Tuning space optimization for multi-core architectures

V. Martínez, F. Dupros, M. Castro,
H. Aochi and P. Navaux

INSTITUTO DE INFORMÁTICA UFRGS

Geoscience for a sustainable Earth
brgm

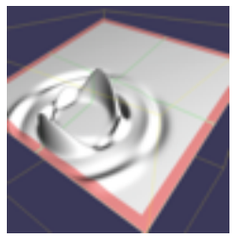
Informática Estatística
ine
CTC • UFSC

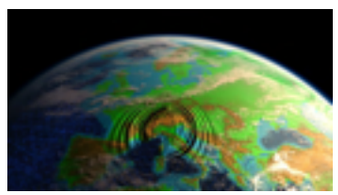
Contents

- **Introduction.**
  - HPC applications Performance
- Stencil Model.
- Testbed configuration.
- Experiments.
- Results.
- Conclusion.

Scientific Applications

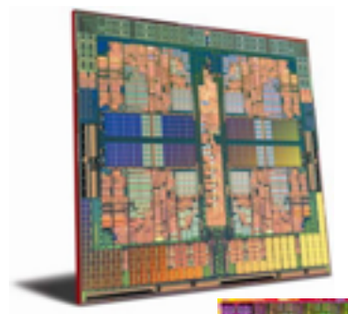
Applications

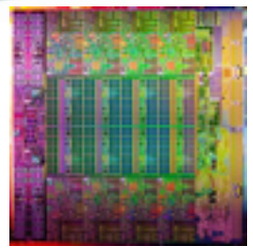




Stencil Applications:
Data dependency

Multi-core





Contribution

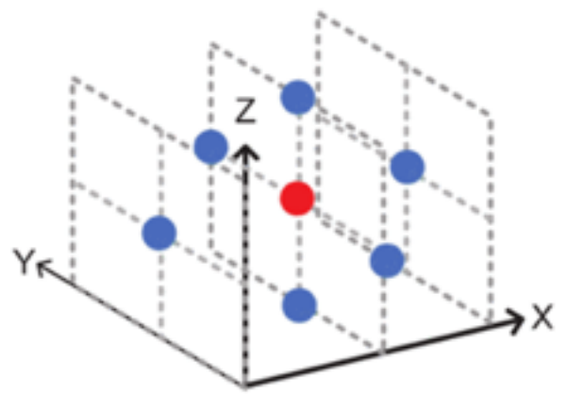
To find best configuration of runtime parameters (tuning) for stencil computations based on number of available threads and L3 CM reduction

Contents

- Introduction.
- **Stencil Model.**
  - Jacobi 7-point
- Testbed configuration.
- Experiments.
- Results.
- Conclusion.

Stencil: 7-point Jacobi

- 3D Stencil.
  - Heat equation
  - Finite difference method.



- Calculate:

$$B_{i,j,k} = \alpha A_{i,j,k} + \beta(A_{i-1,j,k} + A_{i,j-1,k} + A_{i,j,k-1} + A_{i+1,j,k} + A_{i,j+1,k} + A_{i,j,k+1})$$

Algorithm 1 Pseudocode for stencil algorithm

```
1:  for each timestep do
2:      Compute in parallel
3:      for each block in X-direction do
4:          for each block in Y-direction do
5:              for each block in Z-direction do
6:                  Compute stencil(3D tile)
7:              end for
8:          end for
9:      end for
10: end for
```

Contents

- Introduction.
- Stencil Model.
- **Testbed configuration.**
  - Hardware/Application Setup.
- Experiments.
- Results.
- Conclusion.

Experiments (Testbed)

	Node 1	Node 2
Processor	i5-4570	Xeon X7550
Clock (GHz)	3.2	2.0
Cores	4	8
Sockets	1	4
Threads	4	64
L3 Cache size (MB)	6	18
Compiler	gcc-4.6.4	gcc-4.6.4

Contents

- Introduction.
- Stencil Model.
- Testbed configuration.
- **Experiments.**
- Results.
- Conclusion.

Experiments (Setup)

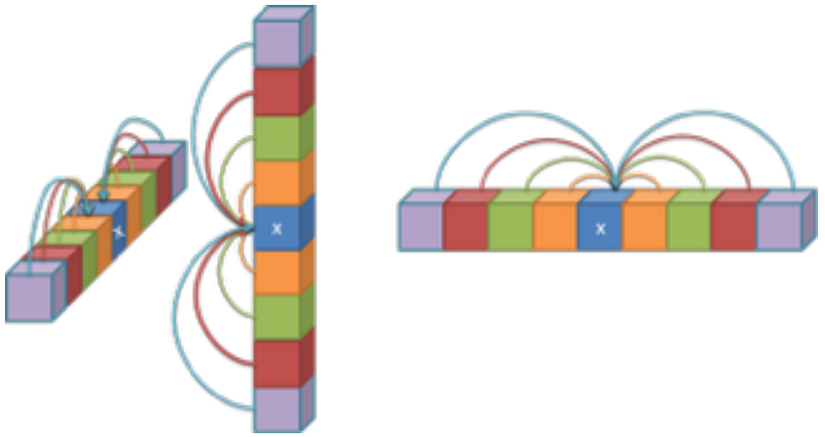
Input vector	Total configurations	
	Node 1	Node 2
Threads	2	6
Looping	2	2
Size	3	3
Chunk	4	4
Scheduling	3	3
Total	144	432

- Output:
  - Cache misses: (PAPI_L3_TCM)
  - Cache Accesses: (PAPI_L3_TCA)
  - Time
  - GFLOPS

Algorithms

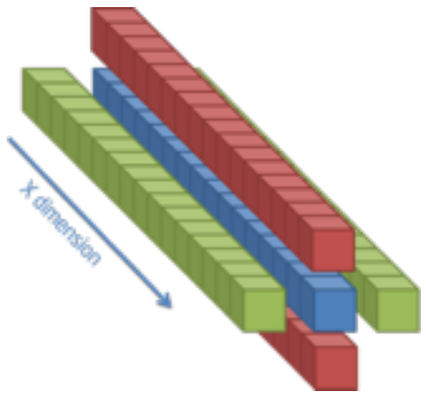
Naive

- Triple nested loops coming from the three spatial dimensions



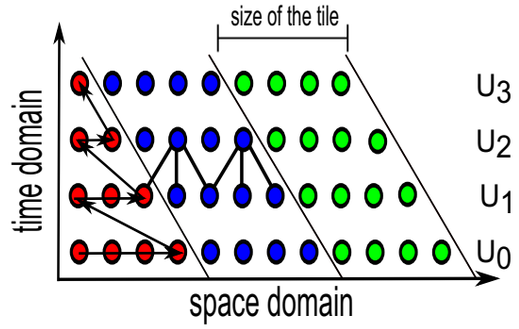
Blocking

- Dependencies between components are exploited to implement a space-time decomposition.



Skew

- Decompose the stencil using both the space and the time directions but in a specific order.

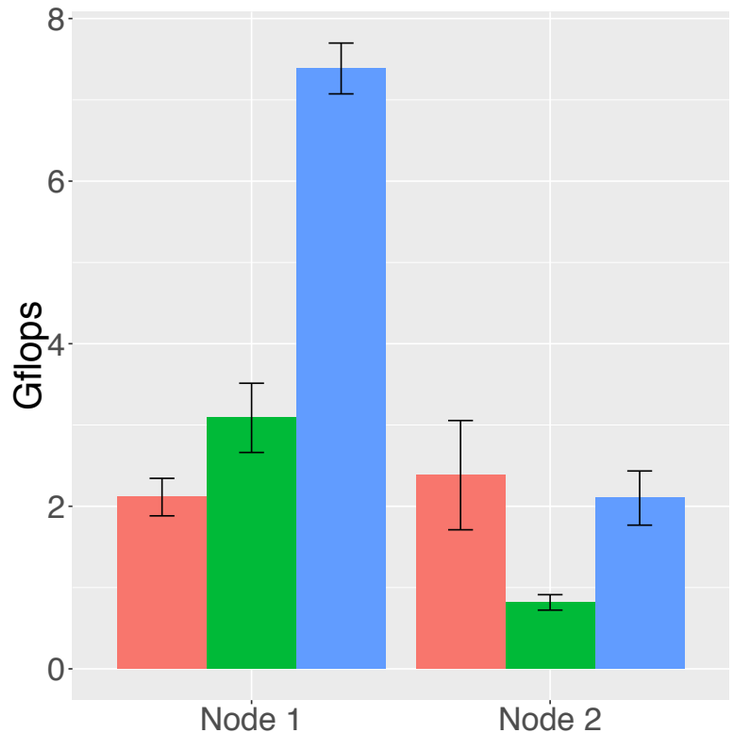


Contents

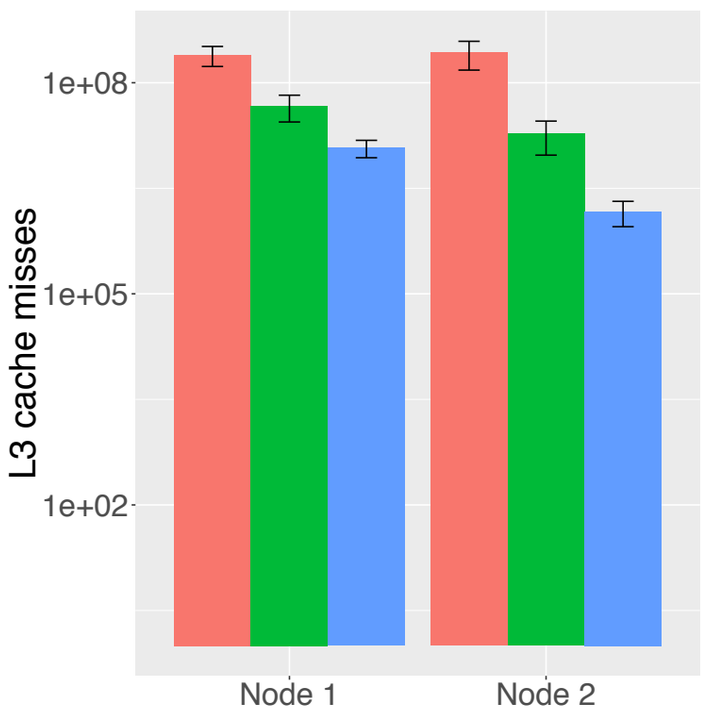
- Introduction.
- Stencil Model.
- Testbed configuration.
- Experiments.
- **Results.**
  - Tuning
- Conclusion.

Results (Algorithms)

Performance



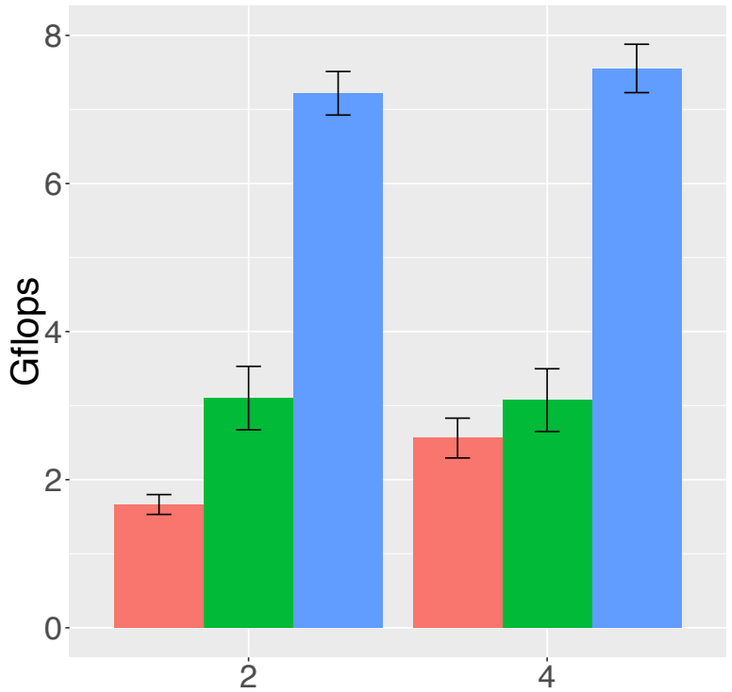
Cache Misses



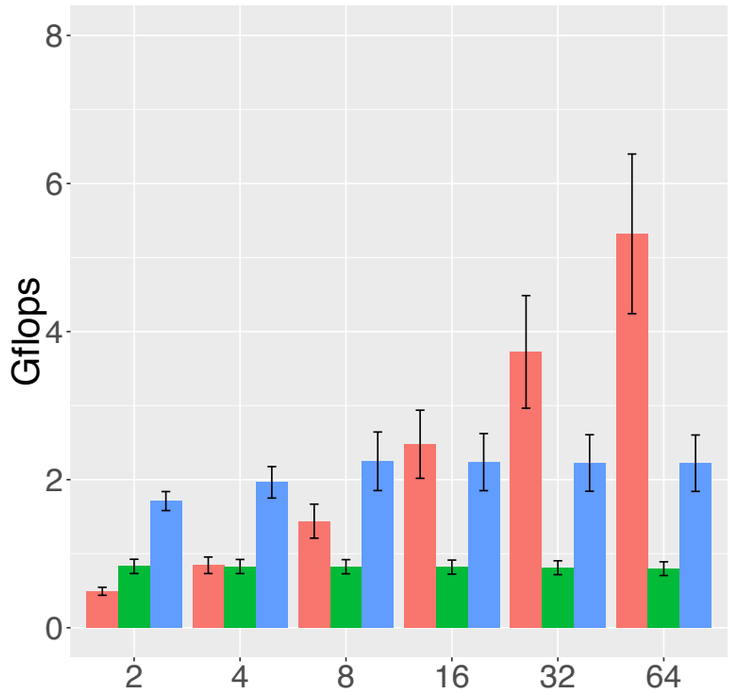
Naive (magenta), Blocking (green) and Skew(cyan)

Results (Scalability)

Node 1



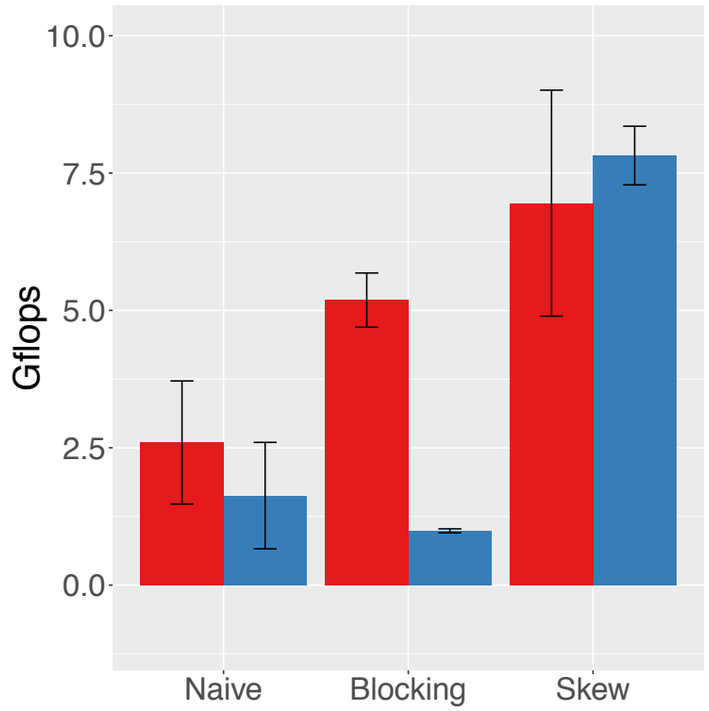
Node 2



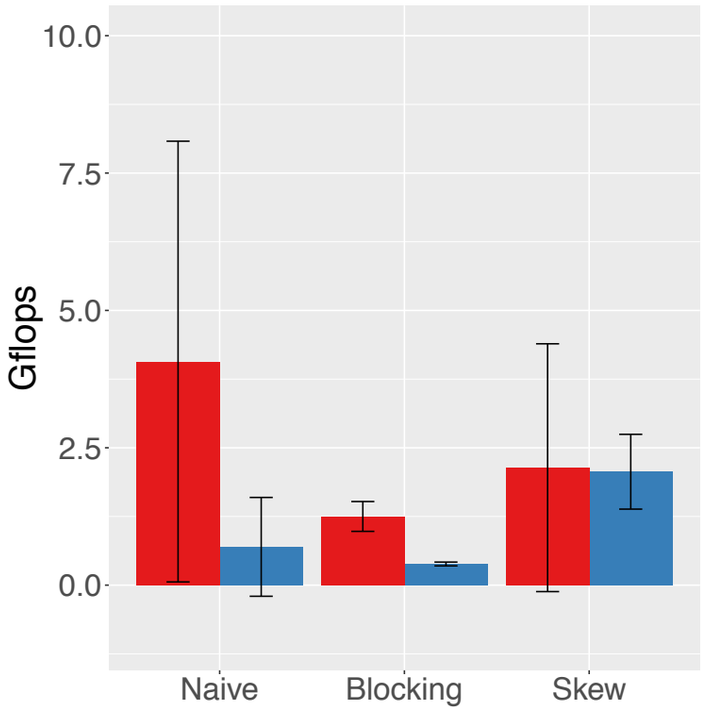
Naive (magenta), Blocking (green) and Skew(cyan)

Results (code optimization)

Node 1



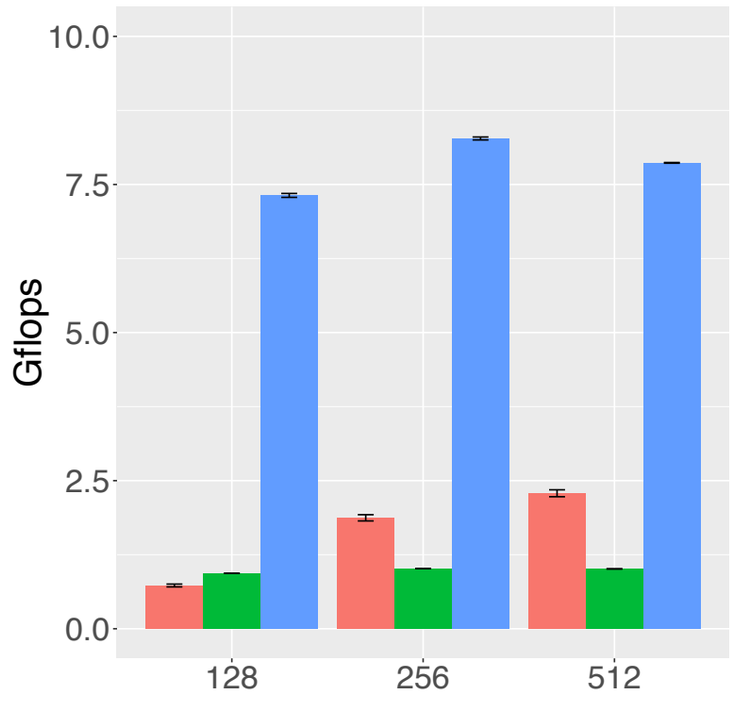
Node 2



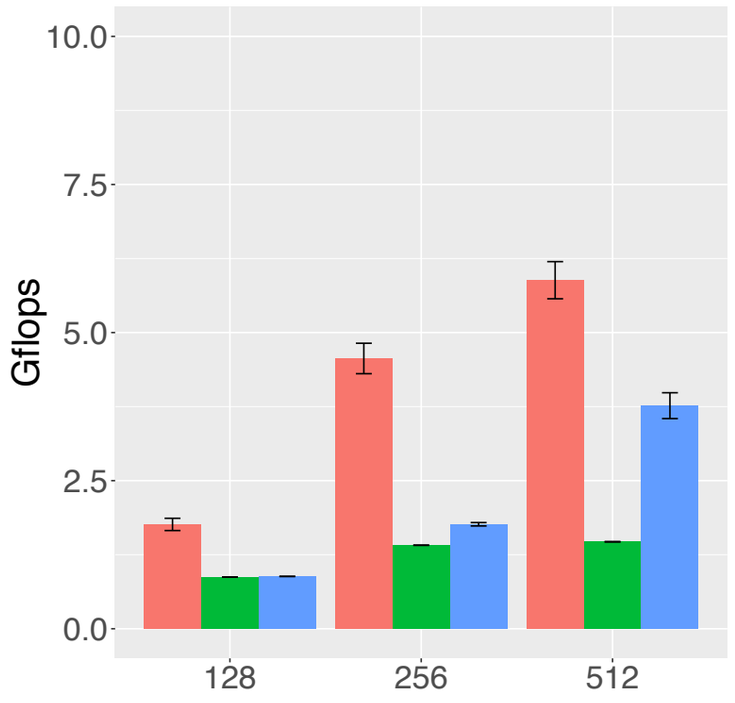
Parallefor (red), Tasking (blue)

Results (Problem size)

Node 1 - Tasking



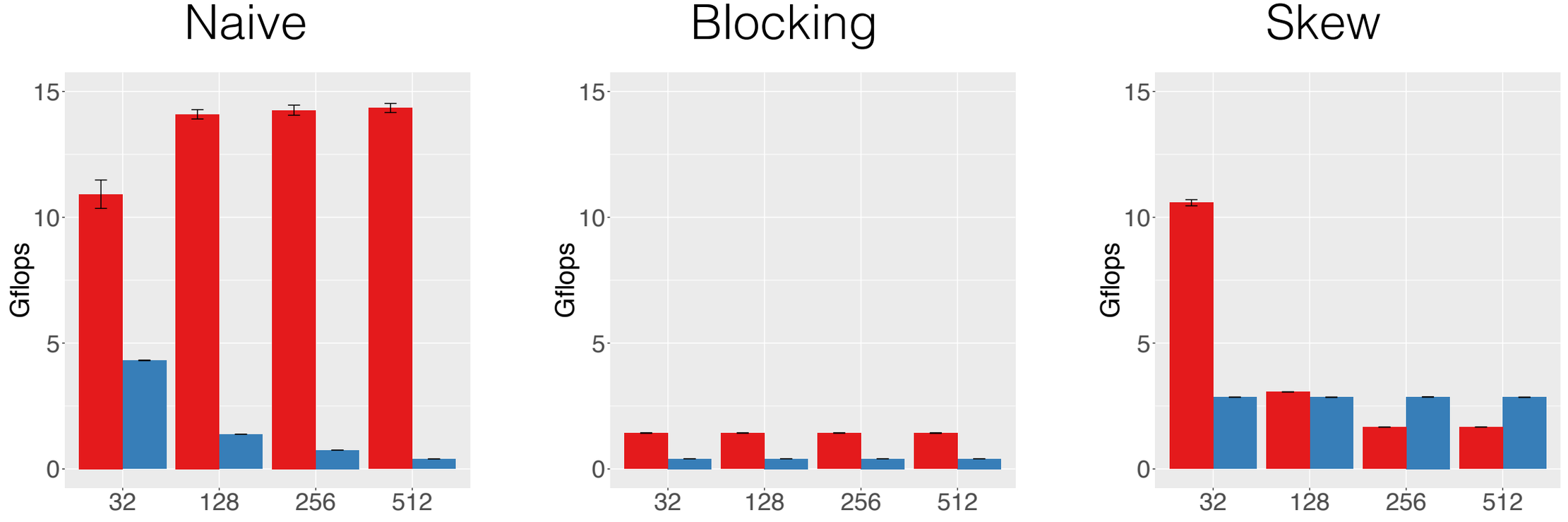
Node 2 - Parallelfor



Naive (magenta), Blocking (green) and Skew(cyan)

Results (Scheduling)

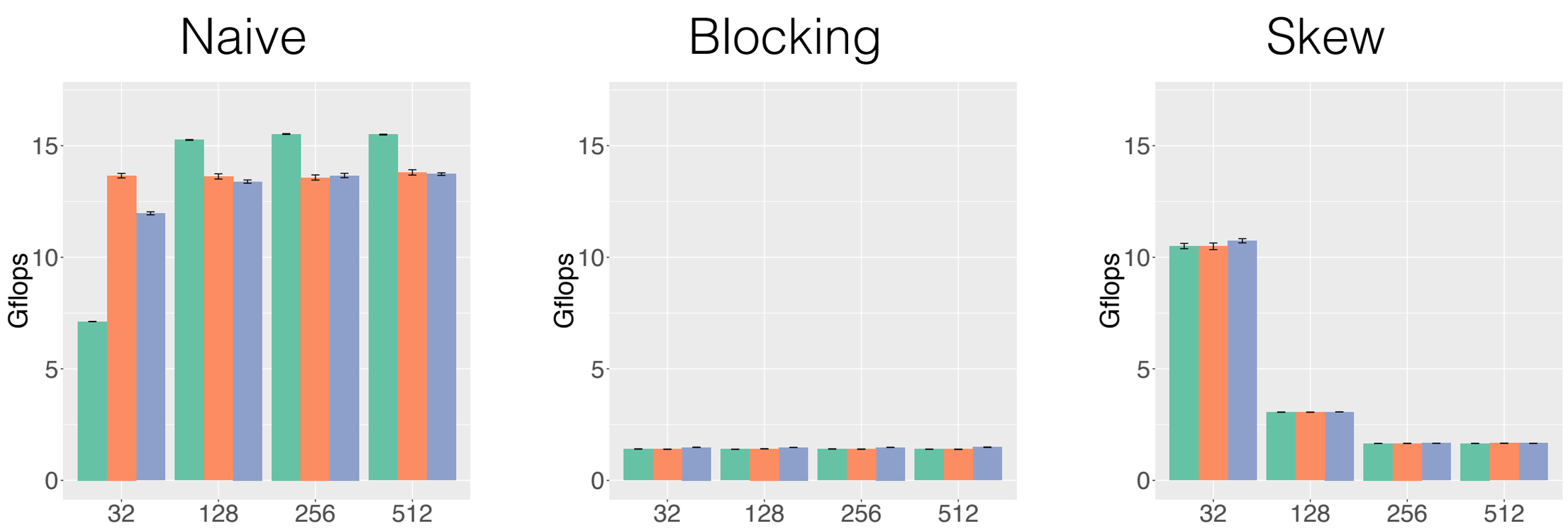
Chunk size (Node 2)



Parallelfor (red), Tasking (blue)

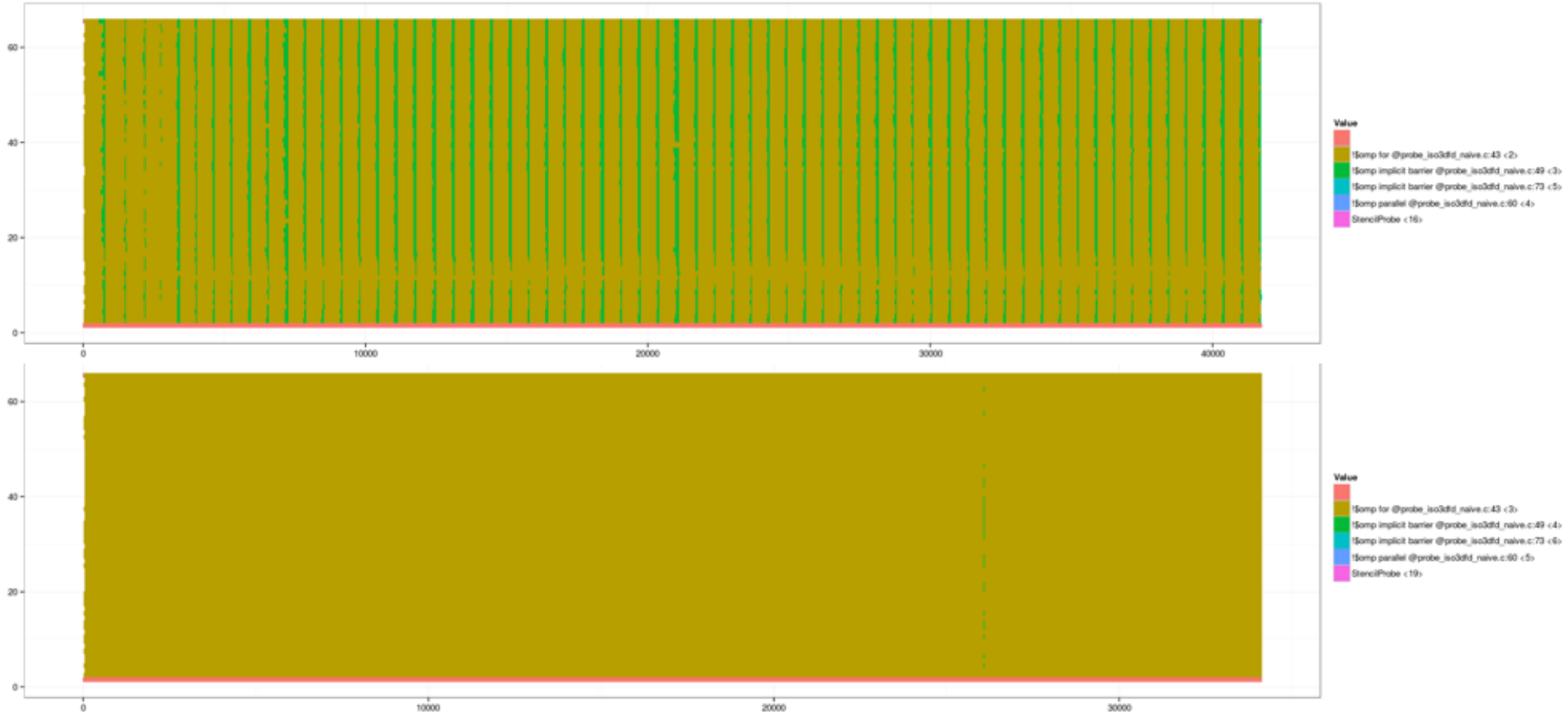
Results (Scheduling)

Policy (Node 2)

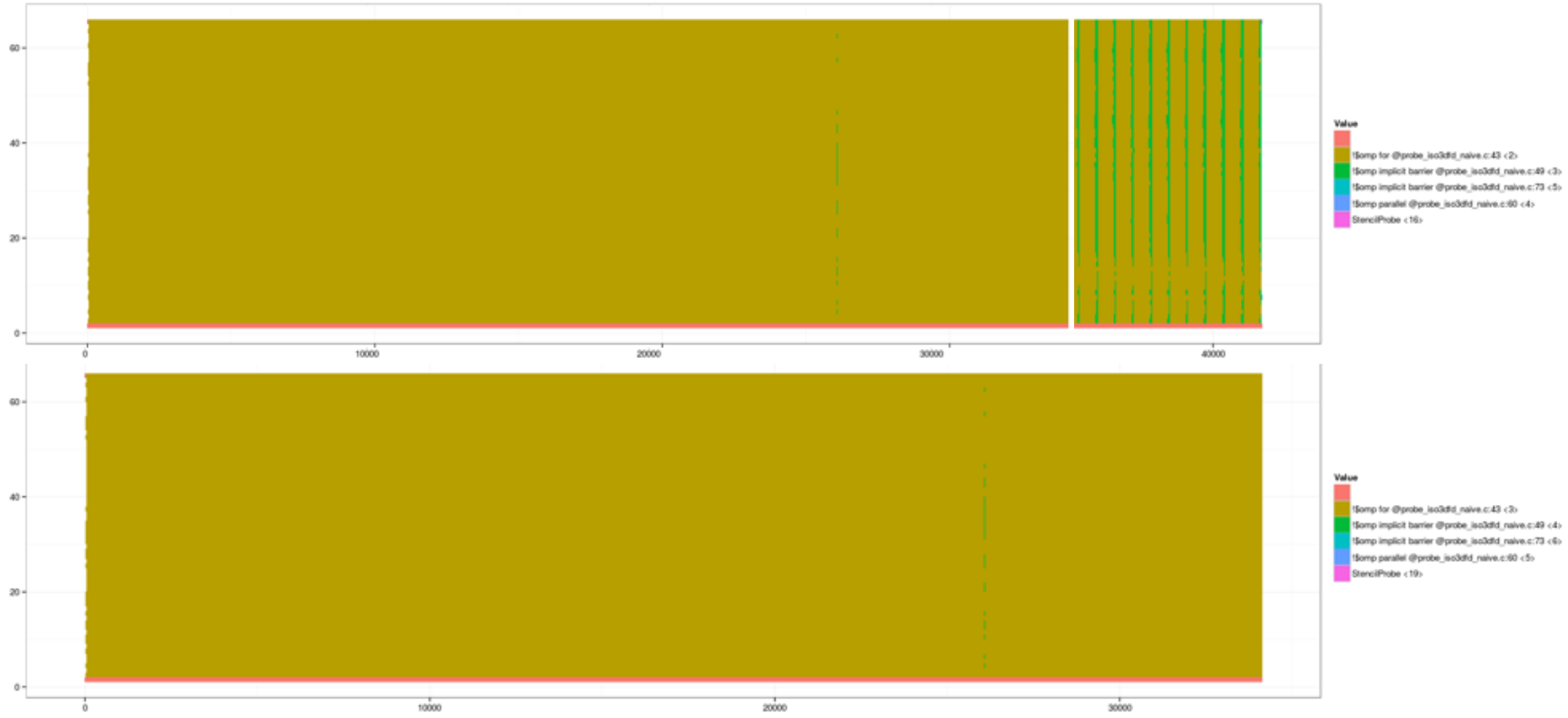


Dynamic (green), Guided (Orange) and Static (Gray)

Makespan (Naive parallelfor)



Makespan (Naive parallelfor)

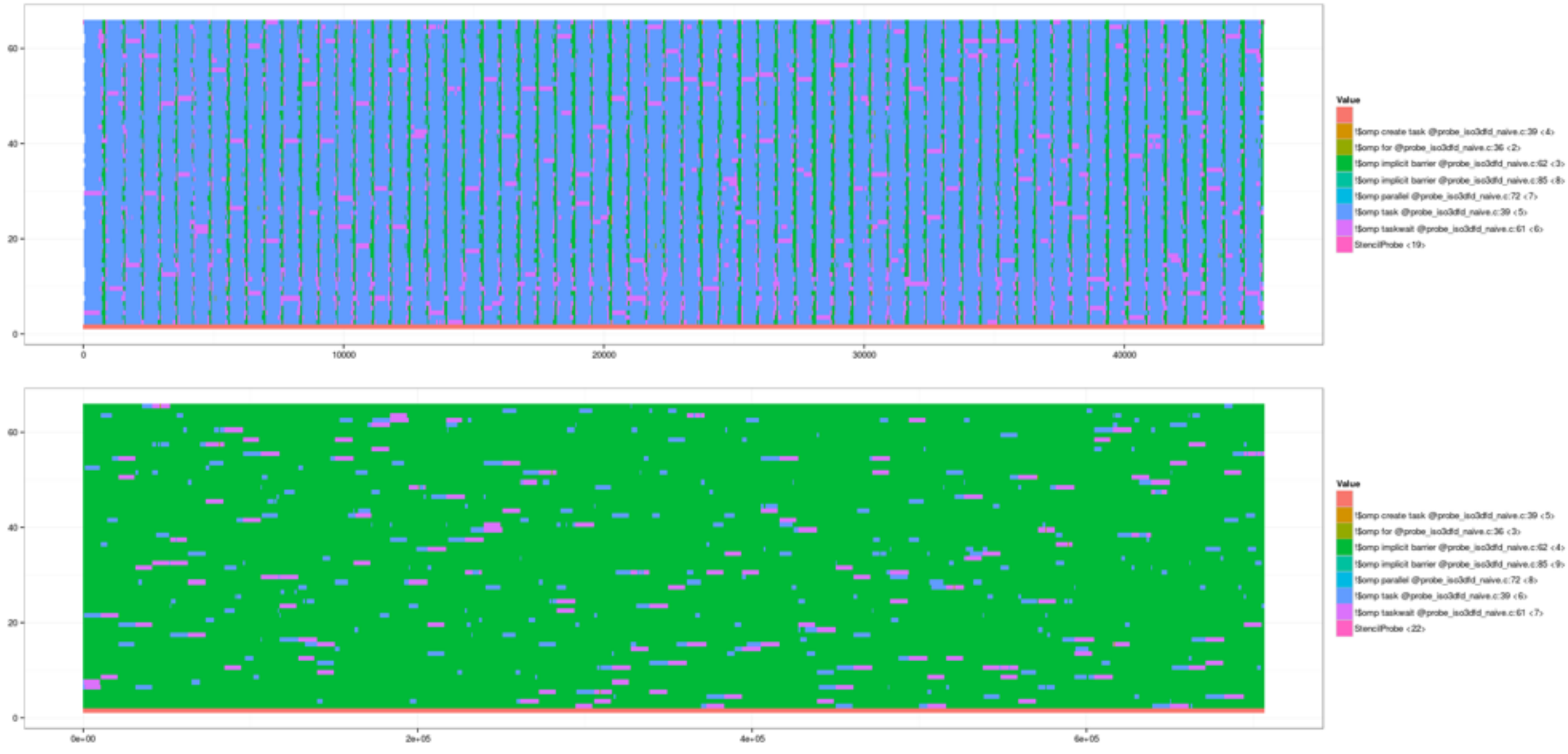


INSTITUTO DE INFORMÁTICA UFRGS

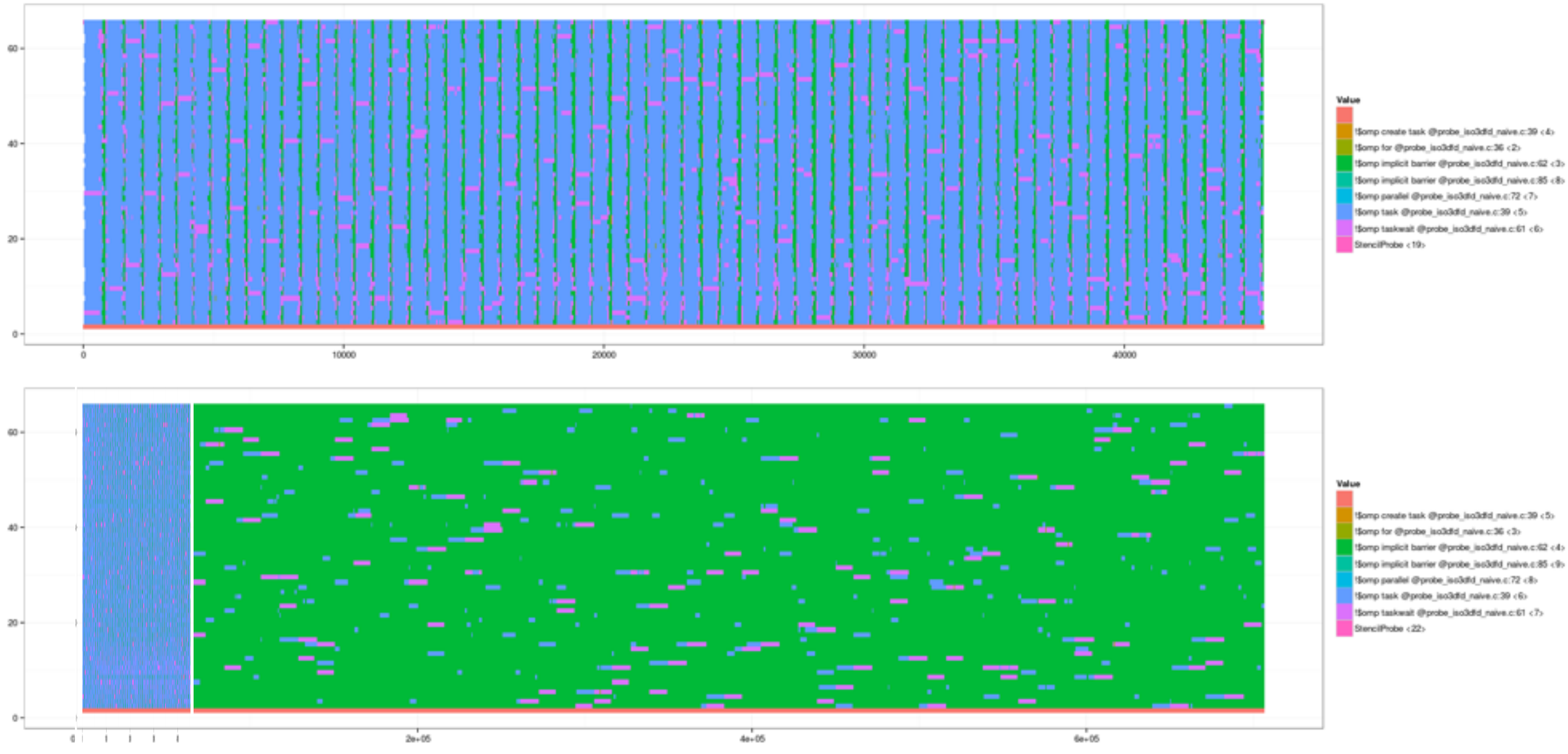
Geoscience for a sustainable Earth brgm

ine Informática Estatística CTC • UFSC

Makespan (Naive tasking)



Makespan (Naive tasking)



Contents

- Introduction.
- Stencil Model.
- Testbed configuration.
- Experiments.
- Results.
- **Conclusion.**

Conclusion

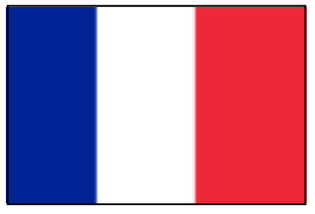
- Tasking achieves good performance when the algorithm does not use cache intensively (Skew).

- Chunk size and scheduling algorithms (OpenMP) play an important role for Naive algorithm and can contribute to achieve a peak of performance.

- Fitting: Cache misses can be approximated by linear fitting whereas the performance could be predicted by exponential fitting

- Future: we intend to develop an auto-tuning approach to automatize the choice of the input parameters





Thanks

Questions?

victor.martinez@inf.ufrgs.br

INSTITUTO DE INFORMÁTICA UFRGS

Geoscience for a sustainable Earth brgm

ine Informática Estatística CTC • UFSC