Report on task 3

**Name: Igenov Temirlan**

**Group: BD-2004**

**E-mail: 201107@astanait.edu.kz**

Step 0:

We will be using the sequential ray tracing program from Task 1. Download and install Mini-Rt library (https://github.com/georgy-schukin/mini-rt), if necessary.

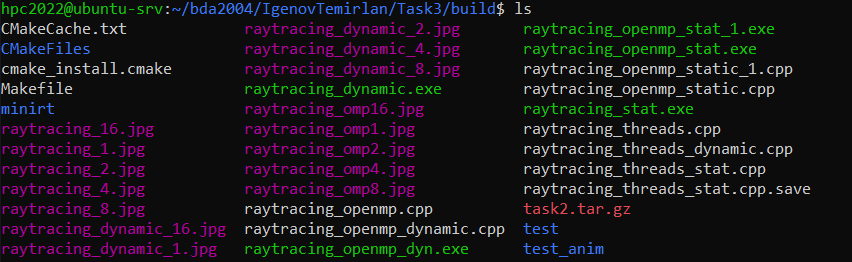
Already exist

Step 1: Prepare a directory for the Task 3

Изображение выглядит как текст

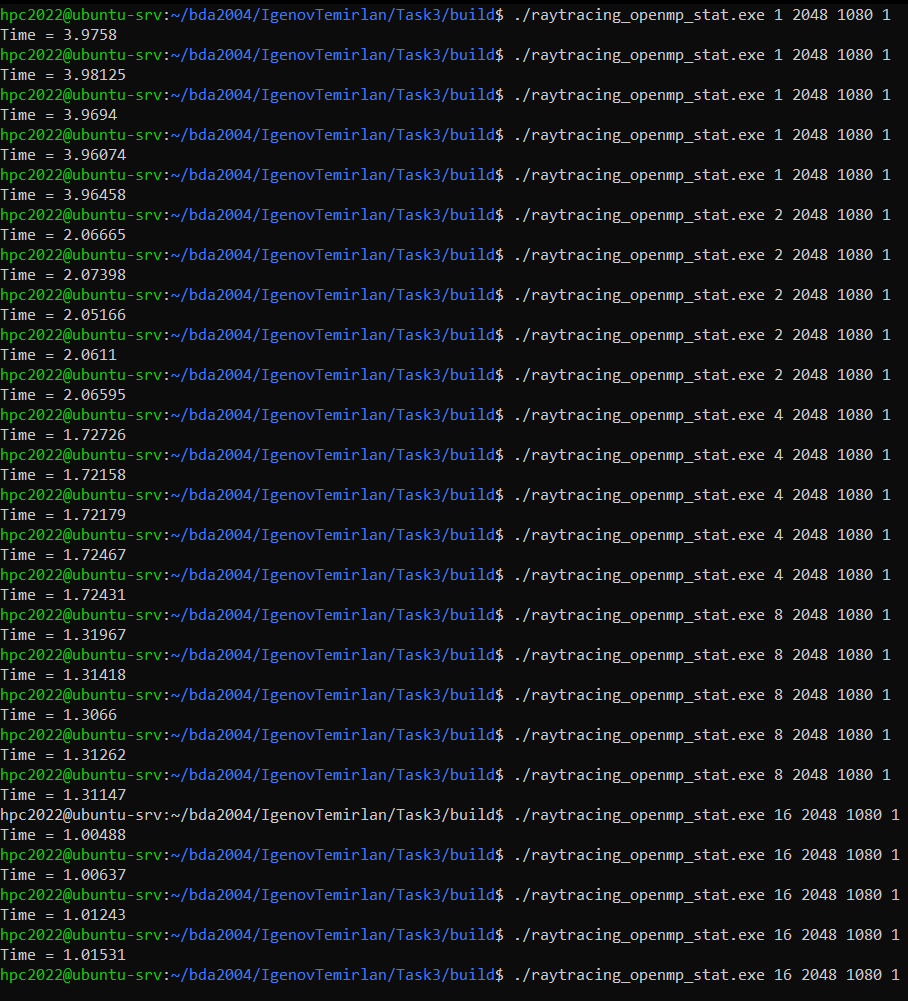
Автоматически созданное описаниеStep 2: Implement parallel program with OpenMP

I just use code from template and create 3 exe cpp and exe file for static, static1 and dynamic openmp.



Step 3: Study performance of your parallel program

1. static

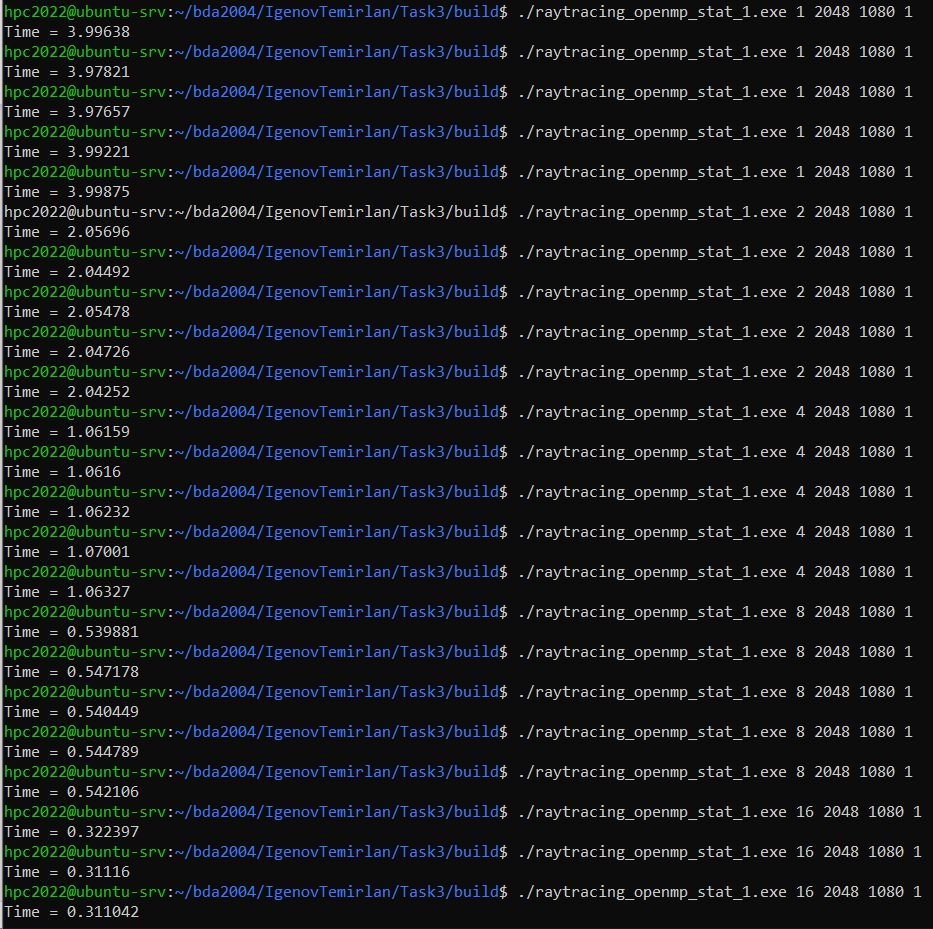


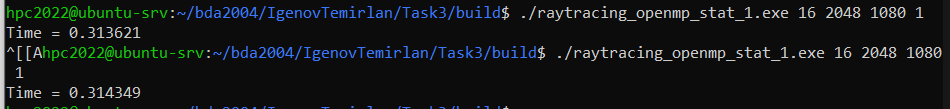


**Table of execution time, speedup and efficiency for static**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Threads | Run 1 (s) | Run 2 (s) | Run 3 (s) | Run 4 (s) | Run 5 (s) | **Min. time(s)** | Speedup | Efficiency |
| 1 | 3.9758 | 3.98125 | 3.9694 | 3.96074 | 3.96458 | **3.96074** | 100% | 100% |
| 2 | 2.06665 | 2.07398 | 2.05166 | 2.0611 | 2.06595 | **2.05166** | 193% | 96.5% |
| 4 | 1.72726 | 1.72158 | 1.72179 | 1.72467 | 1.72431 | **1.72158** | 230% | 54.5% |
| 8 | 1.31417 | 1.31418 | 1.3066 | 1.31262 | 1.31147 | **1.3066** | 303% | 37.875% |
| 16 | 1.00488 | 1.00637 | 1.01243 | 1.01531 | 1.00722 | **1.00488** | 394% | 24.625% |

1. static 1

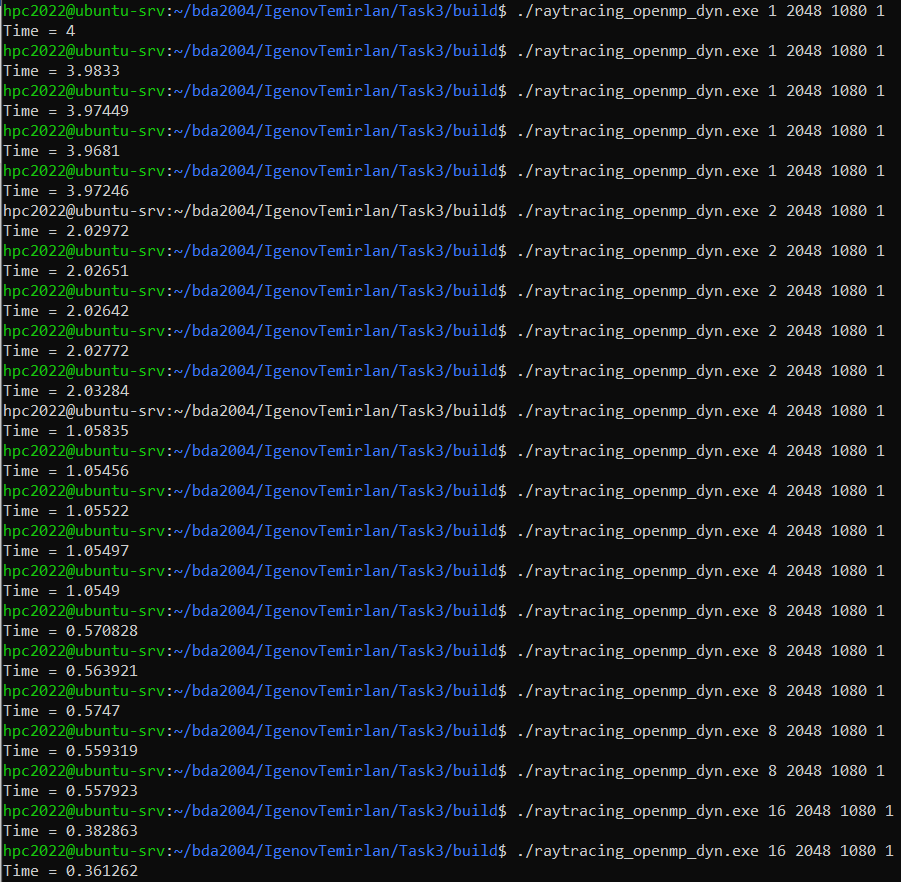
****

****

**Table of execution time, speedup and efficiency for static 1**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Threads | Run 1 (s) | Run 2 (s) | Run 3 (s) | Run 4 (s) | Run 5 (s) | **Min. time(s)** | Speedup | Efficiency |
| 1 | 3.99638 | 3.97821 | 3.97657 | 3.99221 | 3.99875 | **3.97657** | 100% | 100% |
| 2 | 2.05696 | 2.04492 | 2.0548 | 2.04726 | 2.04252 | **2.04252** | 194.7% | 97.35% |
| 4 | 1.06159 | 1.0616 | 1.06232 | 1.07001 | 1.06327 | **1.06159** | 374.6% | 93.65% |
| 8 | 0.539881 | 0.547178 | 0.540449 | 0.544789 | 0.542106 | **0.539881** | 736.6% | 92.075% |
| 16 | 0.322397 | 0.322397 | 0.31116 | 0.311042 | 0.313621 | **0.311042** | 1278% | 79.875% |

1. dynamic



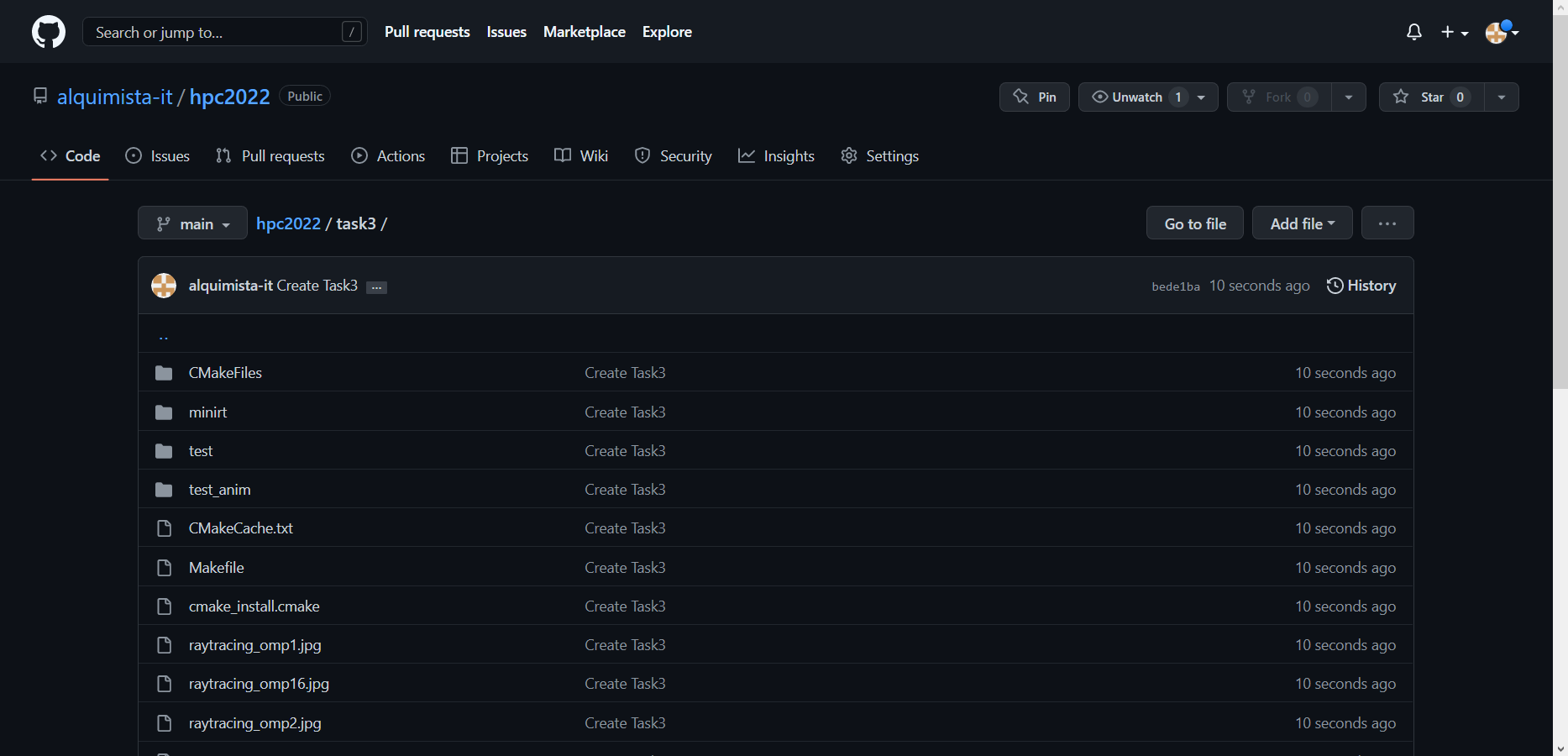
Изображение выглядит как текст

Автоматически созданное описание

**Table of execution time, speedup and efficiency for dynamic**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Threads | Run 1 (s) | Run 2 (s) | Run 3 (s) | Run 4 (s) | Run 5 (s) | **Min. time(s)** | Speedup | Efficiency |
| 1 | 4 | 3.9833 | 3.97449 | 3.9681 | 3.97246 | **3.9681** | 100% | 100% |
| 2 | 2.02972 | 2.02651 | 2.02642 | 2.02772 | 2.03284 | **2.02642** | 195.8% | 97.9% |
| 4 | 1.05835 | 1.05456 | 1.05522 | 1.05497 | 1.0549 | **1.05456** | 376% | 94% |
| 8 | 0.570828 | 0.563921 | 0.5747 | 0.559319 | 0.557923 | **0.557923** | 711% | 88.875% |
| 16 | 0.382863 | 0.361262 | 0.38061 | 0.377014 | 0.386308 | **0.361262** | 1098% | 68.625% |

Step 4: Commit and push your changes to the Gitlab server



<https://github.com/alquimista-it/hpc2022/tree/main/task3>

Step 5: Conclusion in a free form

The difference between dynamic and static is clear. In dynamic parallelization, blocks are executed by threads as they are ready and executed, and in static, everything is executed according to a ready-made distribution. However, there is a difference between static and static 1. They both perform static parallelization, but with one exception, that static1 uses iteration distribution, because default iterations are not ideal, and by changing the procedures for assigning iterations, you can increase the speed of threads.