TWS Class 5: beginning with the familiar

Danilo Carastan-Santos and Enikő Kevi

13th of October 2023

1 Good Example 1[1]

Consider a supermarket that sells a variety of items to customers. Customers select one or more items that they want to buy and, after they are done selecting their items, they arrive at the supermarket's checkout section with their shopping items to be paid. The checkout section can be constituted by one or more cashiers. The cashiers may service customers at different speeds (an experienced cashier may work faster than a beginner one), and can service only one customer at a given time. Once a cashier started servicing a customer, the checkout process can not be suspended, it must go until its completion. The customers can arrive in the checkout section uninterruptedly and at an unknown rate (when the supermarket is open) or all the customers to be serviced at the checkout section are known in advance (when the supermarket closes for new customers at the end of a working day, the customers will be only the ones already in the supermarket). The scheduling problem arises as the customer-cashier assignment in order to minimize some objective. The objective may be for instance the minimization of the average customers' waiting time at the checkout section (customers will overall wait as little time as possible), the maximum waiting time (no one will wait for too long) or even, when the supermarket closes, the objective may be to guarantee that the last customer will be checked out as quickly as possible. The same scheduling problem arises in the context of High-Performance computing (HPC), where multiple computing applications (i.e., the "customers") need to be assigned to a computing machine (i.e., the "cashier"). We consider the case where applications can be parallel, which means that an application may require multiple machines to process.

2 Good Example 2[2]

Related Work

Scheduling in theory. The research community studied the parallel job scheduling problem under a general problem called multiple-strip packing problem [1]. Given a set of rectangles (jobs) and set of strips (set of computing resources), the objective is to find...

Scheduling in practice: scheduling heuristics. In online parallel job scheduling, many works explore list scheduling [2] based algorithms that rely on waiting queue ordering heuristics. These heuristics can be created by...

Machine learning in scheduling. Machine Learning (ML) is used in the context of online job scheduling mainly in two scenarios: (i) to improve scheduling by predicting the jobs' characteristics [3,4], and (ii) to create novel heuristics by using techniques such as non-linear regression...

3 Bad Example 1[3]

The dataset encompasses a comprehensive scope within the system. This encompasses data points spanning across the computing nodes (totaling more than 980), encompassing internal metrics such as core loads, temperatures, frequencies, memory read/write operations, CPU power consumption, fan speeds, GPU usage details, and more.

[1] Fragment adapted from Danilo Carastan dos Santos. Learning about simple heuristics for online parallel job scheduling. Doctoral dissertation. Université Grenoble Alpes; Universidade Federal do ABC, 2019. English. https://theses.hal.science/tel-02928077/file/CARASTAN_DOS_SANTOS_2019_archivage.pdf

[2] Fragment from Rosa, L., Carastan-Santos, D., Goldman, A. (2023). An Experimental Analysis of Regression-Obtained HPC Scheduling Heuristics. In Job Scheduling Strategies for Parallel Processing. JSSPP 2023. Lecture Notes in Computer Science, vol 14283. Springer, Cham. https://doi.org/10.1007/978-3-031-43943-8_6

[3] ChatGPT generated text

4 Bad Example 2[4]

Since the dawn of time, mankind is obsessed with optimisation. It started with the creation of tools from silex stones with cutting edges, bows and arrows to make hunting easier. Then, the wheel appeared, people invented means of transportation, tasks were automatised. Iteratively, humans developed tools and improved their technique in the search for boosting their efficiency to mainly serve one purpose. Why would a caveman decide to go bow hunting? Why would a farmer in Middle Age transport a haystack on a wheel cart? Why would you drive 10 km/h faster than the speed limit on an empty country road? The reason is to perform a task more easily, faster, to save time and/or energy. Hence the optimisation. Now, do not be mistaken! This manuscript does not discuss anthropological questions about the race of humans against the time. We are here to talk about optimisation in computer science, and, more specifically, about the efficient management of jobs and resources in heterogeneous distributed and parallel computing platforms

5 Bad Example 3[5]

In this paper, we review the current large-scale data computational model DOT, DOTA and p-DOT. However, these computing models cannot describe the time-varying state of computation or effectively reflect the data distribution profile variation in computing process. The p-DCOT model is proposed to solve this problem. This novel model from the bulk synchronous parallel model(BSP) is suitable for describing the collaborative computing behaviors between the GPU and CPU. It extends the matrix form in the DOT model to describe the basic computing operations. The p-DCOT model uses the idea from the p-DOT model that divides the computing process into multiple phases. However, it is not a simple extension of p-DOT, the data skew factor is introduced to reflect the real-time changes in data distribution. This further enhances the description capability of computing model. The p-DCOT model is applied to analyze the computation complexity and the computing time cost that are valuable for the workload balance, communication optimization and system scalability of GPU clusters. Finally, simulation datasets with different data skewness are applied to verify the conclusions deduced from the p-DCOT model. The p-DCOT model helps reveal the cooperative computing behaviors in large-scale data analytical processing more accurately.

6 Good Example 3

Information and Communication Technologies (ICT) consume up to 10% of the world's total electricity[6]. Exascale High-Performance Computing (HPC) systems contribute to this consumption. The first exascale supercomputer, hosted at the United States (US), consumes 20 megawatts of power. Such a power is enough to satisfy the electricity needs of a small city of around 14 thousand US citizens[7].

[4] Fragment extracted from Mommessin, C.. Efficient Management of Resources in Heterogeneous Platforms (Doctoral dissertation, Université Grenoble Alpes, 2020.

[5] Fragment obtained from Zhang, Ming and Liu, Luanqi and Wang, Haifeng, Cooperative Computing Model of Gpu Heterogeneous Cluster for Skewed Data Distributions. http://dx.doi.org/10.2139/ssrn.4327868

[6] https://lejournal.cnrs.fr/articles/numerique-le-grand-gachis-energetique

[7] Based on an electrical US power *per capita* of 1387 watts. Source: https://en.wikipedia.org/wiki/List_of_countries_by_electricity_consumption