Concurrency and Parallelism
Cilk$^+$ Parallel Patterns Implementation

André Rosa
48043
af.rosa@campus.fct.unl.pt

João Geraldo
49543
j.geraldo@campus.fct.unl.pt

Rúben Silva
47134
rfc.silva@campus.fct.unl.pt

Abstract—In recent years, the parallel computation paradigm has been emerging, as a consequence to: the huge growth on the amount of data that needs to be processed and analyzed, and to the switching of processors' architecture evolution process from becoming smaller and having faster clock speeds, to integrate parallel functionalities, such as multiple cores, hardware threads and vector operations. Therefore, it is imperative to build algorithms that explore these functionalities to increase the efficiency on processing such huge amounts of data. However, building algorithms on top of these functionalities can sometimes be a difficult and complex task, due to them being low level primitives and sometimes platform dependent. Thus, parallel functions and libraries can provide an easy and generic way to better utilize these resources, when available. Therefore, we developed a library, that implements some of the most well known parallel patterns, and that can be easily integrated in any already existing sequential program, to improve its efficiency. We developed each algorithm to try to achieve the maximum possible parallel slack and ...

In order to be able to use it in a vast amount of use cases, these implementations are independent both from the data types they are manipulating and the parallel functionalities provided by each specific hardware platform.

What was our approach? What were the results? What did you learn?

Finally, we conducted a preliminary experimental evaluation on the performance of the different implemented alternatives, comparing them with their corresponding sequential version, that showed ... .

Index Terms—Parallel Algorithms, Cilk$^+$

1. Introduction

Nowadays, the parallel paradigm is a matter of great importance. Every day the amount of data to be processed grows [meter uma citação], as well as the demand for efficient programs that produce results faster and efficiently. Hardware performance has also evolved —processor architecture has switching from building bigger and clockwise faster to have more cores and threads. For these reasons, build highly concurrent programs is essential to achieve good levels of scability and letency.

However, code such programs can be very difficult, not only because they make use of low level primitives, but also because those primitives can be platform dependent. Therefore (outra palavra para nao ser igual?), we builded a library that implements well known parallel patterns that allow the application-level developer to integrate such patterns and give its application a boost of permormance with little to no difficulty. We did our best to keep the library independent from the system and the applications. So, all of our implementations are independent from data types and specific concurrent

(... alguma cena para ligar)

2. Architecture

3. Implementation

4. Experimental Evaluation

4.1. Experimental Setting

4.2. Experimental Results

5. Conclusion

The conclusion goes here.

Acknowledgments

Comments

References

[1] J. Leitão, P. Ákos Costa, M. Cecília Gomes, and N. Preguiça, "Towards Enabling Novel Edge-Enabled Applications," *ArXiv e-prints*, May 2018.