Department of Electrical, Computer, and Software Engineering

Part IV Research

Compendium Document

Project Number: 116

How much slack is in a multi-processor schedule?

Prabhnandan Singh

Harrison Warahi, Project Partner

Oliver Sinnen, Supervisor

18/10/2021

Declaration of Originality

This report is my own unaided work and was not copied from nor written in collaboration with any other person.

Signature:

Name: Prabhnandan Singh

Abstract—This document provides an overview of our Part IV project's journey. The process followed throughout the project is briefly described. The challenges faced during the project are discussed and a personal reflection on the project's journey is presented. All the supporting files submitted as the compendium are listed and briefly explained which could help those who wish to extend the research scope.

1. Introduction

This document provides an overview of the part IV project's journey and summarizes the supporting files submitted. Our project was to calculate the amount of slack in the multiprocessor schedules. The slack is basically the amount of time the tasks in a schedule can be delayed without affecting the makespan of the schedule. With the high processing needs, the energy requirements for scheduling has increased immensely. The slack in the schedule can be used to procrastinate or slow down the execution of some tasks, which can reduce the energy consumption of the schedule. However, the amount of slack is unknown, therefore, using slack reclamation techniques could cause unnecessary overhead if there is little slack in the schedules.

2. Project Journey

Firstly, a thorough literature review was performed, which helped us establish the current state of research done around the amount of slack. We found that a lot of research is done around reclaiming slack, but there is little to no research done about how much slack is present that can be reclaimed. There are rarely any attempts made at formally defining slack and calculating the total amount of slack in schedules. Moreover, many researchers have not considered idle time in the schedules which unlike slack time cannot be reclaimed. Therefore, filling these gaps in the research gave us the intent of our research. Before beginning the technical part, we formally defined slack time and idle time. Then, an algorithm was constructed to calculate the total slack time and idle time in the schedule. Based on the literature review, we decided to focus on static scheduling on homogeneous processors, as it was one of the most common scheduling problems.

A scheduling and graphs library was given to us by our project supervisor, which was implemented by previous Part IV students. The libraries provided us with multiple graphs of different graph structures and multiple list and cluster scheduling algorithms to schedule those graphs. We carefully selected a mixture of algorithms to schedule all the graphs to be analysed. The proposed algorithm was implemented and incorporated into the scheduling library to calculate the amount of slack and idle time in all the schedules. A script was used to automate the scheduling process for all the graphs and scheduling algorithms. All the important information from the schedules was saved in a CSV file. We later realized that some of the processors were completely unused which affected percentage proportion of slack and idle times, and the efficiency. Therefore, we implemented another script to find the number of unused processors for all the schedules. We used Google Colab to analyse the results. Multiple graphs were created to compare the different properties of the schedule and how they affected the amount of slack, as it was unknown what the amount of slack depended on. We found that there was indeed little slack in most of the graphs. However, a significant of idle time was found in most of the schedules. We also found that the efficiency of the schedule affected the idle time and the maximum amount of slack significantly. Therefore, we concluded that the slack reclamation techniques must target cases where it is known that high efficiency cannot be achieved. Furthermore, we encourage researchers to accumulate idle times in their slack reclamation algorithms as they can used for high energy saving.

3. Challenges

This was the first time I had ever done a project at such a level. Some challenges I faced during the project was managing my time with deliverables from other courses. Also, at some stages we realized that we did not get some properties of the schedules, therefore additional steps had to be performed to get the required information. However, thanks to our project supervisor, we were mostly on track because of the weekly meetings. Completion of the project would have been impossible without the assistance our supervisor provided in terms of sharing knowledge, motivating us and giving us the required resources.

4. Conclusion and Reflection

It was a challenging but interesting journey doing this project. I was able to learn many new skills like, analysing data using Python, writing a formal research paper and performing a literature review, and especially the advanced knowledge I gained about the scheduling theory. Overall, I think the project went smoothly and I look forward to extending the research scope further and writing the conference paper about our topic with our supervisor.

5. List of Files submitted

Table 1 lists all the files submitted as part of the research compendium along with a short description.

Table 1 : Compendium files and description

Folder name	**Description**
Literature Review	The Literature review as PDF and Word Doc
Seminar -LRROI	The PowerPoint presentation used for the seminar presentation
Slack Calculation Library	The scheduling library provided, added our slack calculation algorithm and saving the CSV file. Files changed : “\src\main\java\parcschedule\schedulers\SchedulerApp.java”, “\src\main\java\parcschedule\schedule\OnePerTaskSchedule.java”, and “src/main/java/parcschedule/schedule/IOUtils.java”
Script	The script used to automate the scheduling process for the big dataset. Plus, the script used to calculate the number of unused processors.
Mid-year video	The mid-year video on some of technical information
Result	The resulting CSV file containing all the results
Python Analysis	The python notebook showing a detailed analysis done on the results
Final Report	The final report as PDF and Word DOC
Display Day video	The Display day video summarizing the whole project