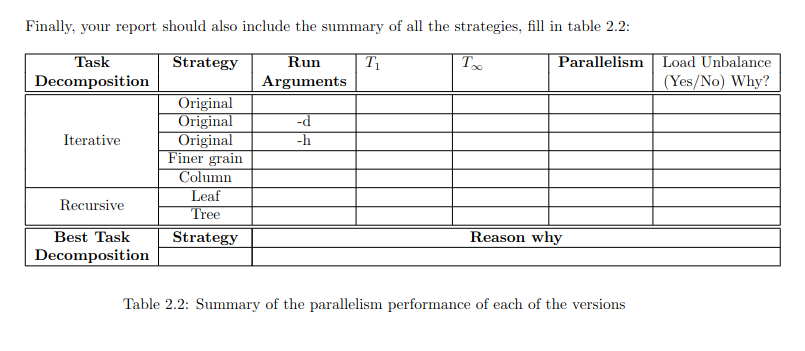
| Section | Description |
| --- | --- |
| Task Decomposition |  |
| Arguments |  |
| Code |  |
| Tareador TDG  (with dependences) : |  |
| Dependence Analysis: |  |
| Tareador TDG  (without ”datasharing” dependences): |  |
| T ∞ Analysis: |  |

Final table:

| **Task Decomposition** | **Strategy** | **Run Arguments** | **T₁** | **T∞** | **Parallelism** | **Load Unbalance (Yes/No) Why?** |
| --- | --- | --- | --- | --- | --- | --- |
| **Iterative** | Original |  |  |  |  |  |
| Original | -d |  |  |  |  |
| Original | -h |  |  |  |  |
| Finer grain |  |  |  |  |  |
| Column |  |  |  |  |  |
| **Recursive** | Leaf |  |  |  |  |  |
| Tree |  |  |  |  |  |
| **Best Task Decomposition** | **Strategy** | **Reason why** | | | | |
|  |  | | | | |



**5.1 Sequential execution**

Run the sequential code to see what it produces:

• Run mandelbrot to only measure its execution time:

sbatch submit-seq.sh ./mandel-seq-iter -i 10000

Computation of the Mandelbrot set with:

center = (-1.81641, -0.203125)

size = 2

maximum iterations = 10000

Total execution time (in seconds): 1.568147

Mandelbrot set: Computed

Histogram for Mandelbrot set: Not computed

• Run to generate histogram (mandel histogram.out) and image (mandel image.jpg) output:

sbatch submit-seq.sh ./mandel-seq-iter -h -o -i 10000

Computation of the Mandelbrot set with:

center = (-1.81641, -0.203125)

size = 2

maximum iterations = 10000

Total execution time (in seconds): 1.574432

Mandelbrot set: Computed

Histogram for Mandelbrot set: Computed

Writing output file to disk: mandel\_image.jpg

• Run it interactively with display:

./mandel-seq-iter -d -i 10000

Computation of the Mandelbrot set with:

center = (-1.81641, -0.203125)

size = 2

maximum iterations = 10000

Mandelbrot set: Computed

Histogram for Mandelbrot set: Not computed

Click on a point in the display to get its coordinates

Press any key (with focus in display) to end the program

• Run tareador analysis. See below.

# Iterative task decomposition

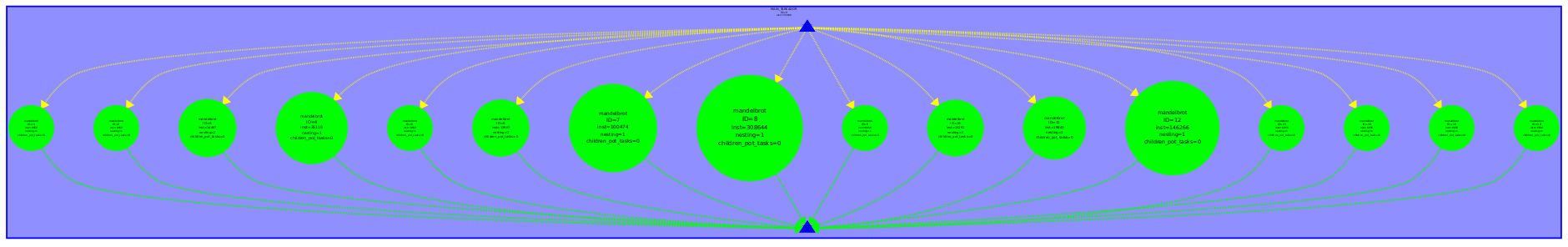
* 1. Original parallel strategy - No arguments
     1. Arguments

None.

* + 1. Code

mandel-seq-iter-tar.c

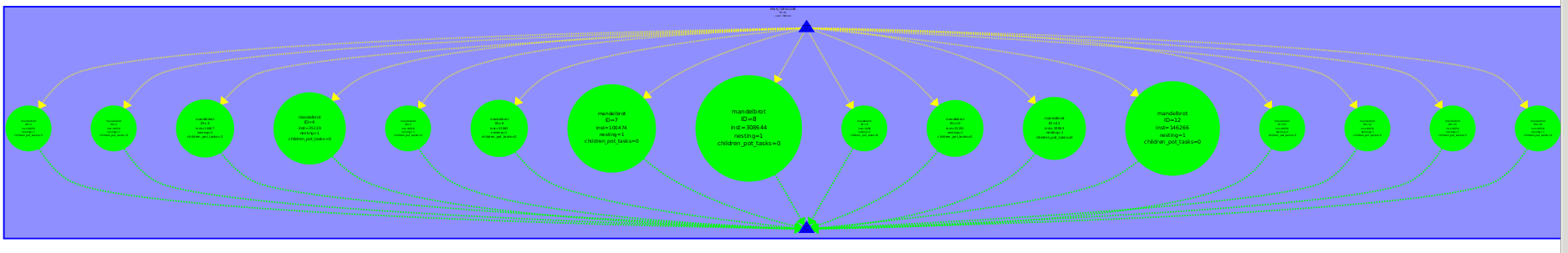
* + 1. Tareador TDG (with dependences) :



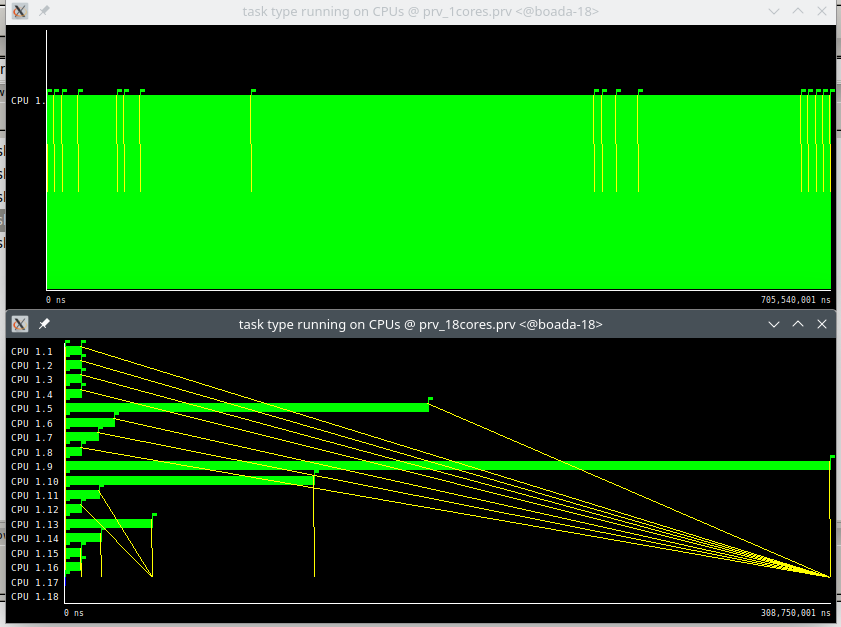
* + 1. Dependence Analysis

As illustrated in the preceding TDG figure, all tasks run simultaneously without any dependencies between them. Additionally, there is a noticeable imbalance in workload, particularly from 4 tasks, where the most demanding task processes 308,644 instructions, while the least intensive ones handle only 6,656.

* + 1. Tareador TDG (without ”datasharing” dependences):



* + 1. T ∞ Analysis



T1 = 705.540.001 ns

T∞ = 308.750.001 ns

Each thread executes one task, and it is noticed that the critical path is the largest task. The load unbalancing can be observed as well with this.

* 1. Original parallel strategy (-d)
     1. Arguments

-d.

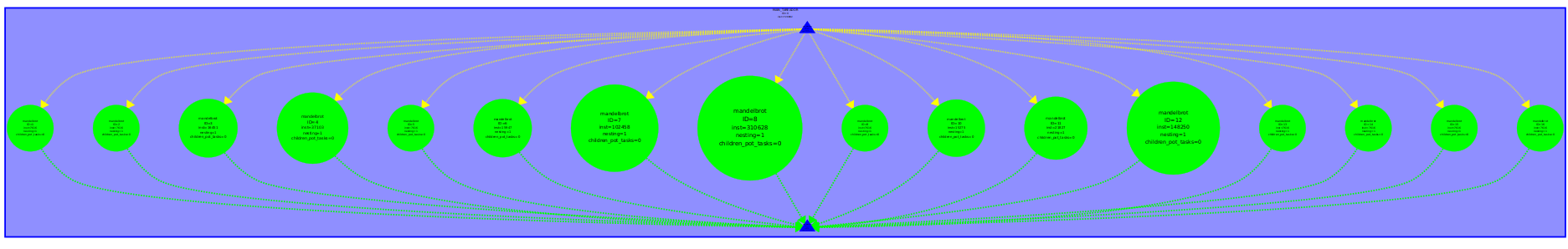
* + 1. Code

mandel-seq-iter-tar.c

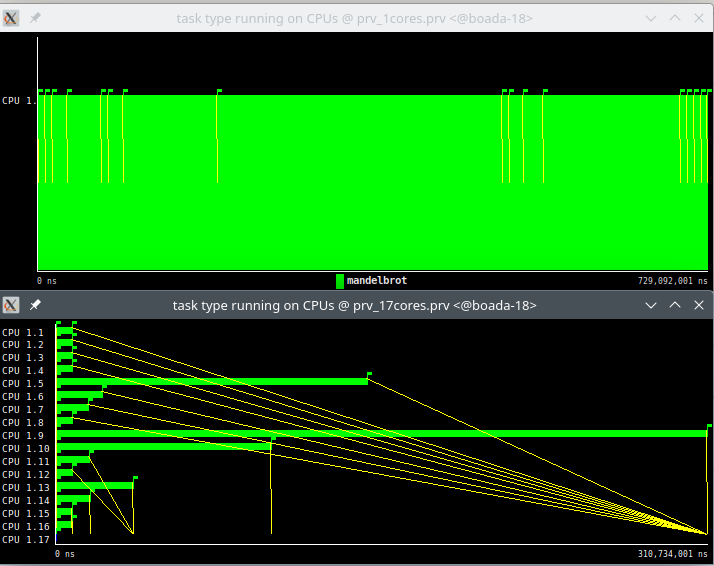
* + 1. Tareador TDG (with dependences) :

The TDG is rotated 90º

* + 1. Dependence Analysis
    2. Tareador TDG (without ”datasharing” dependences):



* + 1. T ∞ Analysis



T1 = 729.092.001 ns

T∞ = 310.734.001 ns

* 1. Original parallel strategy (with -h)
     1. Arguments

-h

* + 1. Code

mandel-seq-iter-tar.c

* + 1. Tareador TDG (with dependences) :



* + 1. Dependence Analysis
    2. Tareador TDG (without ”datasharing” dependences):
    3. T ∞ Analysis
  1. Finer grain parallel strategy
     1. Arguments
     2. Code
     3. Tareador TDG (with dependences) :
     4. Dependence Analysis:
     5. Tareador TDG (without ”datasharing” dependences):
     6. T ∞ Analysis
  2. Column of tiles parallel strategy.
     1. Arguments
     2. Code
     3. Tareador TDG (with dependences) :
     4. Dependence Analysis:
     5. Tareador TDG (without ”datasharing” dependences):
     6. T ∞ Analysis
  3. ddd

**1. Original parallel strategy. You need to perform three analyses:**

• Run with no arguments.

./run-tareador.sh mandel-seq-iter-tar

Use tareador disable object(address of variable) to disable dependences due to data

sharing to exploit parallelism among tasks if possible.

| Section | Description |
| --- | --- |
| Task Decomposition |  |
| Arguments |  |
| Code |  |
| Tareador TDG  (with dependences) : |  |
| Dependence Analysis: |  |
| Tareador TDG  (without ”datasharing” dependences): |  |
| T ∞ Analysis: |  |

• Run with only ”-d” to display the mandelbrot picture.

./run-tareador.sh mandel-seq-iter-tar -d

Note that you have to close (or press any key to close) the window displaying the mandelbrot

set in order to allow tareador to finish its work.

Use tareador disable object(address of variable) to disable dependences due to data

sharing to exploit parallelism among tasks if possible.

• Run with only ”-h” to keep the histogram of colors.

./run-tareador.sh mandel-seq-iter-tar -h

Use tareador disable object(address of variable) to disable dependences due to data

sharing to exploit parallelism among tasks if possible.

The analysis done in this part running with ”-d” and ”-h” is already useful for the rest of strategies and future parallelizations. Therefore, *we don’t ask you and you don’t need to run the following strategies with -d and -h. However you should consider the obtained conclusions above.*