Runtime Adaptivity through Splittable Tasks

Ste||ar Group

Shahrzad Shirzad

July 7, 2020

Louisiana State University

Objective

To provide a runtime adaptive solution to control task granularity.

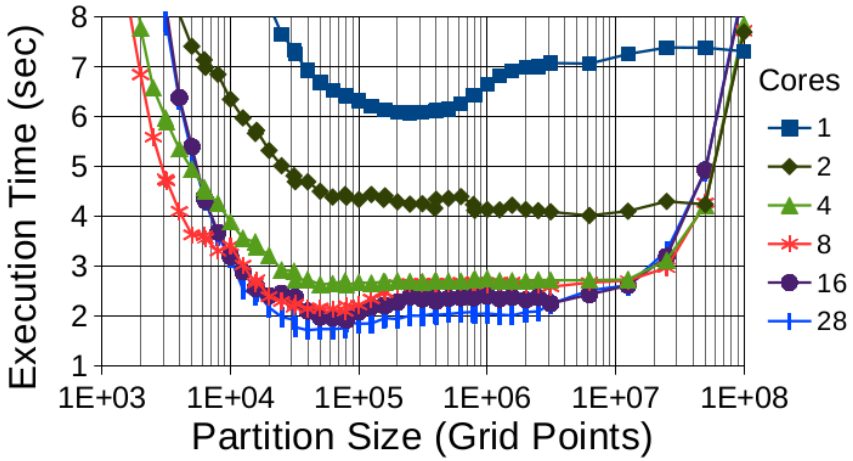


Figure 1: The effect of task size on execution time for Stencil application

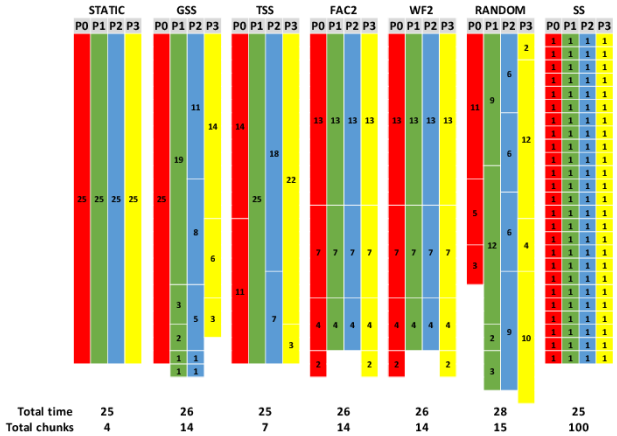


Figure 2: An example of effect of different loop scheduling methods

1 2

[1] Grubel, Patricia, et al. "The performance implication of task size for applications on the hpx runtime system." 2015 IEEE International Conference on Cluster Computing. IEEE, 2015.

[2] Ciorba, Florina M., Christian Iwainsky, and Patrick Buder. "OpenMP loop scheduling revisited: making a case for more schedules." International Workshop on OpenMP. Springer, Cham, 2018.

Splittable Tasks

- Splittable tasks are tasks that could be partitioned into smaller tasks, when sufficient parallelism is available.
  - A task could be splitted into two or more tasks, depending on the splitting strategy.

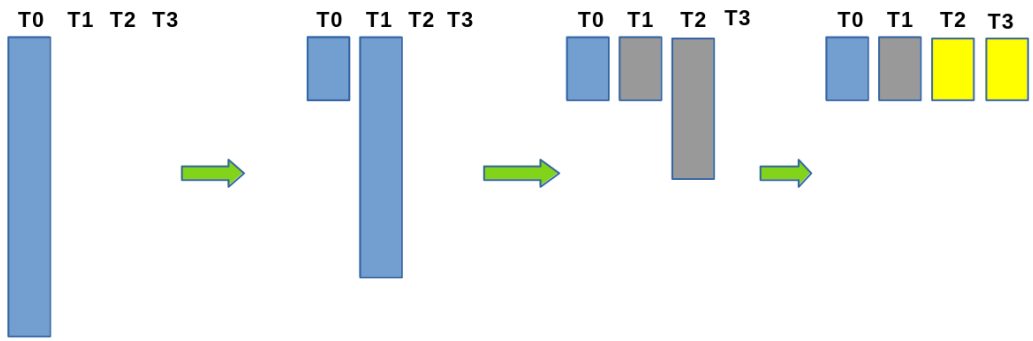


Figure 3: A simple example of splittable tasks

  - Two modes: All, Idle_mask

Splittable Executor Modes: All

Splits the tasks into two parts, $\frac{1}{P}$ of the original task size remains for the current worker, the rest is assinged to the next available worker. P is set to the number of cores and is decremented at each split.

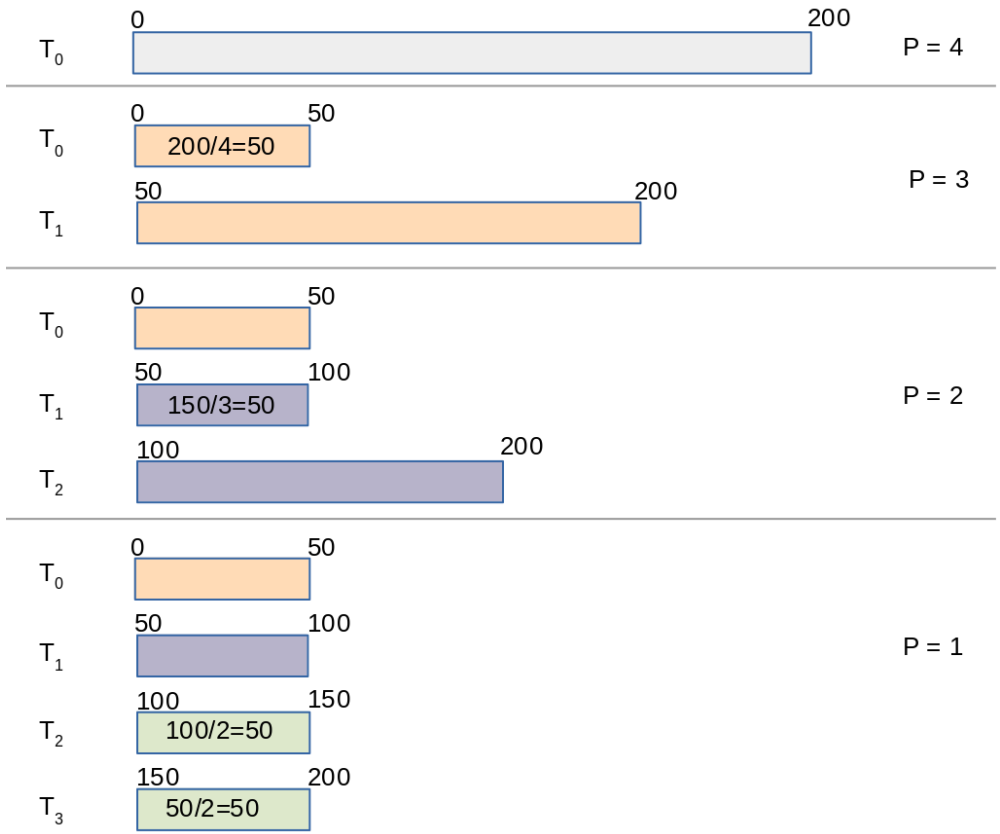


Figure 4: An example of loop scheduling using splittable tasks in ”all” mode with 200 iterations, ran on 4 cores.

Splittable Executor Modes: Idle Mask

Task is equally splitted among the idle cores, and the generated tasks are explicitly assigned to the idle cores.

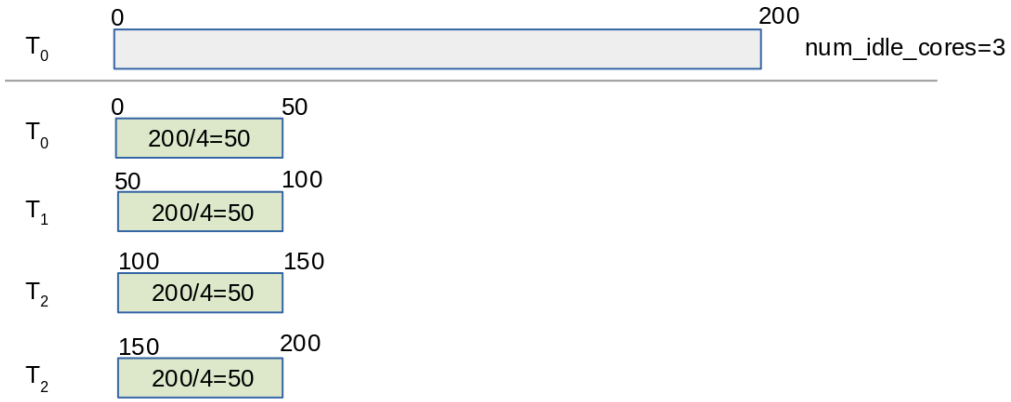
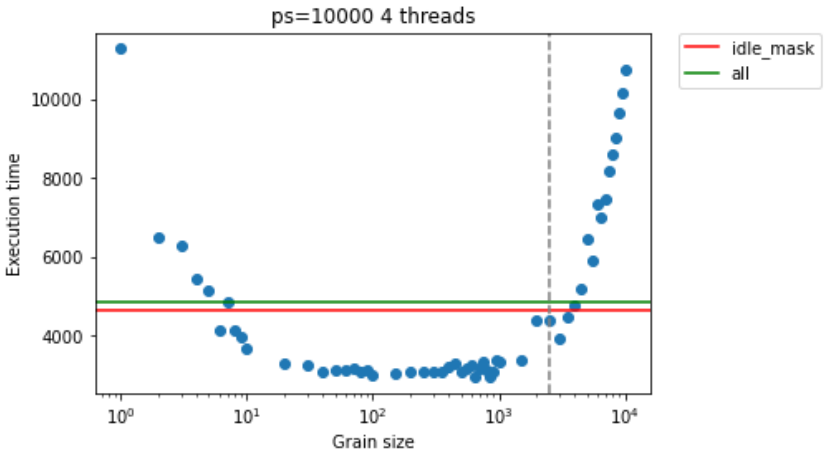


Figure 5: An example of loop scheduling using splittable tasks in ”idle mask” mode with 200 iterations, ran on 4 cores.

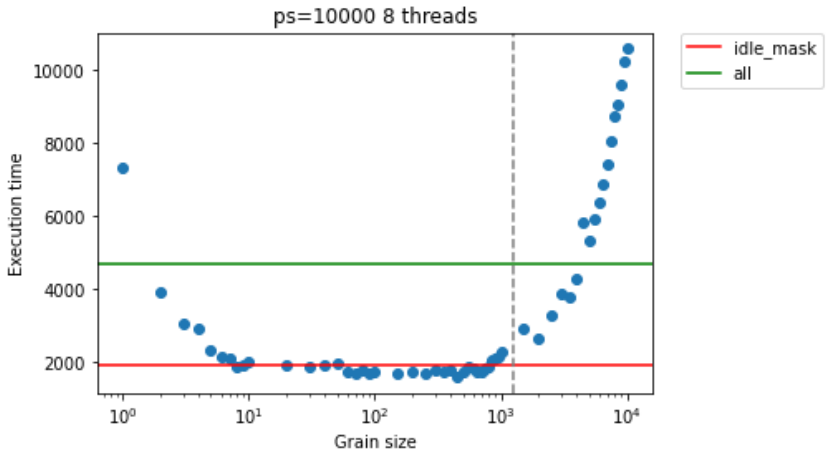
Results

- Benchmark
  - HPX parallel for-loop with iteration length of $1\mu sec$
  - $ps = num\_iterations \times iter\_length = num\_iterations$
  - Run the benchmark on 1, 2,..,8 cores for diffrent problem sizes: 1000, 10000, 100000, 1000000, 10000000

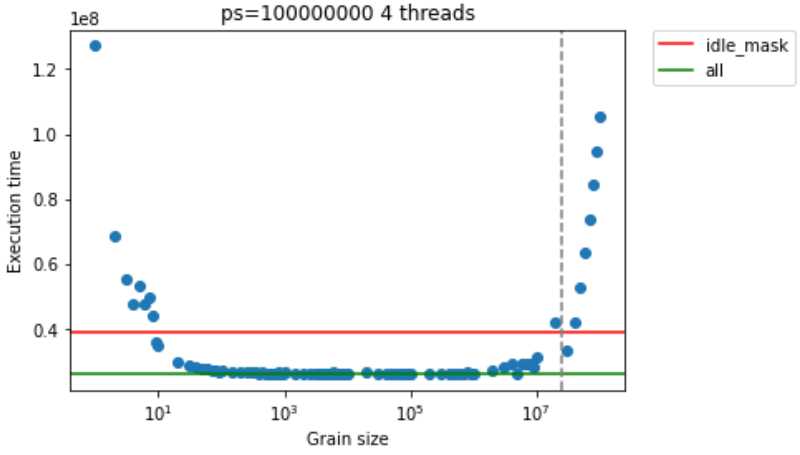
Results



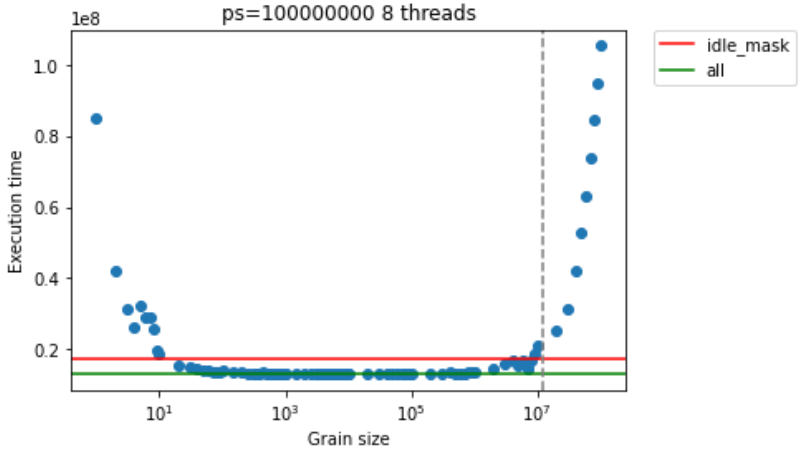
(a)



(b)



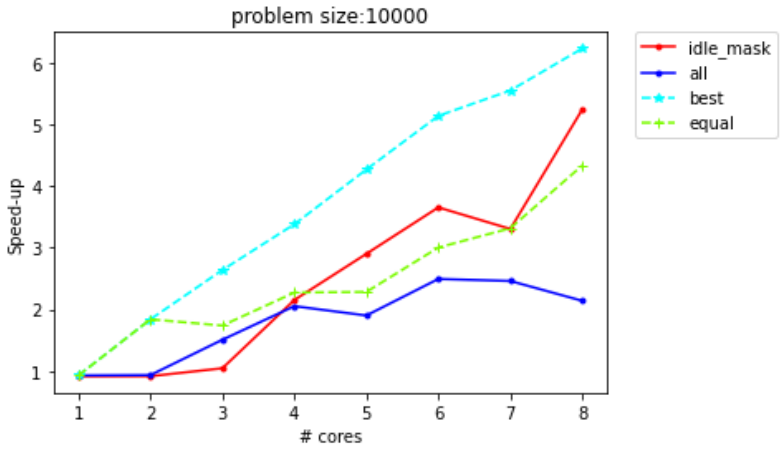
(c)



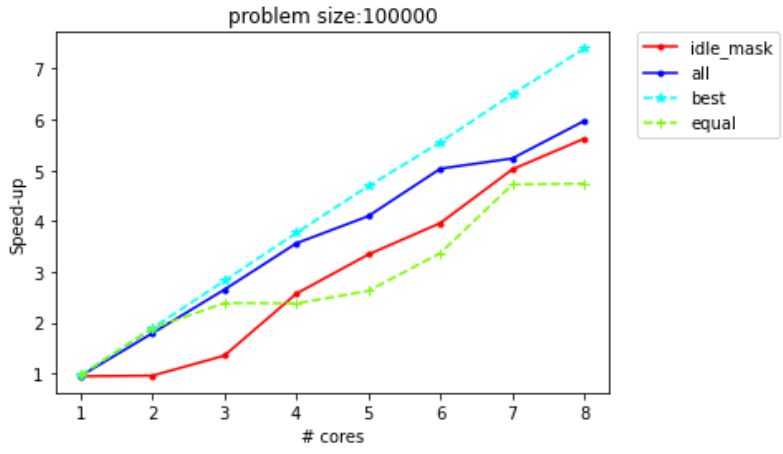
(d)

Figure 6

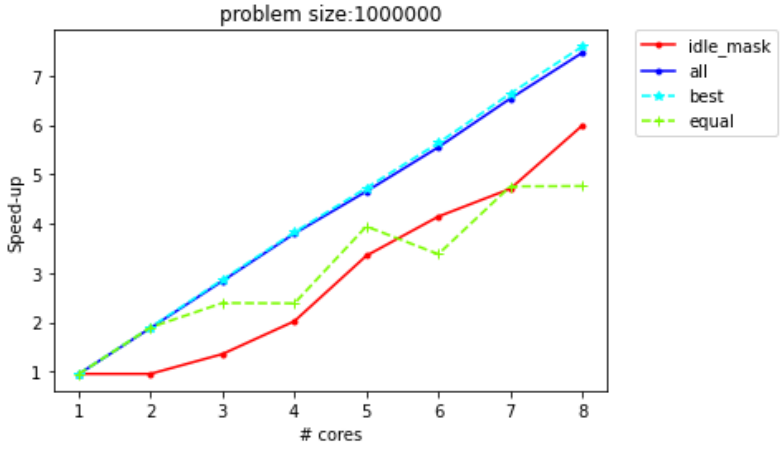
Results



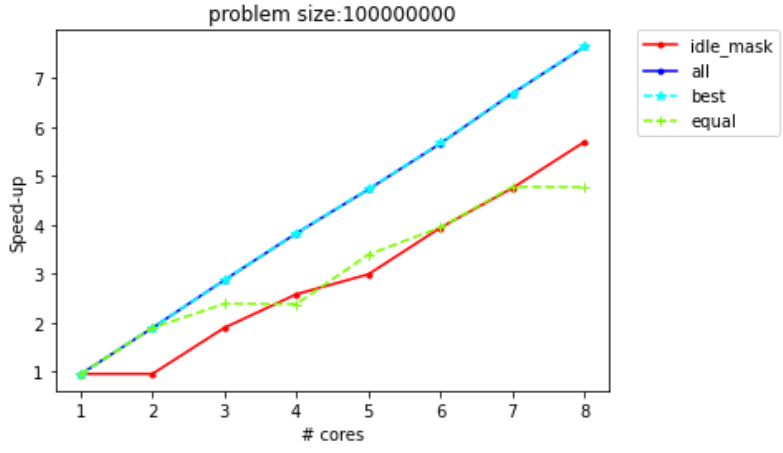
(a)



(b)



(c)



(d)

Figure 7

Thank you!

OTF2 Traces: all mode

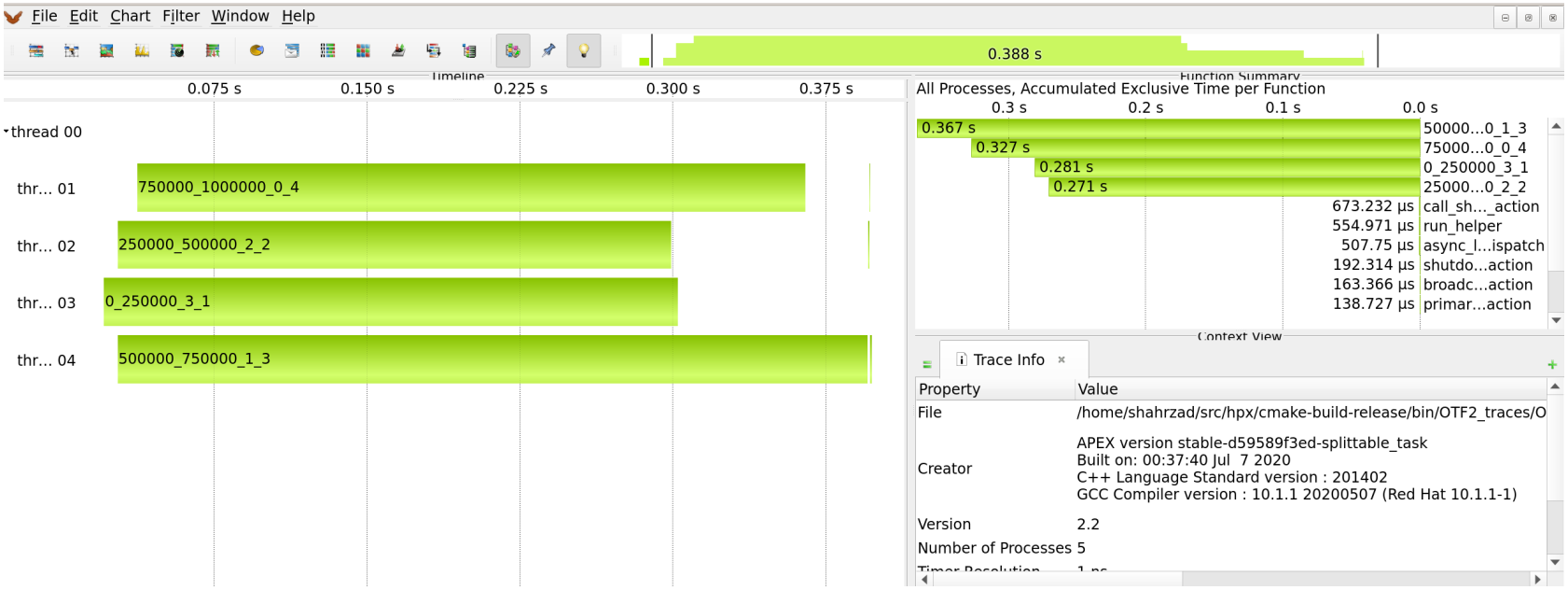


Figure 8

OTF2 Traces: idle mask mode

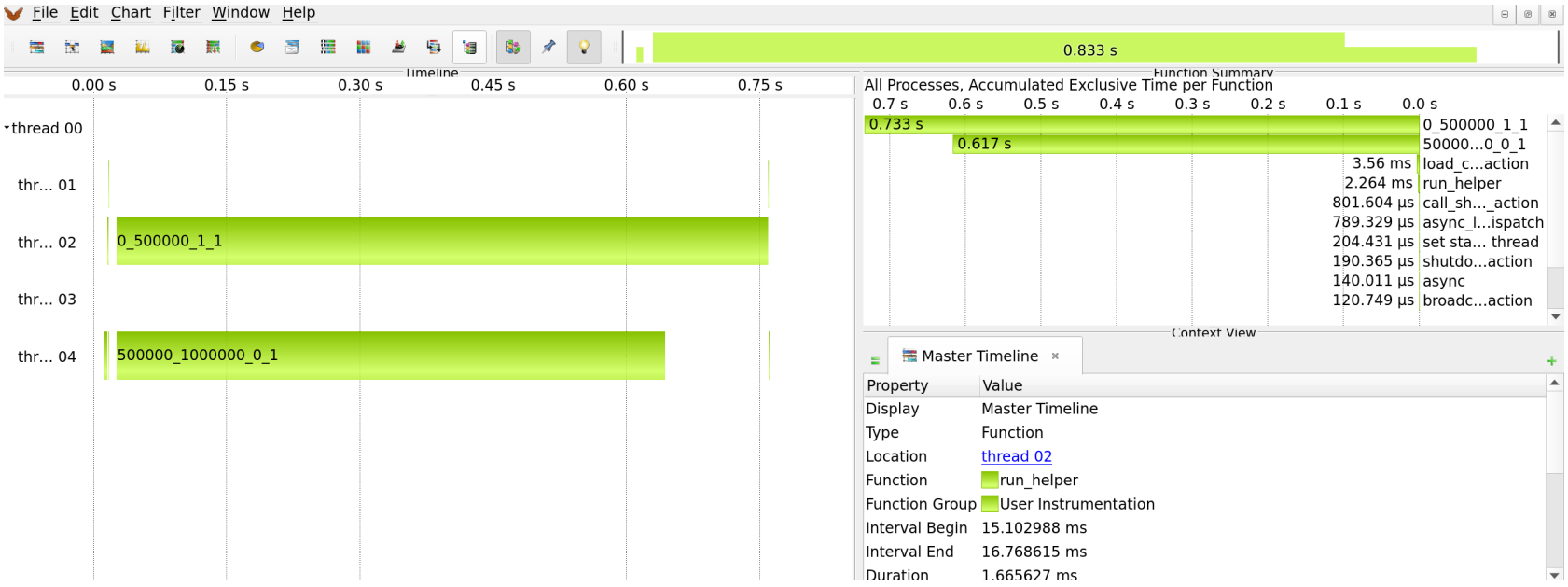


Figure 9