# On the implementation of a flexible, scalable simulator for the family of MHM methods

*In homage to*
*Prof. Luiz Fernando Gomes Soares.*
*NTPPM!*

**Antônio Tadeu A. Gomes**
atagomes@lncc.br

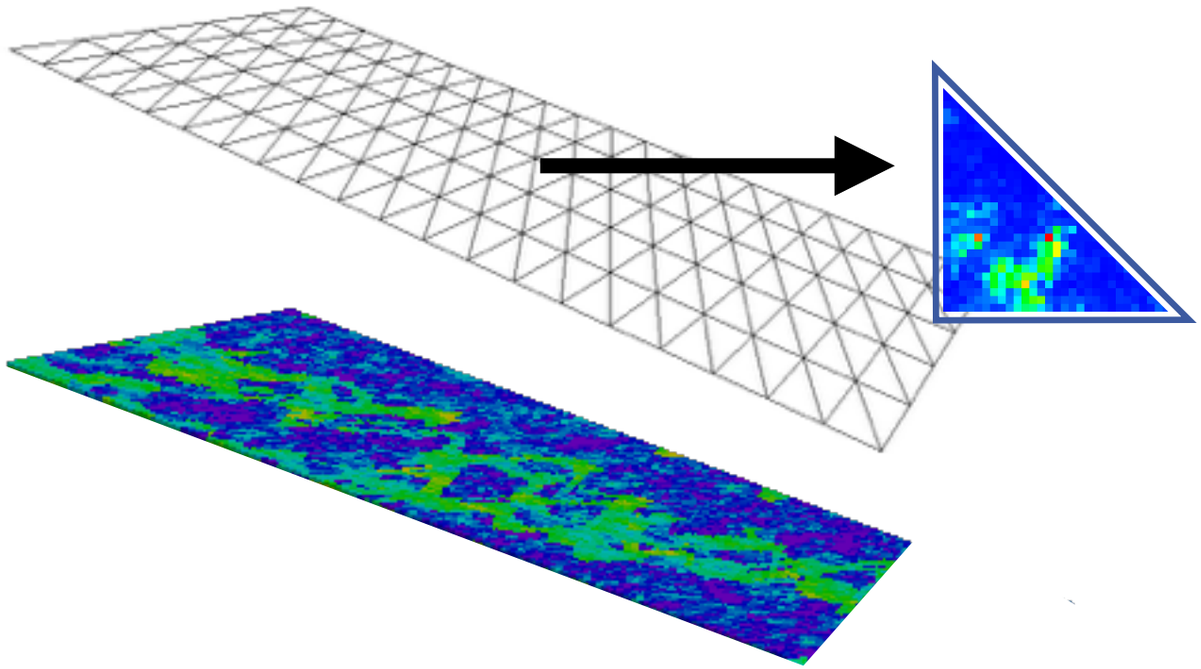
**Frederic Valentin**
**Diego Paredes**

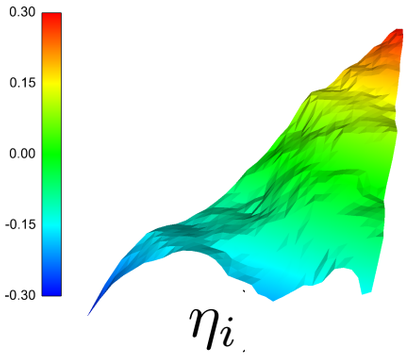
PONTIFICIA UNIVERSIDAD CATOLICA DE VALPARAISO

Laboratório Nacional de Computação Científica

# Multiscale Hybrid Mixed Methods: MHM in a nutshell...

Laboratório Nacional de Computação Científica





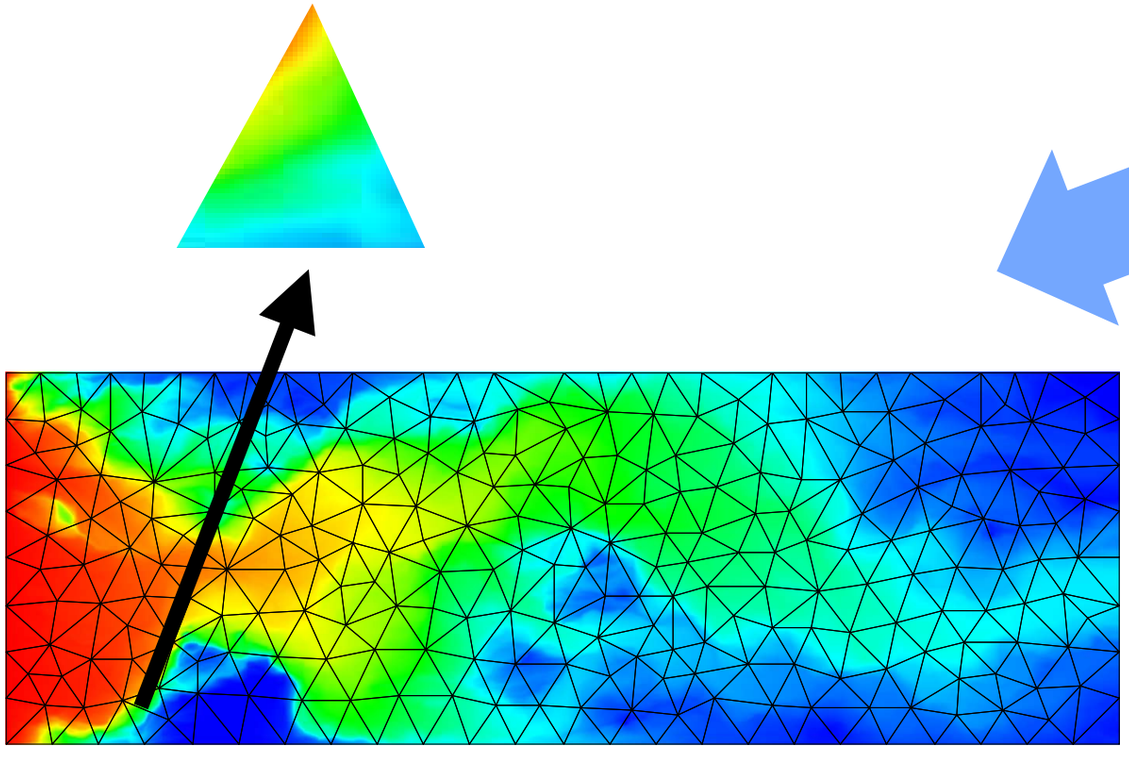

Find $u^f$ such that

$$\left.\begin{aligned} \mathcal{L}u^f &= f, \text{ in } K \\ \boldsymbol{\sigma}(u^f)\cdot\boldsymbol{n}^{\partial K} &= 0, \text{ on } \partial K \end{aligned}\right\}$$

Find $\eta_i$ such that

$$\left.\begin{aligned} \mathcal{L}\eta_i &= 0, \text{ in } K \\ \boldsymbol{\sigma}(\eta_i)\cdot\boldsymbol{n}^{\partial K} &= \psi_i, \text{ on } \partial K \end{aligned}\right\}$$

$$u_h = p_0 + u^{\lambda_h} + u^f = p_0 + \sum_{i=1}^{N_h} c_j\eta_i + u^f$$



$$\lambda_h = \sum_{j=1}^{L_h} c_j\psi_j$$

*Find* $(p_0, \lambda_h) \in V_0 \times \Lambda_h$

$$\begin{aligned} (\lambda_h, q_0)_{\partial\mathcal{T}_h} &= (f, q_0)_{\mathcal{T}_h} \\ (p_0, \mu_h)_{\partial\mathcal{T}_h} + (p^{\lambda_h}, \mu_h)_{\partial\mathcal{T}_h} &= -(\eta^f, \mu_h)_{\partial\mathcal{T}_h} \end{aligned}$$

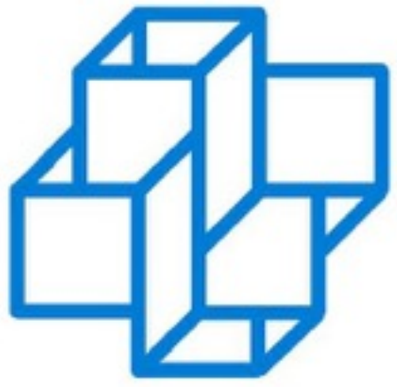
*for all* $(q_0, \mu_h) \in V_0 \times \Lambda_h$.

# Agenda

Laboratório Nacional de Computação Científica

- Architecture
- Preliminary tests (Gramado, 2014)
- Further developments
- What's next?

# Architecture



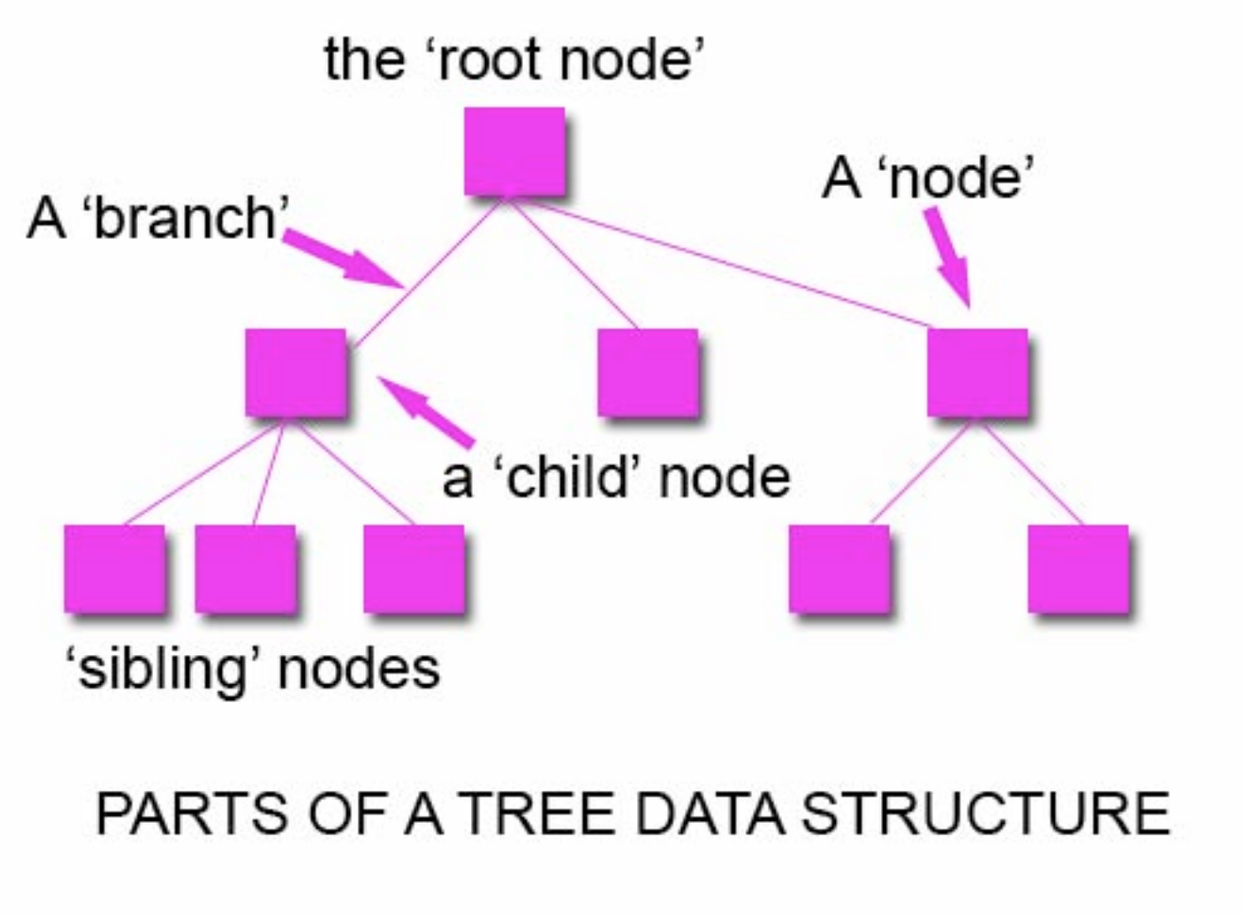
Laboratório
Nacional de
Computação
Científica

# Starting point…

Laboratório Nacional de Computação Científica

- Implementation of a new FEM library specifically crafted for exploring the loosely-coupled strategy of MHM methods to solve global and local problems

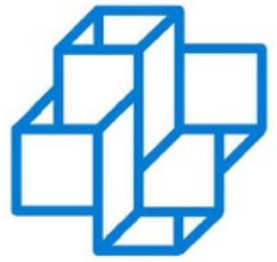
  - Analogy with a tree




PARTS OF A TREE DATA STRUCTURE

**Source:** www.teach-ict.com

# Key design principle: Productivity



Laboratório Nacional de Computação Científica

- Multi-language approach:
  - Erlang for distribution, communication and fault tolerance

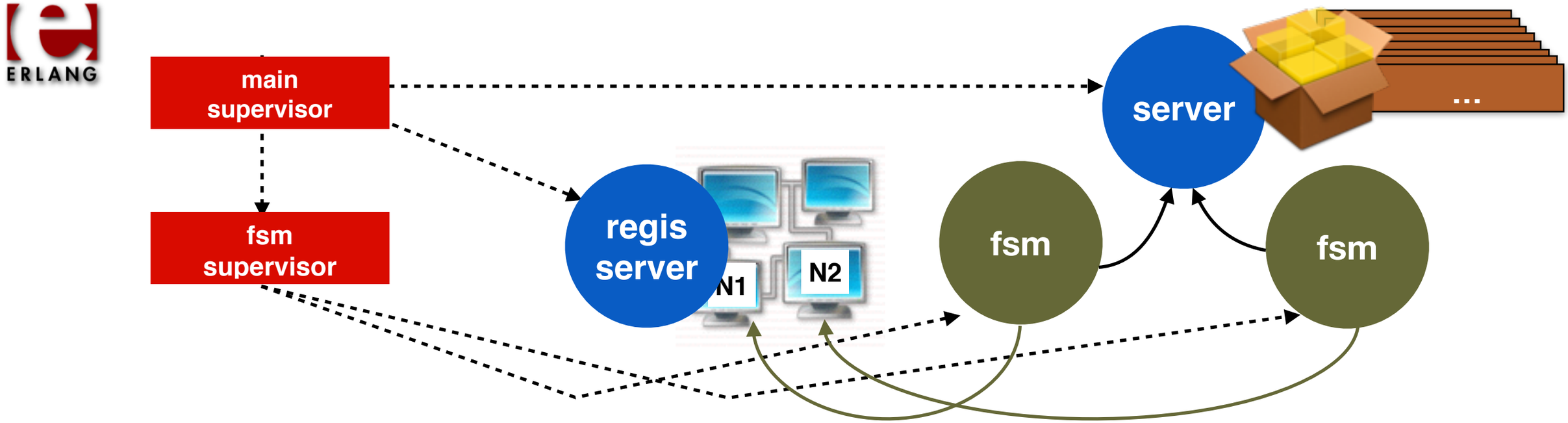

  - C++ for number crunching
    - Reuse of existing efficient libraries (Eigen, METIS, Tetgen…)
    - Could be anything else! (more on this…)

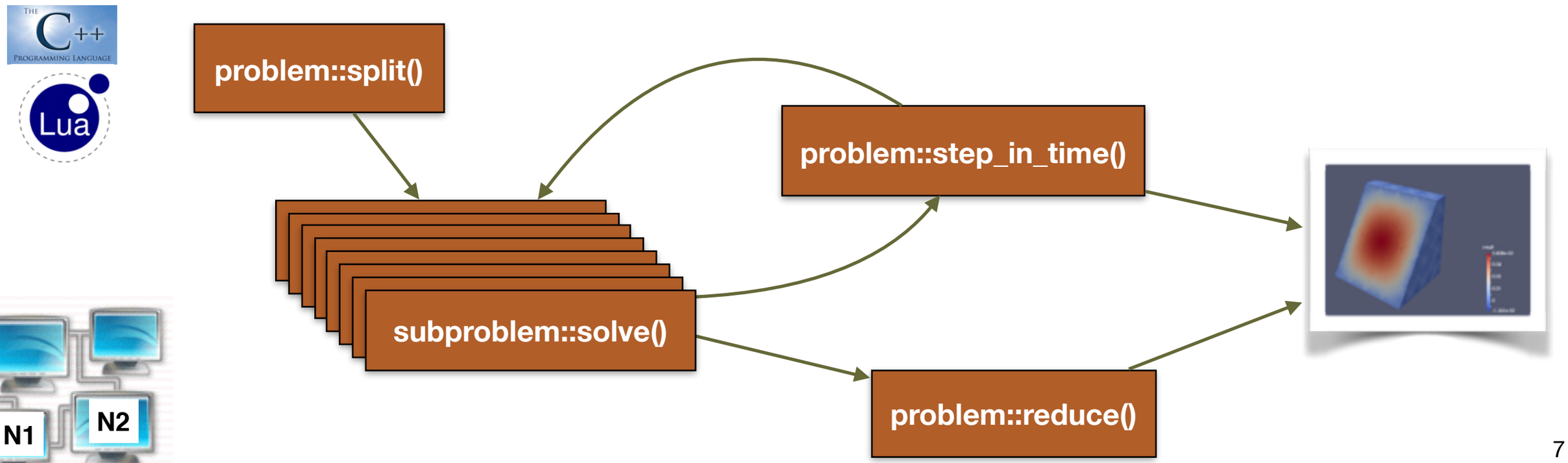


- Lua for rapid prototyping and (in the future) high-level configuration

# Software "architecture"

Laboratório Nacional de Computação Científica

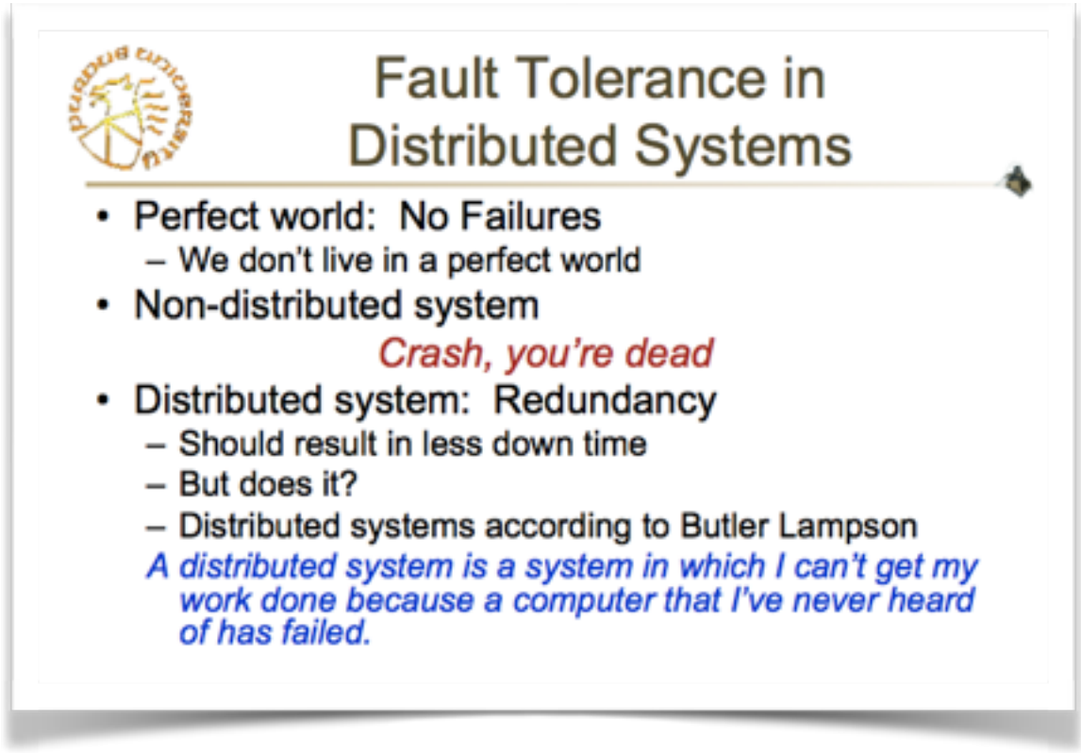


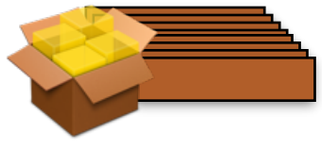
# Fault tolerance: let it crash!

Laboratório Nacional de Computação Científica

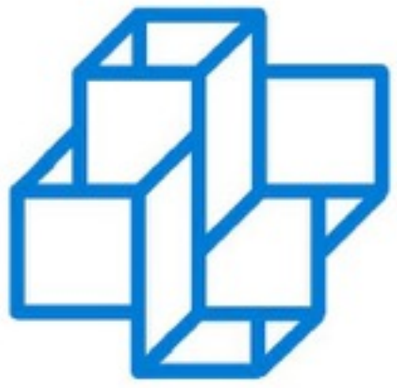
- At the worker (fsm) nodes: **fsm**
  - Network partitions and node failures are considered permanent
  - Related “being solved”/ “being reduced" task is rolled back
- At the master (server) node: **server**
  - Network partitions and node failures are considered transient
  - All tasks in “being solved list”/ “being reduced" are rolled back
  - Backup master node does the failover




Source: https://www.cs.purdue.edu/homes/clifton/cs603/FailureModel.ppt



# Preliminary tests (Gramado, 2014)



Laboratório
Nacional de
Computação
Científica

# Preliminary tests

Laboratório Nacional de Computação Científica

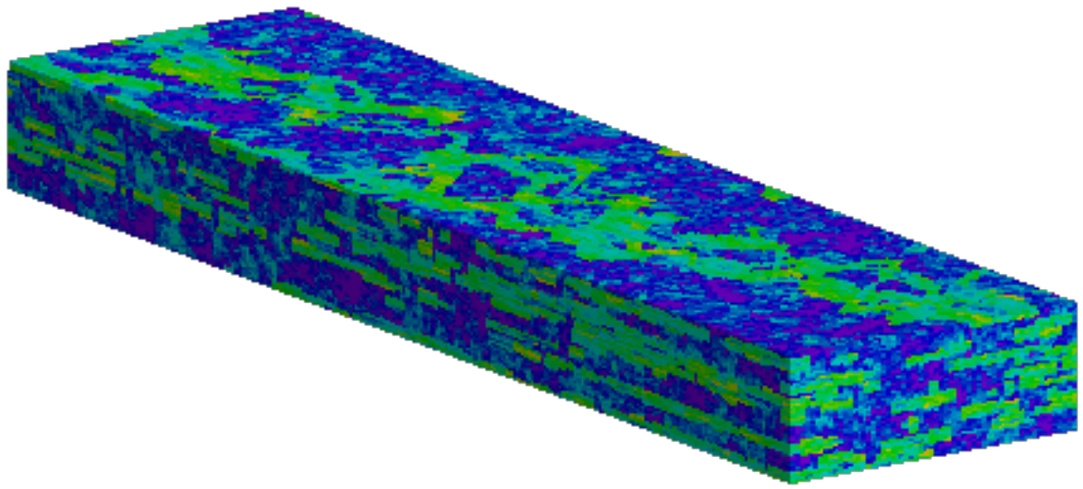
- Testing problem — Darcy (fluid flow through a porous medium)

$$\textit{Find } p = p(\mathbf{x}) \text{ such that}$$

$$\begin{cases} \operatorname{div}(-K\nabla p) = f \text{ in } \Omega \\ p = 0 \text{ in } \Gamma \end{cases}$$

$K$ is a positive definite tensor

$K(\mathbf{x})$ and $f(\mathbf{x})$ **may** have multi-scale characteristics

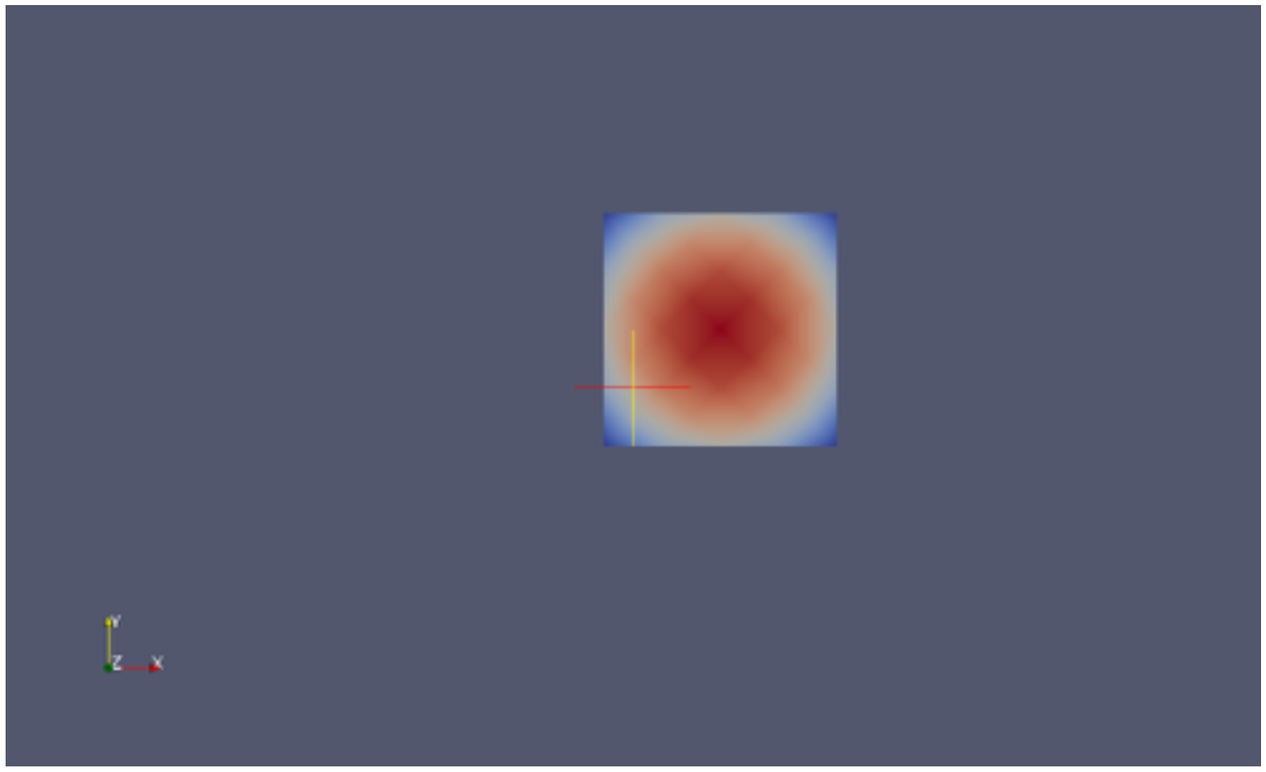


# Preliminary tests

Laboratório Nacional de Computação Científica

In MHM 2D, 256 (G=16, L=16) elements

f(x,y) = 1.0



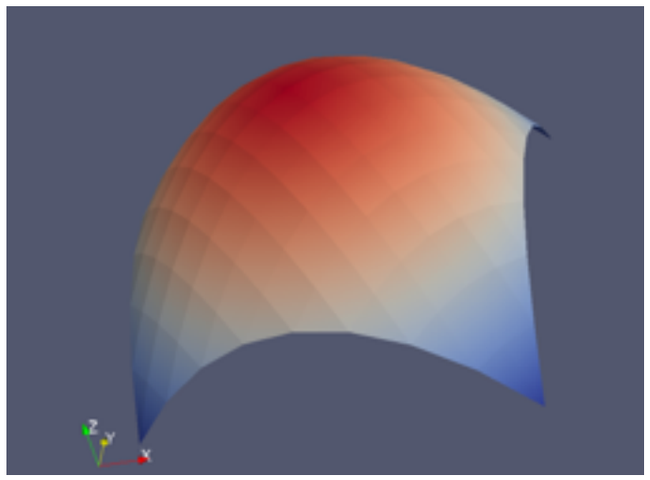
Total number of integration points ($P_E$=2,$P_F$=0):

$$\underbrace{G * 3 * L}_{\text{within 1st level elements}} + \underbrace{G * 2 * 3 * \sqrt{L}}_{\text{on 1st level faces}}$$

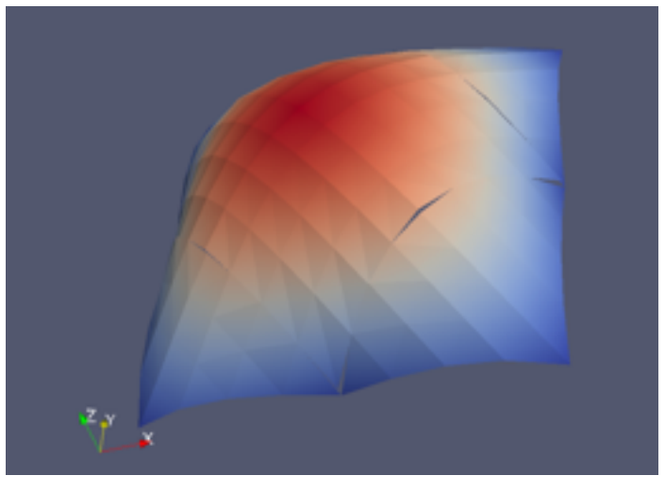
# Preliminary tests

Laboratório Nacional de Computação Científica

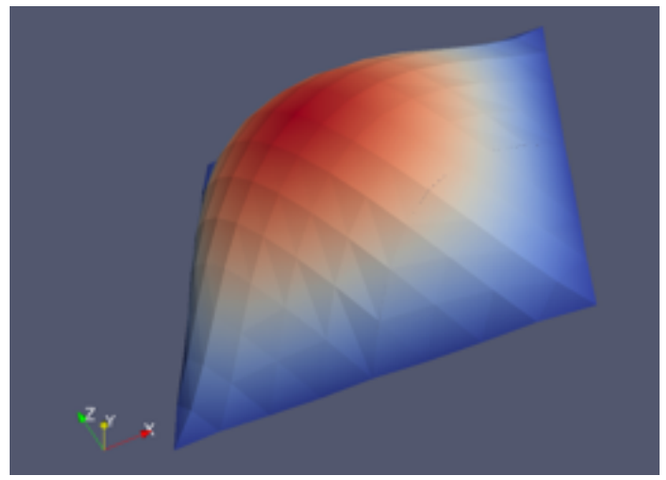
- In MHM 2D, 256 (G=16, L=16) elements



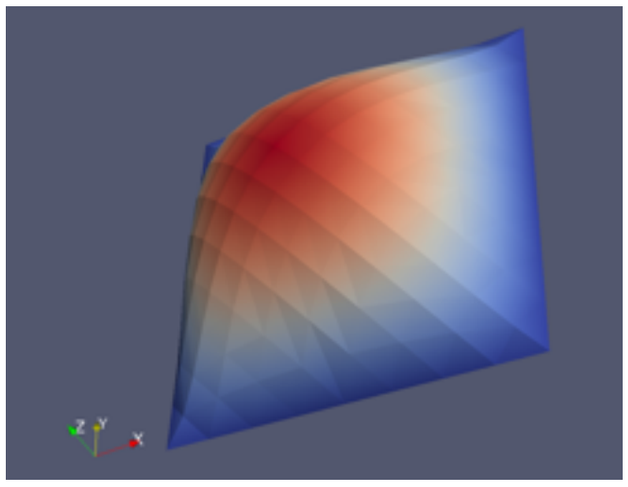
$P_E=2, P_F=0$



$P_E=2, P_F=1$



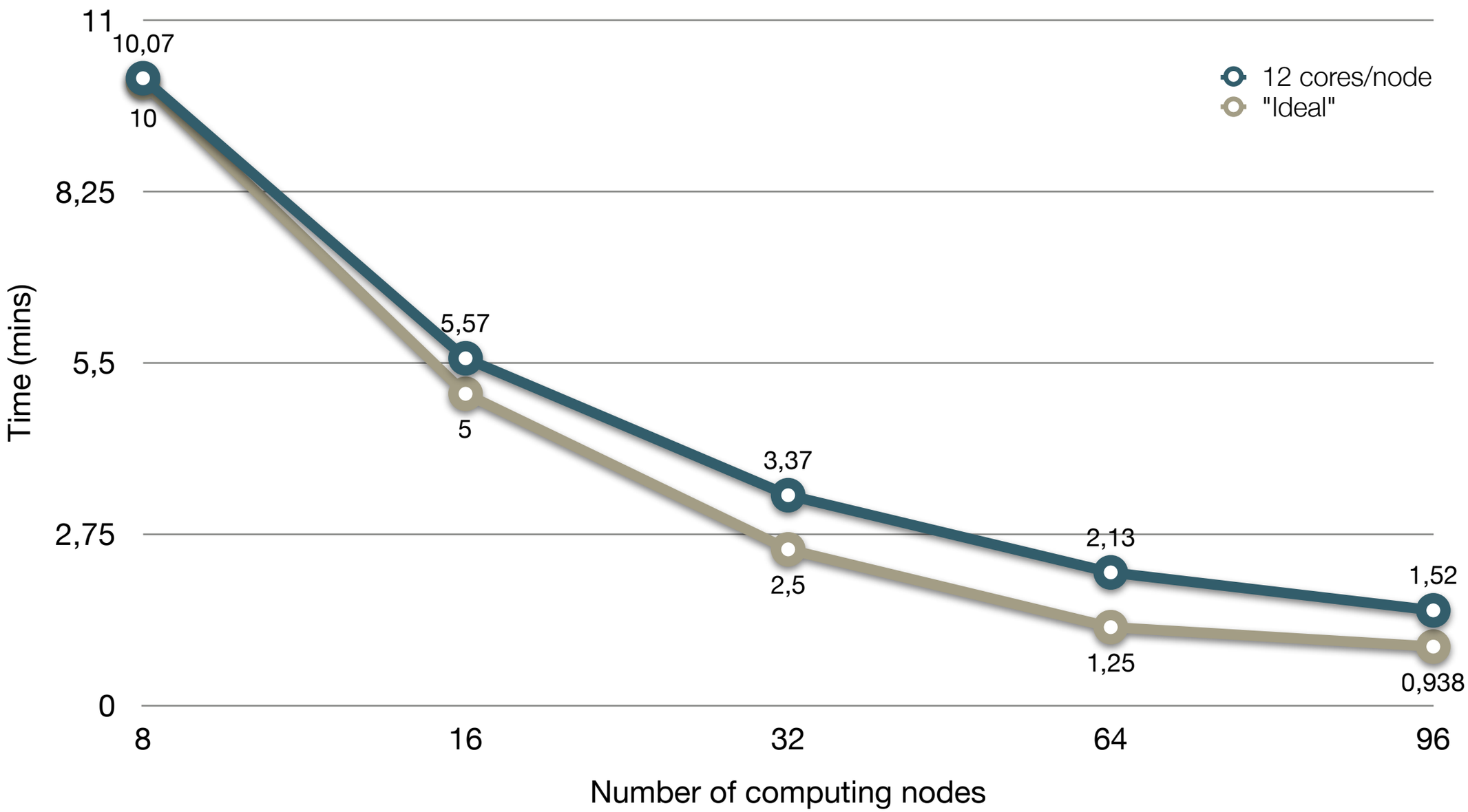
$P_E=2, P_F=2$



$P_E=2, P_F=10$

# 16,777,216 elements (G=1,024 x L=16,384)

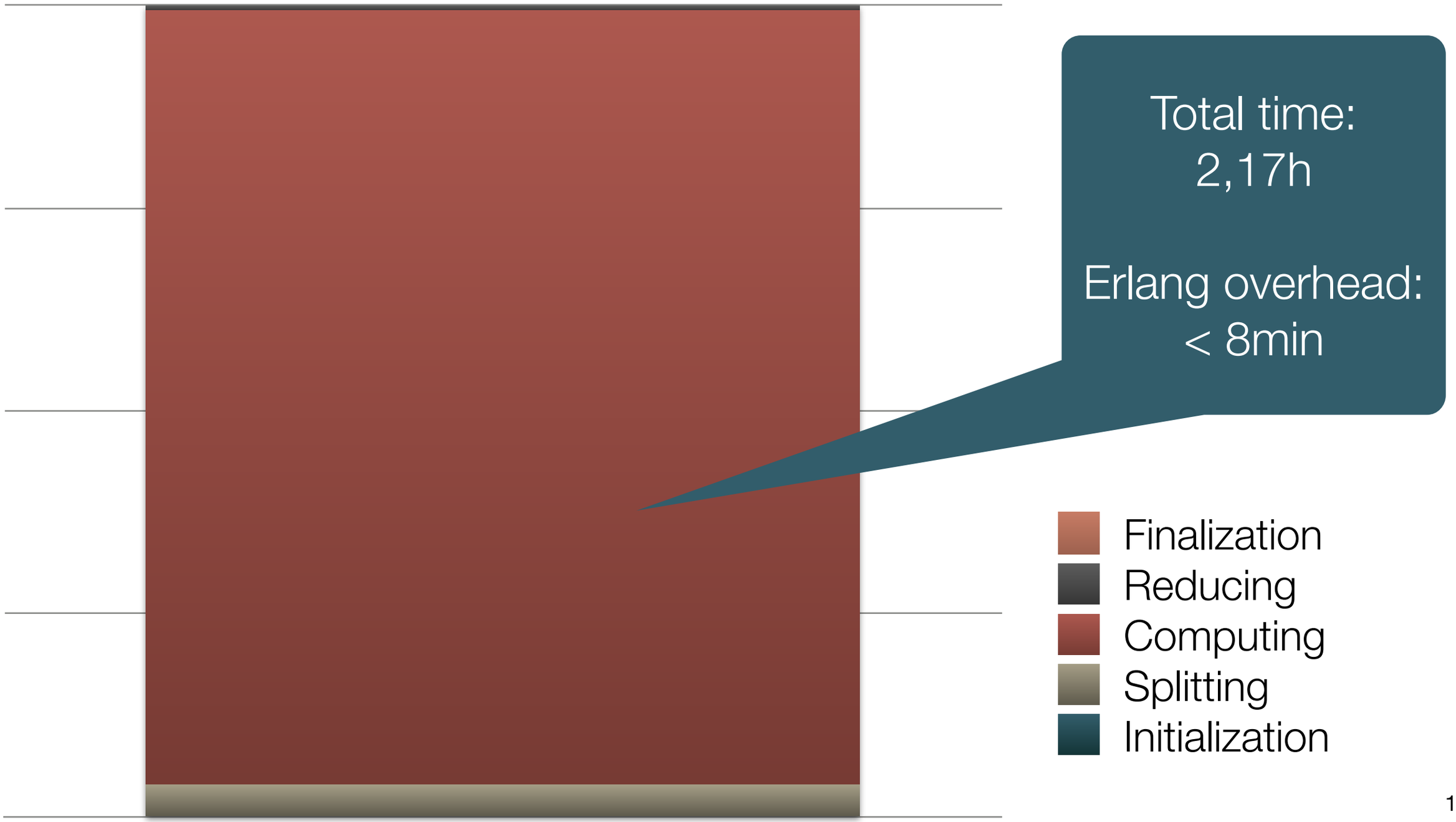
Laboratório Nacional de Computação Científica



# 1,073,741,824 elements (G=16,384 x L=65,536)

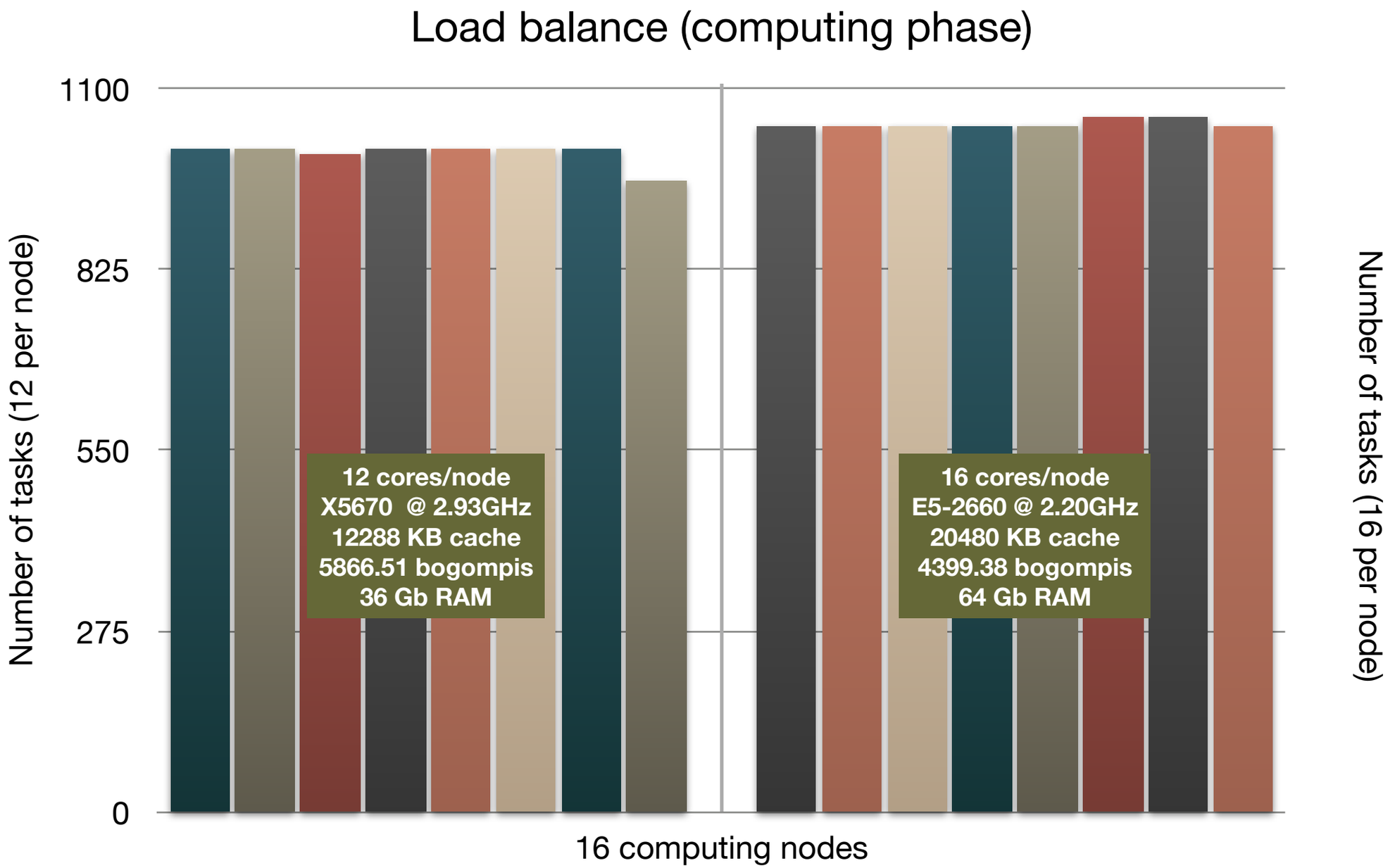
Laboratório Nacional de Computação Científica

Cost share of each phase (64 nodes, 12 cores/node)

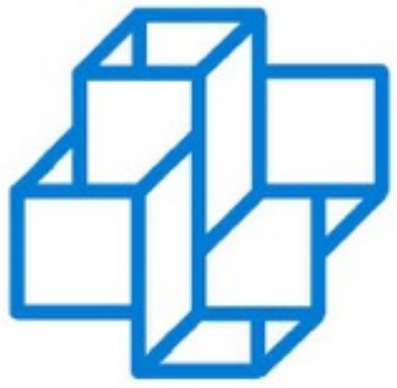


# 1,073,741,824 elements

Laboratório Nacional de Computação Científica



# Further developments



Laboratório
Nacional de
Computação
Científica

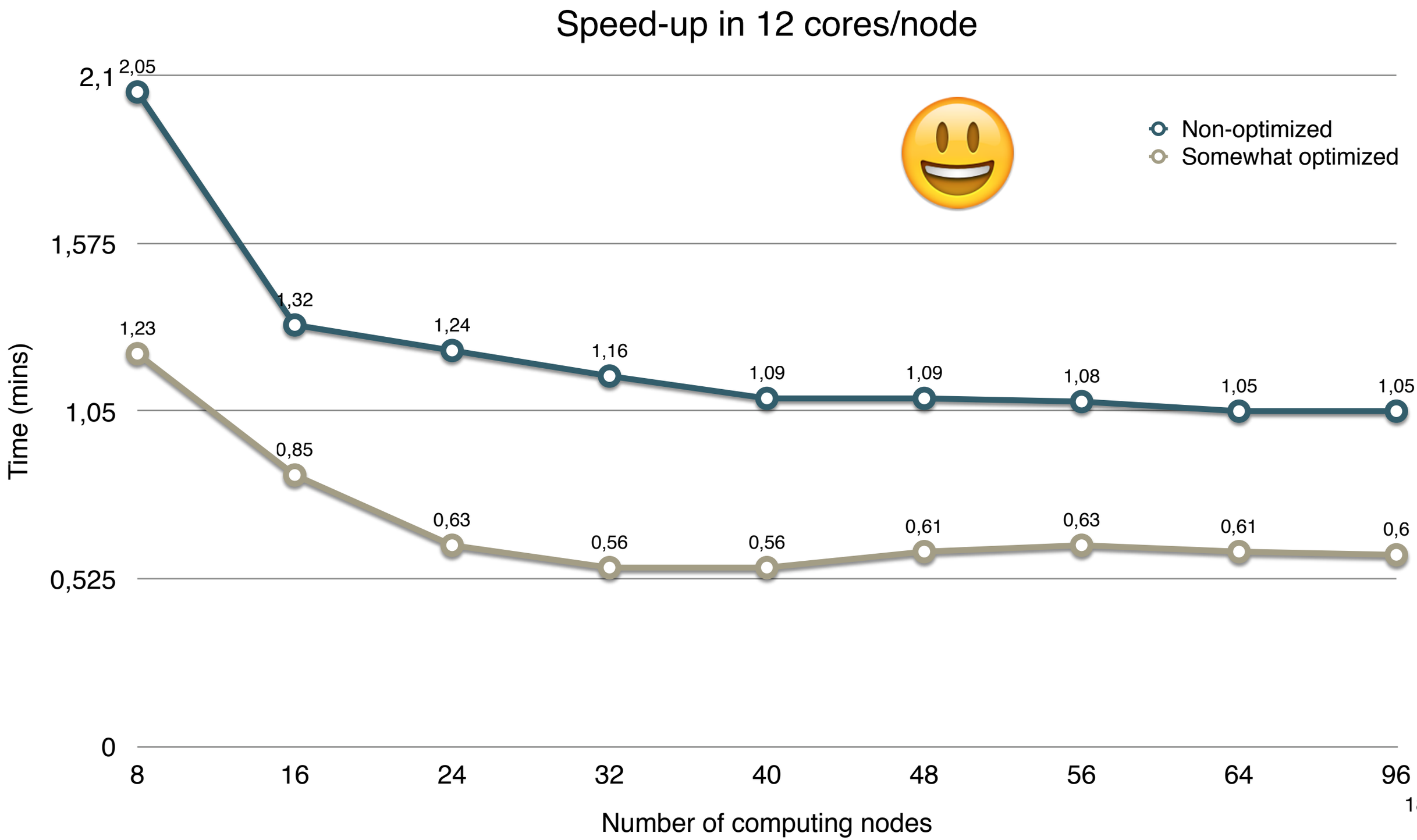
# Let’s "optimize a bit"…

Laboratório Nacional de Computação Científica

- Erlang: iterative splits
- Erlang*C++: persistent C++ jobs
- C++: improvements in numerical integration

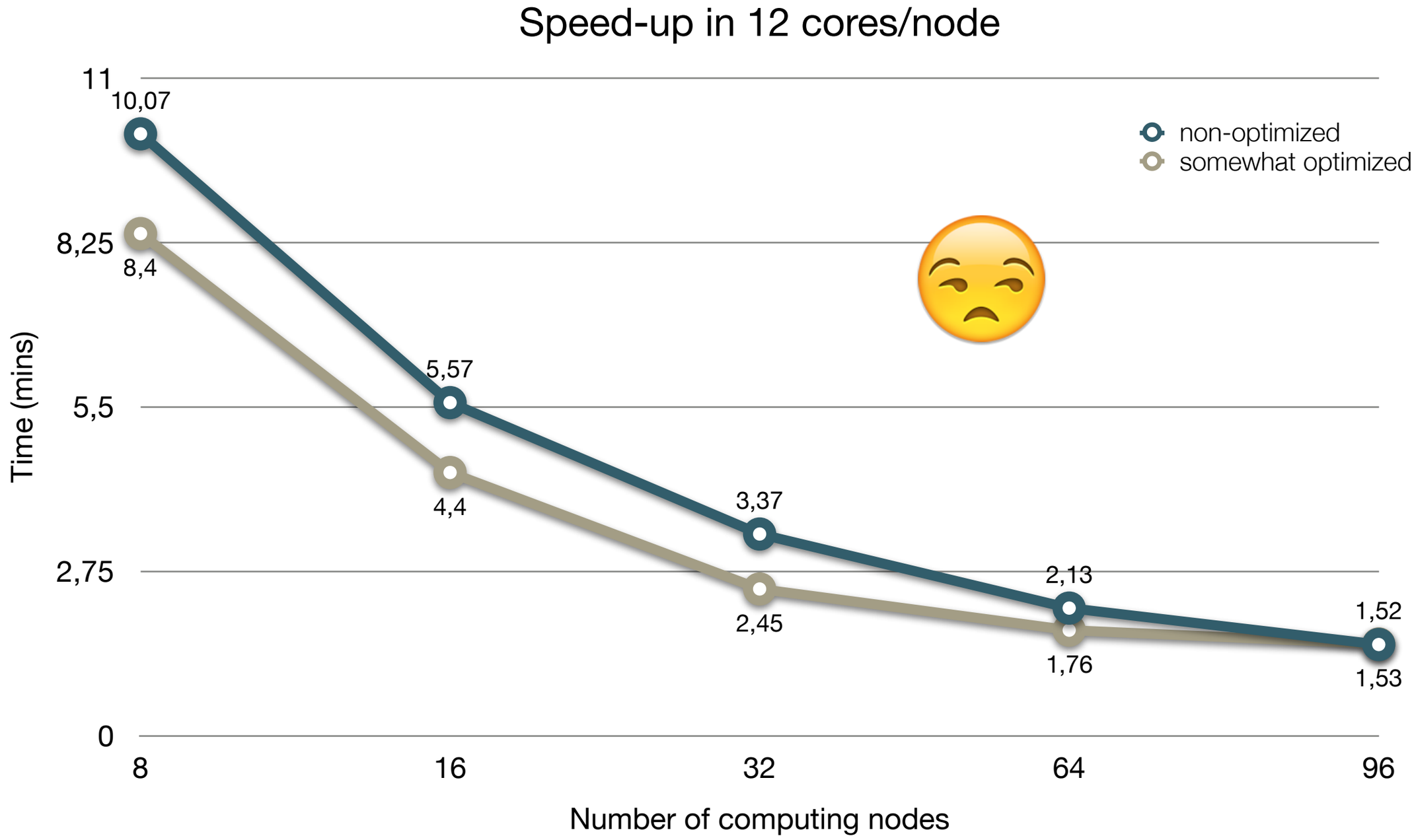
# 4,194,304 elements with opt (G=1,024 x L=4,096)

Laboratório Nacional de Computação Científica



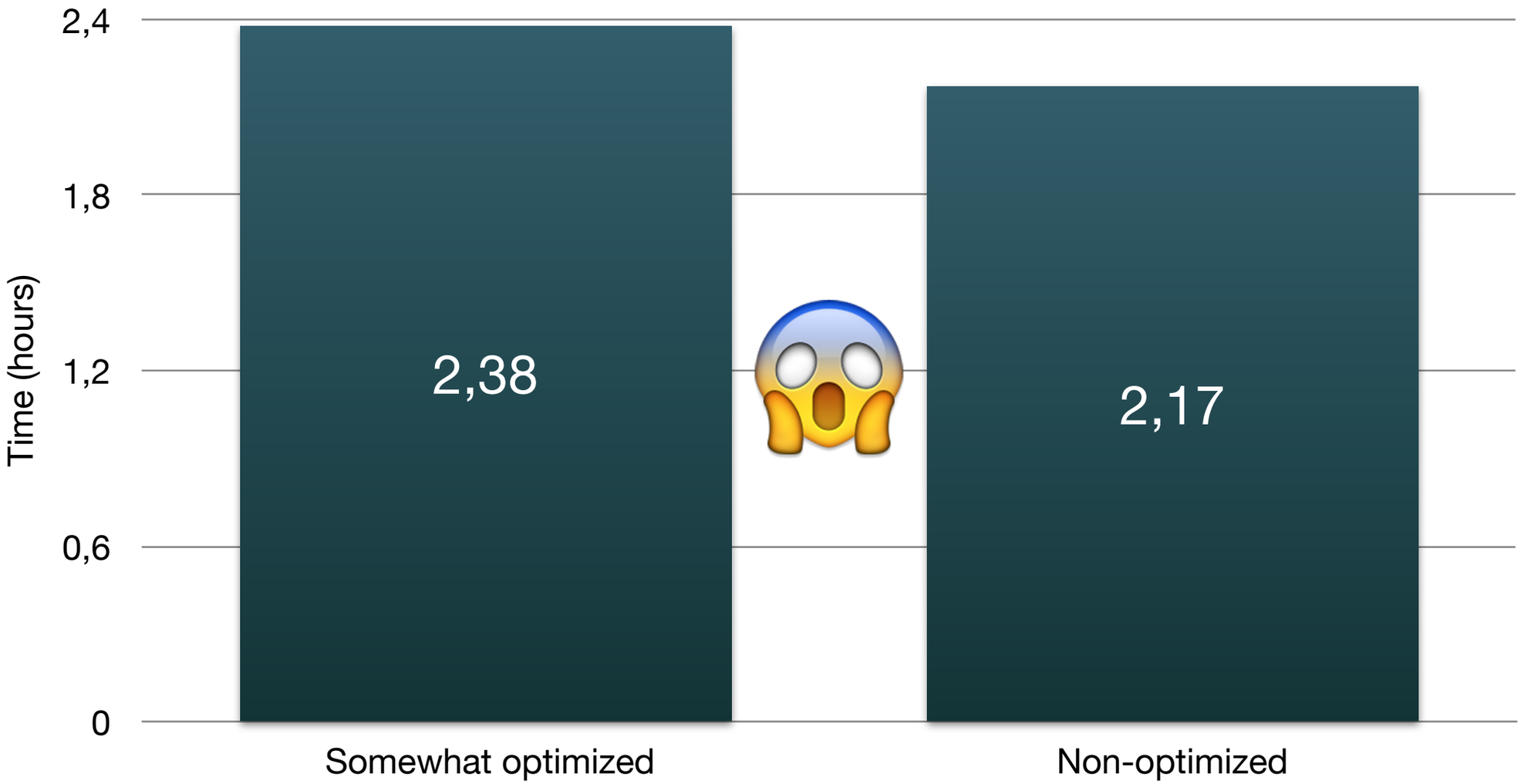
# 16,777,216 elements with opt (G=1,024 x L=16,384)

Laboratório Nacional de Computação Científica



# 1,073,741,824 elements with opt (G=16,384 x L=65,536)

Laboratório Nacional de Computação Científica




- Who is guilty?

# Transient problems

Laboratório Nacional de Computação Científica

In MHM 2D, 256 (G=16, L=16) elements

f(x,y,t) = (1+t*delta_t).cos(2*pi*x)*cos(2*pi*y), $P_e$=2, $P_f$=0

100 timesteps, Δt=2.5e-4, implicit method

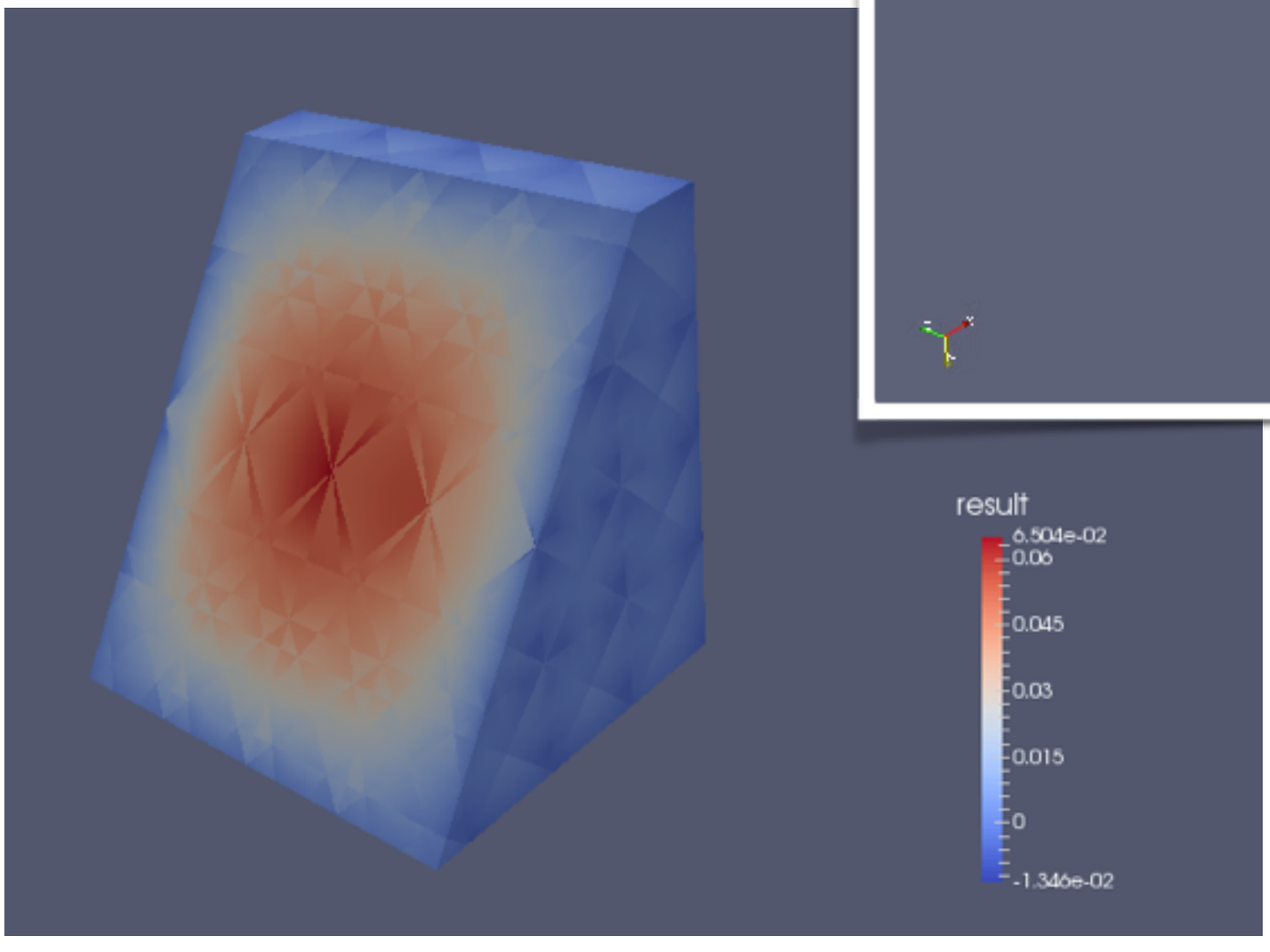
# 3D problems

Laboratório Nacional de Computação Científica

In MHM, 87,552 (G=1,536, L=57) tetrahedra

273,408 LDoFs total

18,994 GDoFs





$f(x,y,z) = 1.0$

$P_e=2$, $P_f=0$

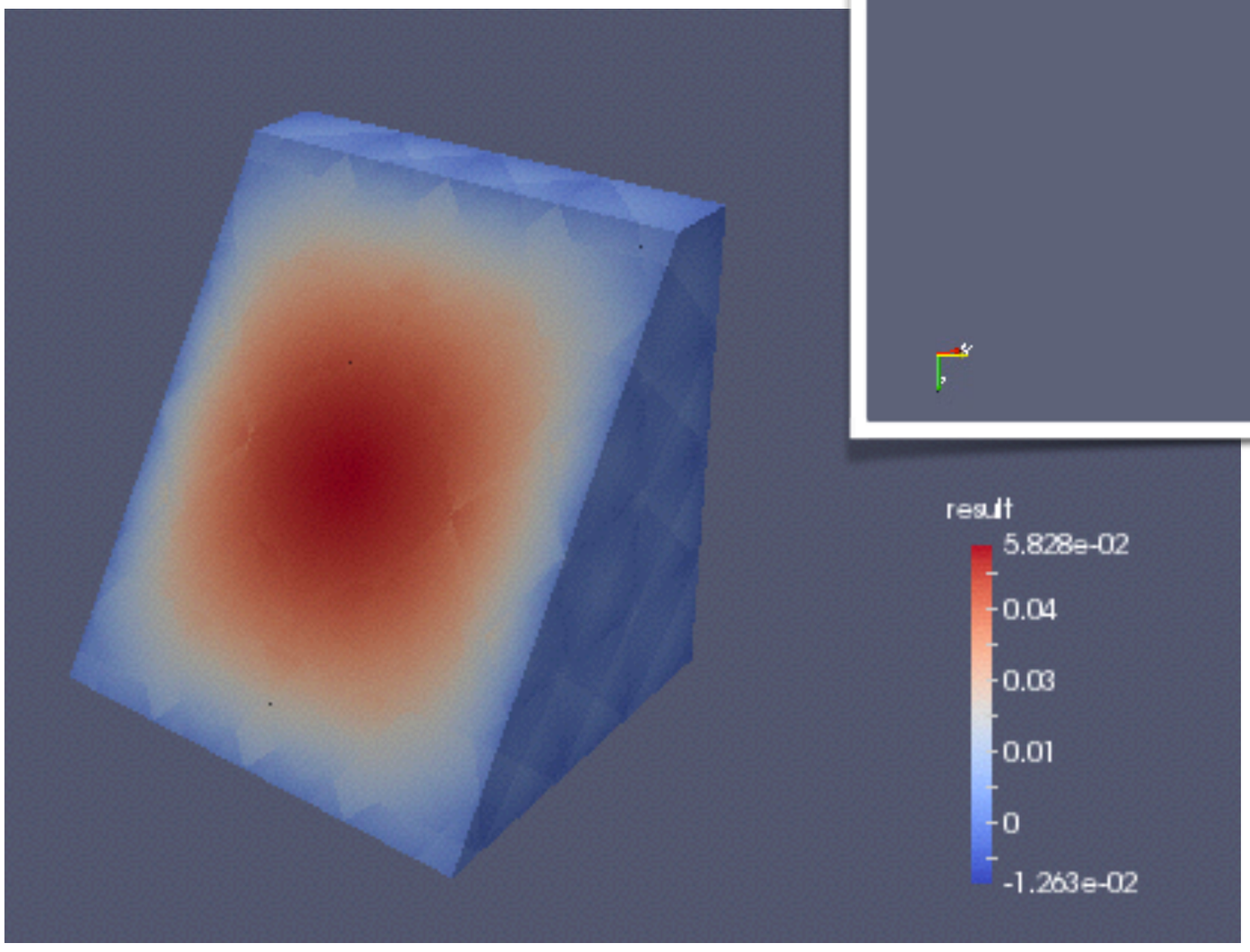
# 3D problems

Laboratório Nacional de Computação Científica

In MHM, 1,069,056 (G=1,536, L=696) tetrahedra

2,116,608 LDoFs total

18,994 GDoFs

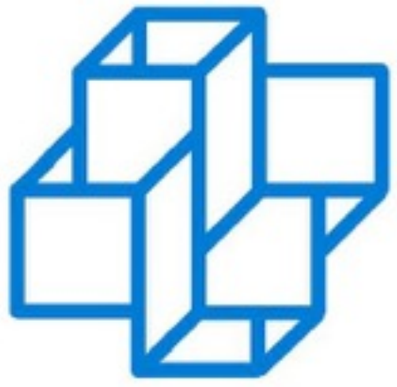




$f(x,y,z) = 1.0$

$P_e=2$, $P_f=0$

# What’s next?

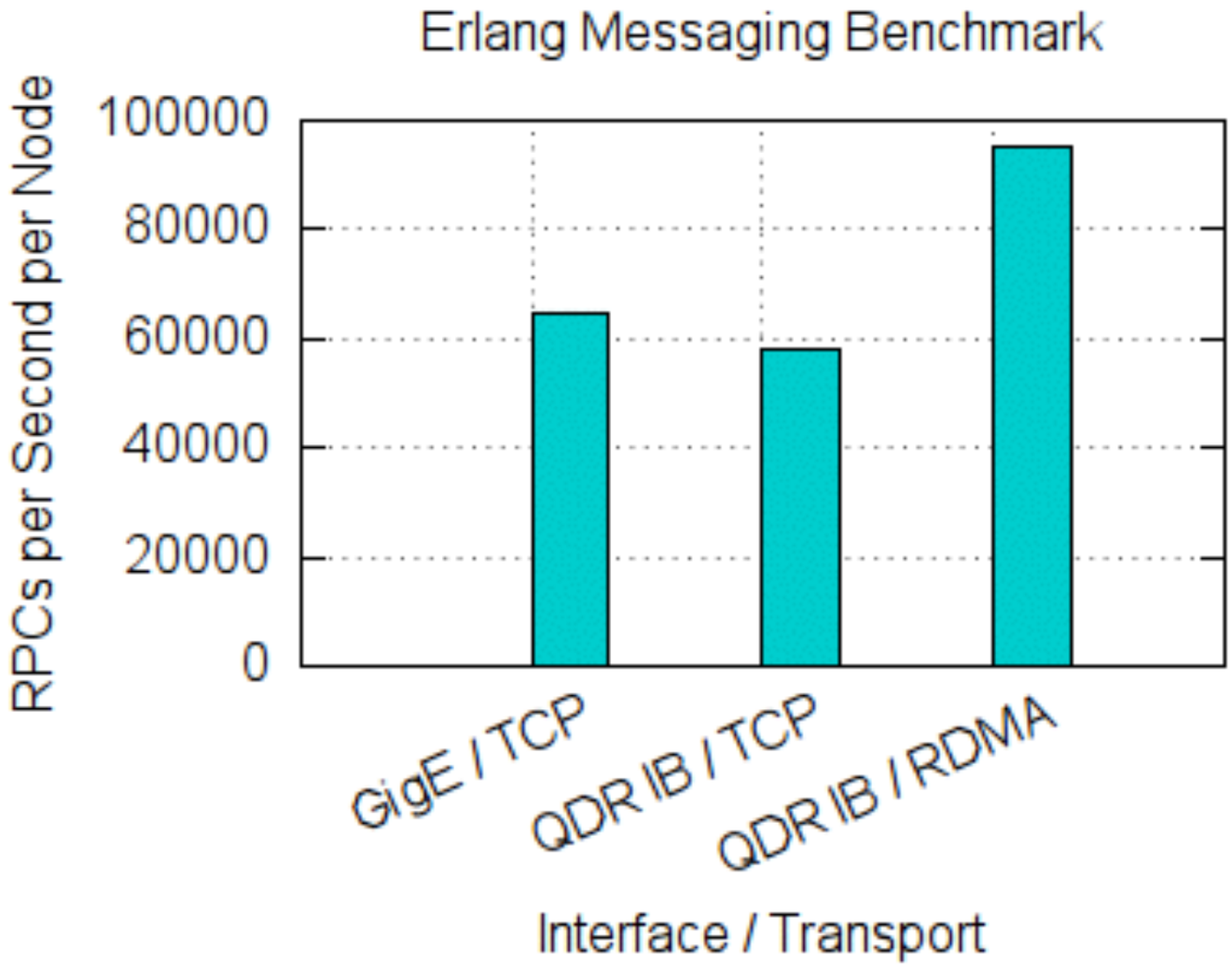


Laboratório
Nacional de
Computação
Científica

# To-do… (Erlang part)

Laboratório Nacional de Computação Científica

- **Smarter scheduling algorithm!!!!!** 😴
- Employ native RDMA protocol over Infiniband (https://github.com/MrStaticVoid/rdma_dist) 😴
- **Arbitrary code to solve local problems** 🏃
  - Some rapid prototyping with FEniCS




# To-do… (C++ part)

Laboratório Nacional de Computação Científica

- Basic optimizations at the processing phase
  - E.g.2) **Use GPUs for computing the many $\eta_i$ in parallel** (stiffness matrix needs to be inverted only once!)

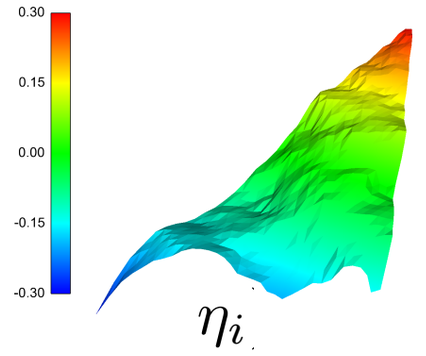
$$
\begin{aligned}
&\text{Find } \eta_i \text{ such that} \\
&\left.\begin{aligned} \mathcal{L}\eta_i &= 0, \text{ in } K \\ \boldsymbol{\sigma}(\eta_i)\cdot\boldsymbol{n}^{\partial K} &= \psi_i, \text{ on } \partial K \end{aligned}\right\} \\
&\text{Find } u^f \text{ such that} \\
&\left.\begin{aligned} \mathcal{L}u^f &= f, \text{ in } K \\ \boldsymbol{\sigma}(u^f)\cdot\boldsymbol{n}^{\partial K} &= 0, \text{ on } \partial K \end{aligned}\right\}
\end{aligned}
$$

- Hotspots at the **splitting (I/O intensive)** and **reducing (computing/memory intensive)** phases
  - Help! 😉

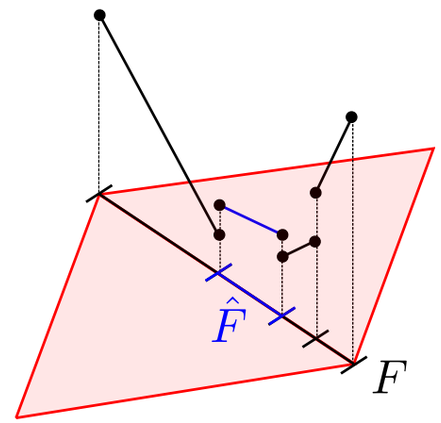
# To-do… (C++ part)

Laboratório Nacional de Computação Científica

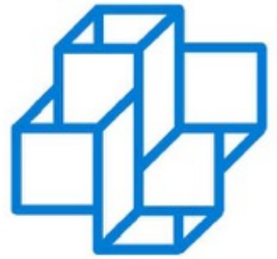
- Give support to:
  - Higher-order element-face basis functions (for 3D) 🏃




  - Partitioned element-face basis functions (for 2D and 3D) 😴



# To-do… (Erlang*C++ part)



Laboratório Nacional de Computação Científica

- Give support to:
  - Vectorial problems 😜
    - Initial target: elasticity/elastodynamics
    - Other execution environments (e.g. clouds) 🚶
    - *A posteriori* adaptation 😴
- Make overall configuration more flexible 😴
- Provide better visualization output 😴

# Summary

Laboratório Nacional de Computação Científica

- [x] Running code for 2D, 3D, stationary and transient problems
- [ ] Functional and non-functional features yet to be added
  - Software architecture carefully [RE]built to facilitate such additions
- [ ] Compare with other simulators in specific scenarios
  - E.g.: HDG…
- [ ] Explore Erlang framework in other contexts

# Thank you!!!

Antônio Tadeu A. Gomes

atagomes@lncc.br

Laboratório Nacional de Computação Científica