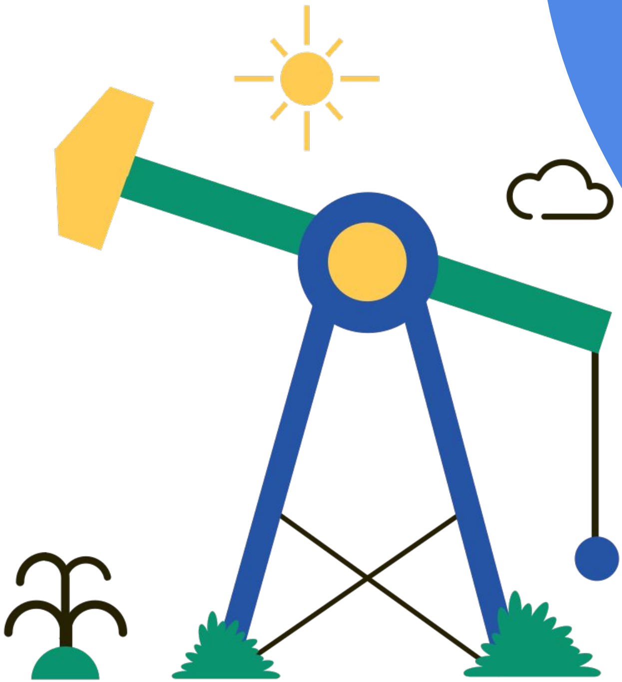
Instituto de Informática - UFRGS
Programa de Pós Graduação em Computação



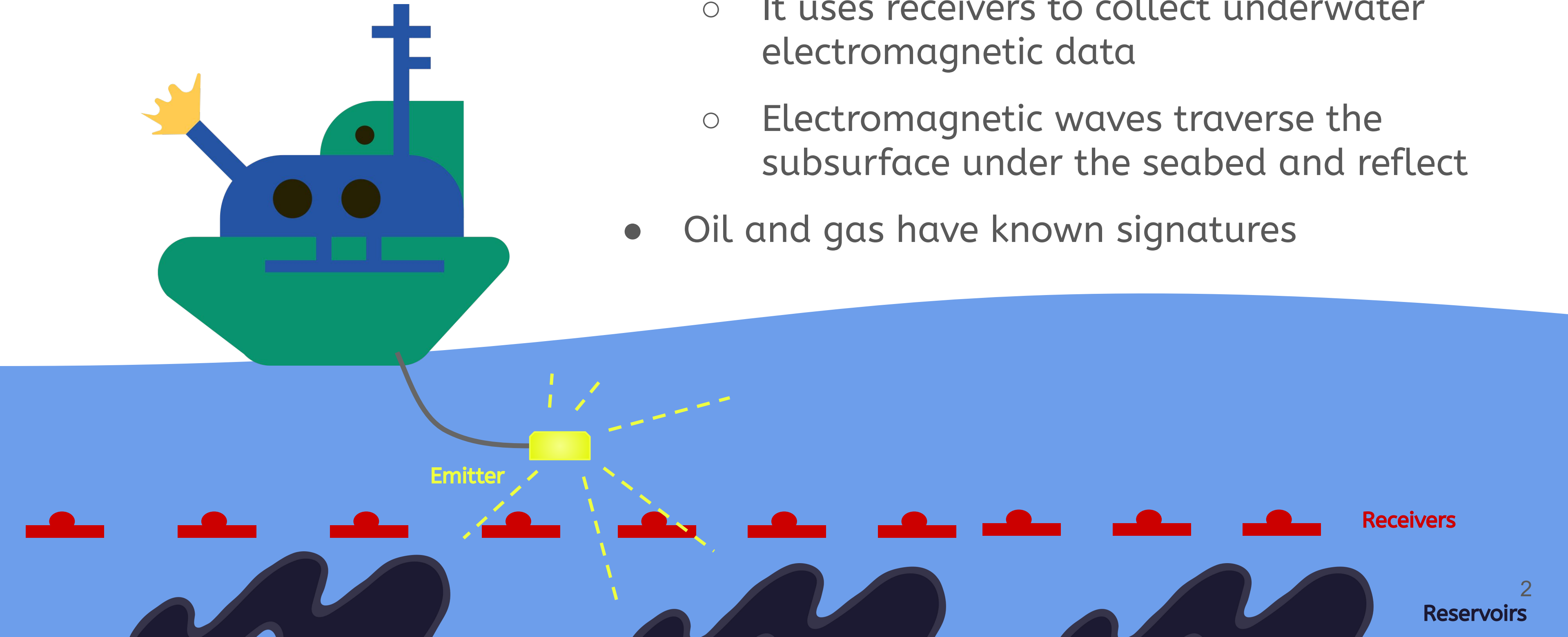
Performance Improvements Applied in an Electromagnetic Inversion Application Focused on Homogeneous and Heterogeneous Computational Environments

Jessica Imlau Dagostini
Advisor: Lucas Mello Schnorr

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
.Inf INSTITUTO DE INFORMÁTICA UFRGS
BR PETROBRAS

Introduction

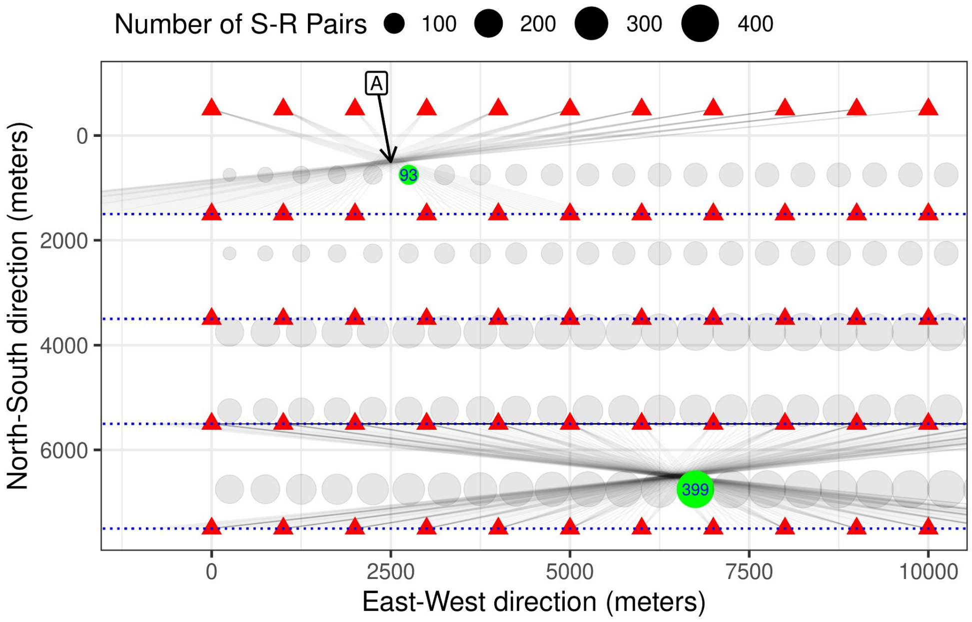
- Marine Controlled-Source Electromagnetic Method (**mCSEM**)
  - It uses receivers to collect underwater electromagnetic data
  - Electromagnetic waves traverse the subsurface under the seabed and reflect
- Oil and gas have known signatures



Introduction

- The data collected from receivers is not “understandable”
- Needs numerical methods to recover data
- Big quantity of data
- Inversion process is computationally expensive
- Run inversion in parallel
- High Performance Cluster

The mCSEM Inversion Application

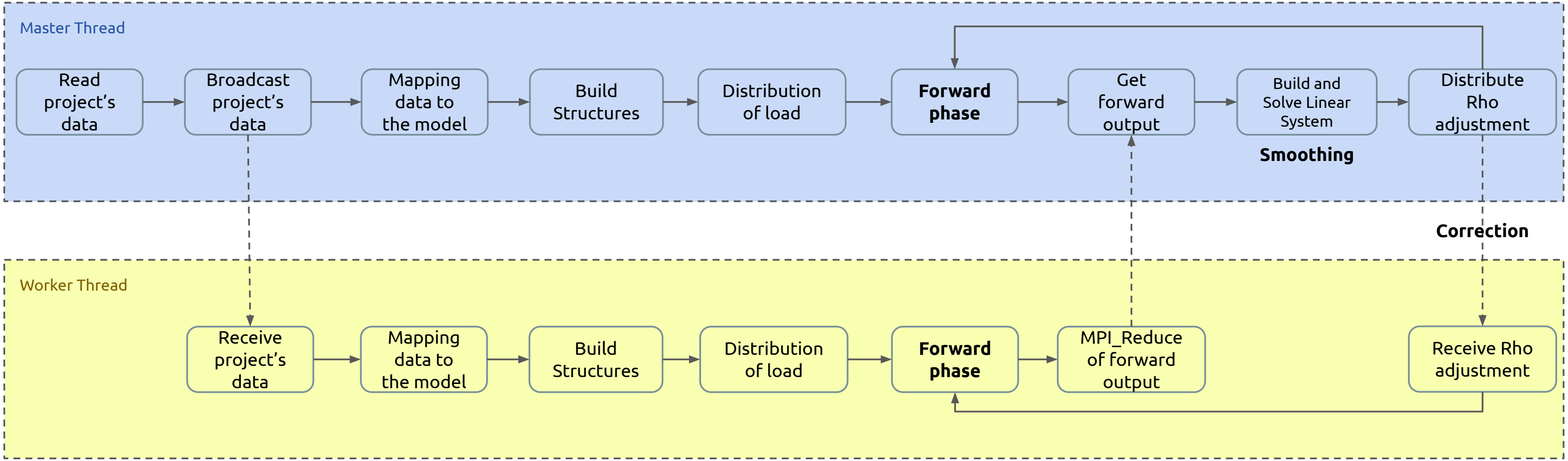


Survey data (red triangles are receivers, small blue squares are emitters), combined with the 2D grid of CMPs (circles).

- Application uses the **Common Midpoint Technique**
  - Map the collected data to the model
- It uses **geographical midpoint** between the **receiver** and the **emitter**
- Mapping will assign **different quantities** of **Source-Receiver pairs** to each CMP cell
- The CMP cell can or cannot match a receiver position

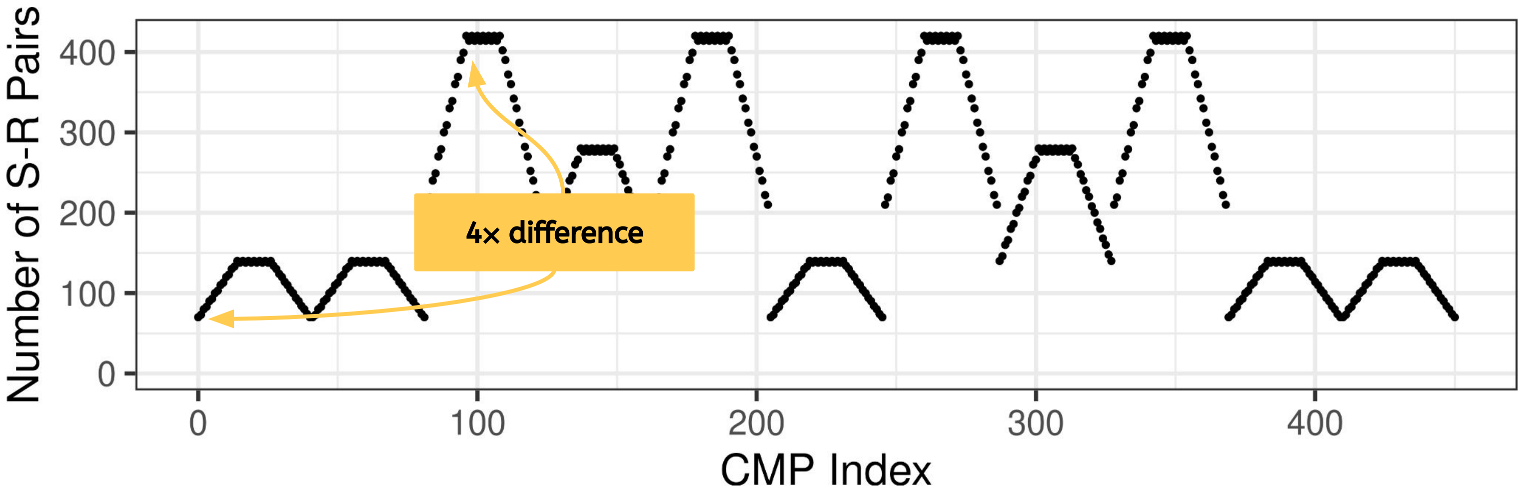
The mCSEM Inversion Application

- Based on the numerical inversion process
- 1D inversion using the CMP domain for 3D data
- Application parallelize two phases: the **forward** and **smoothing**



Finding the Imbalance

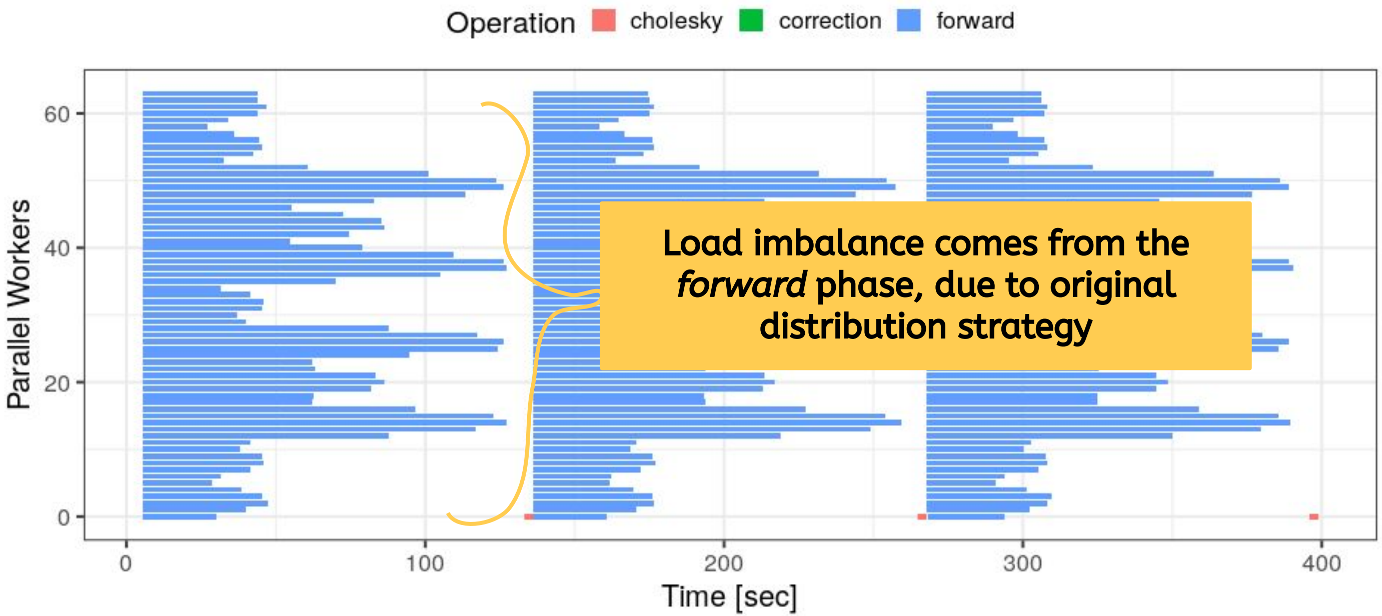
- First instrumentations results give us the organization of the cells
- Outputs with quantity of S-R pairs per CMP



Number of S-R pairs associated for each CMP of the studied model.

Finding the Imbalance

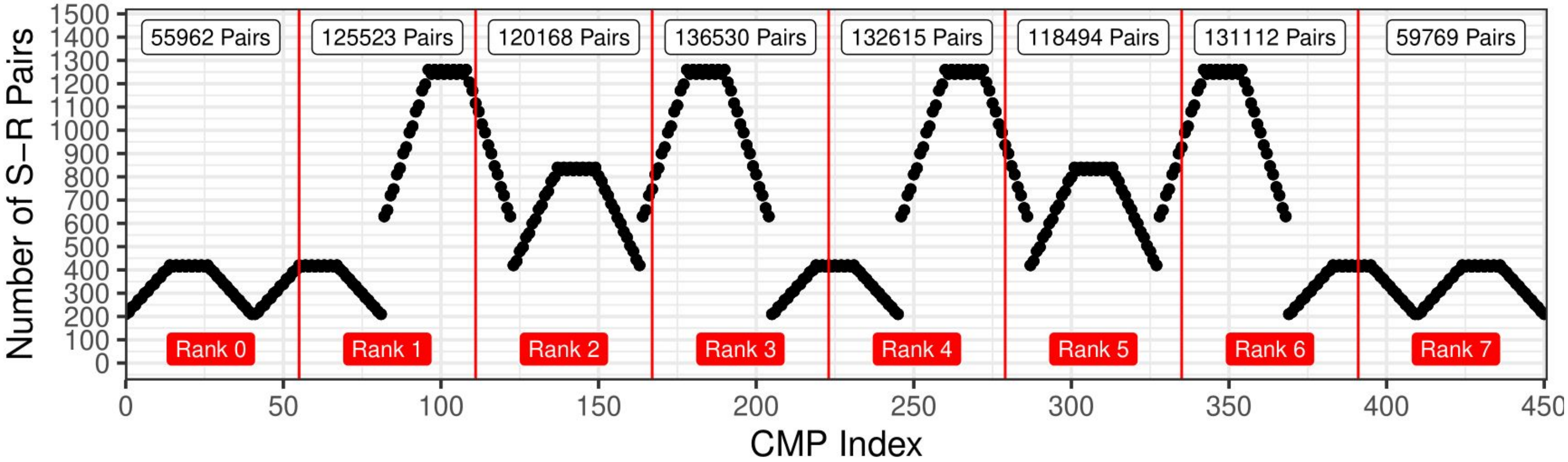
We ran the application with 64 MPI workers, disabling OpenMP threads



Execution behavior of the application executing only with MPI, presenting the timespace of the three main regions of the inversion algorithm.

Finding the Imbalance

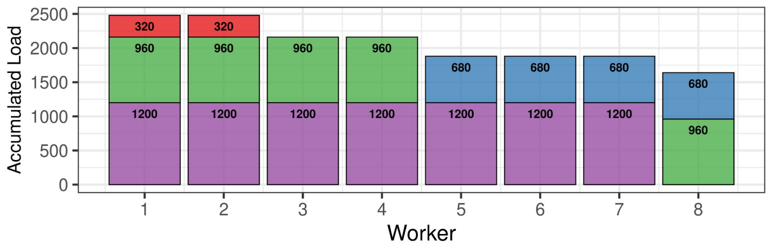
- Original distribution:
  - Considers the total quantity of **CMP** cells
  - **Disregards** the **heterogeneous** load of them



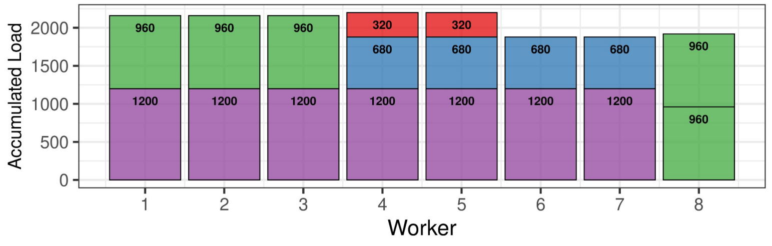
Behavior of the original distribution – it divide the cells among workers, disregarding its own heterogeneous weight.

Load Balancing Heuristics

Round Robin



Sorted-Greedy



ABE-guided

minimize:

$$makespan$$

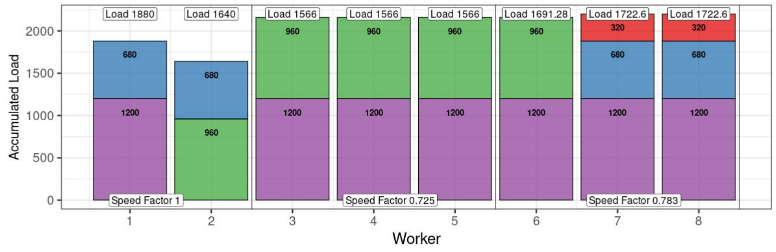
subject to:

$$\forall c \in C: \quad \sum_r n_{c,r} = N_c$$

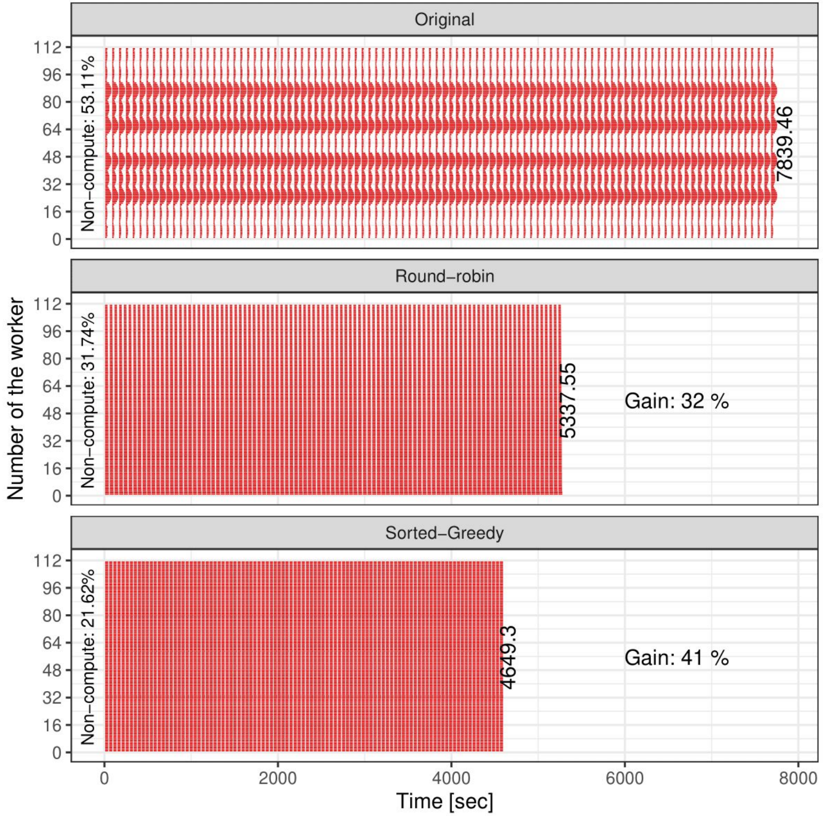
$$\forall r \in R: \quad \sum_c n_{c,r} * w_{c,r} \leq makespan * N_r$$

$$\forall c \in C, r \in R: \quad n_{c,r} \geq 0$$

HEFT



Improvements on the Application



Space/time view of the forward step of 100 iterations for the Original, Round-robin, and Greedy allocation policies.

- **Makespan**
- 3 homogeneous heuristics
- 100 iterations
- Local cluster Draco
- 7 nodes, 16 cores per node
- Total of 112 MPI ranks
- Improvements up to 41%

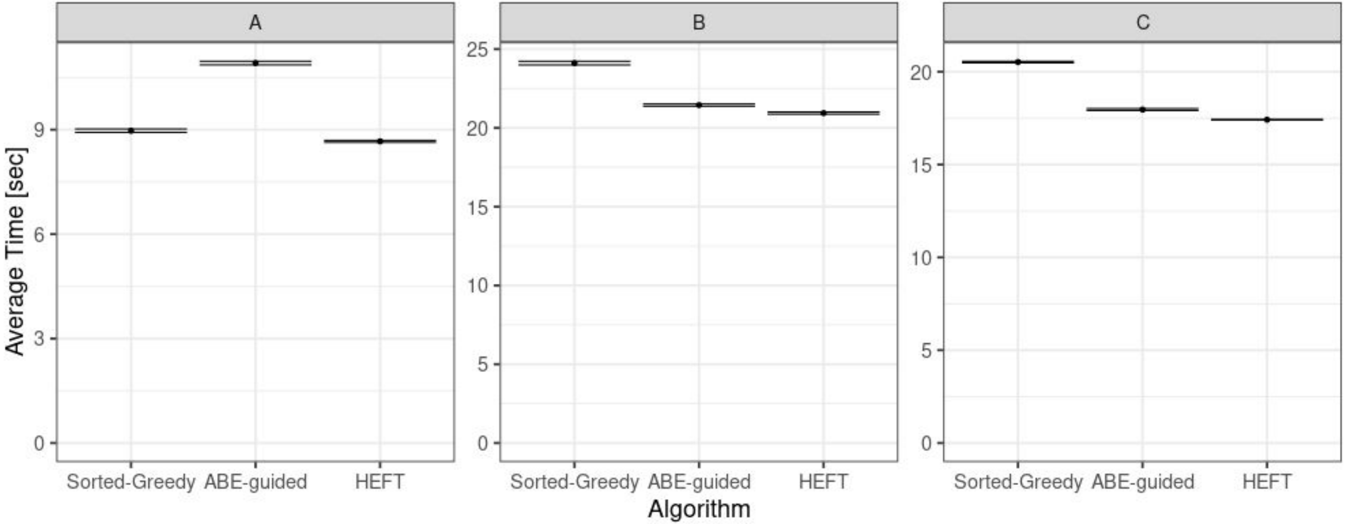
Improvements on the Application

- Previous experiment motivates to **extend** solutions
- Load balancing using **heterogeneous** machines
- **No** changes on processor's frequency
- Two heuristics: **ABE-guided** and **HEFT**
- We selected three different machine combinations scenarios

Study case	draco	hype	cei	blaise	Total Cores
A	7	5	2	1	304
B	3	2	1	0	112
C	0	1	2	1	112

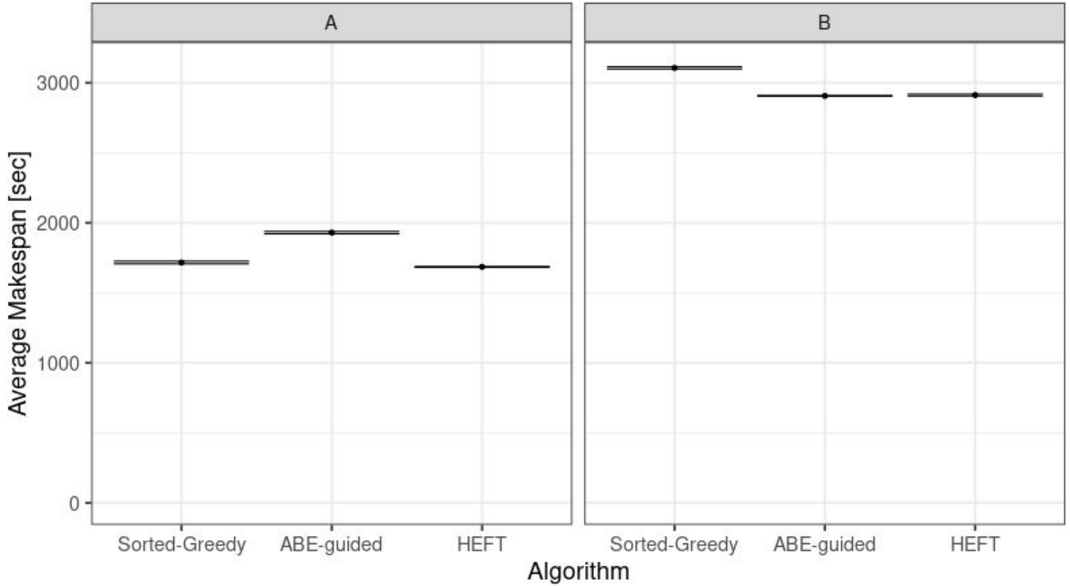
Improvements on the Application

- Compare **iteration time** in each scenario with each heuristic
- 10 repetitions with 10 iterations each



Average time of execution of one iteration in each scheduler running on different heterogeneous configurations.

Improvements on the Application



Average makespan with execution of 100 iterations for each scheduler in two different heterogeneous scenarios.

- **Makespan** reduction with **heterogeneous** solutions
- 100 iterations
- Achieved a performance gain of **78%** against the Original scheduler

Conclusion

- We investigate the computing performance of an MPI-based **inversion application of mCSEM**
- Identified issue regarding **load balancing** and solved it
- Performance improvement
  - **41%** on homogeneous scenarios
  - **78%** on heterogeneous scenarios
- **Capacity planning tool** to further predict best execution scenario
  - Available at bit.ly/interactive-capacity-plan

ACKNOWLEDGMENTS

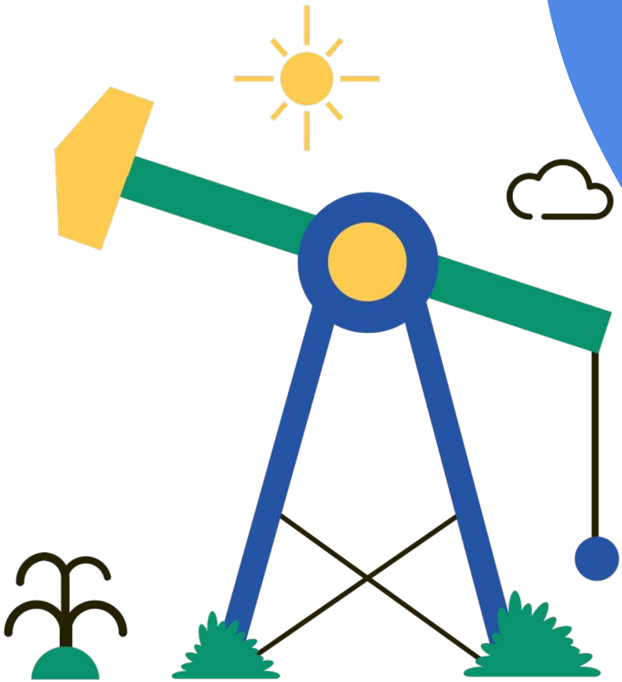
This research was financed by Petrobras (2018/00263-5) and in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Experiments executed in INF/UFRGS's PCAD, http://gppd-hpc.inf.ufrgs.br.

UFRGS UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf INSTITUTO DE INFORMÁTICA UFRGS

BR PETROBRAS

Instituto de Informática - UFRGS
Programa de Pós Graduação em Computação

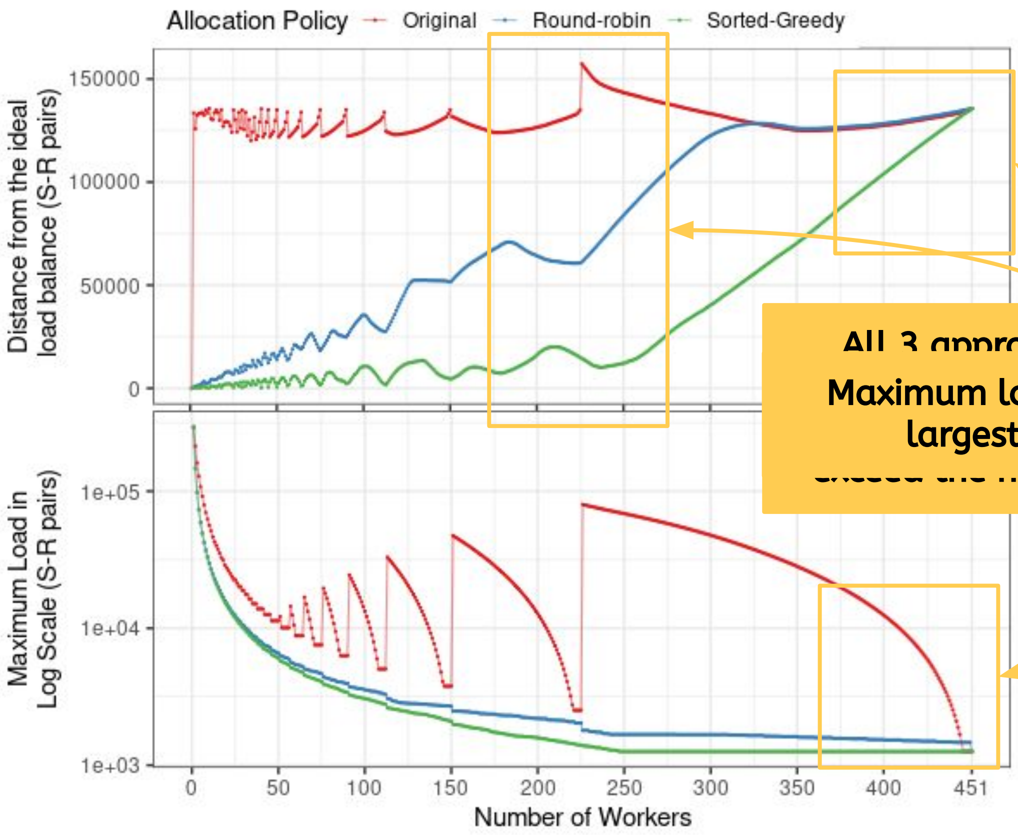


Performance Improvements Applied in an Electromagnetic Inversion Application Focused on Homogeneous and Heterogeneous Computational Environments

Jessica Imlau Dagostini
Advisor: Lucas Mello Schnorr

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
.Inf INSTITUTO DE INFORMÁTICA UFRGS
BR PETROBRAS

Improvements on the Application

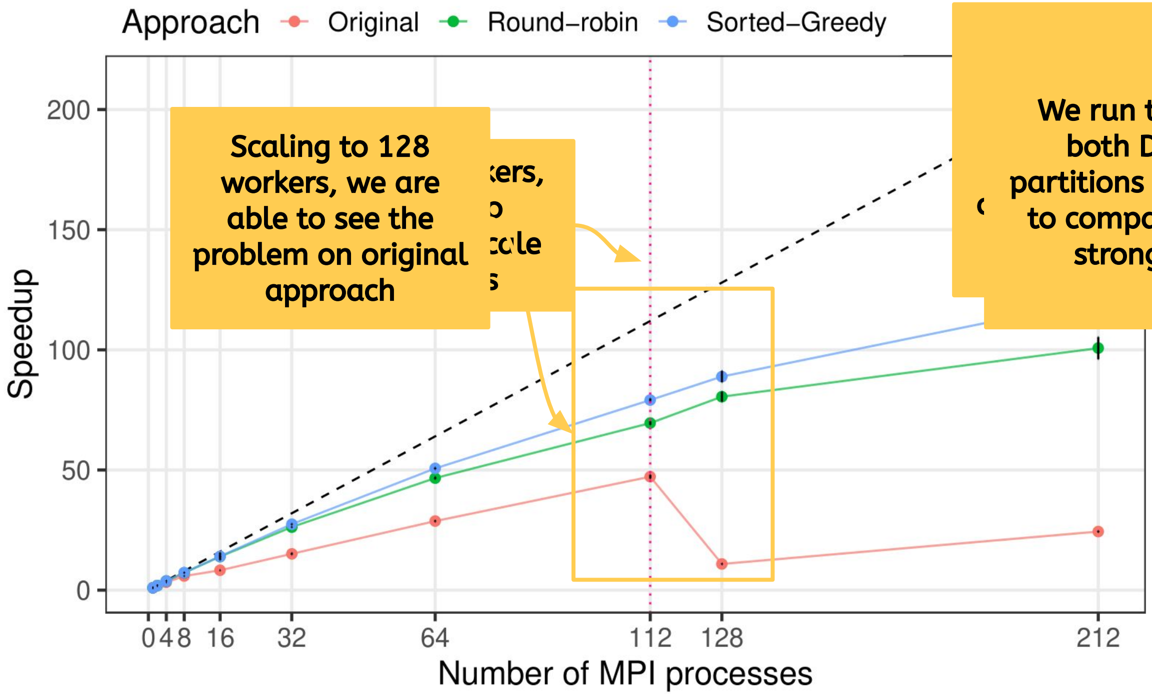


All 3 approaches present an

Maximum load is equal to the largest CMP cell size

Theoretical distribution of load with 1 up to the number of CMP cells of the study case.

Improvements on the Application



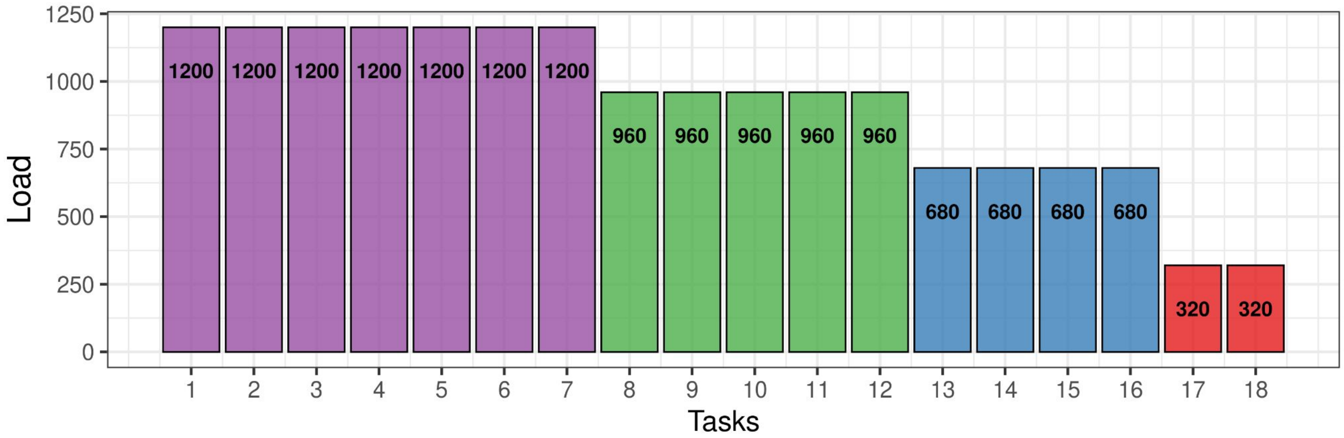
Scaling to 128 workers, we are able to see the problem on original approach

We run the application in both Draco and Hype partitions to have more cores to compare the theoretical strong scaling study

Speedup of many MPI processes from 1 up to 212 using two different partitions with the same processor's frequency.

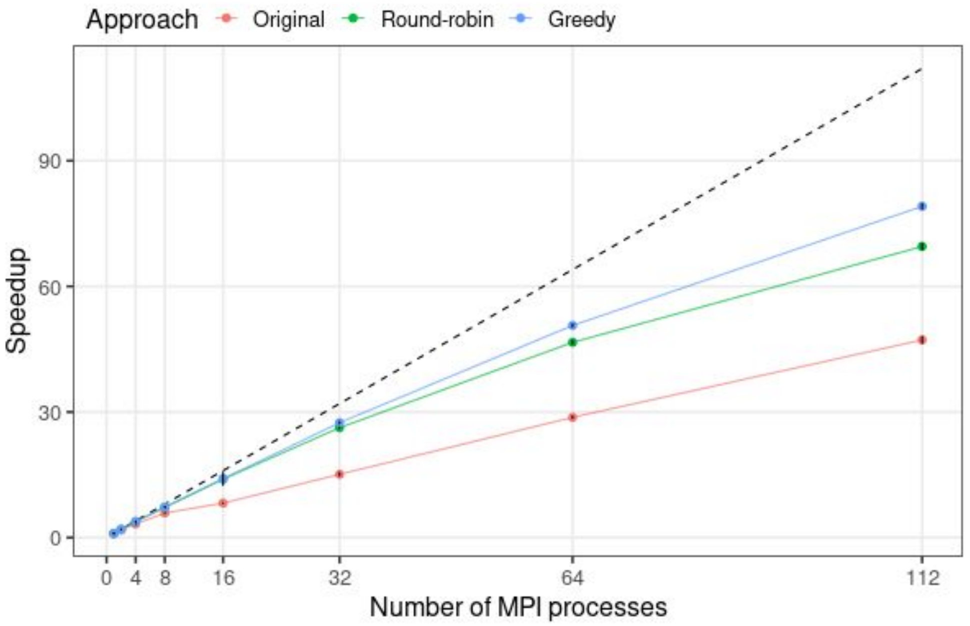
Load Balancing Heuristics

- The characteristics of the load give us a number partitioning problem
- We will use the following synthetic load
- Searched for state-of-art heuristics to solve the problem



A synthetic load characterizing number partitioning.

Improvements on the Application



Speedup of 1 up to 112 MPI processes. We run three iterations of the application. The baseline is Original with one process. The dashed black line represents the ideal speedup.

- **Speedup** of the application
- We compared the two heuristics against the Original using 1 up to 112 MPI processes
- Compute the mean speedup with following equations

$$\mathbf{E}(\text{speedup}) = \mathbf{E}(\text{Original 1 proc}) \times \mathbf{E}\left(\frac{1}{\text{new\_distribution } n \text{ proc}}\right); \text{ and}$$

$$\begin{aligned}\mathbf{Var}(\text{speedup}) = {} & \mathbf{E}(\text{Original 1 proc}^2) \times \mathbf{E}\left(\frac{1}{\text{new\_distribution } n \text{ proc}^2}\right) \\ & - \mathbf{E}^2(\text{original 1 proc}) \times \mathbf{E}^2\left(\frac{1}{\text{new\_distribution } n \text{ proc}}\right).\end{aligned}$$