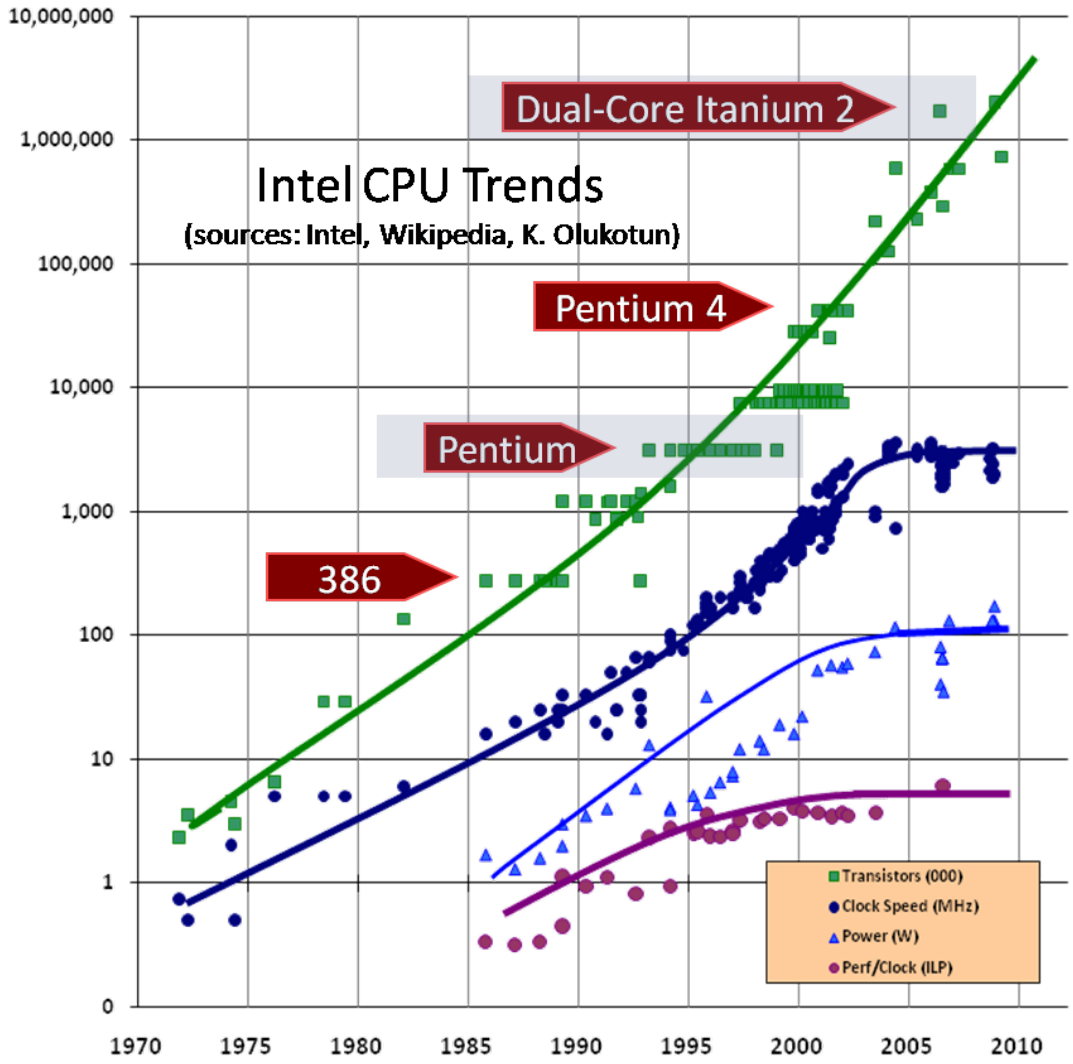
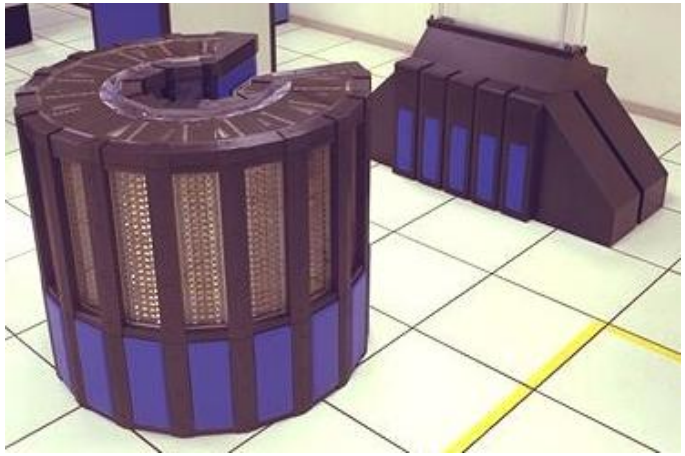
Parallelization issues

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Clock rate history

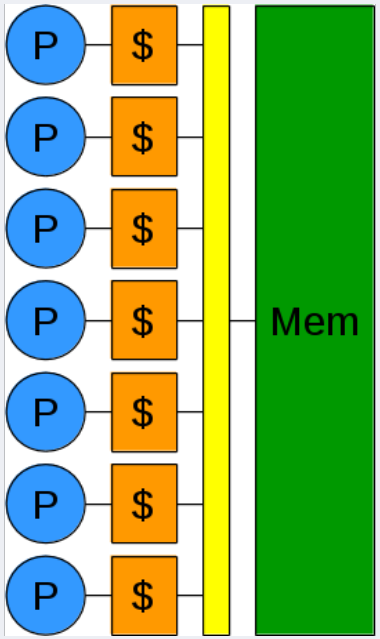


- No clock frequency scaling
  $\Rightarrow$ more performance by faster memory, more cache, **parallelism**
- Parallel structures are scalable
- But need proper algorithms and programming



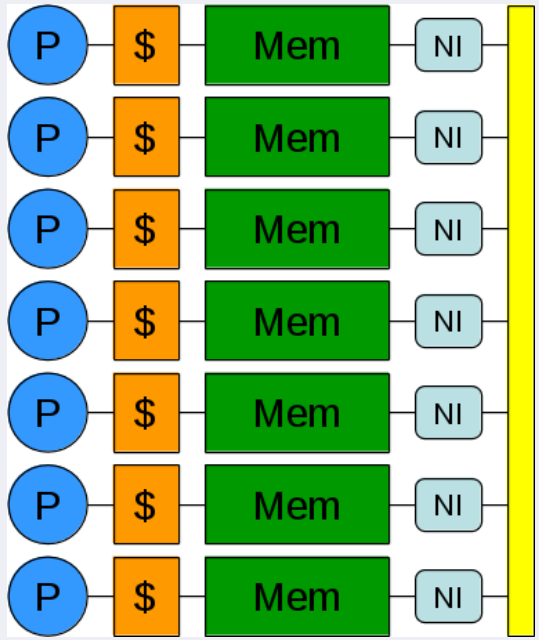
Types of parallelism

Shared memory



Multithreading

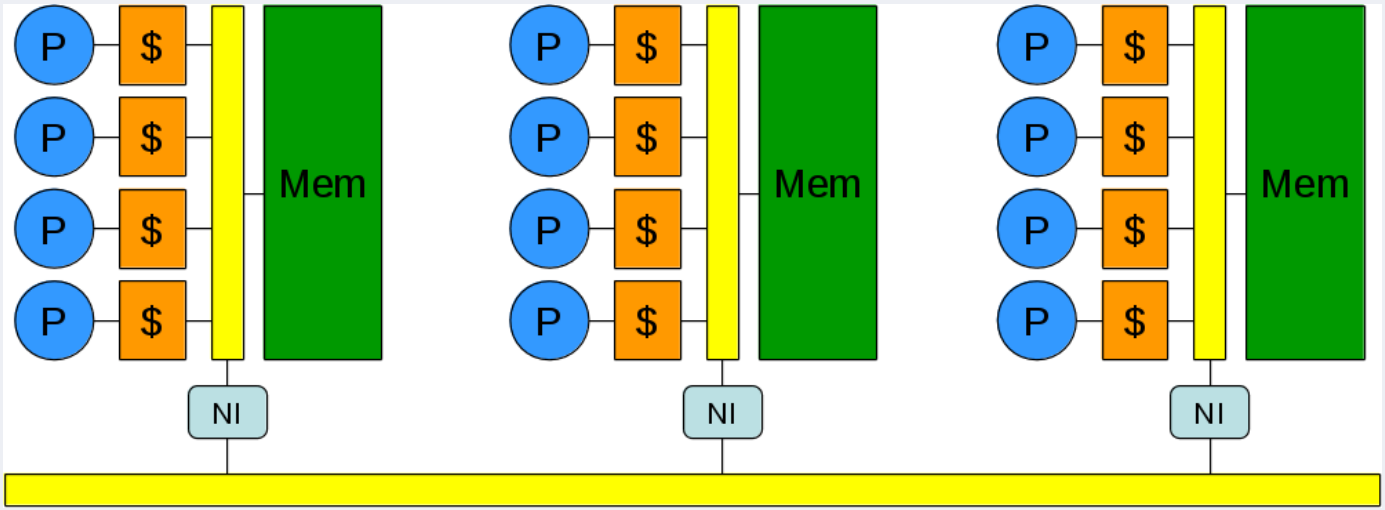
Distributed memory



MPI

Types of parallelism

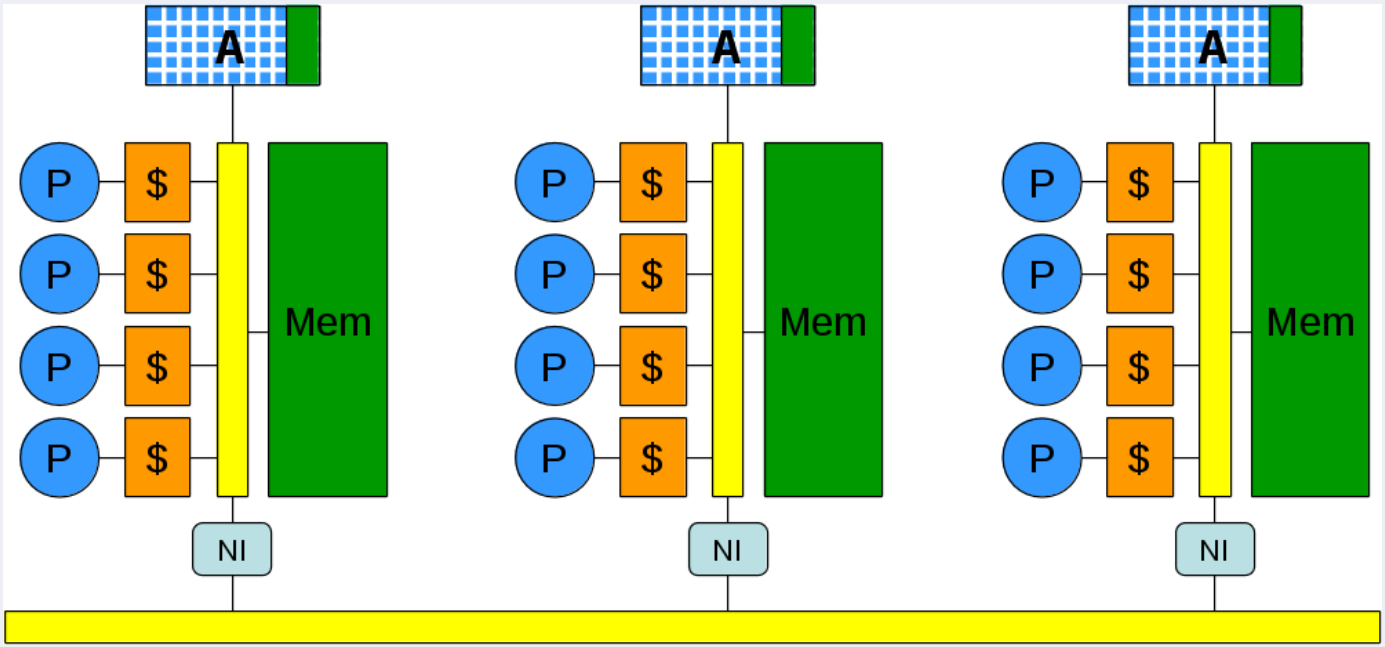
Hybrid systems



Multithreading + MPI

Types of parallelism

Heterogeneous systems



User's tasks

- Centers provide platform
- Code owners provide parallel programs
- User has to deal with:
  - Machine usage: Queueing systems
    optimal use of resources
  - Application configuration

More specialized/optimized code gives better performance
but needs more awareness in its configuration

User's tasks

- Centers provide platform
- Code owners provide parallel programs
- User has to deal with:
  - Machine usage: Queueing systems
    optimal use of resources
  - Application configuration

More specialized/optimized code gives better performance
but needs more awareness in its configuration

User's tasks

- Centers provide platform
- Code owners provide parallel programs
- User has to deal with:
  - Machine usage: Queueing systems
    optimal use of resources
  - Application configuration

More specialized/optimized code gives better performance
but needs more awareness in its configuration

Parallelization in SIESTA

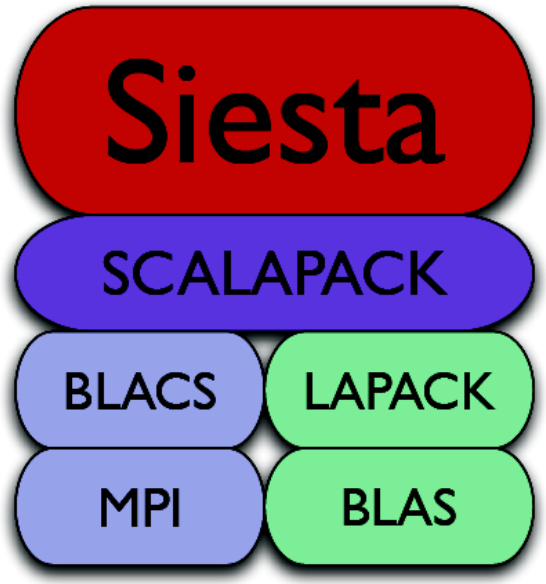
(For now) only distributed memory parallelization.

Several options:

- k-point parallelism
- Distributing orbitals and gridpoints
- 2-level parallelism in PEXSI solver

Parallelization in SIESTA

Software stack:



Parallelization in SIESTA

Compile options:

```
## MPI wrappers to compiler
FC=mpif90

## Flag activating MPI
FPPFLAGS= -DMPI
# eventually also -DBSC_CELLXC and/or -DMPI_TIMING

## Parallel linar algebra libraries
BLACS_LIBS = <yourlibs>
SCALAPACK_LIBS = <yourlibs>
LIBS = $(SCALAPACK_LIBS) $(BLACS_LIBS)

MPI_INTERFACE=libmpi_f90.a
MPI_INCLUDE=.
```

Parallelization in SIESTA

Running with MPI on local system:

```
mpirun -n 8 siesta < Si001+H2.fdf
```

Parallelization in SIESTA

Queue submit script (`launch.sh`) for LSF system:

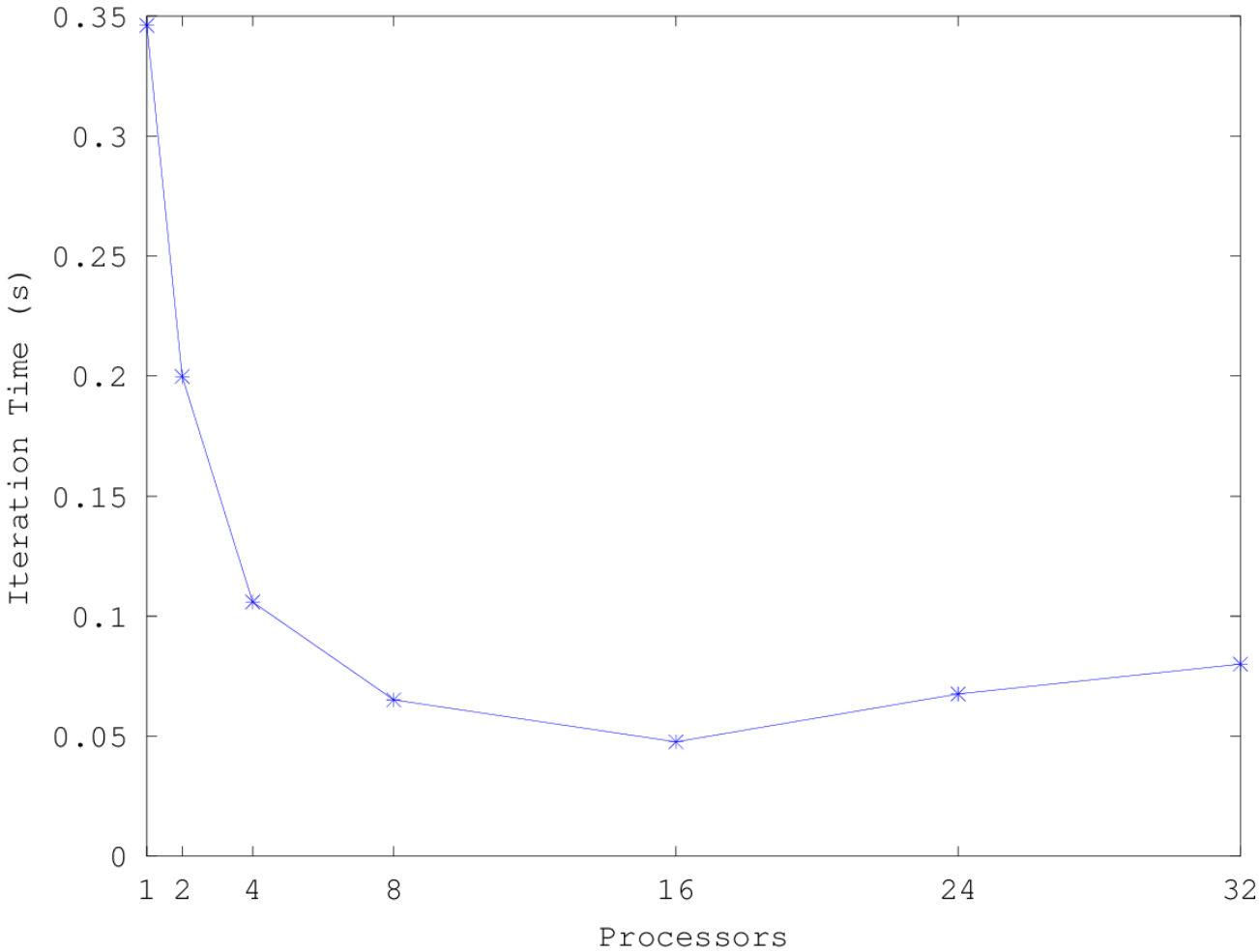
```
#! /bin/bash
#BSUB -J Siesta
#BSUB -cwd ./
#BSUB -n 16
#BSUB -oo std.out
#BSUB -eo std.err
#BSUB -R"span[ptile=16]"
#BSUB -W 00:20

mpirun siesta < Si001+H2.fdf
```

Submitting to the queue by:

```
bsub < launch.sh
```

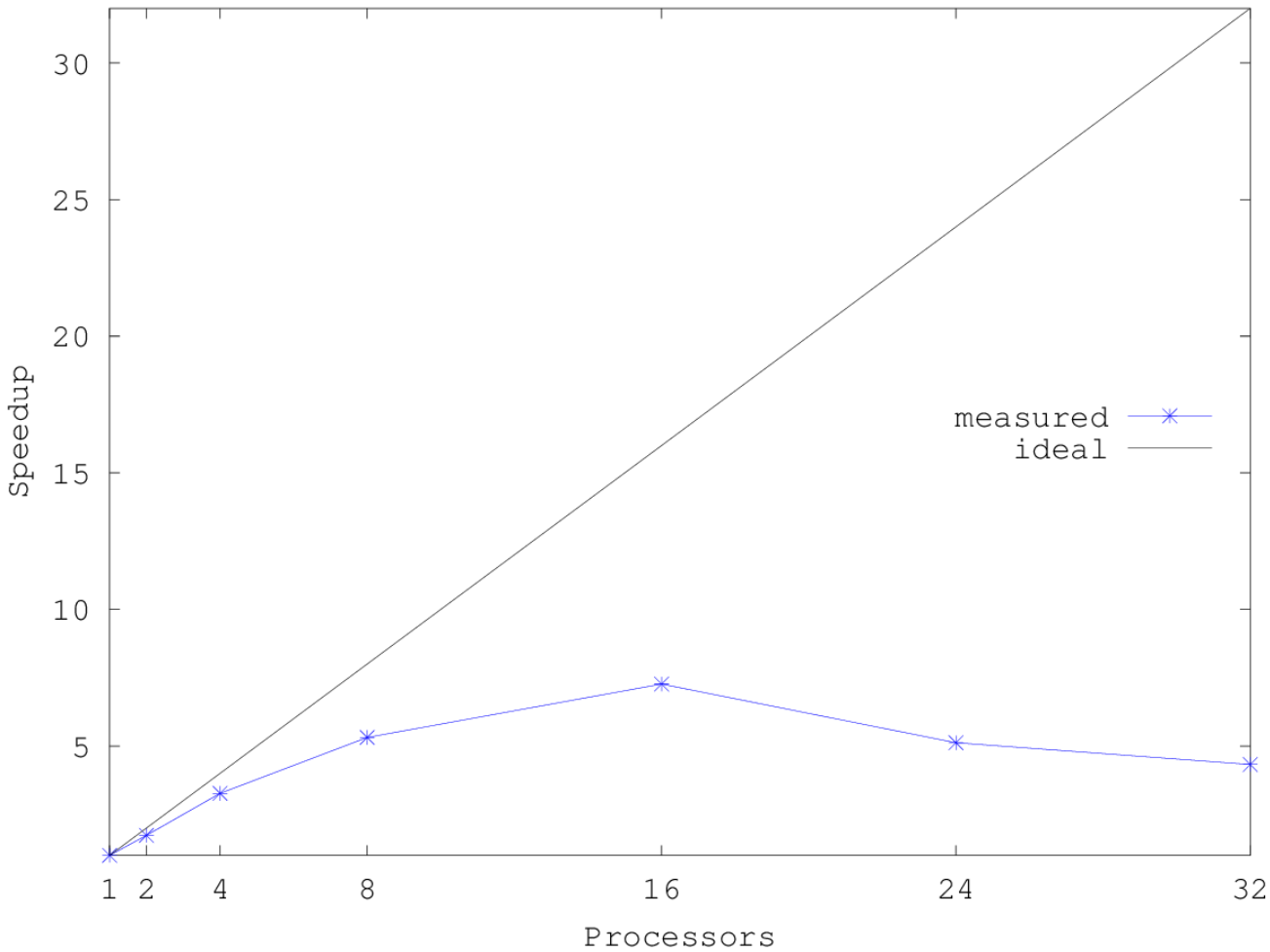
Scaling: Time

Example from the tutorial `visual/MolecularDynamics`



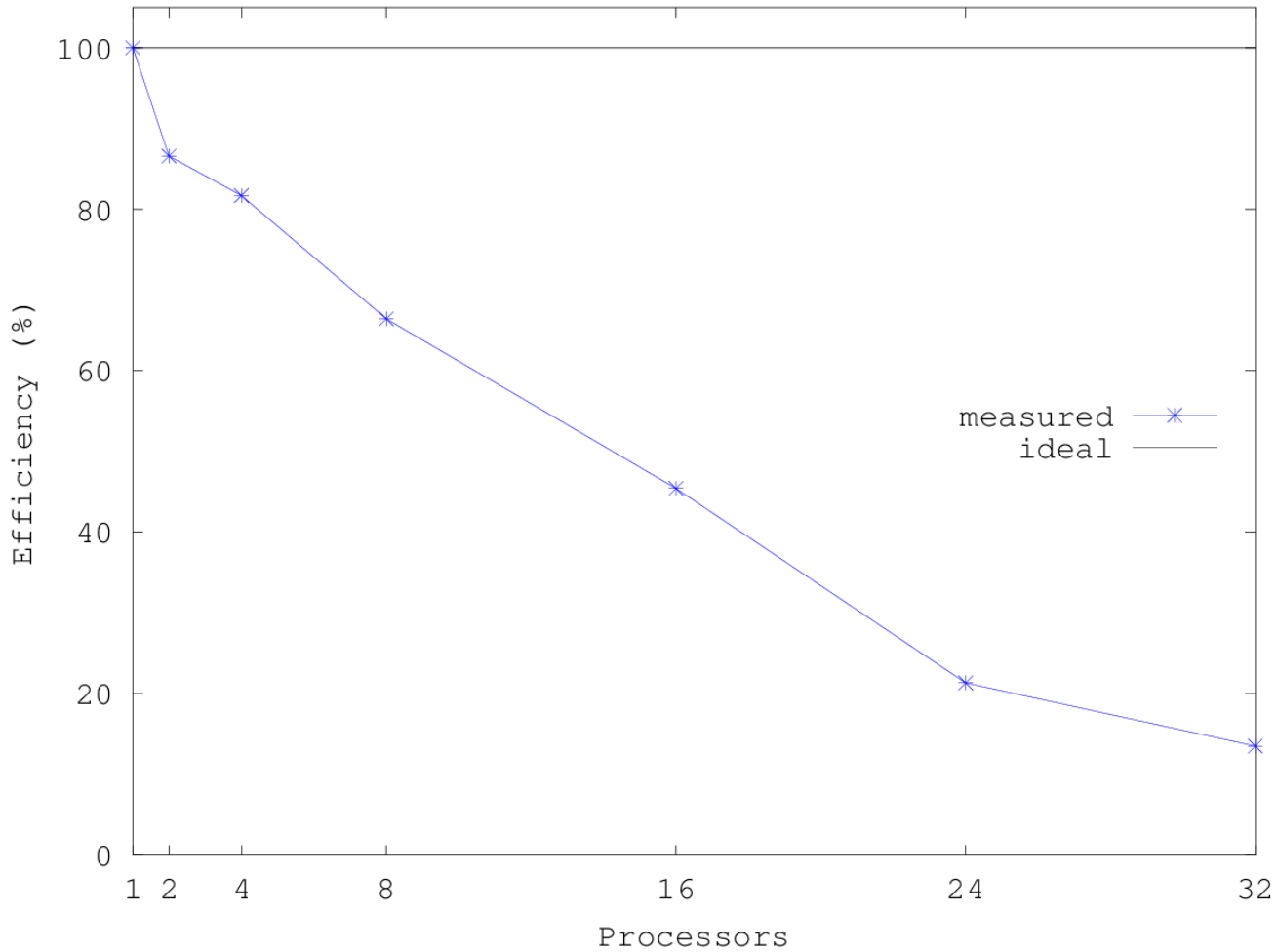
Scaling: Speedup

Example from the tutorial `visual/MolecularDynamics`



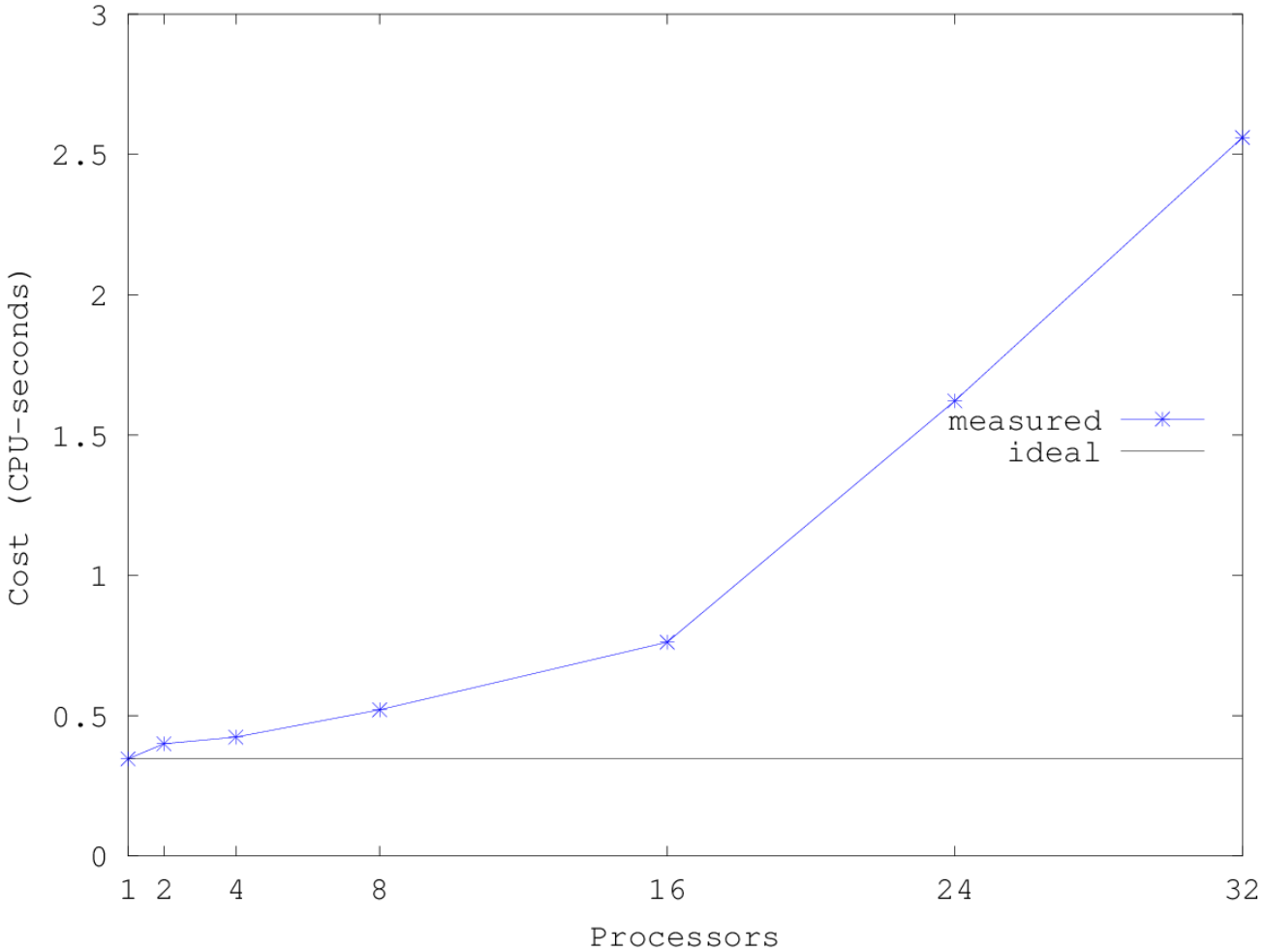
Scaling: Efficiency

Example from the tutorial `visual/MolecularDynamics`

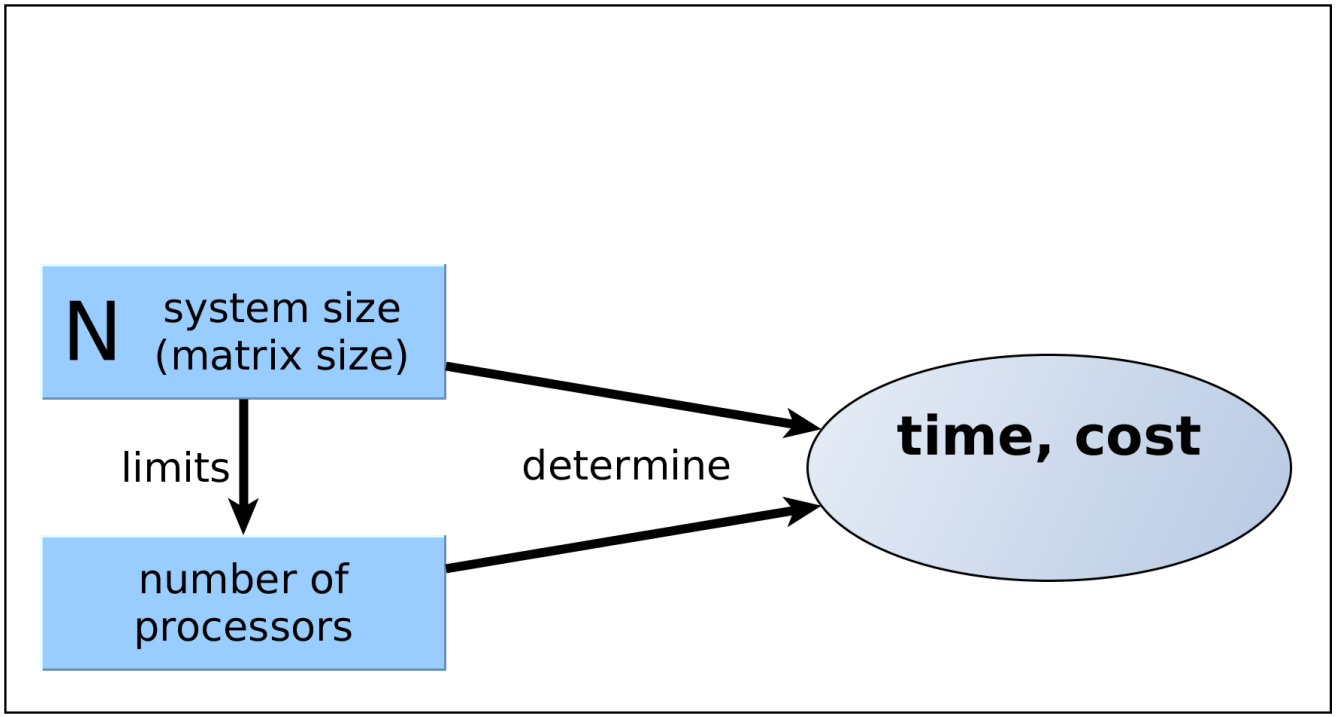


Scaling: Cost

Example from the tutorial `visual/MolecularDynamics`



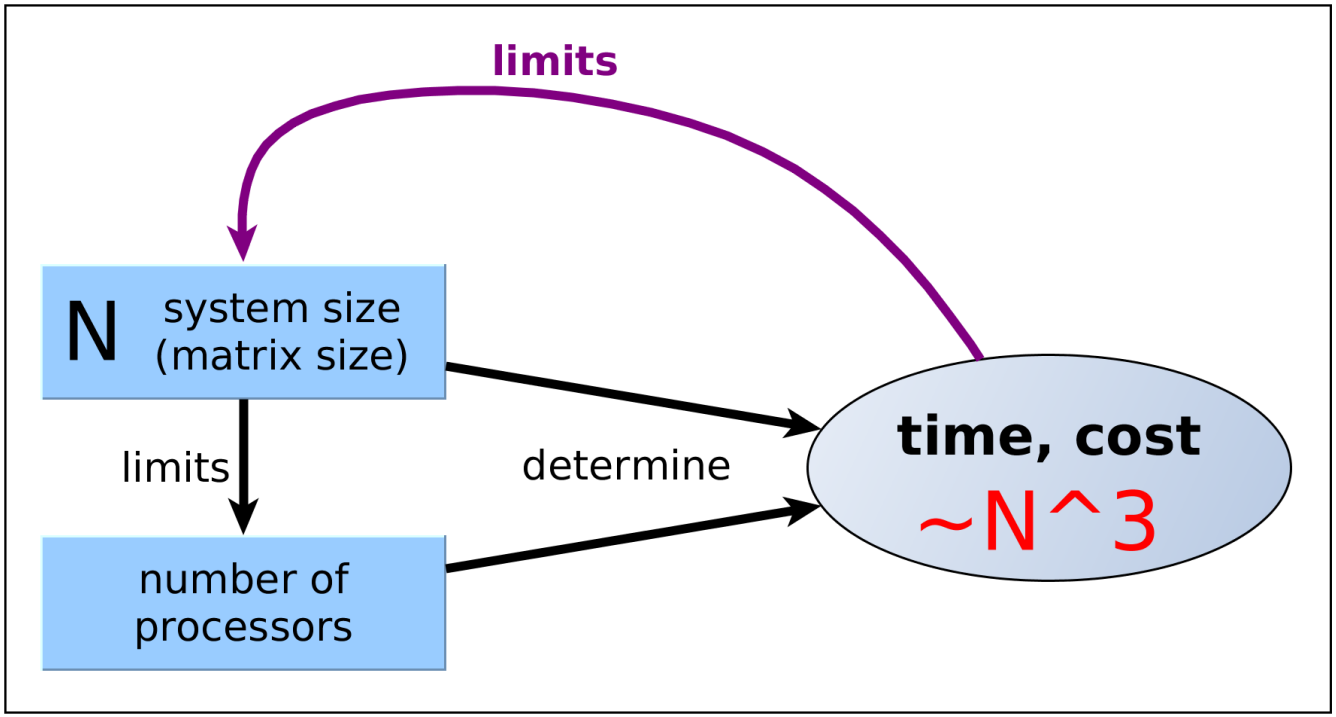
Weak scaling



See also:

`http://departments.icmab.es/leem/siesta/siestimator/siestimator.php`

Weak scaling

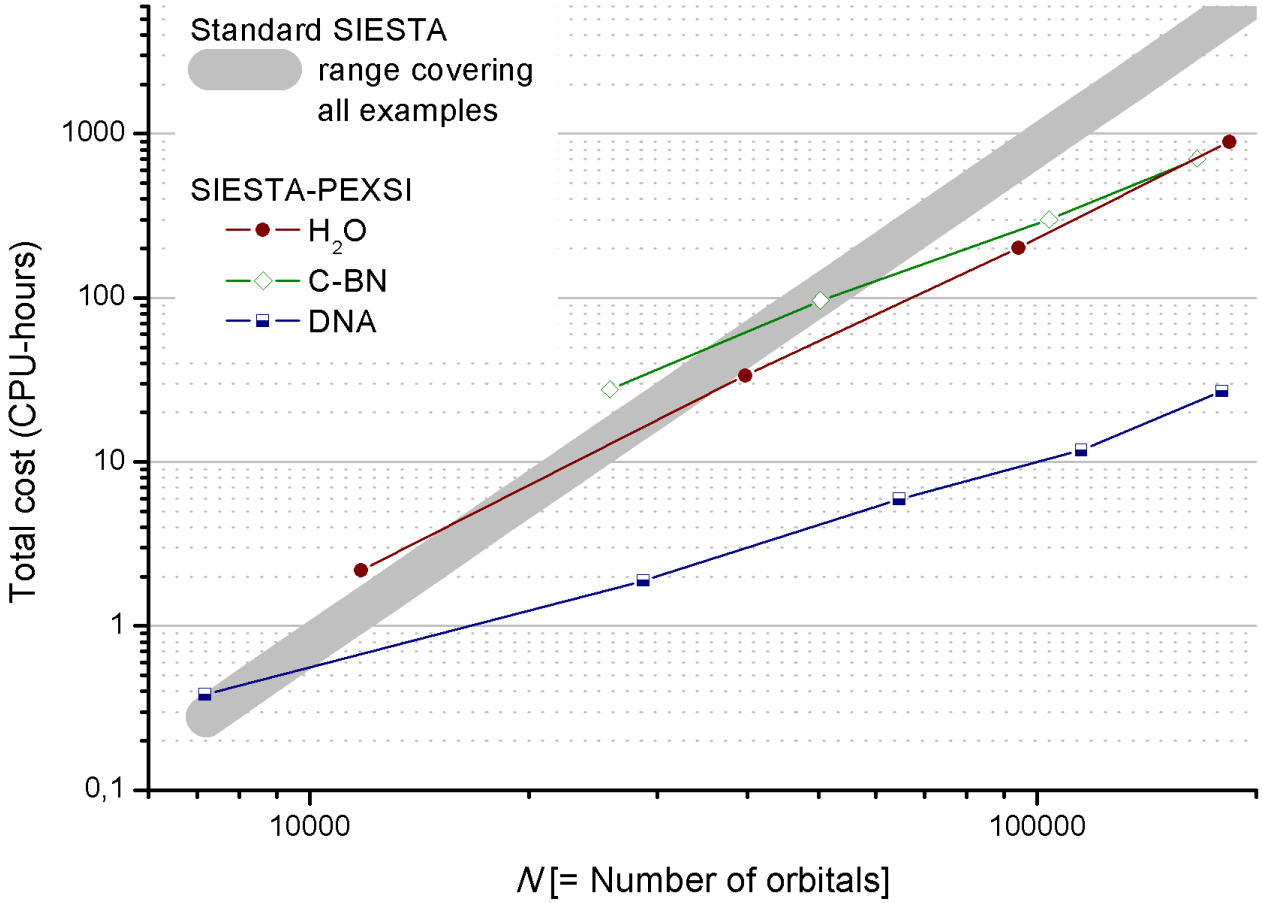


See also:

`http://departments.icmab.es/leem/siesta/siestimator/siestimator.php`

Weak scaling - PEXSI

PEXSI reduces computational complexity



Access to parallel computers

1. Clusters at universities, companies, ...
2. National networks
   in Spain: RES `http://www.res.es/`
3. PRACE (European network)
   `http://www.prace-ri.eu`
   - Preparatory access
   - Project access

RES and PRACE:

- Regular calls
- Significance of project and scalability of software
- Grants certain amount of CPU-hours (free of charge)

Further resources

- This presentation as full-text
- "Access protocol to the equipment of the Barcelona Supercomputing Center and the Spanish Supercomputing Network (RES)"
- PRACE "Guide for Applicants to Tier-0 Resources"


Keep it running!