Universidade: presente!
UFRGS PROPESQ
XXXI SIC
Salão UFRGS 2019 — Conhecimento, Formação, Inovação
21. 25. OUTUBRO • CAMPUS DO VALE

# Computational Experiments on Task-Based Parallel Applications

Henrique Corrêa Pereira da Silva[1] and Lucas Mello Schnorr[2] (advisor)

{hcpsilva[1], schnorr[2]}@inf.ufrgs.br

## Contextualization

- The need of ever-more performant systems led developers into parallelizing their applications;
- However, utilizing *homogeneous* hardware is still the norm, which comes with great investment needs;
- Given this reality, considerable effort has been poured into the development of *middleware* capable of effectively distributing the workload into more *heterogeneous* systems.

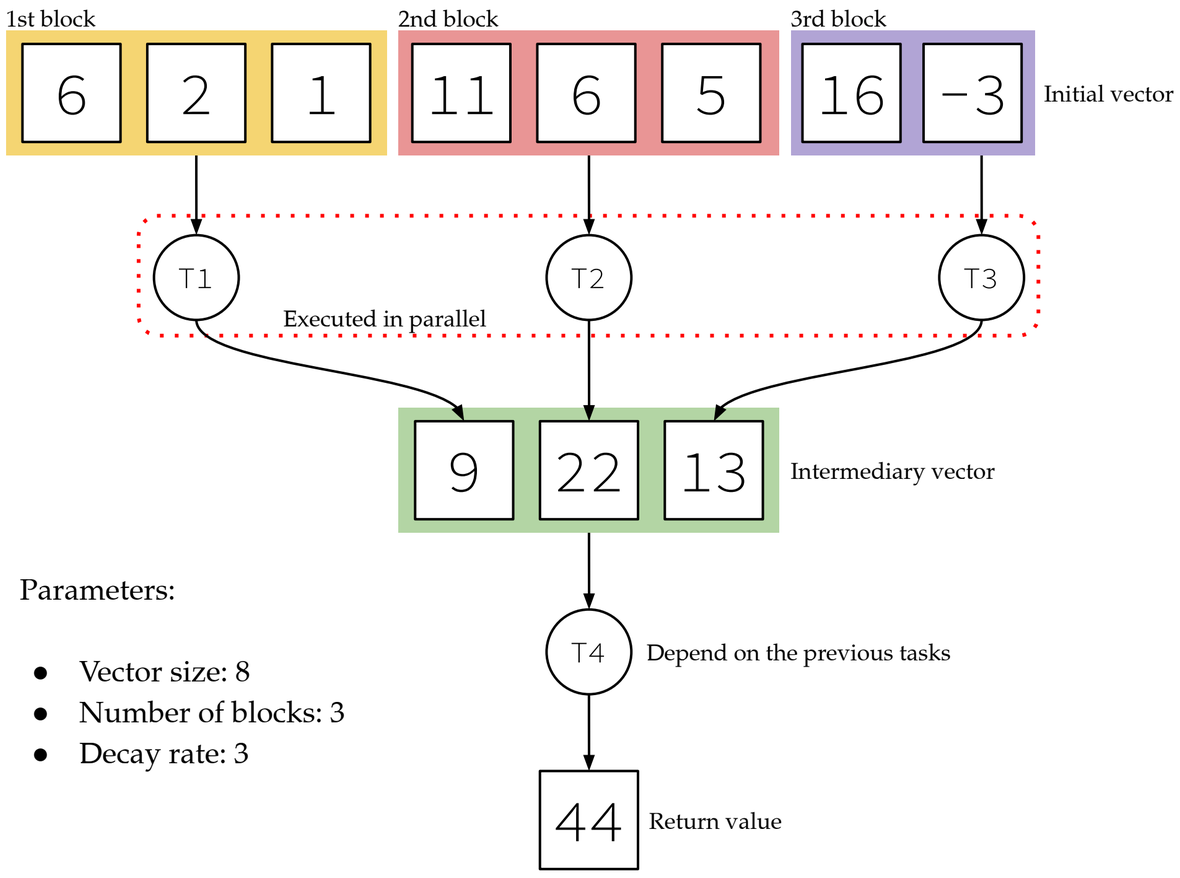
## Motivation

- There are several popular parallel programming models, such as the **message passing**, **shared memory** or a mix of these, known as a **hybrid model**;
- The implementation of such models in modern software is a known difficult, error-prone and time-consuming process;
- Popular implementations of said models try to provide a user-friendly interface through better *APIs* or compiler directives;
- Nonetheless, by the inherent fixed size of the parts in a problem decomposition, we still suffer from dynamic imbalance and, thus, loss of performance;
- We can utilize said models to implement a higher-level approach, called *task-based* model, to divide our problem into smaller kernels of computation.

## Methodology

- One of the *middleware* capable of such *heterogeneous computation* is called **StarPU**, which approaches the problem with the task-based model in hand;
- By defining tasks into what's called an **directed acyclic graph** (shown in Figure 1), the dependencies get naturally exposed and StarPU can then handle the distribution of said tasks onto the underlying hardware;
- Said approach is not only programmer friendly but is also very powerful when it comes to declaring and visualizing dependencies;
- This way, we intend to show that said model is a practical in achieving parallelism in modern software.

### Directed Acyclic Graph




Parameters:

- Vector size: 8
- Number of blocks: 3
- Decay rate: 3

Figure 1: Example of the reduction of an vector into blocks and their respective tasks.

## Results and conclusion

- Thus far, we have implemented a block vector reduction using the task-based model, which example can be observed in Figure 1.
- Using the initial number of blocks and its decay rate as parameters we have achieved a multi-level parallel vector reduction;
- This initial implementation shows that StarPU is a viable and programmer-friendly way to apply the task-based model in modern software.

## References

1. Samuel Thibault. On Runtime Systems for Task-based Programming on Heterogeneous Platforms. *Distributed, Parallel, and Cluster Computing* [cs.DC]. Université de Bordeaux, 2018.
2. Garcia Pinto, V, Mello Schnorr, L, Stanisic, L, Legrand, A, Thibault, S, Danjean, V. A visual performance analysis framework for task-based parallel applications running on hybrid clusters. *Concurrency Computat Pract Exper*. 2018; 30:e4472.

**Acknowledgements:**

UFRGS — Universidade Federal do Rio Grande do Sul
Instituto de Informática UFRGS
CNPq — Conselho Nacional de Desenvolvimento Científico e Tecnológico