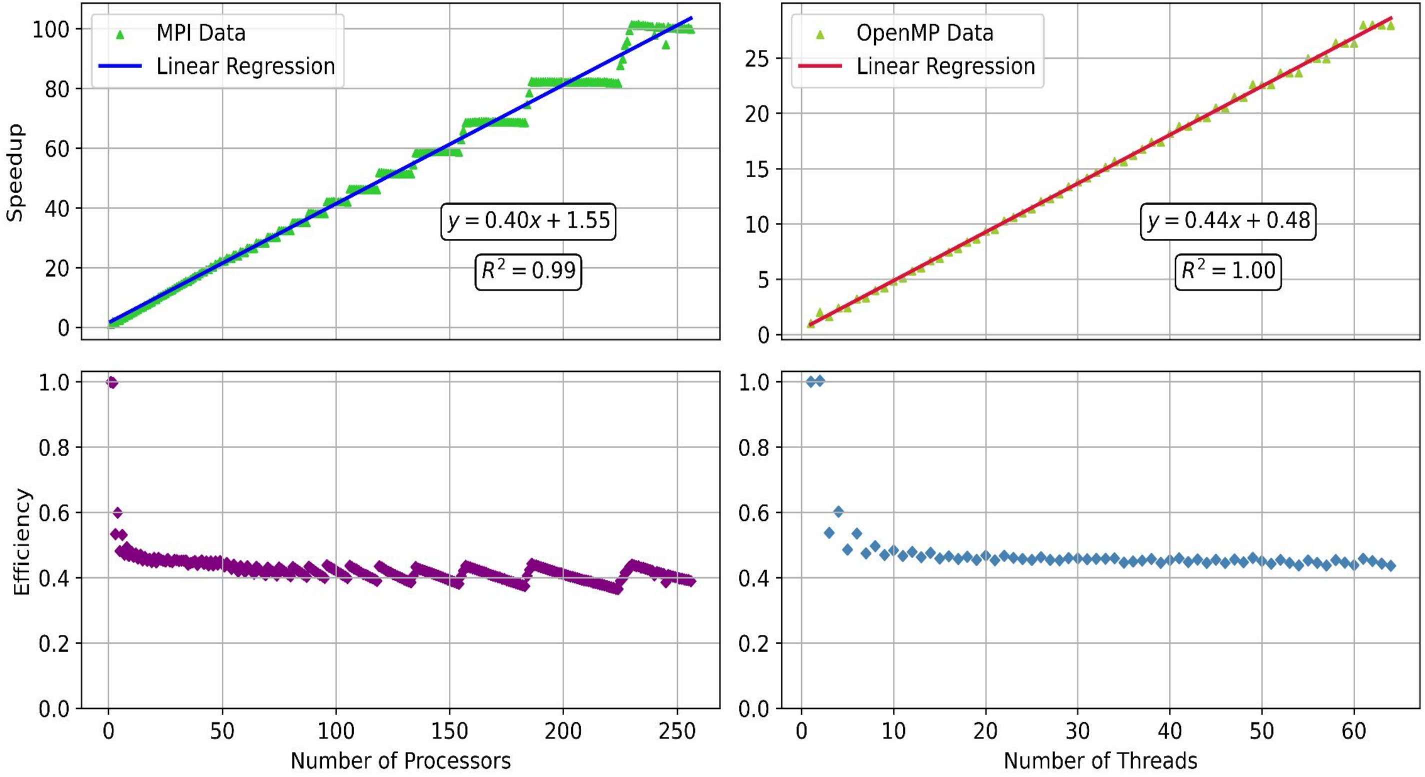
# Results and Discussion




scaling: 100x in MPI, 30x in OpenMP (expected inverse relation between t vs. W)

weak scaling: ~6 minutes stabilization (unexpected behavior)

# Results and Discussion




linear trend as expected

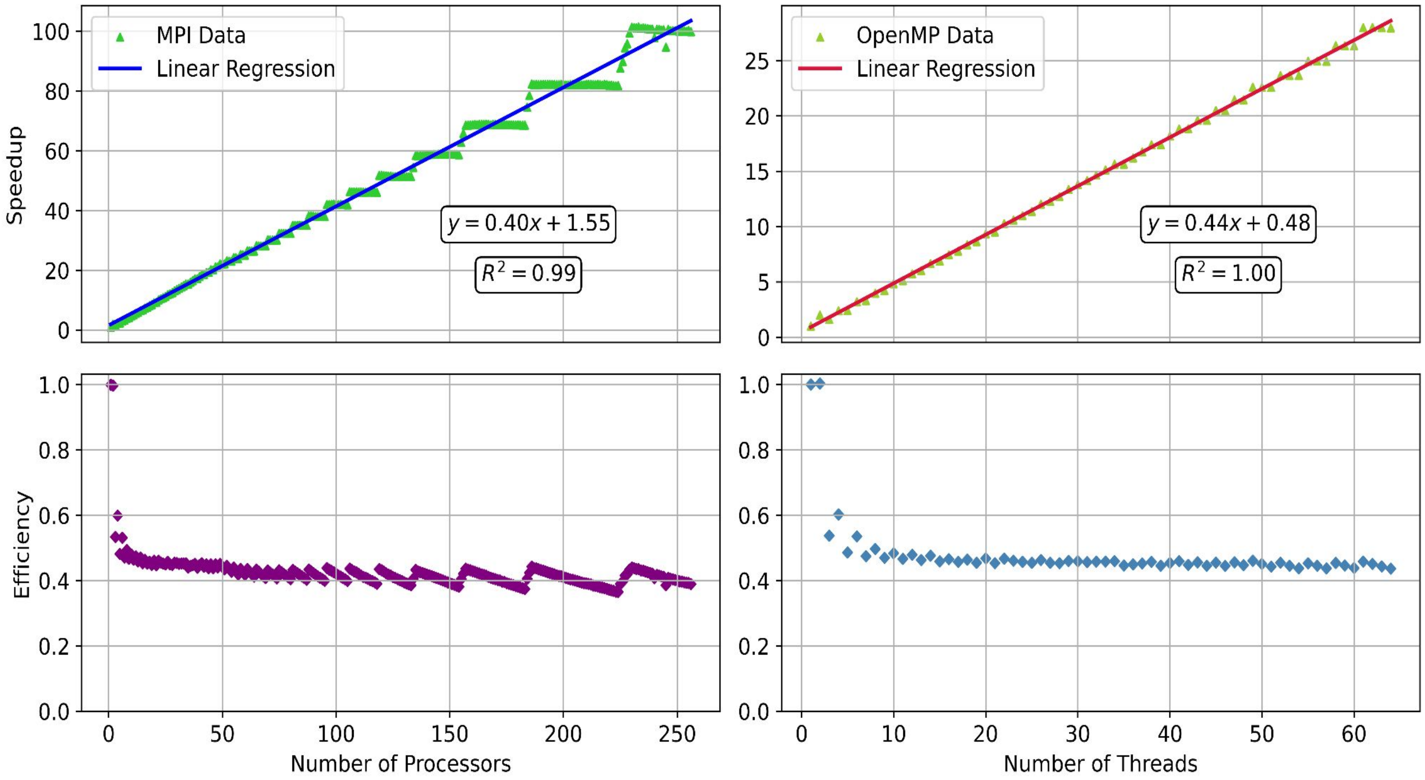
~60% deviation from ideal

>40% efficiency, cores in 1 socket might be sufficient

OpenMP mirrors MPI strategy (to be improved)

# Results and Discussion

Perplexing improvement in performance for OpenMP alone with dynamic scheduling.



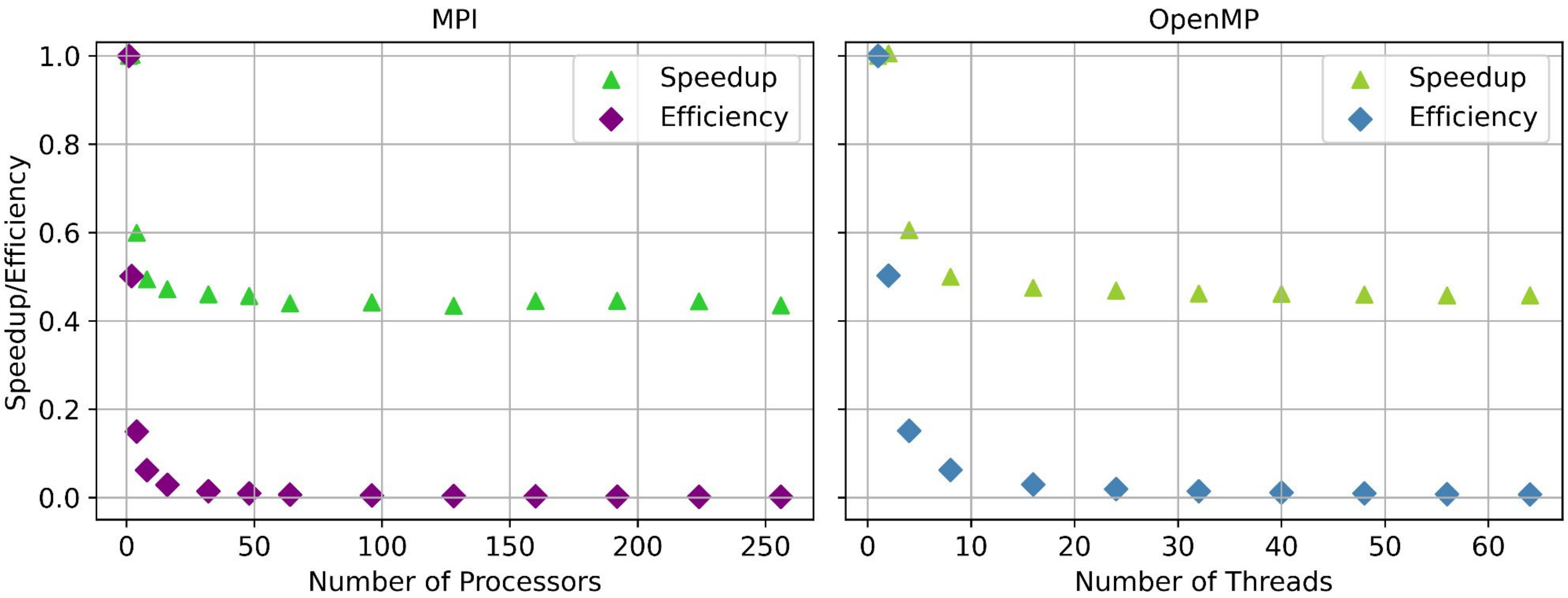

linear trend as expected

~60% deviation from ideal

>40% efficiency, cores in 1 socket might be sufficient

OpenMP mirrors MPI strategy (to be improved)

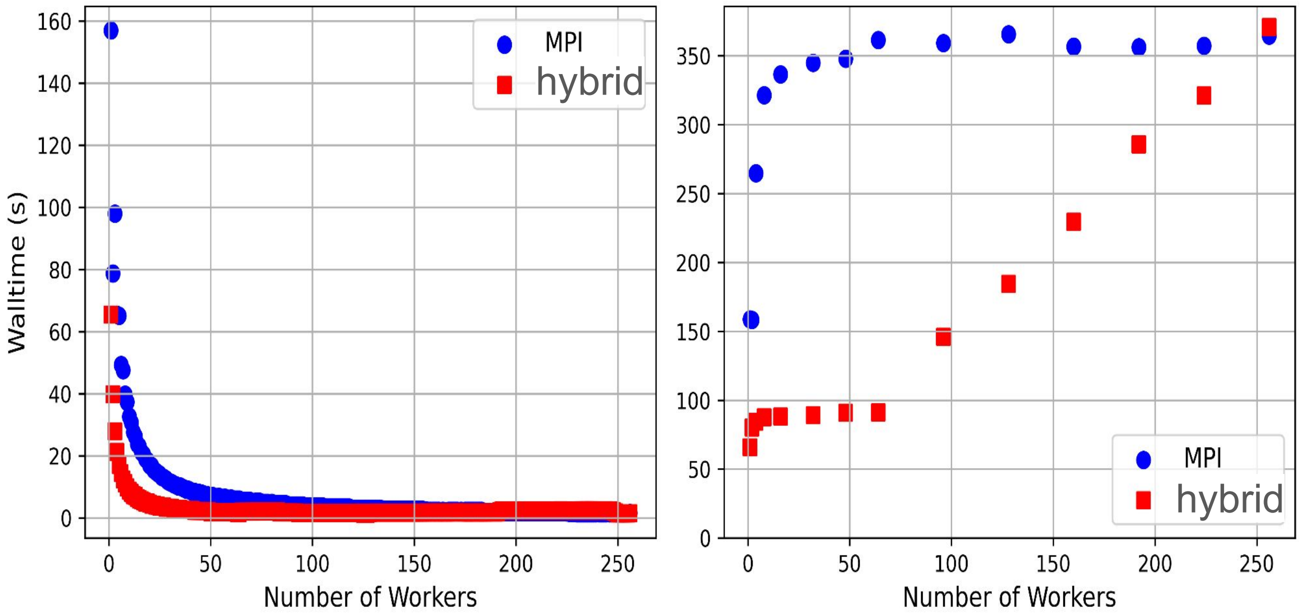
# Results and Discussion




Speedup stagnation at 40%, different from linear behavior of Gustafson's Law

Rapid to zero in efficiency, <1% for more than 50 units -> **improve workload distribution**

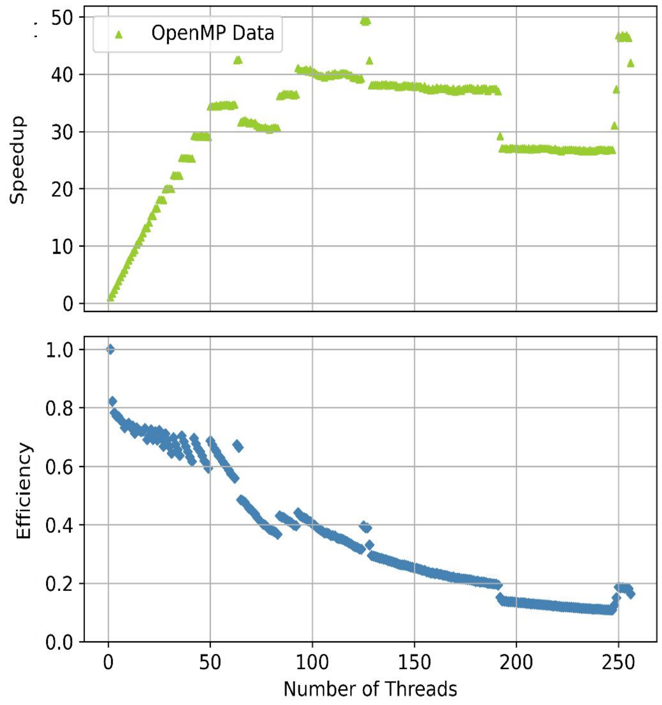
# Results and Discussion

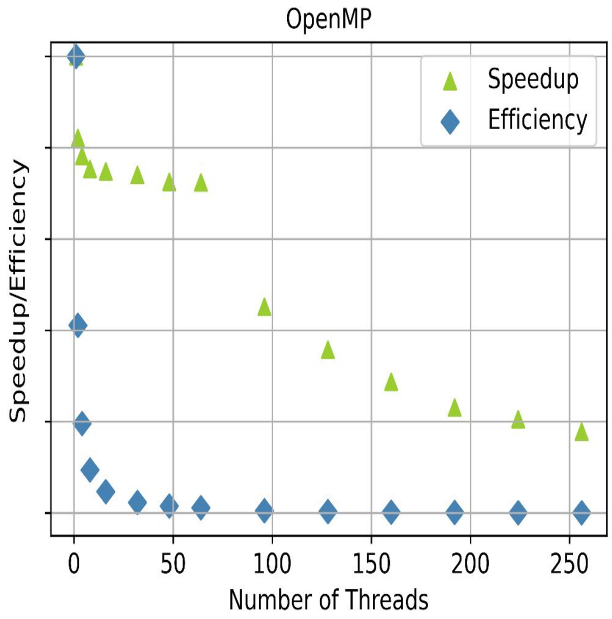



“Hybrid strong scaling”: smaller times with 4 processors in 2 nodes and increasing threads.

Weak scaling: deviation could be attributed to memory overhead, load imbalance, …  more costly to spawn threads.

# Results and Discussion



# Conclusions

Functional MPI + OpenMP Mandelbrot set computation.

In the implementation OpenMP scaling mirrors MPI scaling.

Improvement is necessary to reduce workload imbalance: better partitioning scheme for processes computations and further investigation of scheduling strategies for OpenMP threads.

The results suggest that only the cores in 1 socket might be sufficient for optimal resources use.