TU Wien Informatics

TU Wien
Faculty of Informatics
Research Group for Parallel Computing

Parallel Algorithms
2022W
Assignment 2

January 11, 2023

2 Person Group 8
1: Ferdinand MOND, 11945397
2: David MÜLLER, 01620740

1 ParJudge Submission Information

ID	**#763**
Status	**Accepted**
Solved test cases	**10**
Score	**56.7**

2 Algorithm Description

2.1 Constructing a feasible schedule

Our algorithm is a First Come First Serve (FCFS) algorithm, which means it schedules the jobs in the order that they arrive. In our case, since we know that the jobs in the instance are in real time order, it is sufficient to loop through the total number of jobs and handle each in sequence. When a new iteration of the loop is started, time is moved forward to the release time of the current job. The list of running jobs is checked against the current time, and if any are known to have finished before the current time those are removed and the machines freed. Clearing this running list is the only time when the information about actual runtime that the scheduler is not supposed to know is used. When a new job is slated to be scheduled, the currently free amount of machines are compared to the required number for this job, and if there are at least as many available as required the job is scheduled at the current time. If there are not enough machines available to schedule the current job, the running list is emptied in the order of when the jobs actually finish until there are enough free machines available. Then the current job is scheduled at the current time, which is the time when enough machines became available.

2.2 Finding the best schedule

We tried to find a better method but were not able to implement one that full filed a valid schedule. We tried to use the library heapq to use the priority algorithm but we failed to

implement a vaild solution. The file is attached as well to the submission. There are three attepmts implemented to get a better schedule but we could not get a valid schedule.