COPPE - Federal University of Rio de Janeiro

Third Brazil-France Workshop

On High Performance Computing and Scientific Data Management Driven by Highly Demanding Applications

02-05 September, Bordeaux, France 2013

Emerging Challenges for EdgeCFD Simulations in Massively Many-core Architectures

Renato N. Elias[1]

renato@nacad.ufrj.br

[1] High Performance Computing Center (NACAD)

www.nacad.ufrj.br

Civil Engineering Department (PEC/COPPE)

www.pec.coppe.ufrj.br

Federal University of Rio de Janeiro (UFRJ)

www.ufrj.br

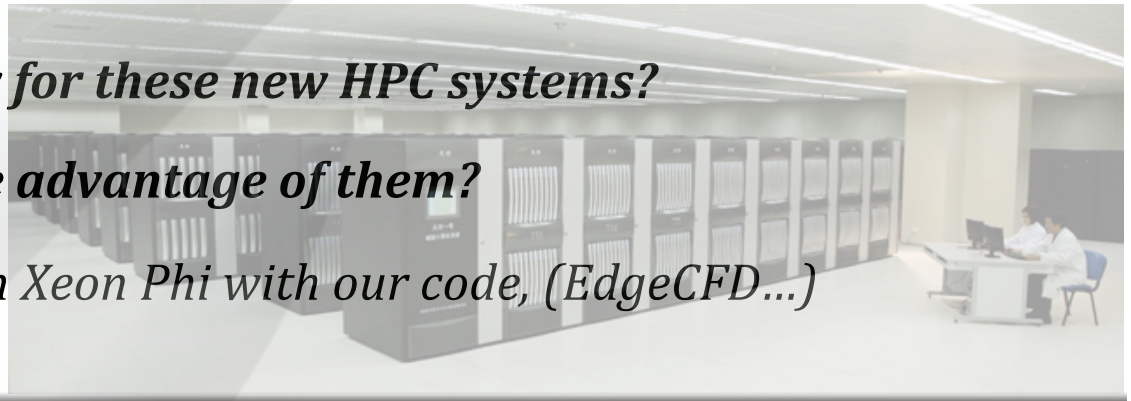
NACAD

Outline

- ***Introduction***
- ***What's EdgeCFD?***
- ***EdgeCFD Parallel Models***
- ***Emerging Platforms: issues vs alternatives***
- ***Playing with new architectures***
  - *Preliminary tests with Intel Xeon Phi*
- ***Closure and discussions***

Introduction

- ***HPC systems are changing…***
  - *Massive in slower cores = "Many-cores"*
  - *Several parallelism and vectorization levels → how to exploit?*
  - *CoProcessors came to stay… (Intel Xeon Phi and Nvidia Kepler)*
- ***1st on top500.org***
  - *Tianhe-2/China: 3.12M of cores (Intel Xeon Phi)*
  - *33.86 Pflops in HPL*
- ***Are old applications ready for these new HPC systems?***
- ***How could/should we take advantage of them?***
  - *Let's play a little bit with Xeon Phi with our code, (EdgeCFD…)*



What's EdgeCFD?

- ***Just an Academic Stabilized Finite Element Simulator for Fluid Flow Problems (in general).***
  - *Inherit/incorporate results from theoretical and applied researches*
  - *Helps other projects as a simulation tool*

***Edge*: Data structure used in the software**

+

***CFD*: Computational Fluid Dynamics**

what comes under the hood...

Main and General Features

- ***General:***
  - ***Hybrid parallel (MPI, OpenMP or both)***
  - ***Edge-based*** *data structure.*
  - *Entirely written in* ***Fortran90***
  - ***Unstructured*** *linear tetrahedra (only)*
  - *Input from "competitors" (Ansys, IcemCFD, CFX) and GMSH*
  - *Output to: ParaView, VisIt and/or Ensight*

- ***Physics (coupled or not):***
  - *Incompressible Flow;*
  - *Advection-diffusion transport of multiple species*
  - *Fluid-object interaction*
  - *Compressible Flow*

(Very) Brief History of EdgeCFD

- ***B.2007***
  - *Earlier works from **Catabriga**, **Martins**, **Elias** & **Coutinho** on edge based solvers for solid and fluid dynamics on HPC platforms;*
  - *Code started from scratch! (incompressible flow only)*
- ***2007***
  - *Advection-Diffusion and Eulerian (VOF/LS) free-surface flows*
- ***A.2011***
  - *Mesh update and Fluid-Object-Interaction*
  - *Advection-Diffusion of multiple species*
  - *Compressible Flows (earlier this year from **Mendonça**, DSc)*

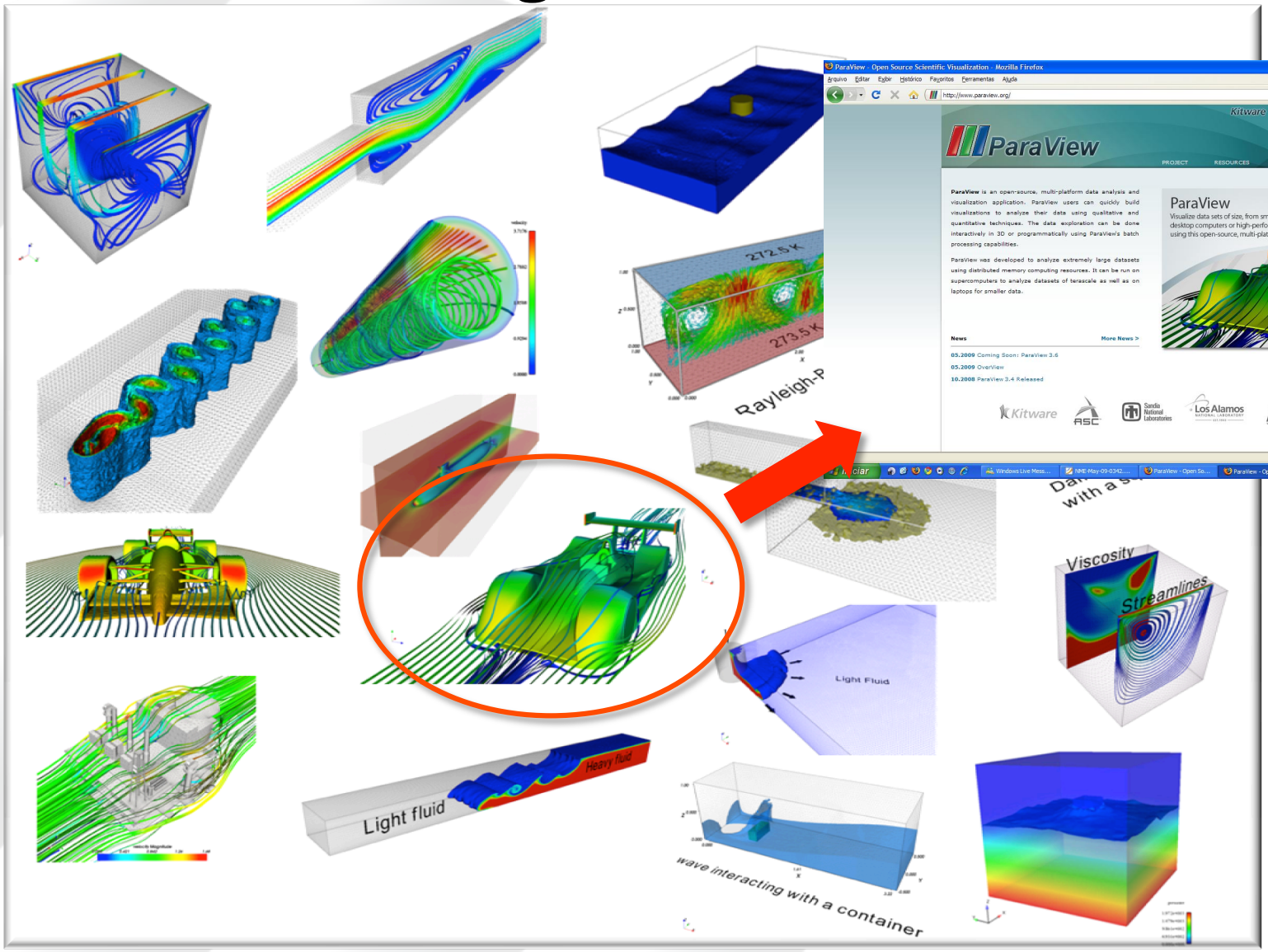
Development Teams

- ***Team 1 (NACAD)*** *~ 4 people*
  - *Fluid mechanics core*
  - *Supporting HPC tools/algorithms*
- ***Team 2 (MECANICA):*** *~ 4 people*
  - *Uncertainty Quantification*
  - *Verification and Validation*
  - *Turbulence by RBVMS formulation*
- ***Team 3 (LAMCE):*** *~ 5 people*
  - *ALE*
  - *Solid mechanics core*

Collaboration with Martas' Team:

- **Chiron®**, Provenance and Scientific Workflows

EdgeCFD in Action



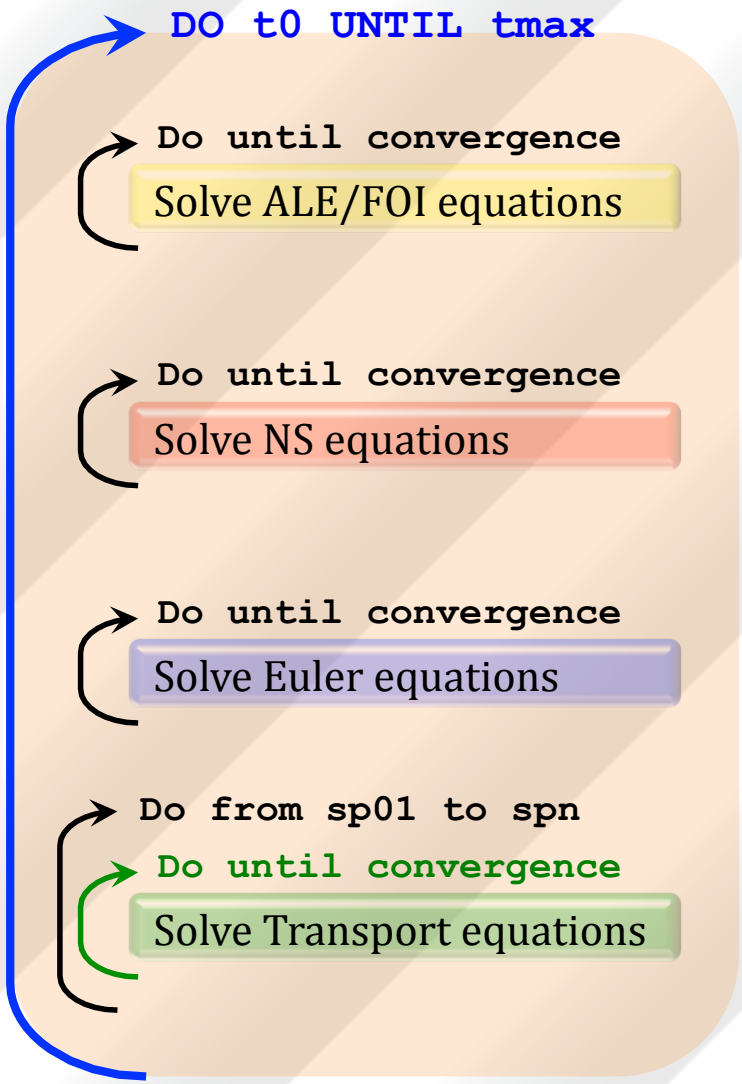
Some "Goodies and Baddies"

- ***Goodies:***
  - *Built as a hybrid parallel code* ***from scratch***
  - ***Highly portable:*** **any* Fortran compiler is enough… (really!)*
  - ***Efficient:*** *at least, Intel Vtune and TAU say that… (and we believe ☺)*
- ***Baddies:***
  - *No Open Source ☹*
  - *No "fancy"* ***UI's*** *(yet)*
  - *Only* ***linear tetrahedra*** *supported*
  - *Lack of linear solver "****accelerators****" (****see Benaias's talk****)*

Main Kernels

- ***Incompressible Flow:***
  - *Equal-order, linear tetrahedra, Edge-based, SUPG/PSPG*
  - *Turbulence by Smagorisnky or RBVMS*
  - *Inexact-Newton*
- ***Free Surface interface capturing***
  - *Marking Function (VOF or LS)*
  - *Edge-based, Y-Zbeta, SUPG*
  - *Predictor-Multicorrector scheme*
- ***Fluid-Object-Interaction (FOI):***
  - *with Rigid body translations/rotations*
  - *Mesh update with Edge-based, Laplace smoothing*
- ***Compressible Flow:***
  - *linear, tetrahedra, Edge-based, Y-ZBeta/SUPG*

Loops and Costs...



`residua() < stiffness() < matvec()`

PCG (3 dofs)

`residua() < stiffness() < matvec()`

GMRES (4 dofs)

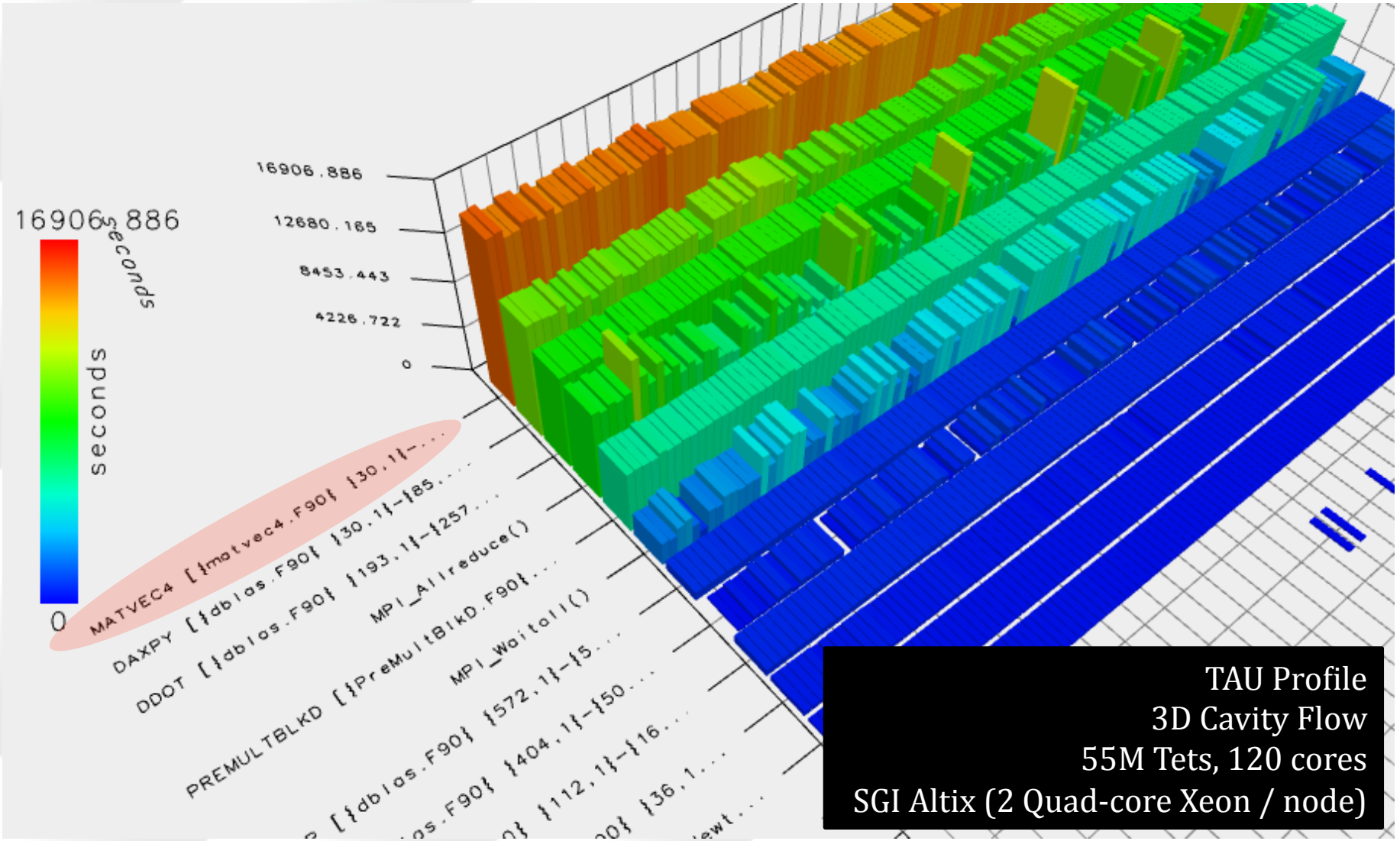
`residua() < stiffness() < matvec()`

GMRES (5 dofs)

`residua() < stiffness() < matvec()`

GMRES ("n" dofs)

Computational Costs



TAU Profile
3D Cavity Flow
55M Tets, 120 cores
SGI Altix (2 Quad-core Xeon / node)

Software Stack

1. ***Ansys Classic, ICEM-CFD, CFX and/or GMSH***
   - *Computational model*
   - *Mesh Generation*
   - *Boundary and Initial conditions*
2. ***Preprocessor (EdgeCFDPre)***
   i. *Takes a serial mesh;*
   ii. *Creates partitions with **Metis** (**could be Scotch...**)*
   iii. *Extracts edges and reorders data with **EdgePack***
   iv. *Stores prepared data to solver*
3. ***Solver (EdgeCFDSolver)***
4. ***ParaView, VisIt, Ensight***
   - *Visualization: Ensight, Xdmf/HDF5 or Parallel VTK*

***Blue**: "Home made"*
***Green**: Third party*

*Workflow management and provenance by **Chiron***

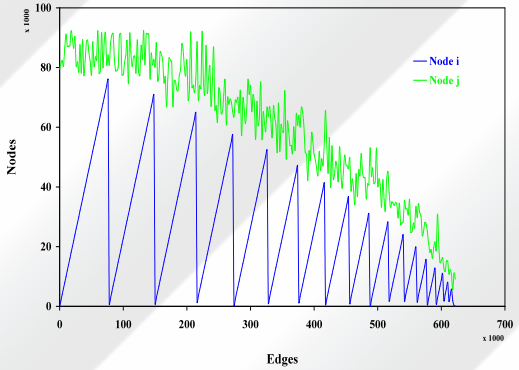
Performance Optimization in EdgeCFD

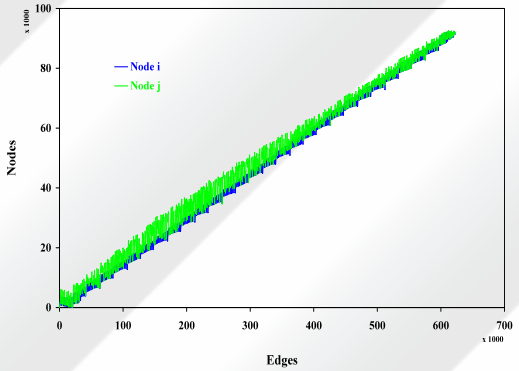
- *Exploring memory access → **EdgePack***
- *Exploring **Parallelism** and **Vectorization***
- *Programming practices that ***used to be*** good…*
  - *"Intel is changing this rule…" ☹*
- *Fast solvers:*
  - *Inexact Newton / relaxed linear solutions;*
  - *Time Step Control by PID controller*
  - *Dynamic Deactivation*
  - *Mesh Multiplication + Multigrid (**Benaias duty…** ☺)*

NOTE: No concerns to specific optimizations for architecture X, Y or Z

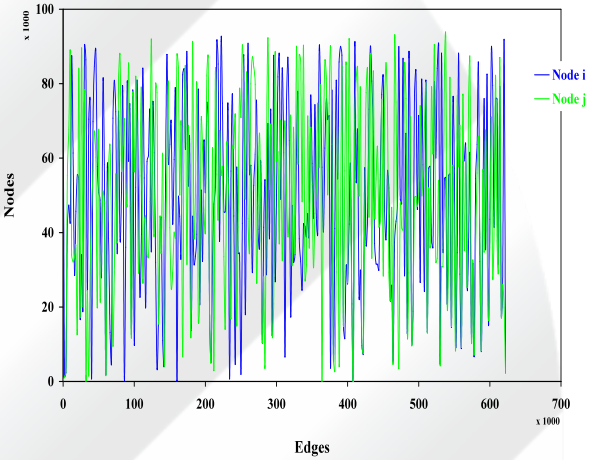
EdgePack and Memory Access Patterns

Reordering for minimizing i/a





Improving data locality with NO memory dependencies

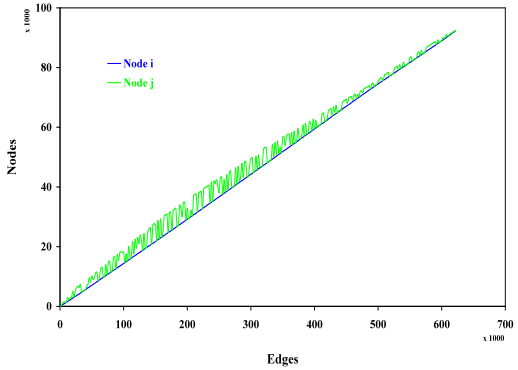


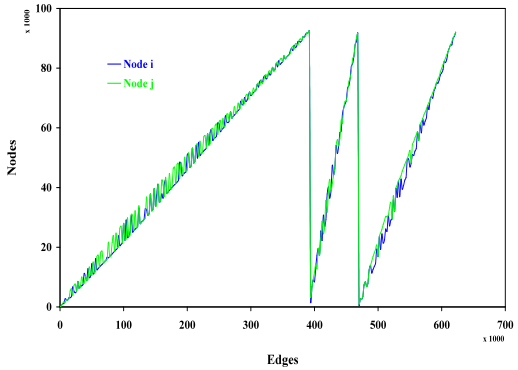
Original edge order

Data reordering schemes provided by

EdgePack®

Improving data locality: RCM vertex reordering and edge reordering in ascending vertex order



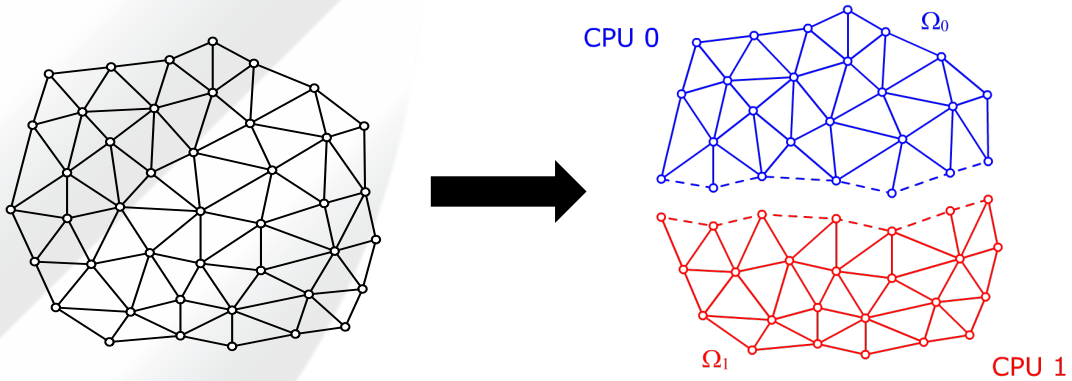


Improving data locality for each superedge type. Superedge improves the use of CPU registers

More details in Coutinho *et al*, IJNME, 2007

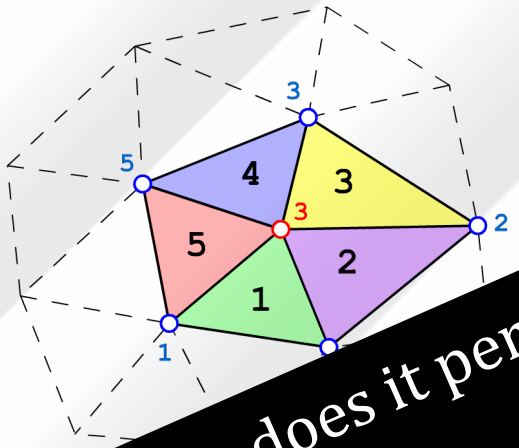
Distributed Memory Parallelism

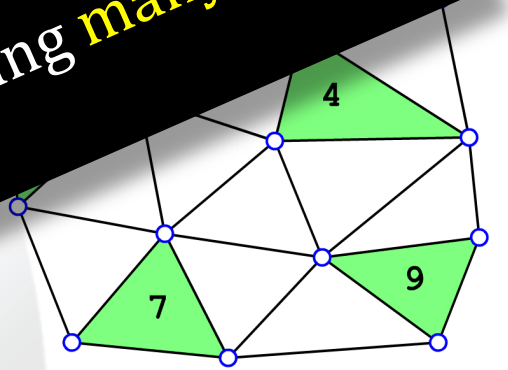
- ***Standard Approach:***
  - *Takes a mesh and give it to a partitioner (**Metis**, **Scotch**, **Jostle**…)*
  - *Once partitioned, **keeps the data parallel** (forever…)*
  - *Global IDs → **Local IDs***
  - *Each process records its own files*
  - *Shared information is synchronized by **p2p non-blocking communications***
  - *No ghost/halo information employed*



Shared Memory Parallelism

- *Standard blocked loops to remove memory dep[illegible]ency*





How does it perform on emerging many-core architectures?

```
!$
do
    ! rieve element nodes
    x(no) = x(no) + a
enddo
!$OMP END PARALLEL DO
```

```
ielm = 0
do icor = 1, ncores
   nvec = ielblk(icor)
!$OMP  PARALLEL DO
   do i = ielm+1, ielm+nvec
      ! Retrieve element nodes
      x(no) = x(no) + a
   enddo
!$OMP END PARALLEL DO
   ielm = ielm+nvec
enddo
```

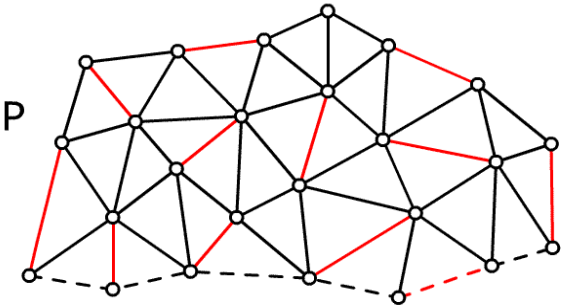
Hybrid Matrix-Vector Product

```
iside = 0
DO iblk = 1, nedblk
nvec  = ia_edblk(iblk)
!dir$ ivdep
!$OMP PARALLEL DO
   DO ka = iside+1, iside+nvec, 1
      ...MATVEC computations...
   ENDDO
!$OMP END PARALLEL DO
ENDDO
```

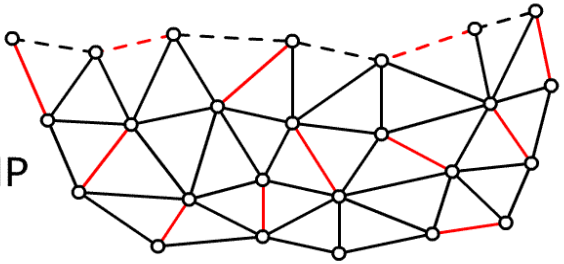
```
...over interface nodes...
#ifdef MPICODE
call MPI_AllReduce
#endif
```

Edge-by-Edge

OpenMP



OpenMP



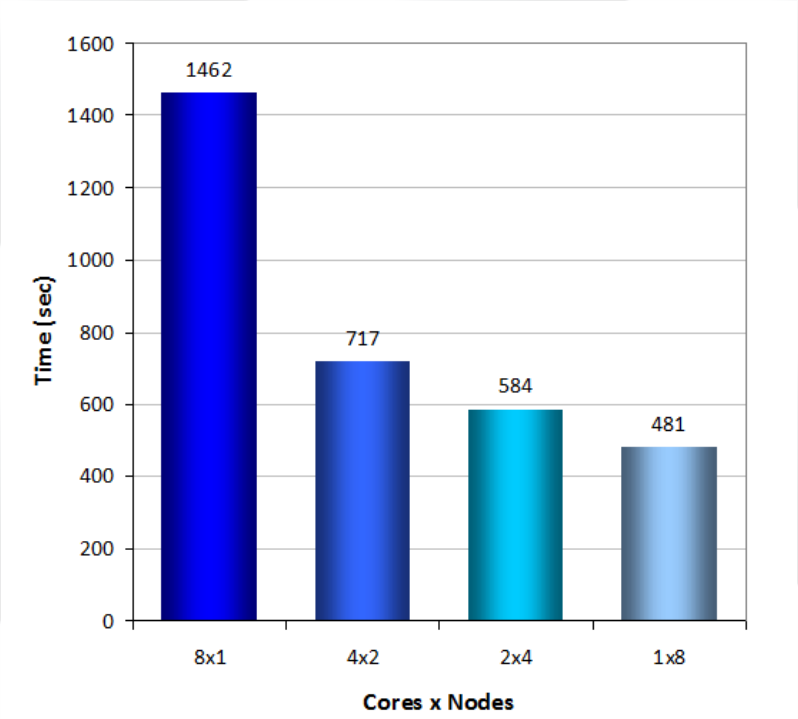
Moore's Law Breakthrough

Intel's courtesy

In order to keep Moore's law, applications will have to exploit Parallelism in many-core systems

Multicore CPUs and Process Placement

- *In 2010 we presented these results at VECPAR @ Berkeley*
  - *Same system (2 quad-core Xeon) varying the number of MPI processes x Node*



1 Computing Node = 2 Quad-Core Xeon
8 x 1 = 8 Cores x 1 Node
1 x 8= 1 Core x 8 Nodes

Weird conclusion (or behavior):
"Do not fill up the computing nodes's.
They don't like pressure!!! ☹

Elias *et al*, LNCS - 2010

... and we (Sandia and TACC) presented :

THE MULTICORE DILEMMA:

SANDIA and TACC's statements:

- *"...more chip cores can mean slower supercomputing..."*;
- *"...16 multicores perform barely as well as two for co... applications..."*;
- *"...Process placement in multi-core ... fluence on performance...."*;
- *"...more cores on a ... ty mean faster clock speeds..."*.

But It's changing...

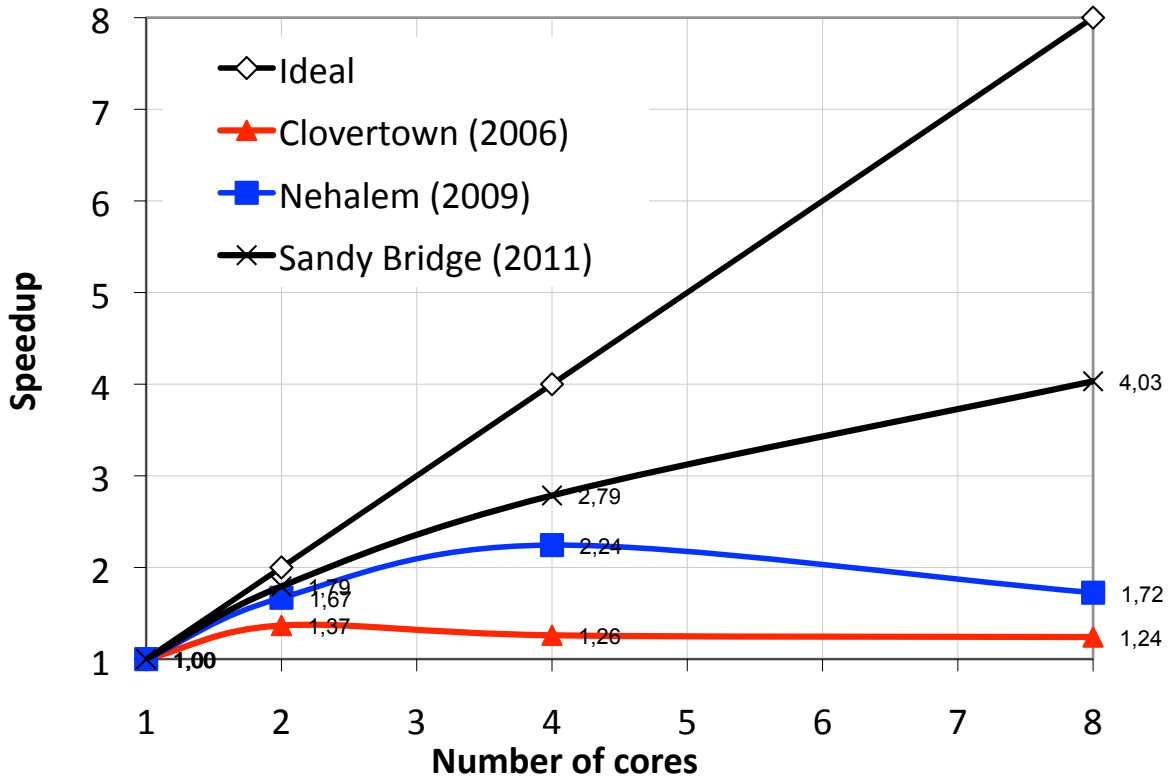
(by Sandia)

... the same checkout counter are processing your food instead of one, ...kout process should go faster. The problem is, if each clerk doesn't have ...cess to the groceries, he doesn't necessarily help the process. Worse, the clerks may get in each other's way.

SOURCE:
Diamond, J. et al, ''Multicore Optimization for Ranger'', TACC, 2009
https://share.sandia.gov/news/resources/news_releases/more-chip-cores-can-mean-slower-supercomputing-sandia-simulation-shows/

Intel and *OpenMP* Performance Evolution

- ***Same code (EdgeCFD), Same run/problem, Intel Fortran compiler, different Intel QUAD-core Xeon Generations***



It's getting better...
- *Software (Compiler)?*
- *Hardware (CPU/Core)?*
- *Or both?*

New Trends in Parallel Processing

- ***Massive number of (hybrid) cores***
  - ***Multicore*** *computing nodes with* ***MIC's*** *boards (****coProcessors****)*
  - *MICs ("Mike") stands for* ***Many-Integrated-Core;***
- ***OpenMP is gaining muscles*** ☺
  - *(and has some folks: CILK, OpenCL...)*
- ***Several parallel and vectorization levels and possibilities***
  - *Parallel/vector cores* ***inside*** *parallel boards* ***inside*** *parallel machines*
  - ***What's the best combination****???*
- ***New problems, new questions, (and we hope,) new answers...***

CoProcessors and Massive Parallelism

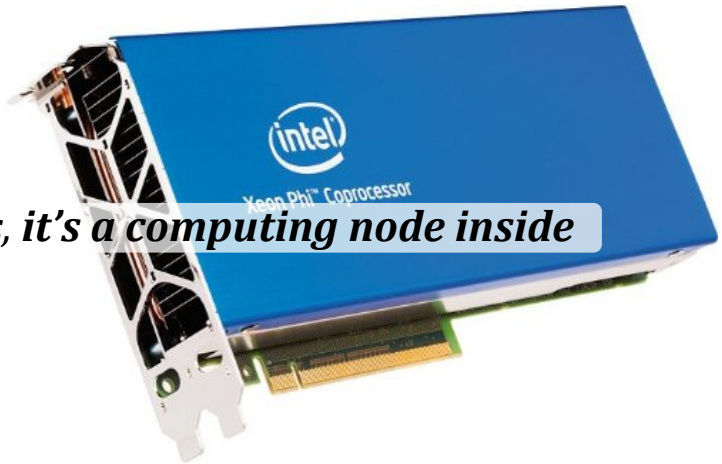
- ***CoProcessor Def. (from Wikipedia):***
  - *processors used to supplement functions of the primary processor (CPU)*
- ***Simple idea:***
  - *"reduced clock, simplify instructions and a put a huge number of cores in the same space occupied by a general purpose CPU"*
- ***CoProcessors's "War"***
  - *Intel Xeon Phi x Nvidia Kepler*

EdgeCDF on Many-Cores

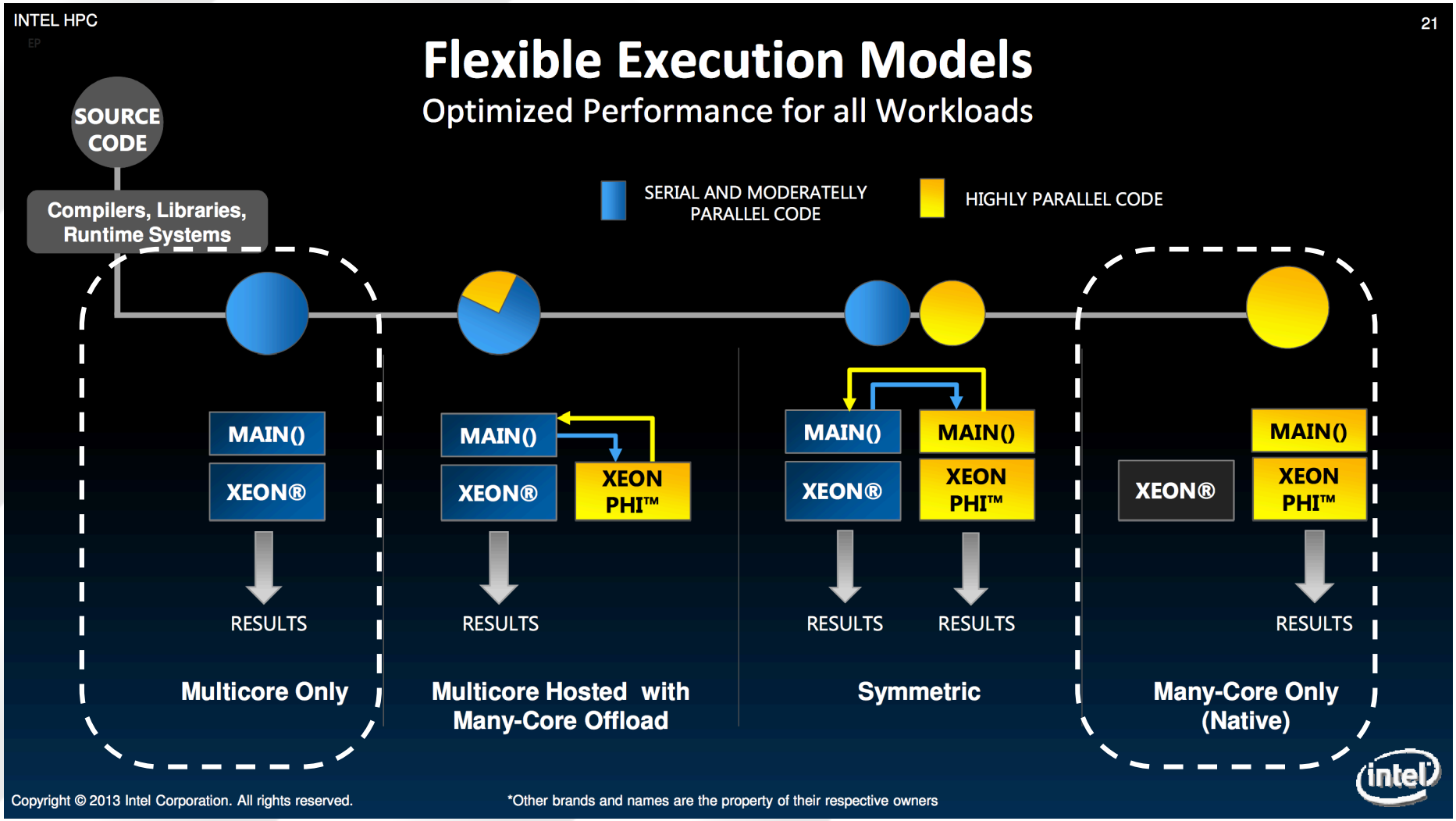
- ***Intel Xeon Phi***
  - ***A straight step for EdgeCFD toward CoProcessors.***
    - ***As simple as*** *call the compiler with a new compilation flag (`-mmic`);*
    - *Ok, but does it* ***naturally scale****? We'll see on next slides…*
- ***Why not GPGPU's (yet)?***
  - *Would require deep structural code changes (data structure, reordering schemes…)*
  - *Remember: No concerns with specific optimizations for architecture X, Y or Z. We seek for a* ***good and overall*** *performance to keep portability!*

Intel Xeon Phi

- ***Intel's Many-Integrated-Core (MIC) architecture***
- ***In few words:***
  - *"A PCI express board with 61 cores supporting up to 4 threads each (244 threads per board)*
- ***Dense and simplified processor***
  - *based on **Pentium P54c**, x86 architecture with 64-bits, vector capabilities and cache coherent*
- ***Each board runs a simplified Linux***
  - *Has ip address and is fully functional. In other words, **it's a computing node inside a computing node***
- ***First impression: "A Kepler's killer"***



Execution Models



EdgeCFD Tests on Xeon Phi

- ***"Blind" test***
  - *Upload, compile and run!*
  - *Nothing changed and/or tuned*
- ***Execution Models***
  - *Host only*
  - *Mic only*
- ***Parallel Models***
  - *OpenMP and MPI on Host*
  - *OpenMP on Mic*
- ***Only strong scalability!***

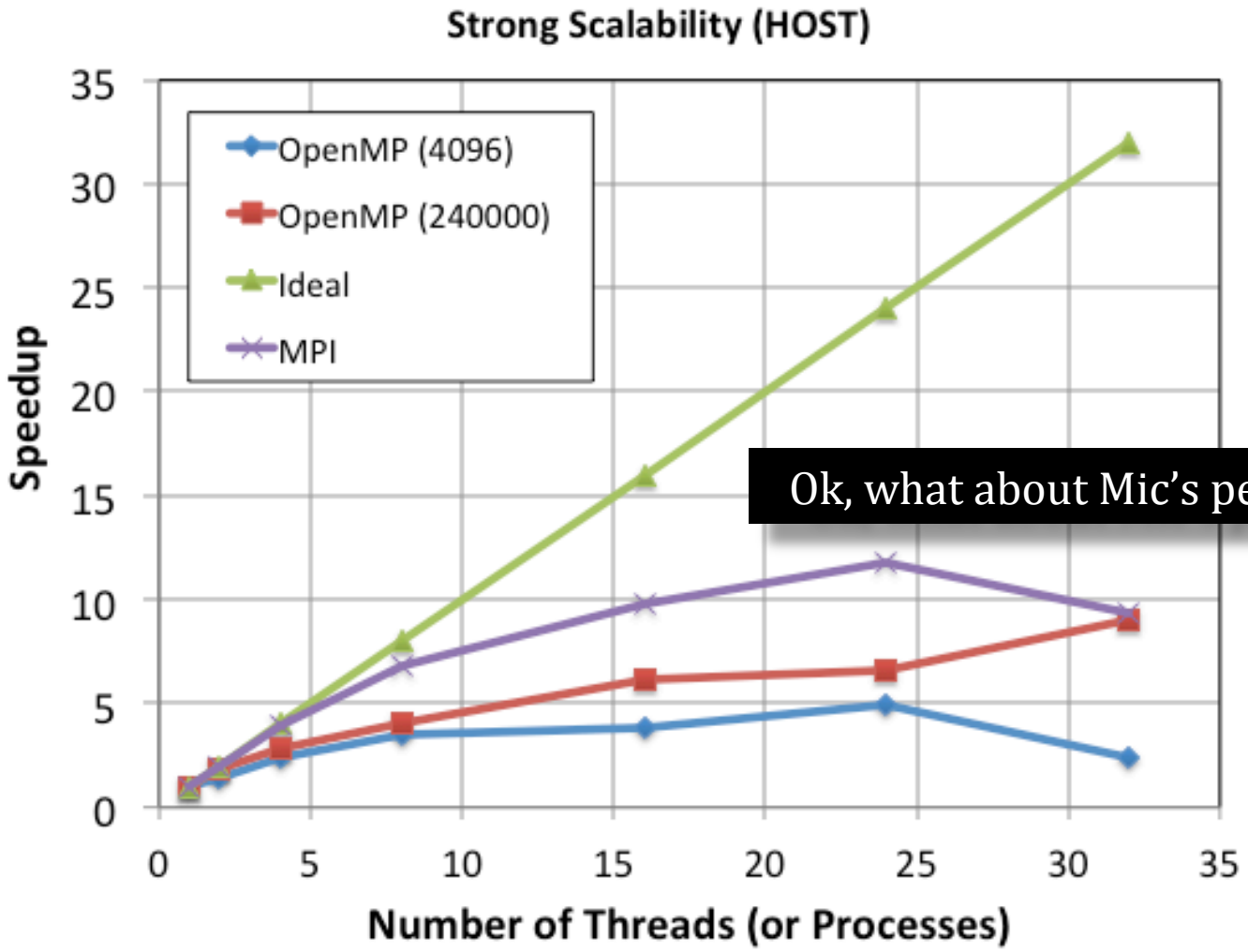
Test Problem and Mic's System

- ***Dambreak (Free Surface)***
  - *Incompressible flow coupled to an Advection Equation (VOF)*
  - **`50 Time steps`**
  - **`Edges: 306,597`**
  - **`Tets: 251,807`**
  - **`Nodes: 46,766`**

It's a small problem!
Could be easily ran on a laptop!

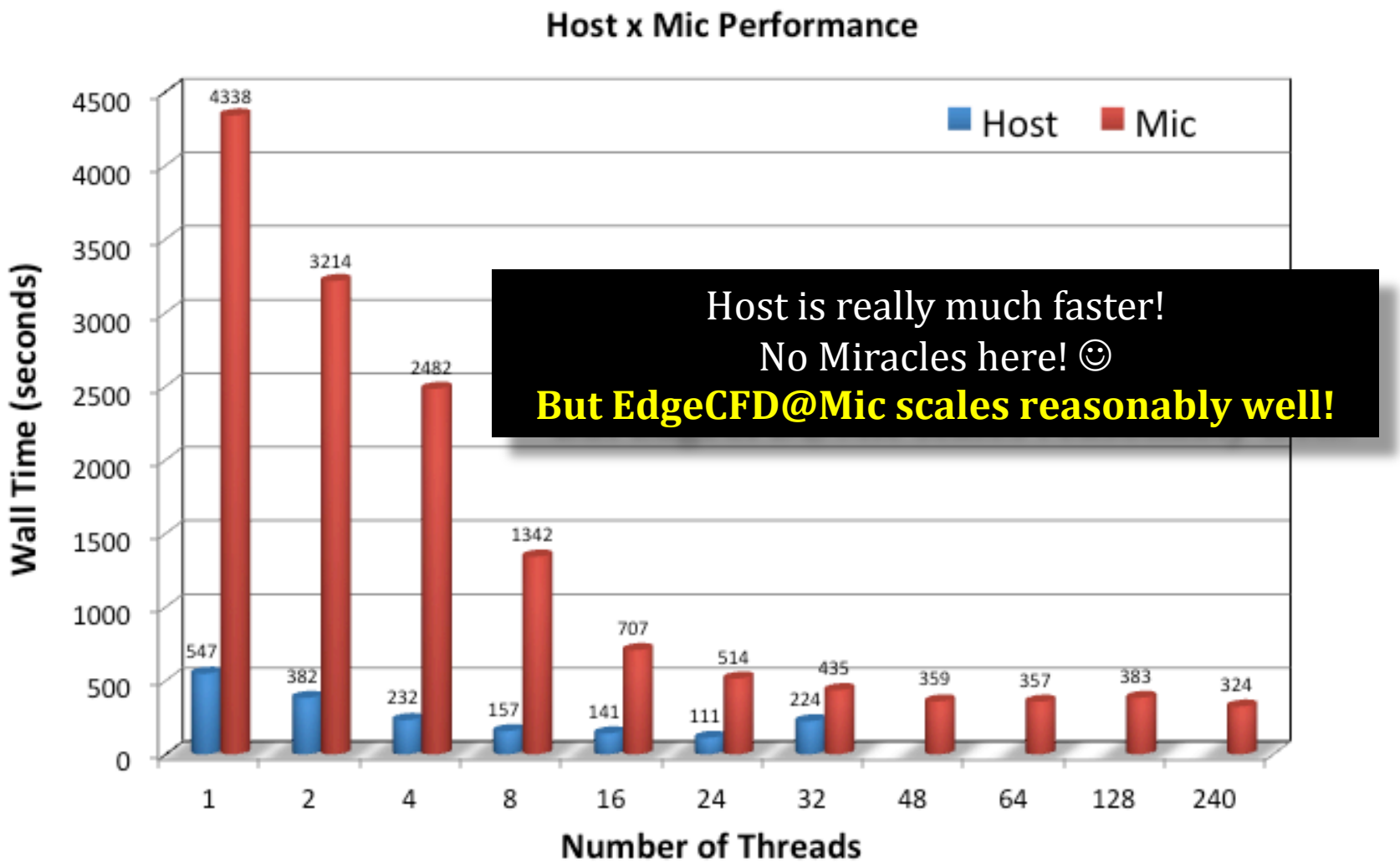
- ***Intel Xeon Phi 5110P Coprocessor***
  - *PCI-Express 3.0 x16*
  - *8GB of memory*
  - *1.053 GHz Clockspeed*
  - *60 Cores / 240 Threads*

Strong Scalability (HOST)

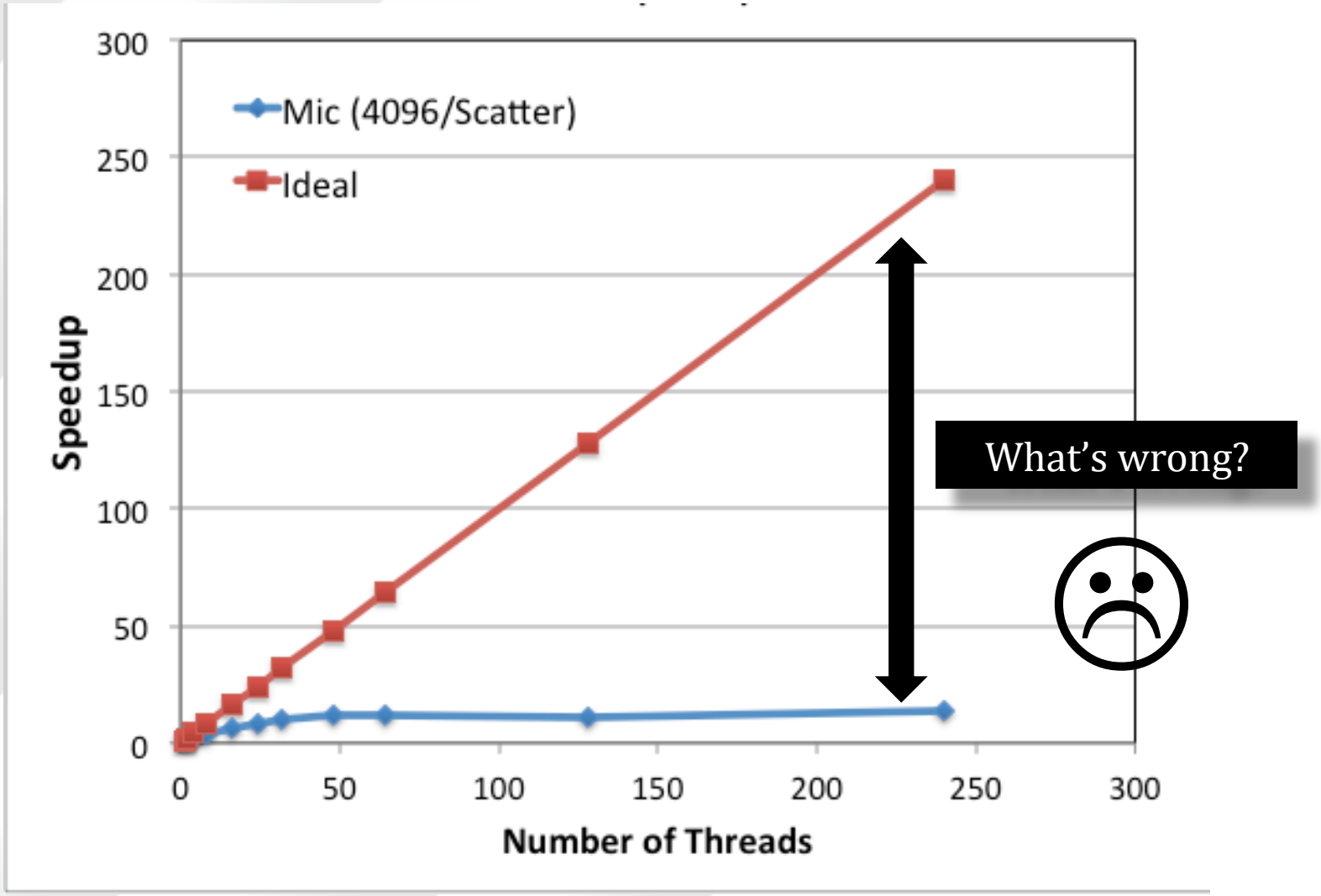


Ok, what about Mic's performance?

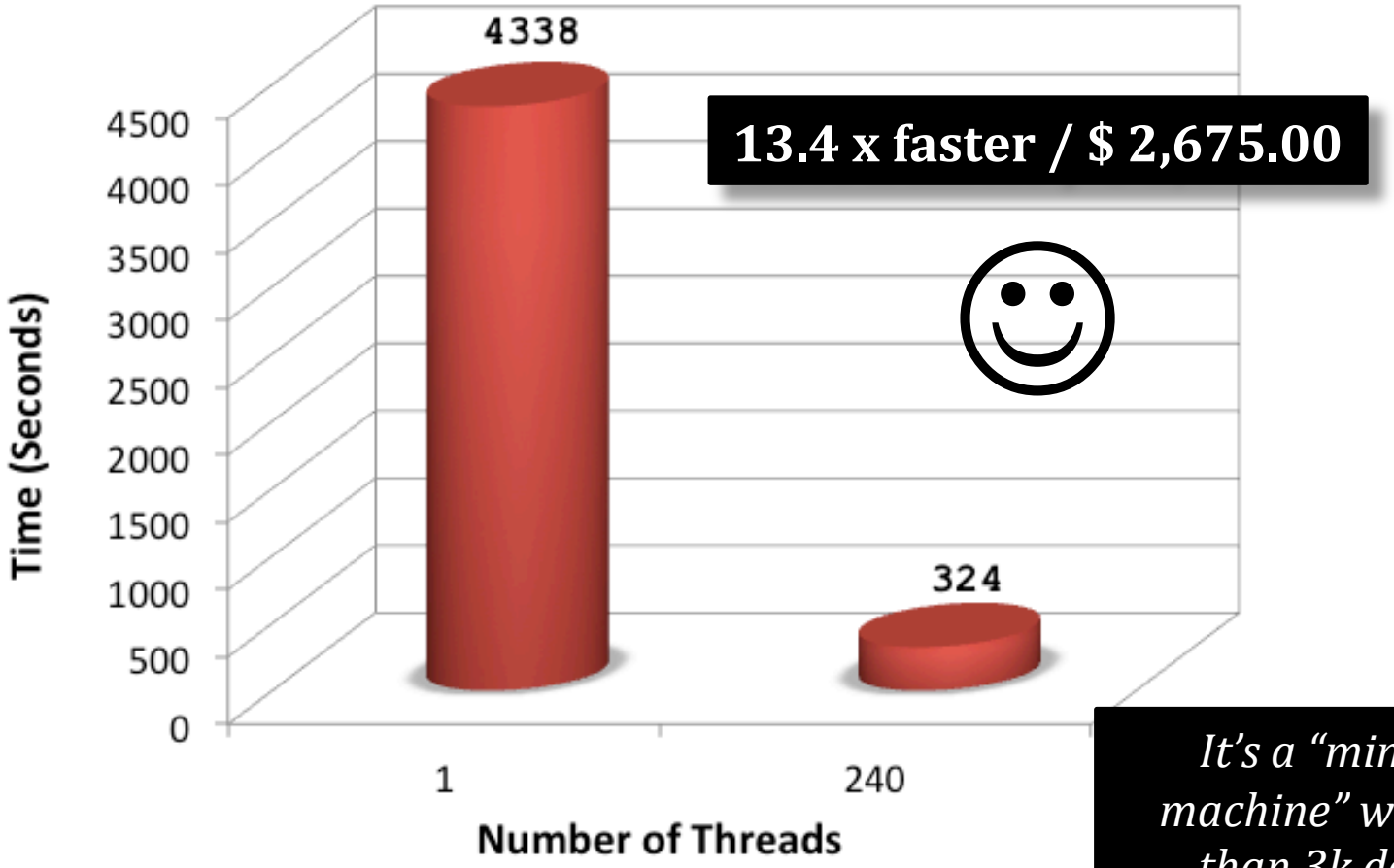
Host x Mic Performance



Mic's Speedup: How do we see it?



Mic's Speedup: How would Intel show it?

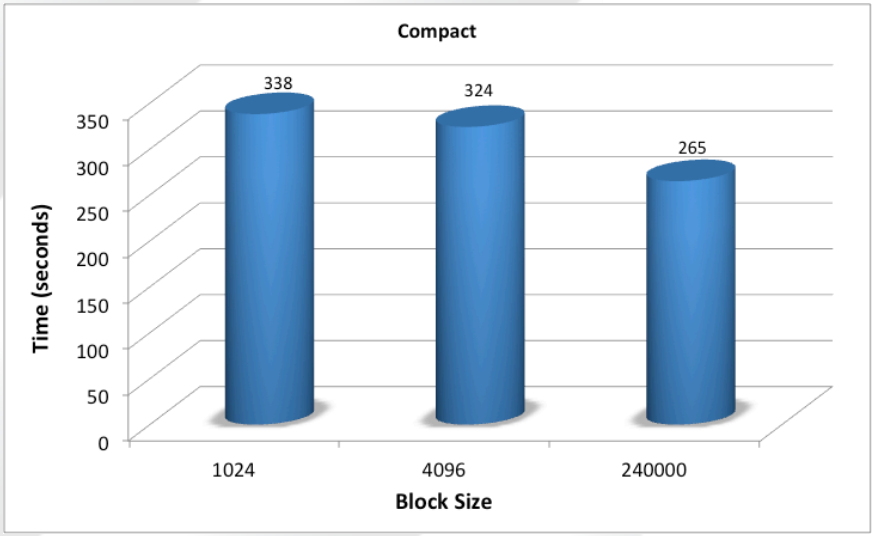


13.4 x faster / $ 2,675.00

It's a "mini shared memory machine" with 61 cores for less than 3k dollars?! Not bad...

Improving Many-Cores Performance

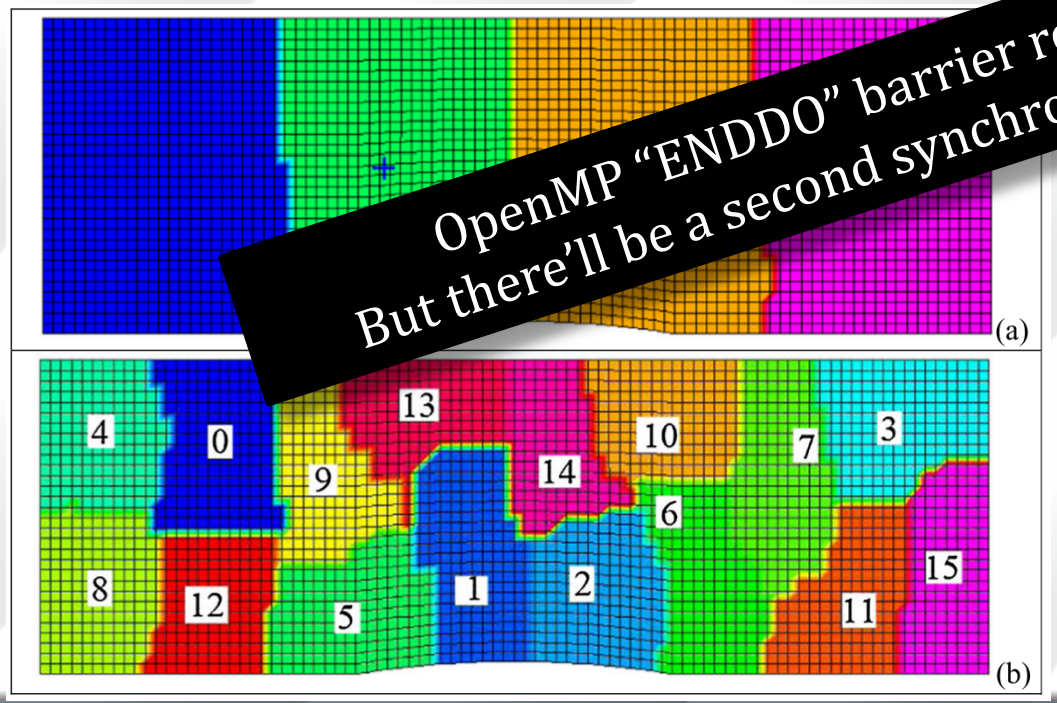
- ***OpenMP performance, in EdgeCFD, is affected by:***
  - *Block sizes → Bigger blocks means denser inner loops and less barrier synchronization.* ***Good! But it's not enough... ☹***



- ***What could/should be done?***
  - *Change the way the mesh is blocked/colored! → Partitioned!*

Cache Friendly Mesh Coloring

- ***Apply mesh partitioning in two steps***
  - *First step for MPI and second step for OpenMP*
- ***See Moureau, V., Comptes Rendus Mecanique, 2011 and Kannan, R., IJNMF, 2013 for details***



OpenMP "ENDDO" barrier removed!
But there'll be a second synchronization step

```
ielm = 0
do icor = 1, ncores
   nvec = ielblk(icor)
!$OMP  PARALLEL DO
   do i = ielm+1, ielm+nvec
      ! Retrieve element nodes
      x(no) = x(no) + a
   enddo
!$OMP END PARALLEL DO
   ielm = ielm+nvec
enddo
```

Closure for Discussion

- ***Emerging Challenges for EdgeCFD on Many-Cores architectures***
  - *Improve OpenMP performance! OpenMP is alive (and strong)*
  - *Evaluate new blocking schemes on MICs*
- ***Xeon Phi seems promising…***
  - *"A 20 years-old shared memory machine in a PCIe board!"*
- ***How should we combine many parallel and vectorization levels?***
- ***Application MUST get more Threaded-based;***
- ***Other challenges (not covered in this talk):***
  - *Visualization (when, where and how?)*
    - *Coprocessing (again)*
    - *Data formats and layouts to save disk space*
    - *Keep necessary information, forget unnecessary data*

Thanks for your Attention!

We're trying to figure a way to get out of the jungle

;-)

COPPE - Federal University of Rio de Janeiro

Third Brazil-France Workshop

On High Performance Computing and Scientific Data Management Driven by Highly Demanding Applications

02-05 September, Bordeaux, France 2013

Emerging Challenges for EdgeCFD Simulations in Massively Many-core Architectures

Renato N. Elias[1]

renato@nacad.ufrj.br

[1] High Performance Computing Center (NACAD)

www.nacad.ufrj.br

Civil Engineering Department (PEC/COPPE)

www.pec.coppe.ufrj.br

Federal University of Rio de Janeiro (UFRJ)

www.ufrj.br

NACAD