**Moscow Institute of Physics and Technology**

**(National Research University)**

**Department of Molecular and Biological Physics**

Exam presentation at the department of foreign languages

Optimal Load Balancing

based on original article "Optimal Load Balancing and Assessment of Existing

Load Balancing Criteria"

by Anthony Boulmier, Nabil Abdennadher, Bastien Chopard,

*University of Geneva, Departement of Computer Science, Route de Drize 7, 1227*

*Carouge, Switzerland*

*University of Applied Sciences and Arts, Western Switzerland (HES-SO), Rue de la*

*Prairie 4, 1202 Geneva, Switzerland*

<https://arxiv.org/abs/2104.01688>

Amir Mennibaev

Gr. Б05-822

Moscow

2021

**Good morning! My name is Amir Mennibaev. I’m a third-year student at MIPT, Phystech School of Applied Mathematics and Informatics.**

**The subject of my presentation is load balancing. I find this topic interesting because parallel iterative applications are quite popular today and often suffer from load imbalance. Firstly, I will give you an overview of the load imbalance problem. Then we will take a look at existing solutions. And finally we will compare the performance results of existing and suggested solutions. My presentation will take 7-8 minutes. I’ll be glad to answer all your questions at the end of my presentation.**

**Let’s start.**

Parallel iterative applications often exhibit an irregular computational scheme that may create load imbalance over time, which is a major performance degradation factor. For that purpose, dynamic load balancing mechanisms are used throughout the application execution to keep processing elements’ workloads evenly distributed and their communications minimized. Those mechanisms involve two separate questions “how” and “when” to load balance. “How to load balance” is related to finding the algorithm that divides the computational domain into several pieces that are distributed on the available processing elements while minimizing their communications. “When to load balance” defines the particular iterations at which the load balancing mechanism is required. Their goal is to minimize the application wall time.

**Let’s discuss the computation model.**

Consider an iterative parallel application that consists of γ iterations and uses P processing elements. The time per time step is equal to the time of the slowest processing element due to synchronization mechanisms at the end of each iteration. To maximize efficiency, the workload attributed to each processing element must be roughly equal at each iteration. This is achieved through load balancing algorithms. Let’s call the set of iterations at which the load balancing algorithm is used the “load balancing scenario”. Now we can define the load balancing problem. Given an application comprising of γ iterations and P processing elements, find the set of iterations σ\* at which the load balancing mechanism must be activated such that the application wall time is minimized.

**Let’s take a look at existing solutions.**

Here you can see straightforward load balancing criteria. Fattebert, Offenhäuser, Lieber suggest to rebalance every 100, 1000, and 180 iterations respectively. While Ishiyama in his paper proposed rebalancing after every iteration. Marquez suggests an acceptable workload variation range. Procassiny uses another strategy: his global criterion redistributes workload when it is predicted that it will lead to performance improvement. Menon considered a model with linear growth of difference in time-per-iteration between the slowest processing element and an average one. In that case, an optimal period of rebalancing can be calculated by the following formula.

**Let’s talk about the solution proposed by the authors.**

Now the time spent can be calculated as the following integral. According to our definitions, one can make the following transformations. And then we need to solve the simple differential equation. This leads us to the global load balancing criterion.

**Let’s move *to the final part of my presentation*** and speak about the comparison of the criteria.

Here you can see benchmarks on static workload. There are execution times on the top graph and a load imbalance rate on the bottom one. As you can see on the constant and linear imbalance growth all the criteria show acceptable results. Now let’s proceed to sublinear imbalance growth and linear growth with autocorrect. Herein Menon criterion rebalances too often. In the autocorrect neither Menon criterion nor proposed one can understand when the optimal time to rebalance is. Let’s move to an irregular workload. In the constant and linear cases the performance of criteria is almost unchanged. On the sublinear experiment Menon criterion improves its performance, whereas Procassini criterion’s performance decreases.

In order to use Procassini criterion one should choose an appropriate constant. On the benchmark results there is the best constant among 5000 values.

**So, I have come to the end of my presentation. Now I want to summarize what I have said**. In the present paper authors proposed state-of-the-art load balancing criteria, which was compared with current solutions. As we saw it showed almost optimal performance. The authors plan to run criteria on production codes.

**In conclusion, I’d like to say** that load balancing is a very complicated and important problem. And it’s great that such an elegant solution had been found.

**Thank you for your attention.**

**Now I’m glad to answer all your questions.**