Slurm Configuration Impact on Benchmarking

José A. Moríñigo, Manuel Rodríguez-Pascual, Rafael Mayo-García

CIEMAT - Dept. Technology
Avda. Complutense 40, Madrid 28040, SPAIN

Slurm User Group Meeting'16 - Athens, Greece – Sept. 26-27, **2016**

Overview

- Introduction
- Facility
- Design of experiments
- NAS Parallel Benchmarks
- Results
- What's next?

Introduction

- **Prospect on HPC environments?**
  - **App's with very different requirements may coexist**
    - Execution time
    - Degree of paralelism
    - Required computational resources
    - % serial App's in clusters – perturbations?
    - **....**

Introduction

- **Prospect on HPC environments?**
  - **App's with very different requirements may coexist**
    - Execution time
    - Degree of paralelism
    - Required computational resources
    - % serial App's in clusters – perturbations?
    - **....**
  - **A place for sharing resources is foreseen**

    …but how App's performance is affected?:

    - Weak – scaling $\leftrightarrow$ Strong – scaling
    - CPU – bounded $\leftrightarrow$ Memory – bounded

Introduction

- **This scenario throws a bunch of questions:**
  - What to do with partially-filled multicore-CPUs?
    - Sharing $\Rightarrow$ Competition for resources, slow-down
      Always happens?
  - How is the performance sensitivity to the Application itself?
    - CPU- vs. Memory-bounded
  - To which extent the system architecture drives App's behaviour?
  - Best / Optimum strategy?

  ….

  Answers will lead to better exploitation and cost-effective strategies using HPC facilities

Introduction

- **Motivation of this work**
  - **Focus**: results (answers) will impact on

specific scheduling decisions

Maximize performance  Minimize energy consumption

Sweet point?

Introduction

- **Motivation of this work**
  - **Focus**: results (answers) will impact on

specific scheduling decisions

Maximize performance  Minimize energy consumption

Sweet point?

  - **Trade-off** to clarify **sensitivity to Slurm setups**:
    - **Behaviour of sci-App's** on modern HPC facilities
    - Measuring what happens **at ideal and "production"** computing conditions at definite scenarios.

Introduction

- **Motivation of this work**
  - **Focus**: results (answers) will impact on

specific scheduling decisions

Maximize performance  Minimize energy consumption

Sweet point?

  - **Trade-off** to clarify **sensitivity to Slurm setups**:
    - **Behaviour of sci-App's** on modern HPC facilities
    - Measuring what happens **at ideal and "production"** computing conditions at definite scenarios.

… and what about **exploring tasks migration as a tool** to improve computational efficiency of App's?

Overview

- Introduction
- Facility
- NAS Parallel Benchmarks
- Design of experiments
- Results
- What's next?

- **Starting Point: Our HPC + Slurm characterization**

  2 families of Benchmarks:

  1.- **System** Benchmarks: "raw" performance of HPC components:

  • **OSU Micro-Benchmark** (Ohio State Uni)
  • **STREAM**
  • **Bonnie++**
  • **Intel Memory Latency Checker**

  2. – Scientific **Application** Benchmarks: behaviour when running real applications:

  NAS (more on this later…)

Facility

- **Cluster ACME**: state-of-the-art, for research
  - 10 nodes:
    - **8 compute nodes (128 cores)**
    - 2 Xeon-Phi nodes

Basic data:

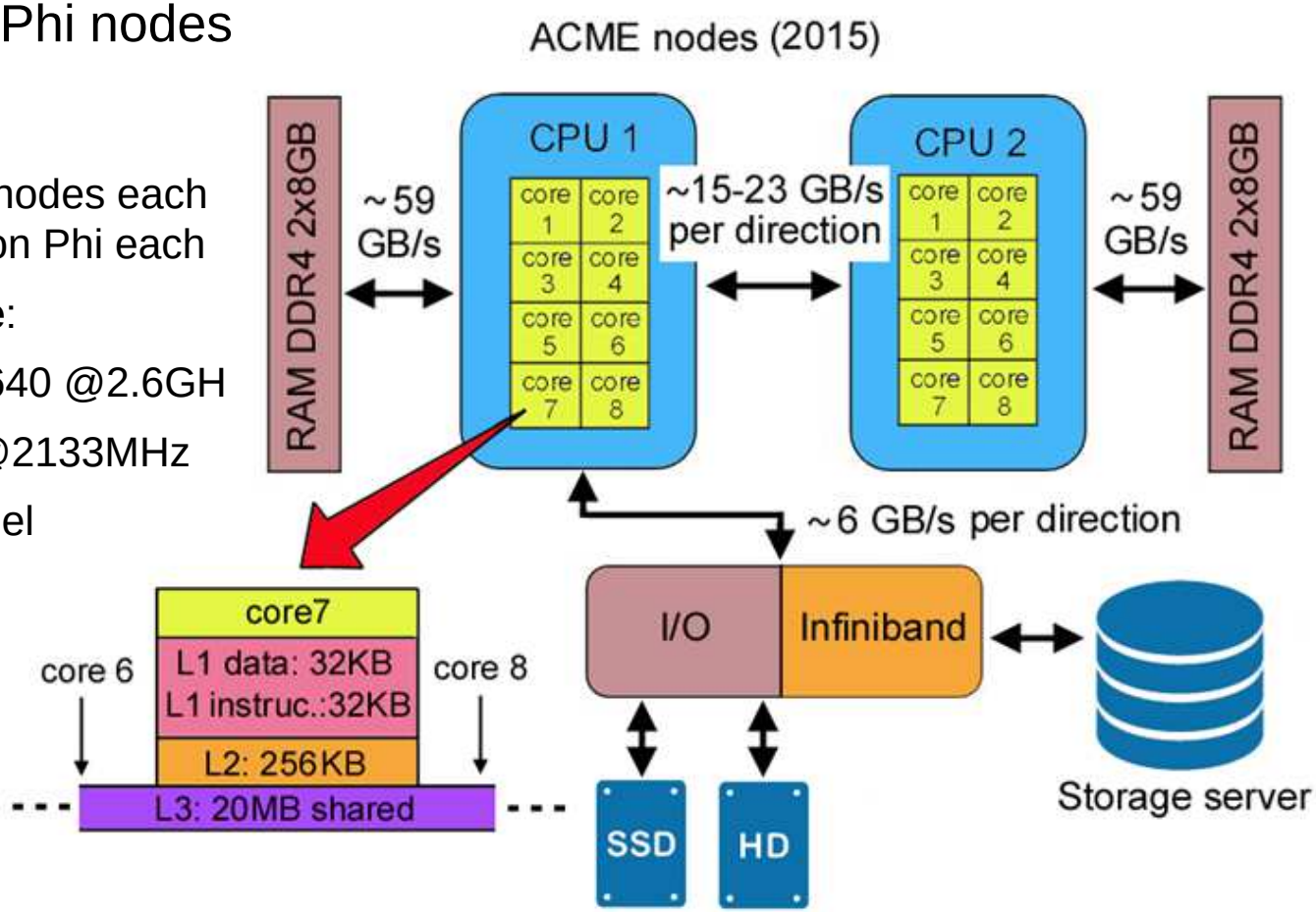
2 Bull chassis with 4 nodes each
2 nodes with one Xeon Phi each

Each node:

2 x 8-core Xeon E5-2640 @2.6GH

32GB DDR4 RAM @2133MHz

NUMA model



Facility

- **ACME cluster**: state-of-the-art, for research
  - 10 nodes:
    - **8 compute nodes (128 cores)**
    - 2 Xeon-Phi nodes

$$\textbf{Ratio} = \frac{\text{Intra-node BW}}{\text{Inter-node BW}} \approx \textbf{3 - 3.5}$$

Basic data:

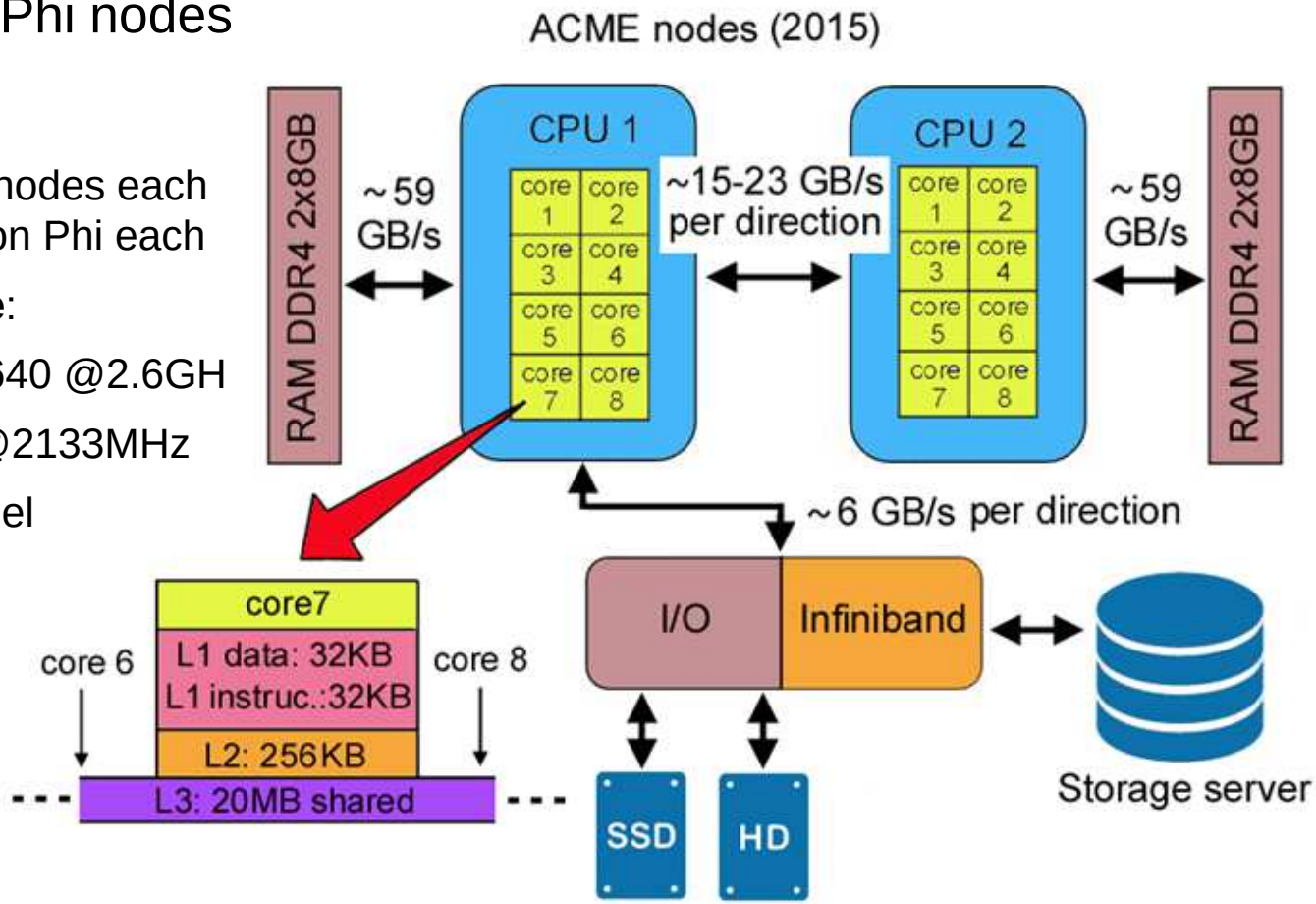
2 Bull chassis with 4 nodes each
2 nodes with one Xeon Phi each

Each node:

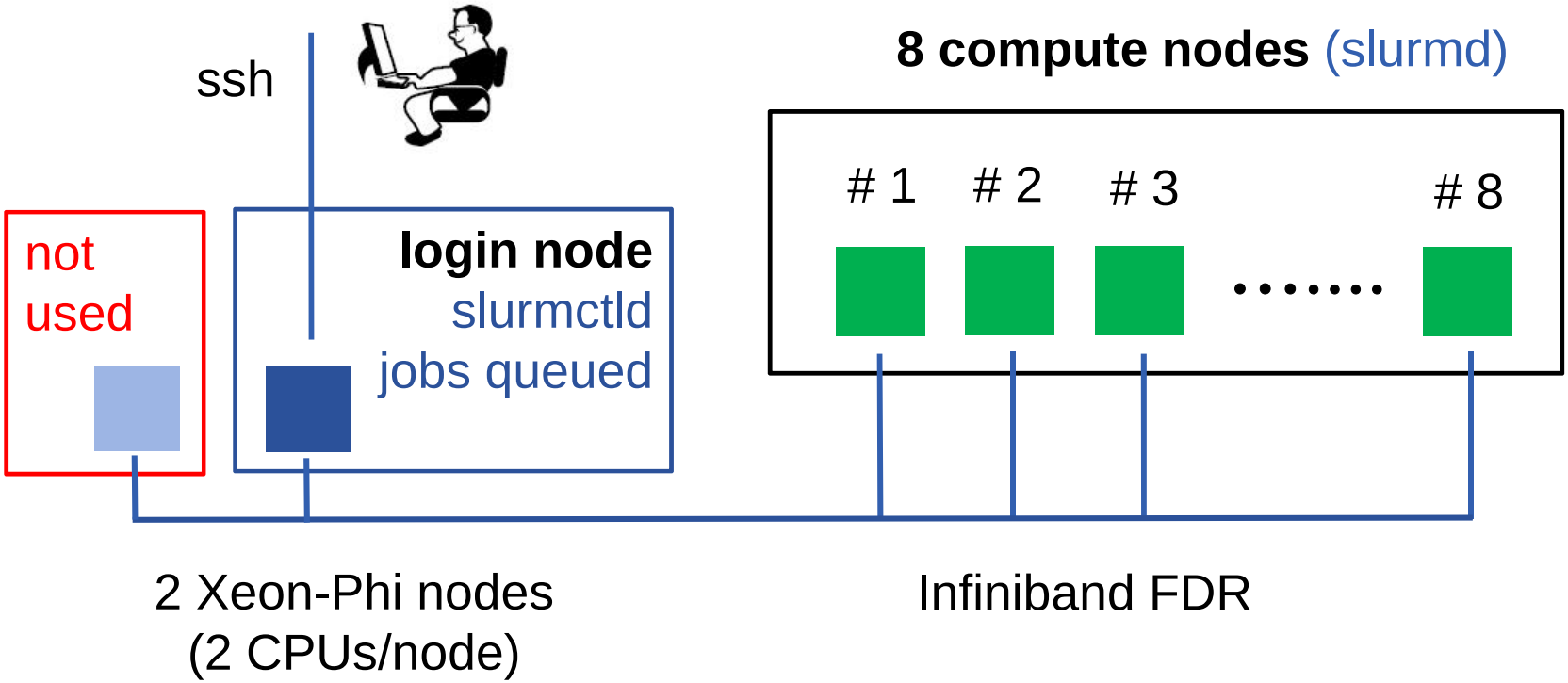
2 x 8-core Xeon E5-2640 @2.6GH

32GB DDR4 RAM @2133MHz

NUMA model



- **ACME cluster with Slurm**
  - version 16.05
  - mvapich2-2.2b
  - Requirement: minimize perturbations during benchmarking



Overview

- Introduction
- Facility
- NAS Parallel Benchmarks
- Design of experiments
- Results
- What's next?

Parallel NAS Benchmarks

- **NAS Parallel Benchmarks** (NPB) v. 2.0:

  Numerical **A**erodynamic **S**imulator (NASA)
  (1991 – present, added kernels in recent versions).

  - Focus on Aerosciences
  - Fortran
  - MPI-based
  - **Building blocks**:

Benchmark code	Class A	Class B	Class C
Embarrassingly parallel (EP)	2^{28}	2^{30}	2^{32}
Multigrid (MG)	256^3	256^3	512^3
Conjugate gradient (CG)	14000	75000	150000
3-D FFT PDE (FT)	$256^2 \times 128$	512×256^2	512^3
Integer sort (IS)	2^{23}	2^{25}	2^{27}
LU solver (LU)	64^3	102^3	162^3
Pentadiagonal solver (SP)	64^3	102^3	162^3
Block tridiagonal solver (BT)	64^3	102^3	162^3

  7 kernels: BT, CG, LU, IS, **EP**, FT, **MG**

  CPU- bounded ⟺ **Memory-bounded**

- **Usefulness:**

  It is expected that results, in terms of **computational efficiency,** be of interest as feedback to sci-groups of production clusters.

Parallel NAS Benchmarks

- **NAS** kernels are **scalable in size**: A, B, C, D,… → **Classes**

Eg. - **MG** (Multi-Grid):

Iterates a 3D scalar Poisson eqn. using a set of nested grids.

Problem size ↓ ↑ **Memory** ↑ **CPU**

Class	*Problem Size*	*Iterations*
A	256^3	4
B	256^3	20
C	512^3	20

MG is a typical case of Memory-bounded algorithm…

…most kernels are **a mixture** of Memory plus CPU-demanding

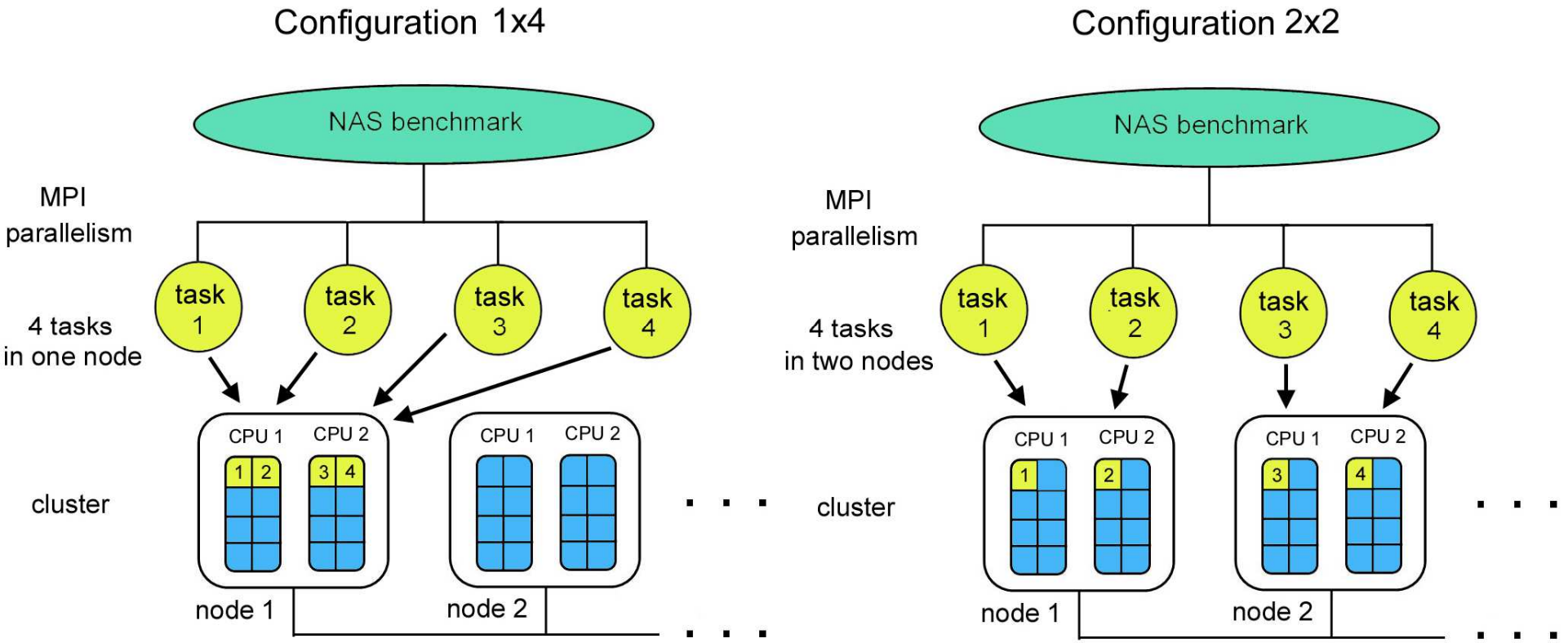
Overview

- Introduction
- Facility
- NAS Parallel Benchmarks
- Design of experiments
- Results
- What's next?

Design of experiments

- Some definitions
  - ***Mapping*** NAS kernels as MPI tasks onto groups of cores
  - **Configuration**, linked to the job:

nN x nT = [# Nodes] x [# MPI Tasks]



Design of experiments

- **Executions under Slurm setups:**
  - **Dedicated Network (reference setup):** 1 job at the same time running in the cluster.
  - **Dedicated Cores:** one-to-one assignment of cores to MPI tasks of the job.

Design of experiments

- **Executions under Slurm setups:**
  - **Dedicated Network (reference setup):** 1 job at the same time running in the cluster.
  - **Dedicated Cores:** one-to-one assignment of cores to MPI tasks of the job.
  - **Dedicated Nodes**: entire node asigned to execute MPI task of the same job.
  - **Dedicated Sockets**: a socket executes MPI tasks of the same jobs (no part of another job may share it during execution).

An experiment?: one NAS kernel (+ class)

\+

definite *nN x nT* configuration

\+

definite Slurm setup

~4000 exec's at each cluster setup

10 executions, averaged

Standard deviation < 1%

Design of experiments

- Slurm configuration
  - Modify **`slurm.conf`** :

Setup:

```
# SCHEDULING
#SchedulerType=sched/backfill
#SchedulerType=sched/builtin
SelectType=select/linear
…
…
```

Nodes definition:

```
NodeName=acme[11-14,21-24] CPUs=16 Sockets=2
  CoresPerSocket=8 ThreadsPerCore=1 State=UNKNOWN
```

  - **No preemption**

Dedicated Nodes:

SelectType=select/linear

Dedicated Sockets:

SelectType=select/cons_res
SelectTypeParameters=CR_Socket

Dedicated Cores:

SelectType=select/cons_res
SelectTypeParameters=CR_CPU

(cons_res: consumable resources)

Design of experiments

- Parameters
  - **Range** of partitions: 1 (serial), 2, 4, ….. 64, 128 tasks (MPI ranks)

Degree of parallelism	1 (serial)	2	4	8	16	32	64	128
Number of jobs	210	360	630	720	840	540	400	170
Number of jobs (%)	5.4	9.3	16.3	18.6	21.7	14	10.3	4.4

> 30%

  - **Consistency:** 3 repetitions of experiments, then averaged.
  - **Cost:**

    4 setups x {3 repetitions x ~4000 jobs/setup} = ~ **48,000 sent jobs**

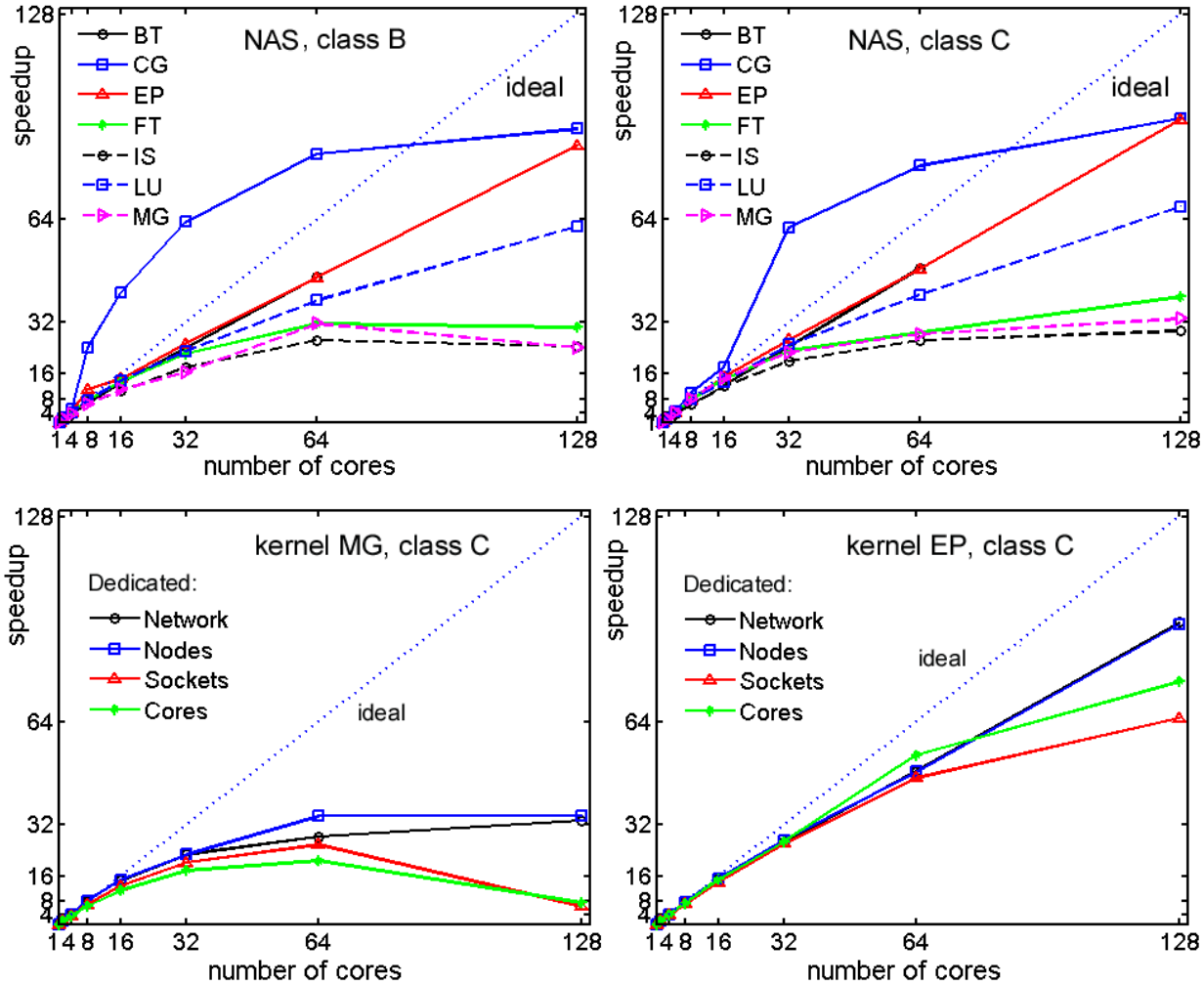
    (~ 4000 jobs sent to the queue at once)

Overview

- Introduction
- Facility
- NAS Parallel Benchmarks
- Design of experiments
- Results
- What's next?

Results

- **NAS Strong-scaling:** Far from optimum!
  - **Resource competition** is higher for Memory-bound kernels (**MG**)

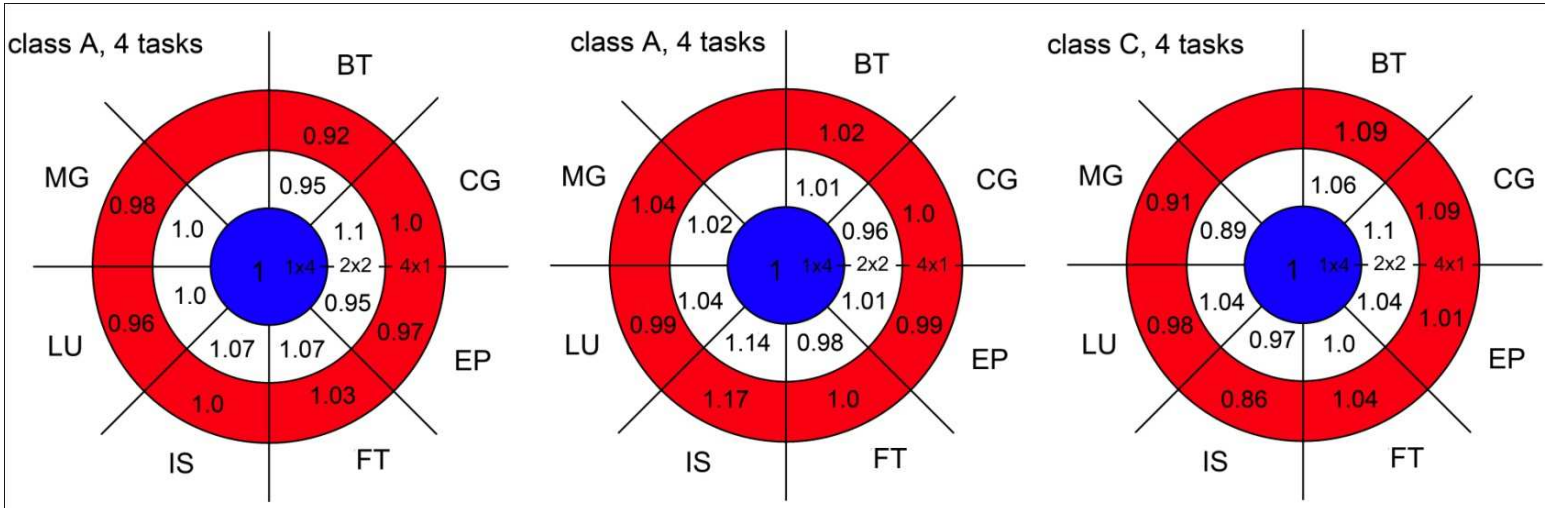


Results

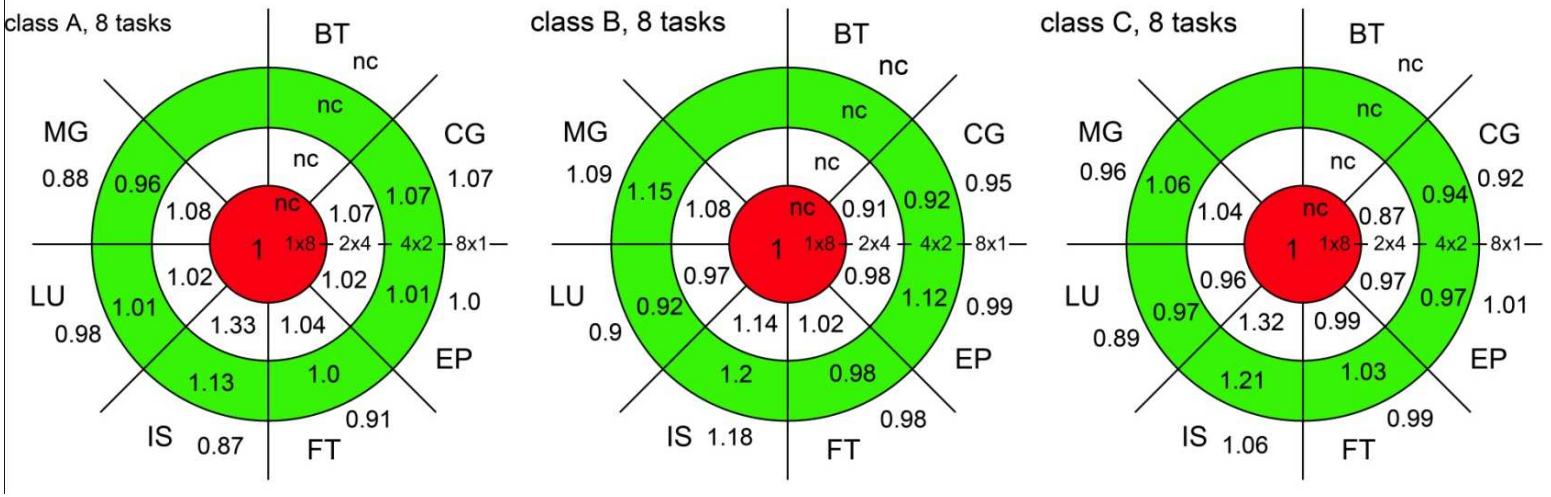
- **Dedicated Cores** setup: Realistic scenario in production

  - Nondimensional execution time (**4, 8**, 16, 32 tasks):

4 tasks



8 tasks

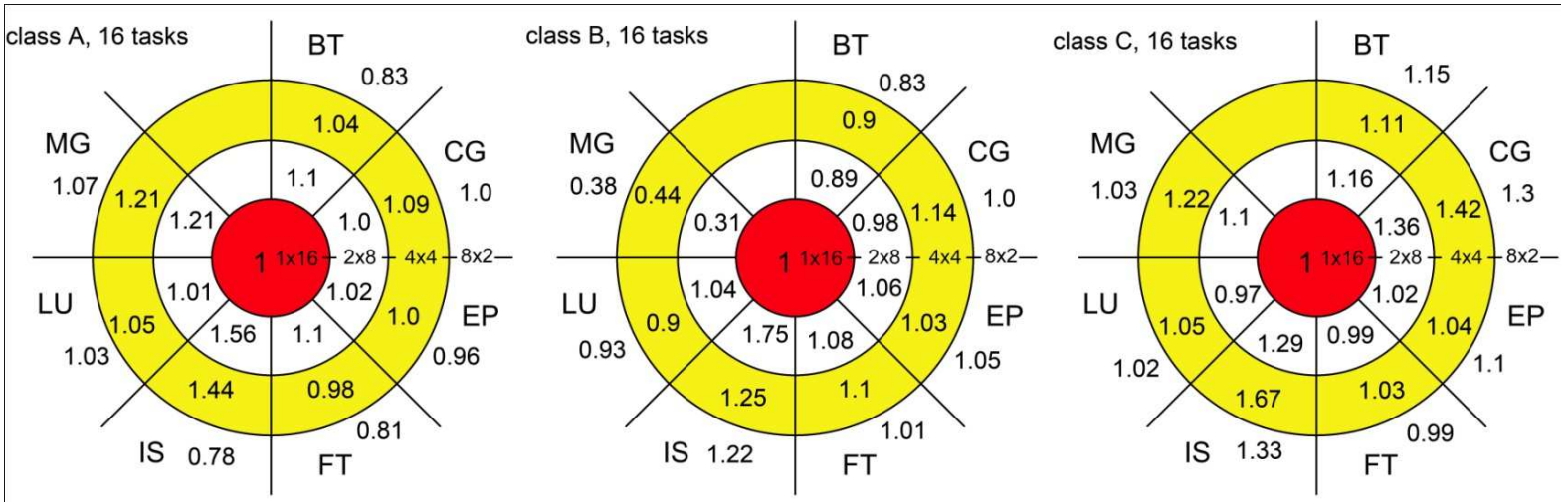


Results

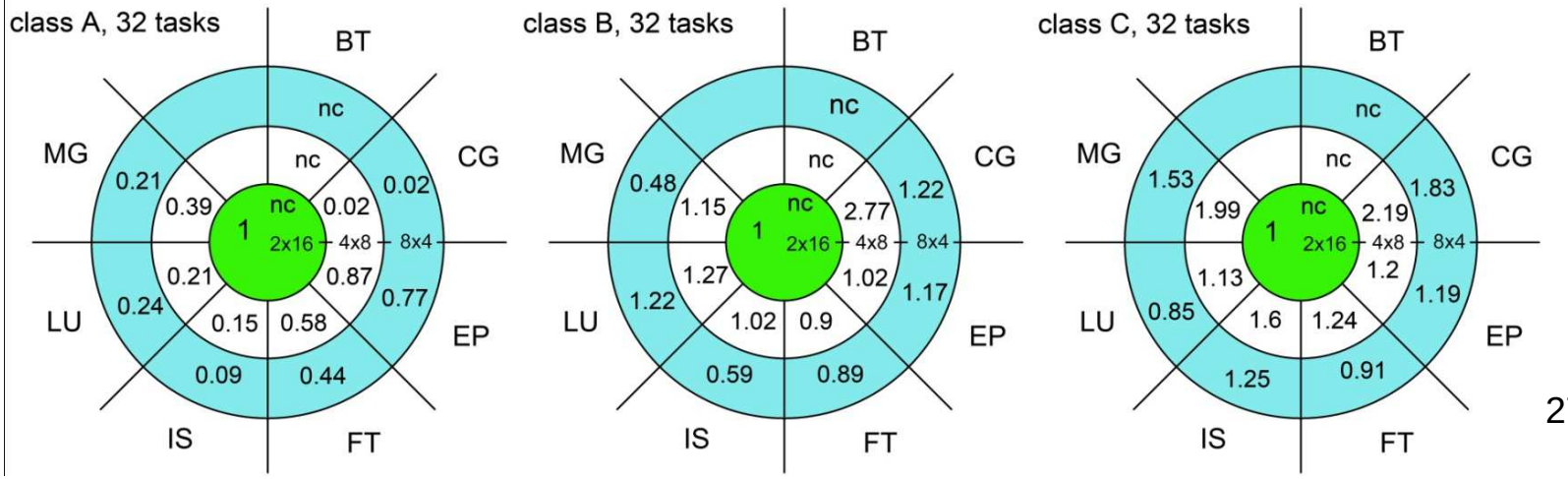
- **Dedicated Cores** setup (Cont.)

  - Nondimensional execution time (4, 8, **16, 32** tasks):

16 tasks



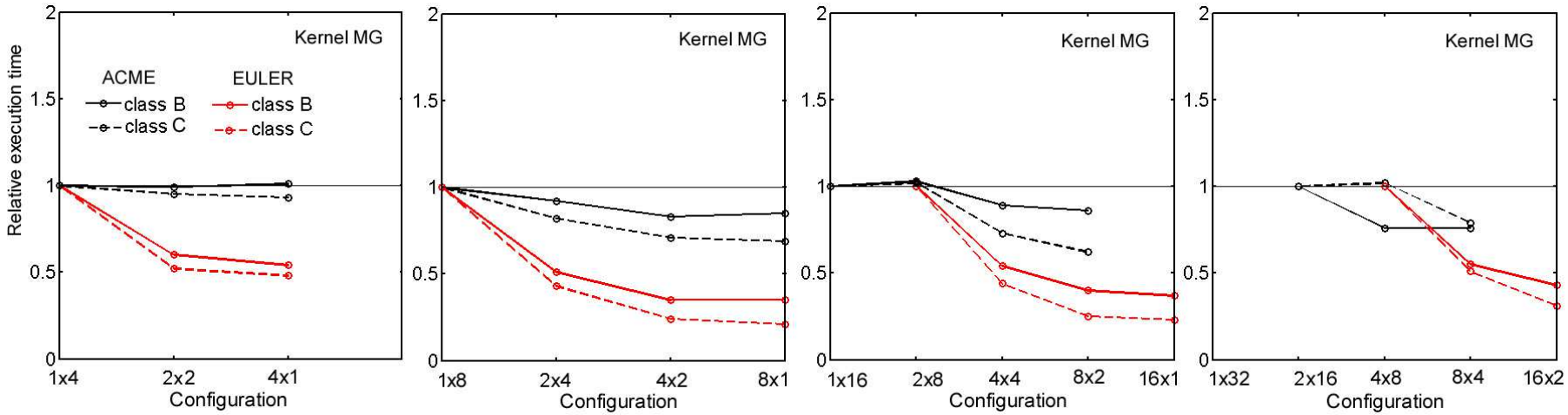
32 tasks



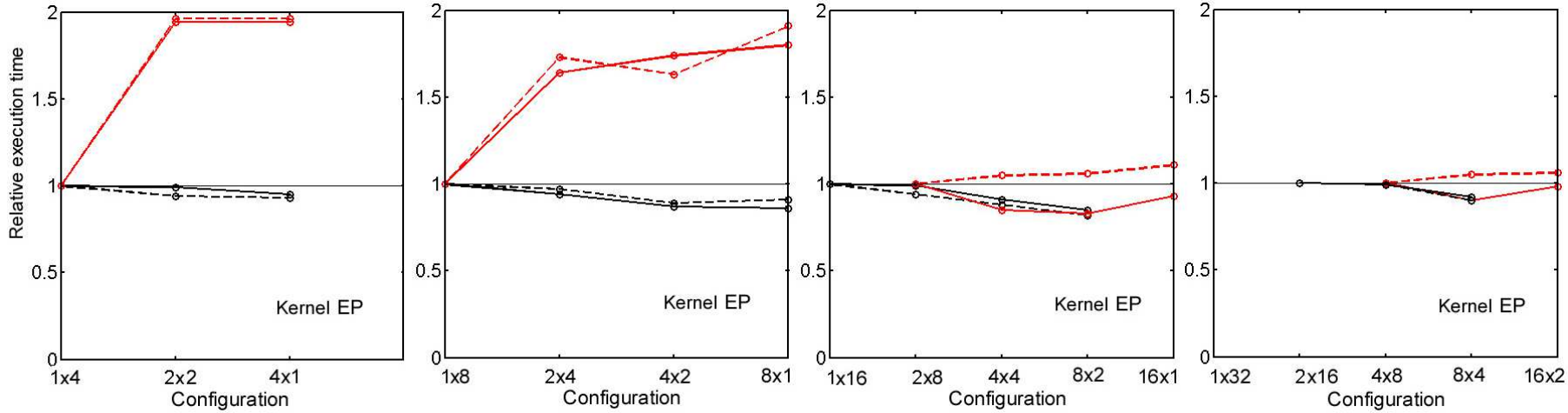
Results

- Focus on kernels **EP** and **MG** & **Dedicated Nodes** setup

MG



EP

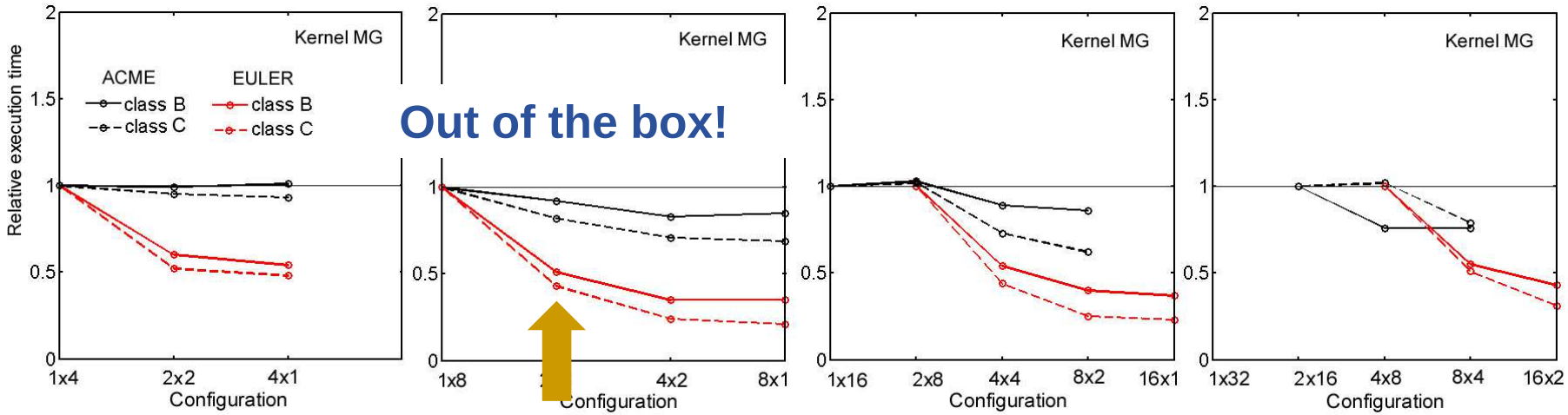


Results

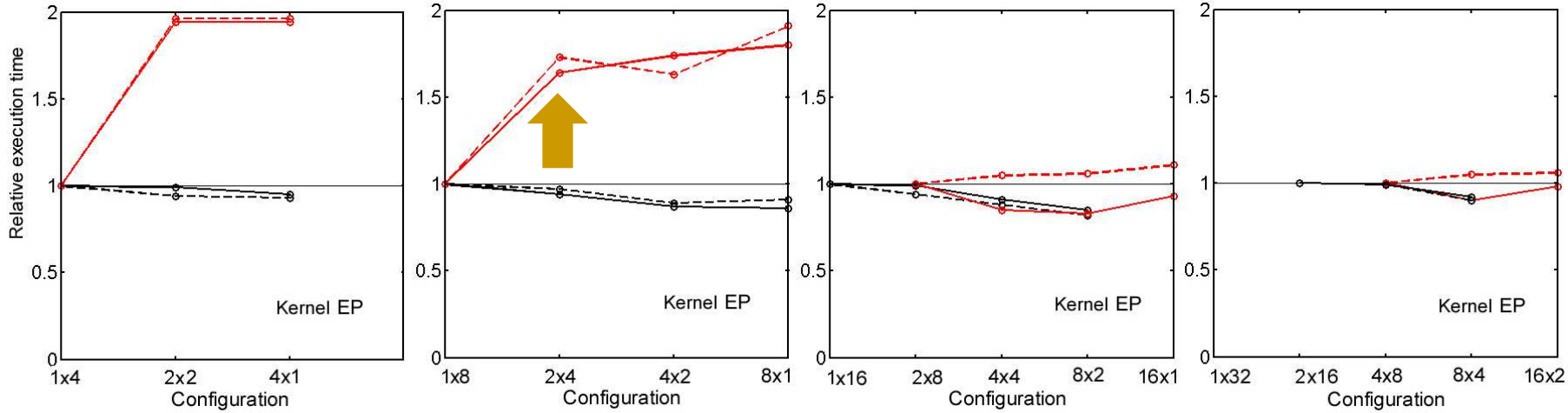
- Focus on kernels **EP** and **MG** & **Dedicated Nodes** setup

MG

Out of the box!



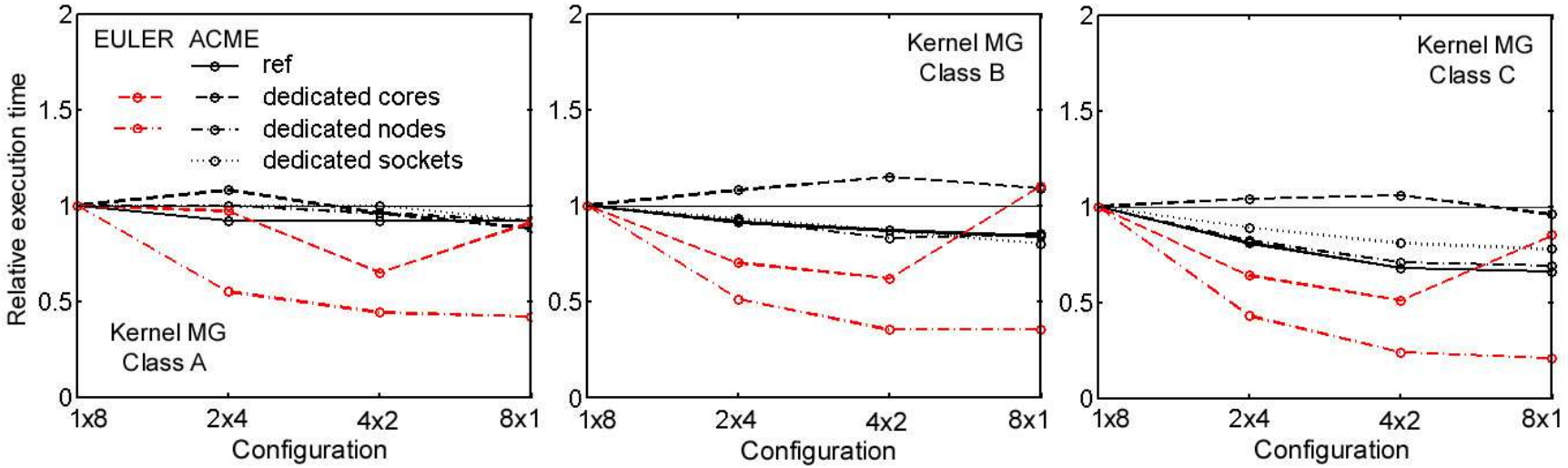
EP



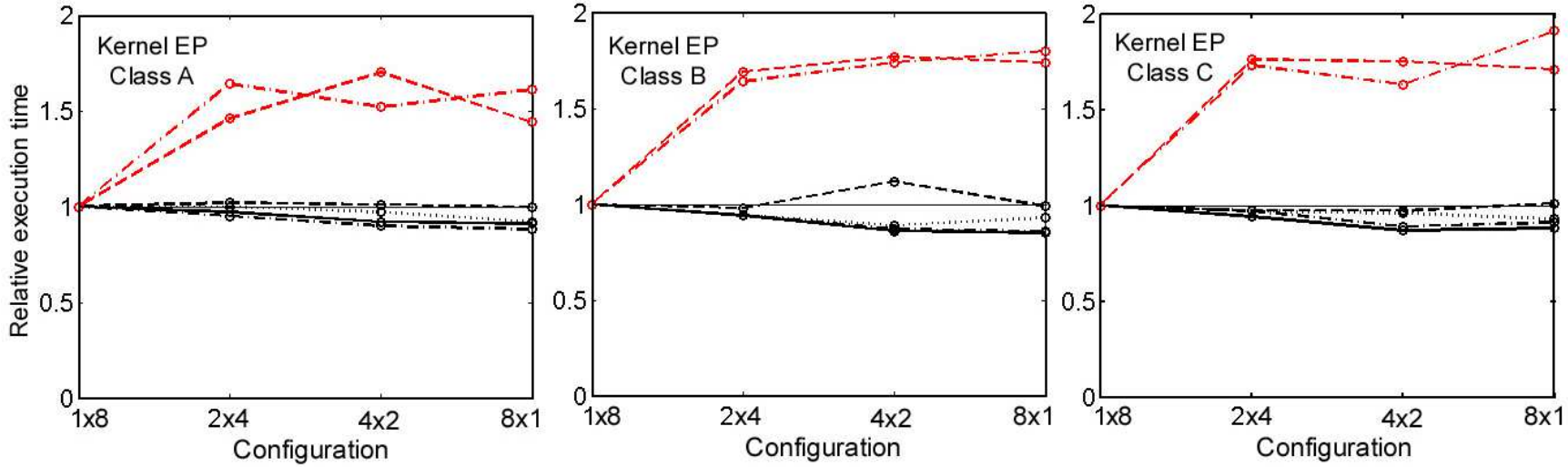
Results

- **Sensitivity to cluster setup:** kernels **MG** and **EP**

MG

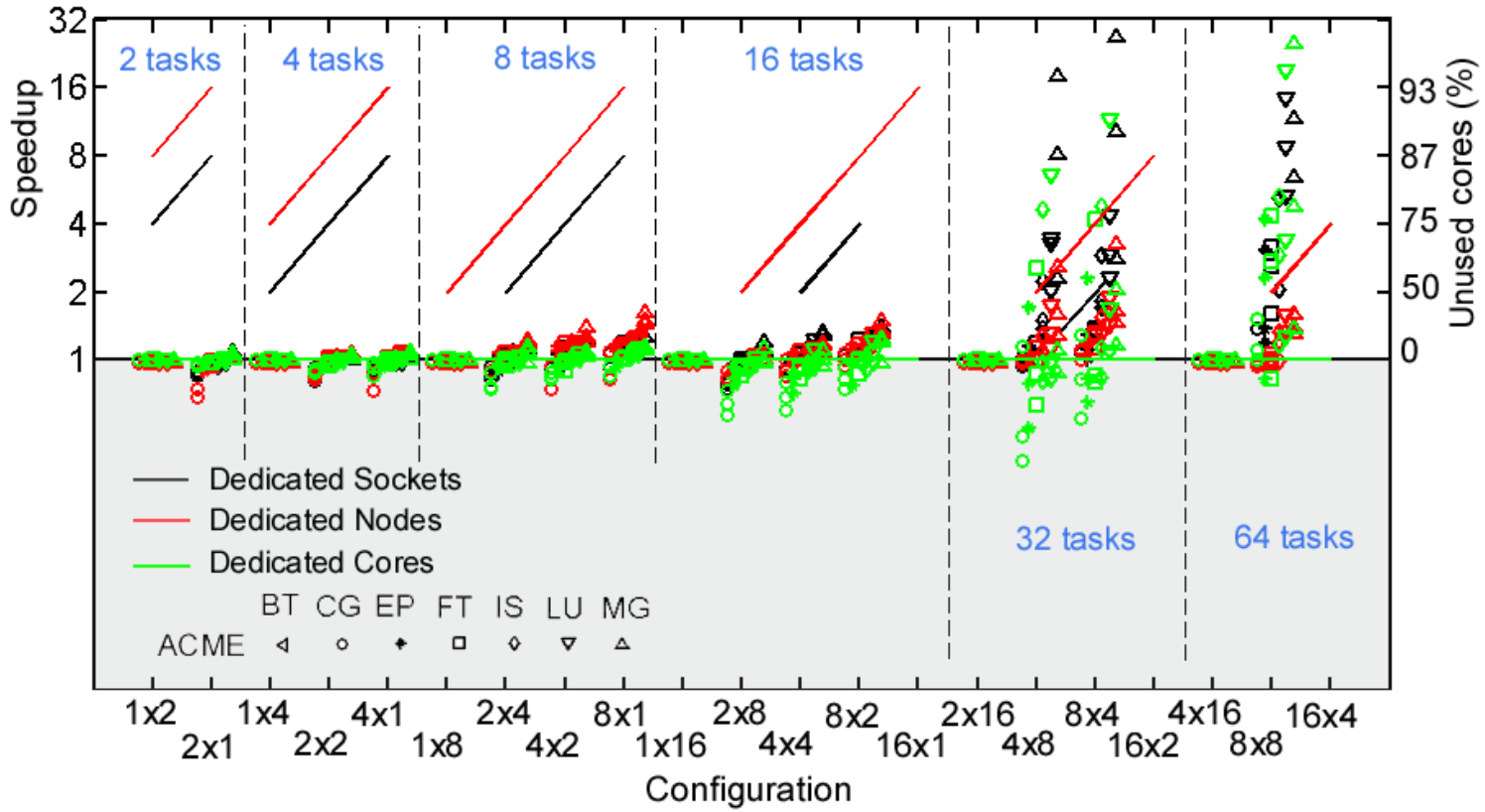


EP



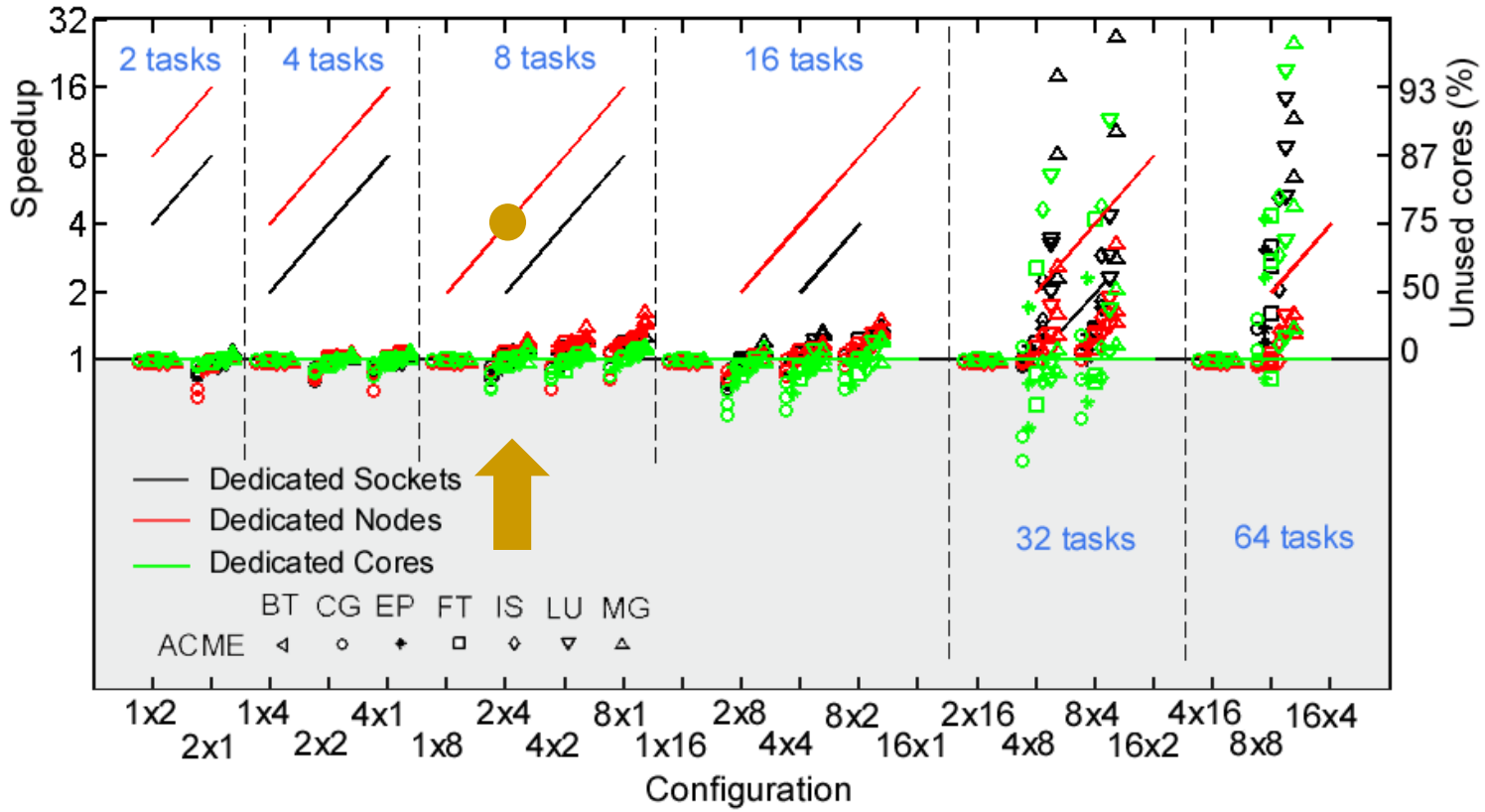
- **Maps of Speedup VS. Computational resources**
  - How many unused resources?

⇒ Performance / Energy saving decisions



Results

- **Maps of Speedup VS. Computational resources**
  - E.g.: **2x4 configuration**
    - Dedicated **Nodes** → 4 tasks per socket: **75%** unused
    - Dedicated **Cores** → (idem): **0%** unused



Overview

- Introduction
- Facility
- NAS Parallel Benchmarks
- Design of experiments
- Results
- What's next?

What next?

- **Lessons learnt:**
  - **Computational efficiency?**
    - **No one-rule** regarding grouping MPI tasks.
    - Depends on the Slurm setup, but some statistical tendences.
    - Under **Dedicated Cores setup**, there are **more** cases at which **grouping improves** computational efficiency.
  - **Not enough!**…
    - Interest in **MPI application codes (Materials, Fusion, Wind Energy,..)**
    - How do **OpenMP + MPI** hybrid codes modify the picture?

What next?

- **In perspective**…

  Execution of **NAS,** a first step (present study)

  - Initial characterization of our HPC facility.

  Ongoing: Execution of **application (scientific) codes**

  - A real scenario
  - Very different MPI-based codes at hand:

    LAMMPS (MD) ⟺ MC-based solver, …

  - How does the picture change?

THANK YOU!!!

CIEMAT – Avda. Complutense, 40 – 28040 Madrid, SPAIN

e-mail:
{josea.morinigo, manuel.rodriguez, rafael.mayo} @ciemat.es

Take a look at our group site!
http://rdgroups.ciemat.es/web/sci-track/