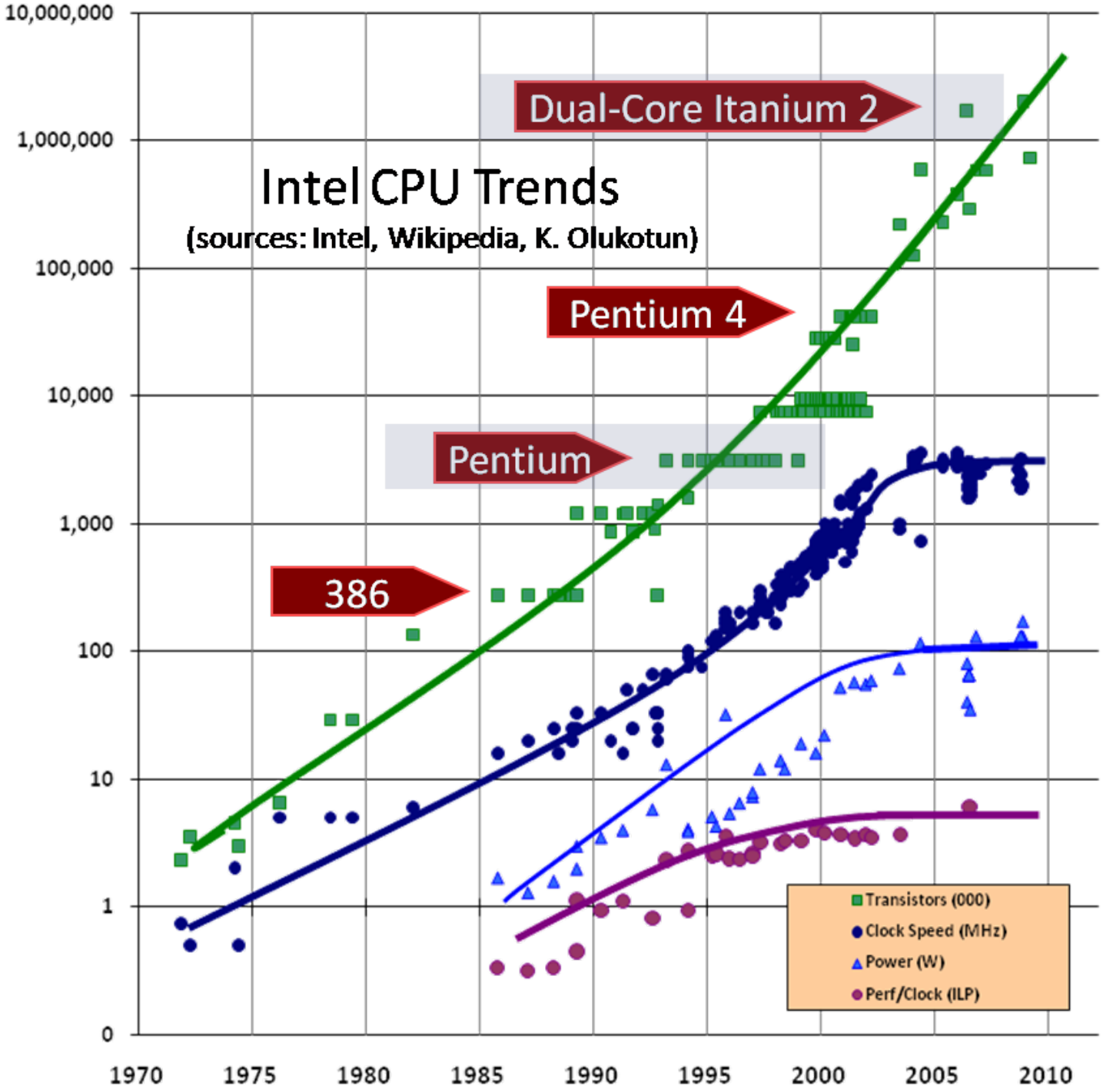
Some advanced issues in SIESTA

Alberto García
Institut de Ciencia de Materials de Barcelona
(ICMAB-CSIC)

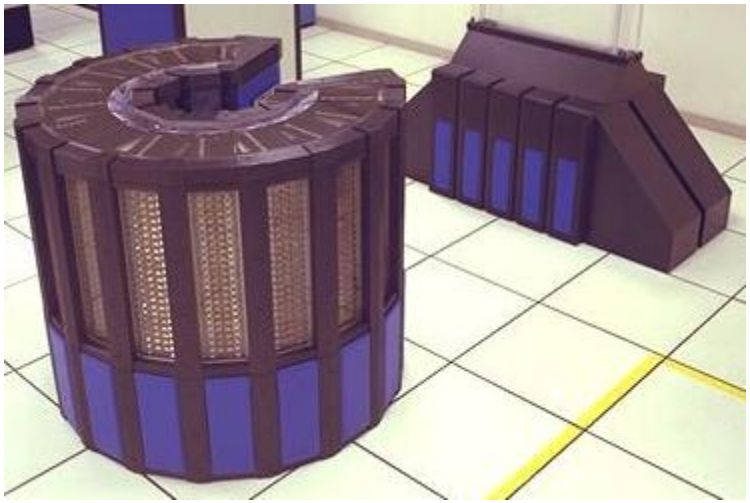
- Parallelization
- New solvers
- Examples of special options and features
- Analysis tools and other utilities
- Special versions of SIESTA

Parallelization

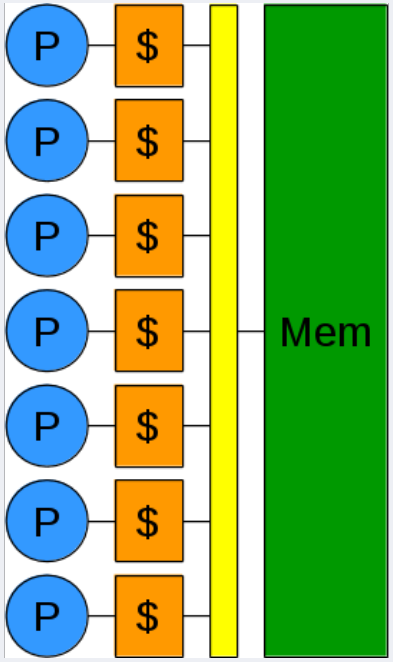
(with thanks to Georg Huhs, BSC)


Intel CPU Trends
(sources: Intel, Wikipedia, K. Olukotun)
Dual-Core Itanium 2
Pentium 4
Pentium
386
Transistors (000)
Clock Speed (MHz)
Power (W)
Perf/Clock (ILP)
10,000,000
1,000,000
100,000
10,000
1,000
100
10
1
0
1970
1975
1980
1985
1990
1995
2000
2005
2010

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

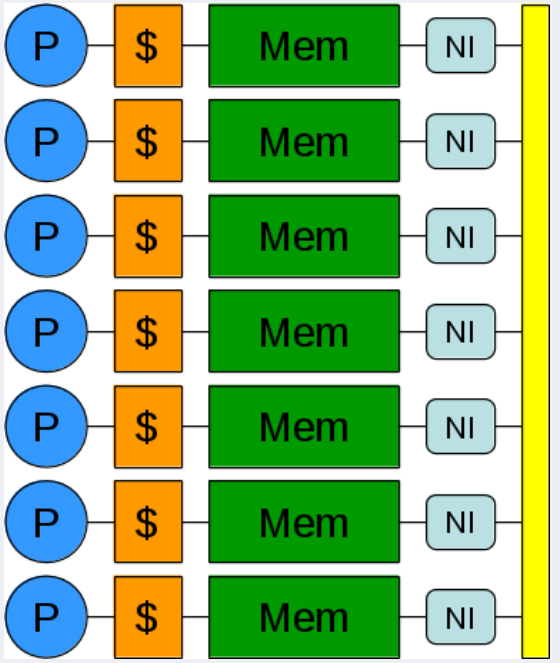


Shared memory



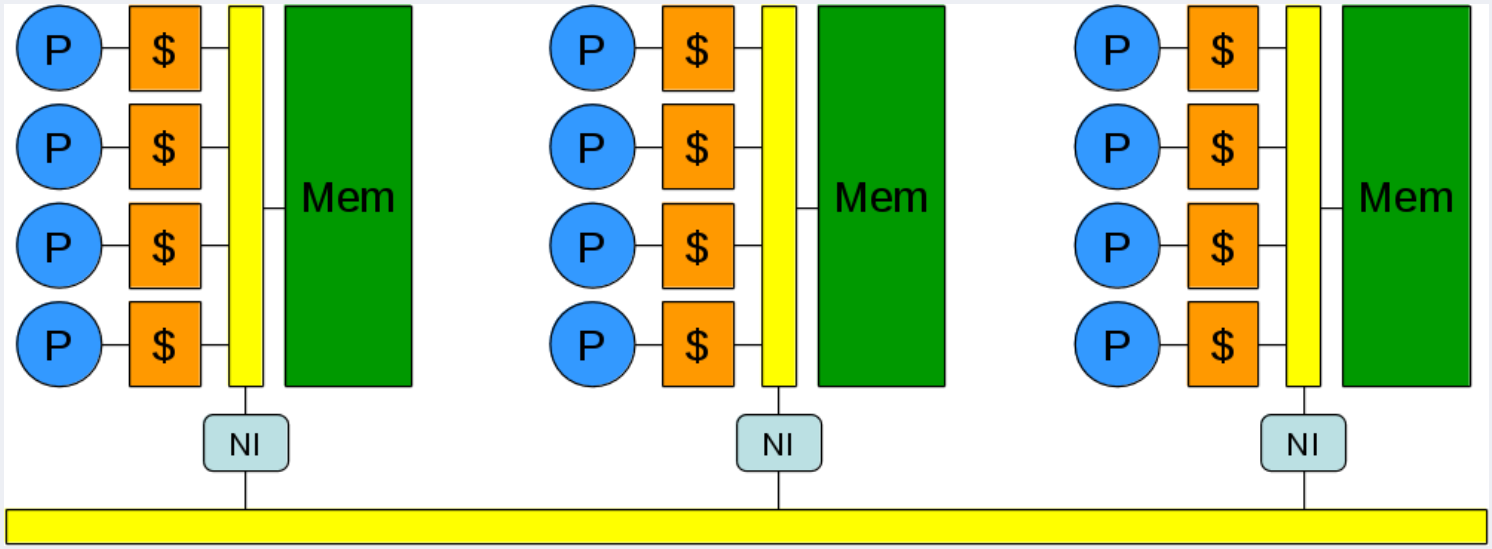
Multithreading

Distributed memory



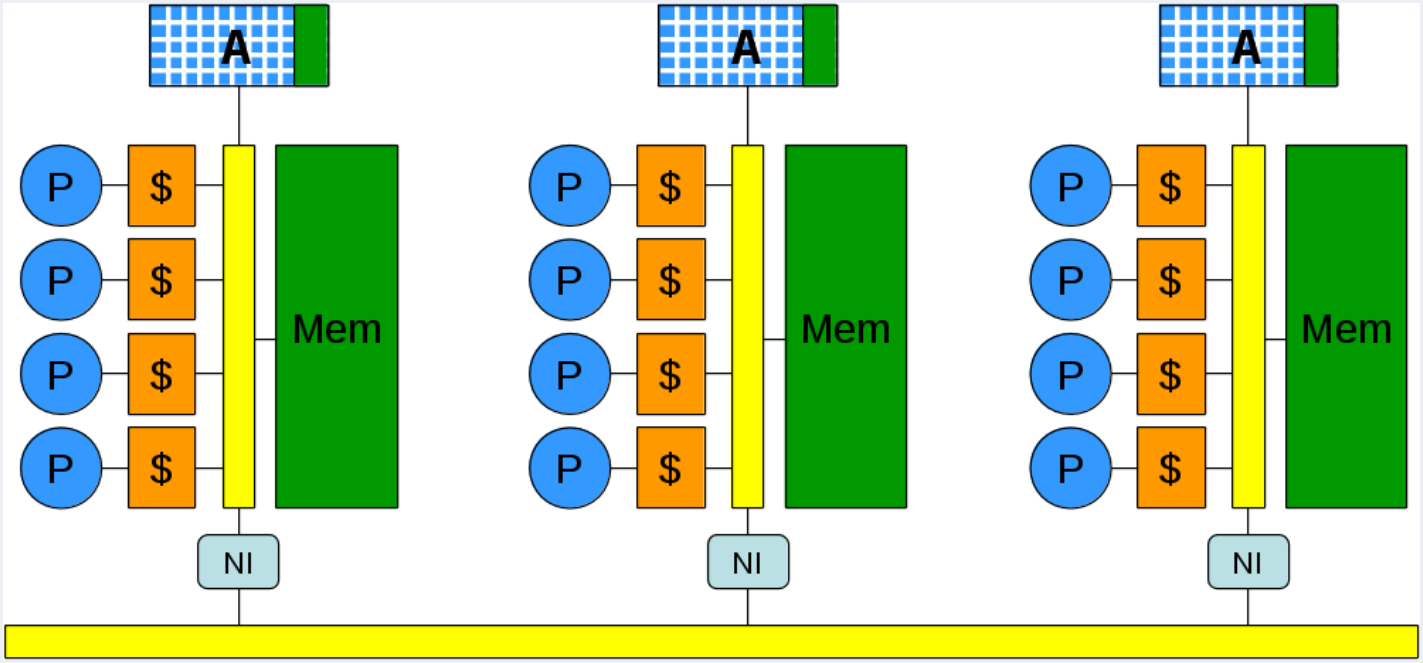
MPI

Hybrid systems



Multithreading + MPI

Heterogeneous systems



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

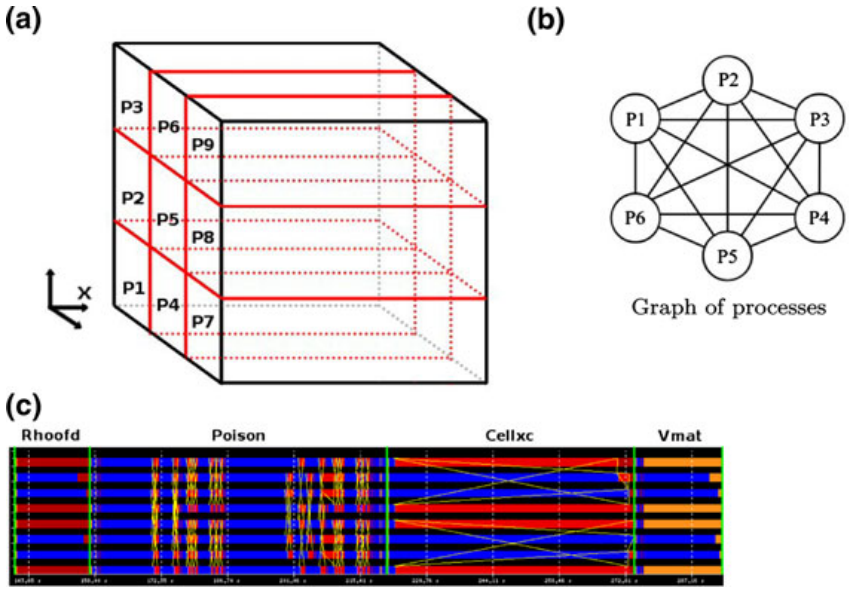
More specialized/optimized code gives better performance
but needs more awareness in its configuration

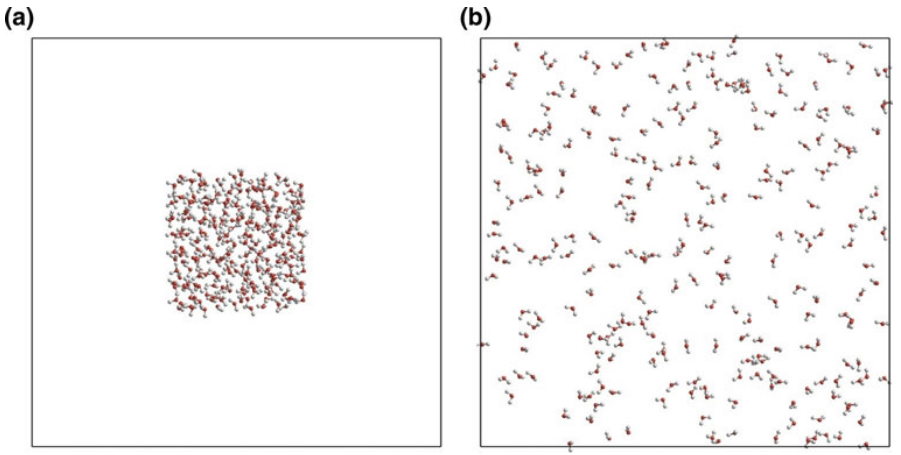
Parallelization in SIESTA

(For now) only distributed memory parallelization.

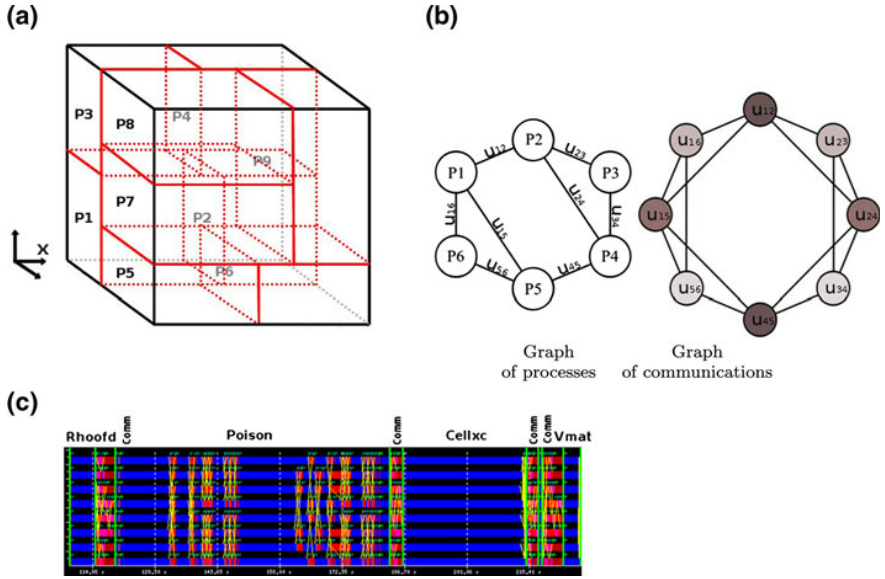
Several options:

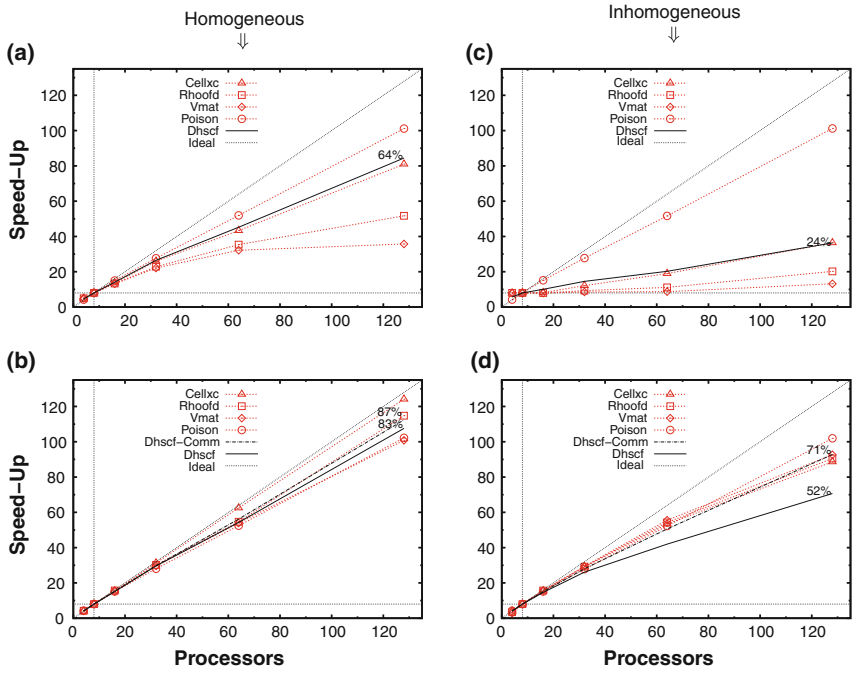
- k-point parallelism
- Distributing orbitals and gridpoints
- 2-level parallelism in PEXSI solver





(Rogeli Grima, J.M. Cela, BSC)





Orbital distribution

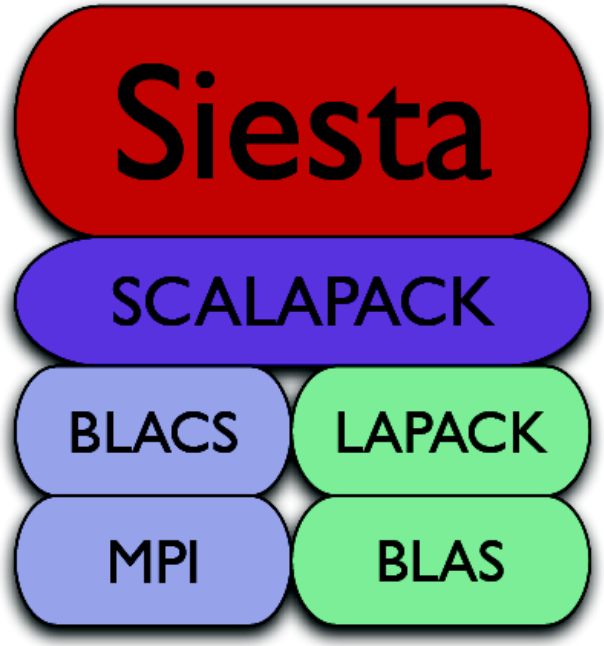
P0

1	1	1	Mg	1	3	0	0	1	F	s	6.620	0	0	0	1
2	2	1	Mg	1	3	0	0	1	F	s	6.620	0	0	0	2
3	3	2	C	1	2	0	0	1	F	s	4.192	0	0	0	3
4	3	2	C	2	2	1	-1	1	F	py	4.870	0	0	0	4
5	3	2	C	3	2	1	0	1	F	pz	4.870	0	0	0	5
6	3	2	C	4	2	1	1	1	F	px	4.870	0	0	0	6
7	4	2	C	1	2	0	0	1	F	s	4.192	0	0	0	7
8	4	2	C	2	2	1	-1	1	F	py	4.870	0	0	0	8
9	4	2	C	3	2	1	0	1	F	pz	4.870	0	0	0	9
10	4	2	C	4	2	1	1	1	F	px	4.870	0	0	0	10

P1

11	5	3	0	1	2	0	0	1	F	s	3.305	0	0	0	11
12	5	3	0	2	2	1	-1	1	F	py	3.937	0	0	0	12
13	5	3	0	3	2	1	0	1	F	pz	3.937	0	0	0	13
14	5	3	0	4	2	1	1	1	F	px	3.937	0	0	0	14
15	6	3	0	1	2	0	0	1	F	s	3.305	0	0	0	15
16	6	3	0	2	2	1	-1	1	F	py	3.937	0	0	0	16
17	6	3	0	3	2	1	0	1	F	pz	3.937	0	0	0	17
18	6	3	0	4	2	1	1	1	F	px	3.937	0	0	0	18
19	7	3	0	1	2	0	0	1	F	s	3.305	0	0	0	19
20	7	3	0	2	2	1	-1	1	F	py	3.937	0	0	0	20

Software stack:



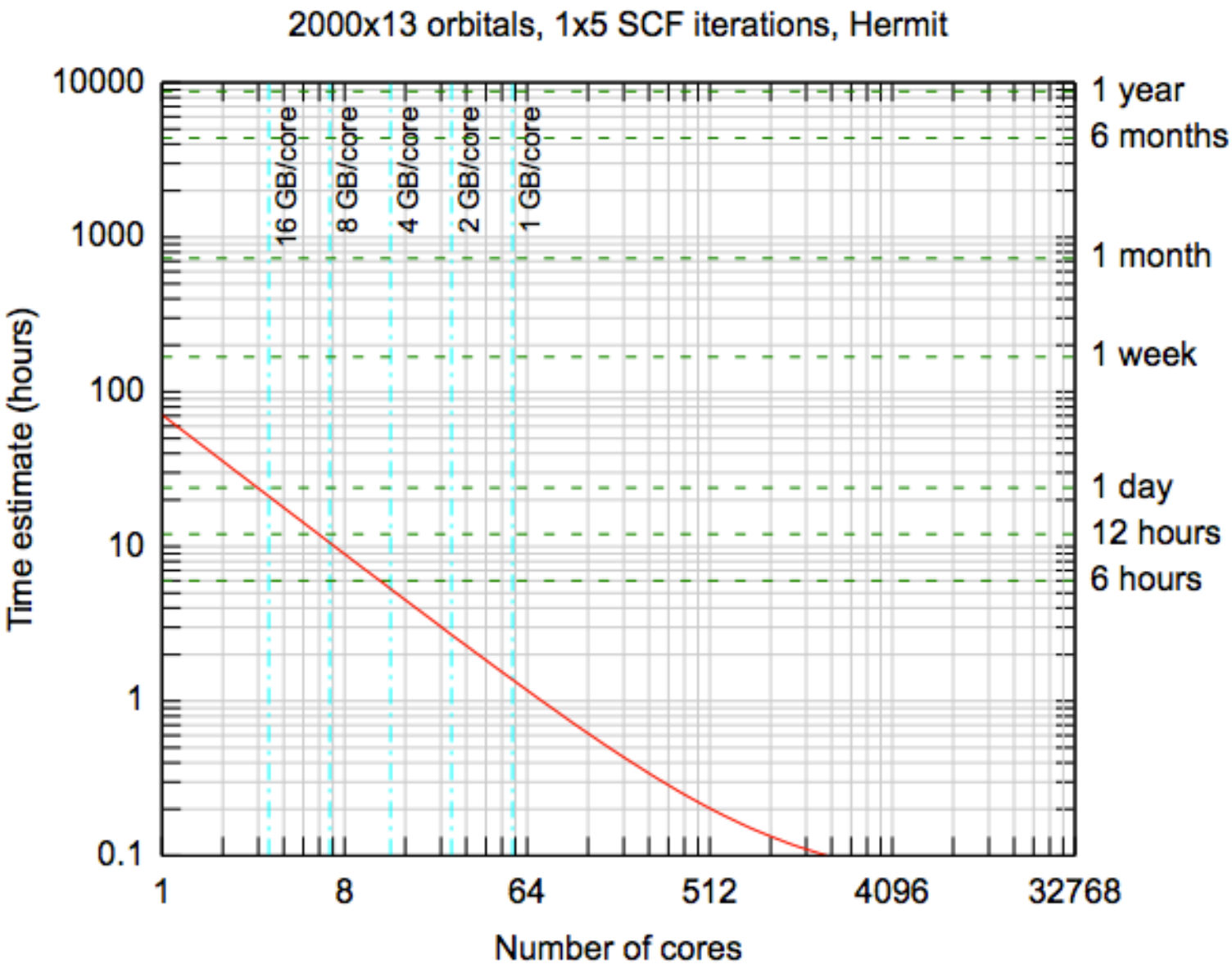
Compile options:

```
## 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=.
```


2000x13 orbitals, 1x5 SCF iterations, Hermit
Time estimate (hours)
10000
1000
100
10
1
0.1
16 GB/core
8 GB/core
4 GB/core
2 GB/core
1 GB/core
1 year
6 months
1 month
1 week
1 day
12 hours
6 hours
1
8
64
512
4096
32768
Number of cores

Massive parallelization: The PEXSI solver

Diagonalization

$H|\psi_i >= \epsilon_i S|\psi_i >$ ScaLaPack $O(N^3)$

$$|\psi_i >= \sum_\mu c_i^\mu |\phi_\mu >$$

$$\hat{\rho} = \sum_i f_i |\psi_i >< \psi_i| = \sum_{i\mu\nu} f_i c_i^\mu c_i^\nu |\phi_\mu >< \phi_\nu|$$

$\rho_{\mu\nu} = \sum_i f_i c_i^\mu c_i^\nu$ Density matrix

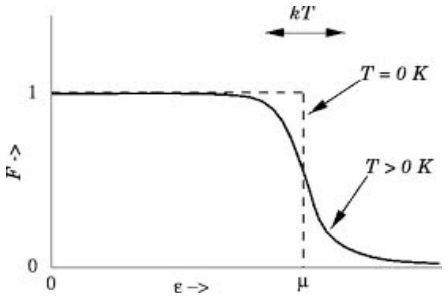
$$N = \mathrm{Tr}[\hat{\rho}S] \qquad E_{BS} = \mathrm{Tr}[\hat{\rho}H]$$

Formal solution to the electronic-structure problem

$$\hat{\rho} = f_\beta(\hat{H} - \mu)$$

$$f_\beta(\epsilon_i - \mu) = \frac{2}{1 + \mathrm{e}^{\beta(\epsilon_i - \mu)}}$$

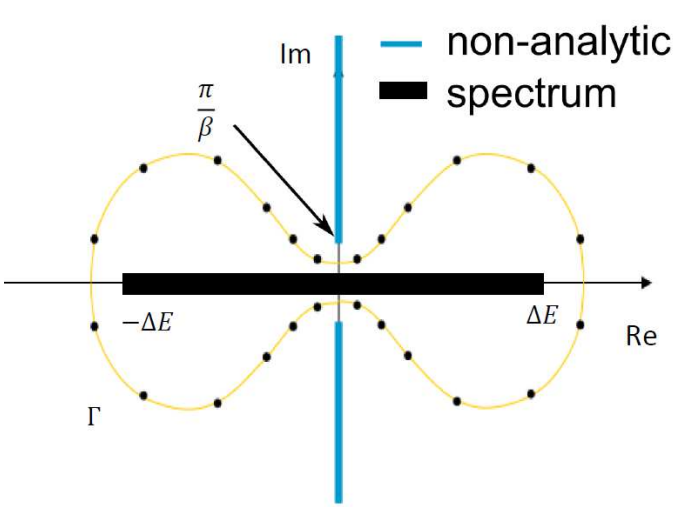
Fermi-Dirac function



We just need a computationally convenient representation of the Fermi-Dirac function

Pole (rational) expansion of the Fermi function

$$f_\beta(\epsilon_i - \mu) = \frac{2}{1 + \mathrm{e}^{\beta(\epsilon_i - \mu)}}$$



$$f_\beta(\epsilon_i - \mu) \approx Im \sum_{l=1}^{P} \frac{\omega_l}{\epsilon_i - (z_l + \mu)}$$

$$\hat{\rho} = \quad Im \left(\sum_{l=1}^{P} \frac{\omega_l}{H - (z_l + \mu)S} \right)$$

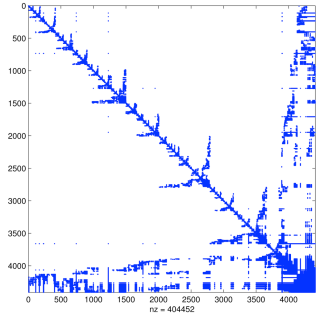
Fewer terms are needed (typically 40 poles are enough)

Pole Expansion plus Selected Inversion

(Lin Lin, Chao Yang, LBNL)

$$\hat{\rho} = Im\left(\sum_{l=1}^{P} \frac{\omega_l}{H-(z_l+\mu)S}\right)$$

One inversion per pole



Only a limited number of elements is needed in the density matrix!!

Use of a very efficient Selected Inversion algorithm

$$A = \begin{pmatrix} a & b^T \\ b & \hat{A} \end{pmatrix}$$

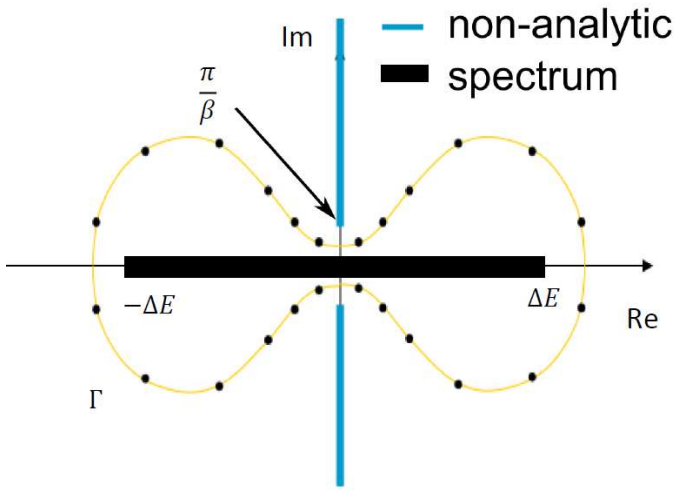
$$(A =)LDL^T = \begin{pmatrix} 1 & \\ l & \hat{L} \end{pmatrix} \begin{pmatrix} \alpha & \\ & \hat{A}-bb^T/\alpha \end{pmatrix} \begin{pmatrix} 1 & l^T \\ & \hat{L} \end{pmatrix}$$

There are no DM locality approximations, as in O(N) methods

For sufficiently big problems

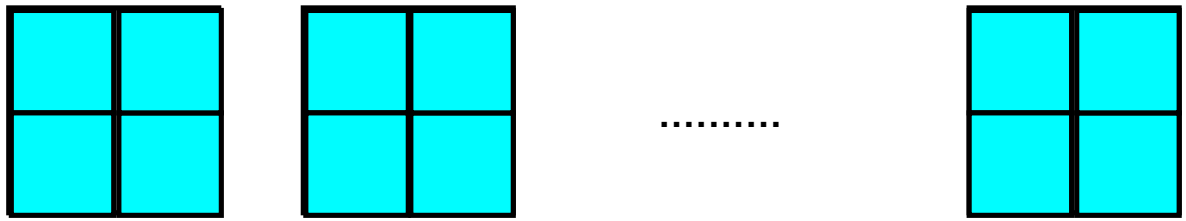
$$\begin{aligned} \text{(quasi-)1D:} &\quad \mathcal{O}(N) \\ \text{(quasi-)2D:} &\quad \mathcal{O}(N^{3/2}) \\ \text{3D:} &\quad \mathcal{O}(N^2) \end{aligned}$$

(Due to sparsity of the target density matrix)



The method is applicable to metals using low effective temperatures

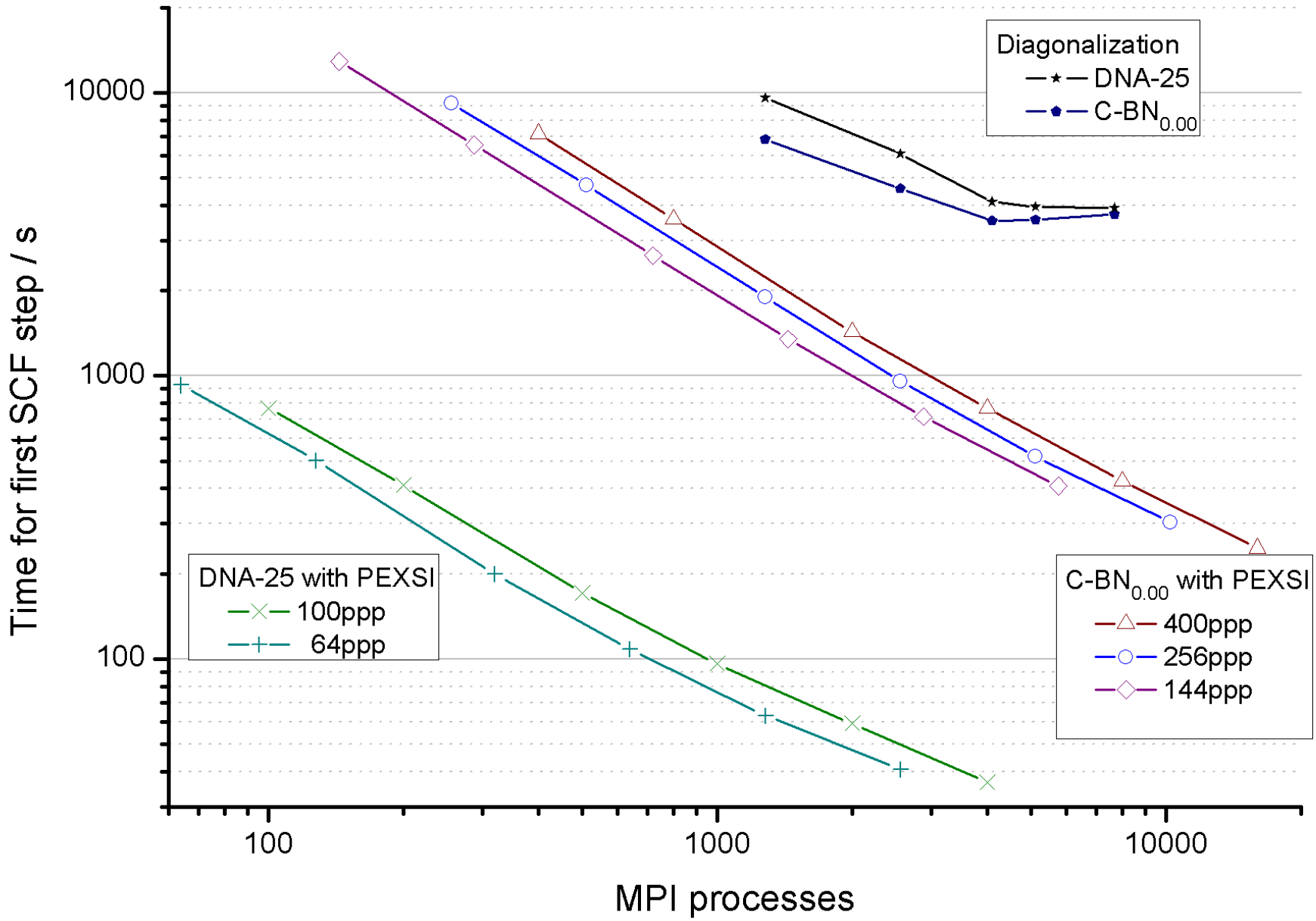
Trivially parallel over poles, with perfect load balancing



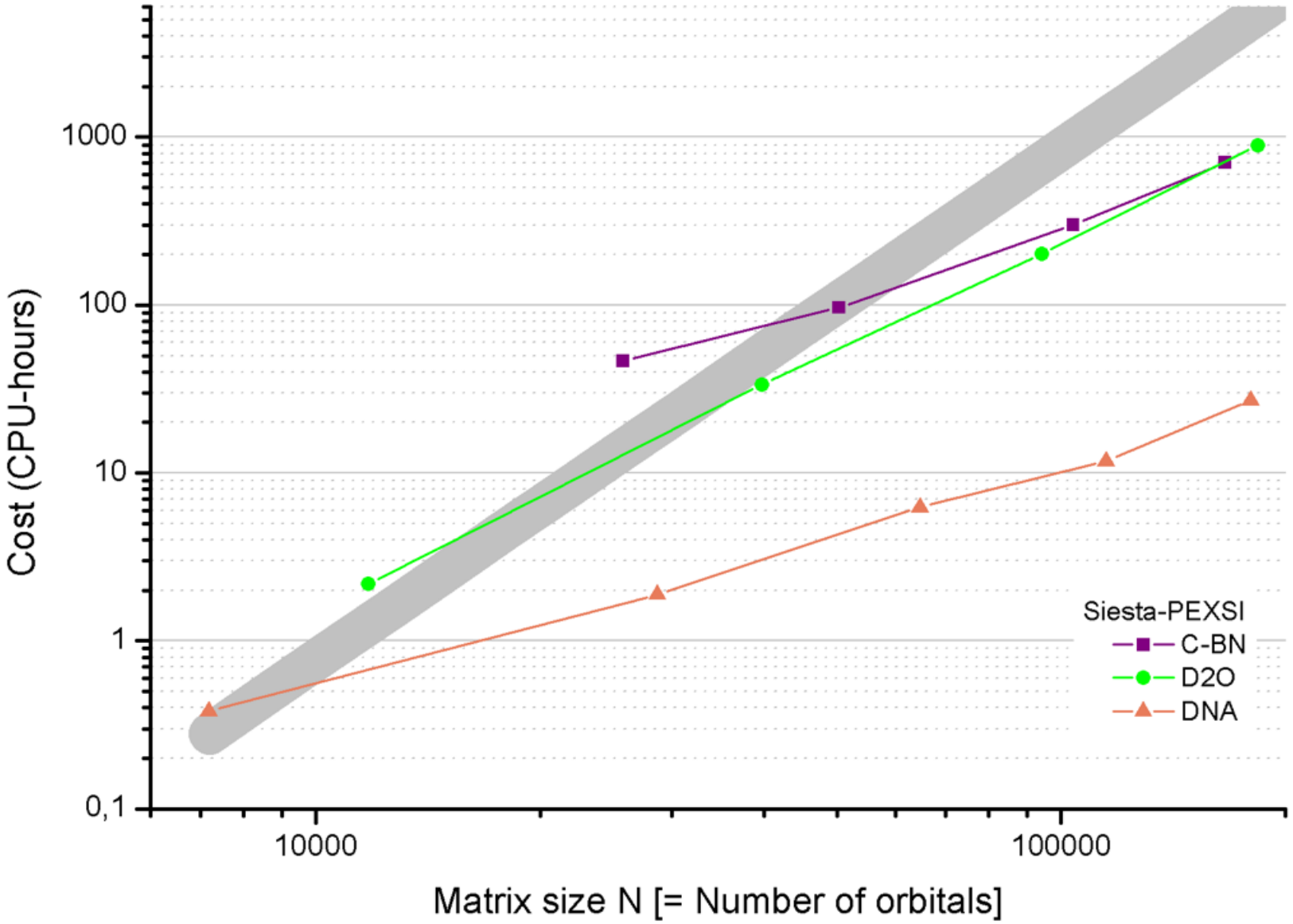
4 processors per pole x 40 poles :: 160 processors

The number of processors per pole is a parameter of the calculation.
Lower limit: as needed to fit the problem in memory.

Strong scaling



Weak scaling



The PEXSI solver will be included soon in the Siesta distribution

SIESTA-PEXSI: Massively parallel method for efficient and accurate ab initio materials simulation without matrix diagonalization

Lin Lin, Alberto García, Georg Huhs, Chao Yang
J. Phys.: Condens. Matter **26** 305503 (2014)
doi:10.1088/0953-8984/26/30/305503

arXiv:1405.0194 [physics.comp-ph]

Examples of special options and features

New mixing options:

Hamiltonian

Charge density in Fourier space

Re-starting options:

Density matrix

Charge density

Replica calculations:

MPI-based dispatch

External scripting

(Download "trunk" (development) version)

Analysis tools and utilities

- Basis optimization (Util/Optimizer)
- Population analysis: Mulliken, Voronoi, Hirshfeld, Bader (see Manual)
- COOP-COHP analysis (Util/COOP)

Special SIESTA versions

LDA+U

Distributed on the SIESTA web page.
Synchronization to the latest version in progress.

Spin-orbit coupling

Implementation based on work by Jaime Ferrer's group (Univ. of Oviedo, Spain).
Distributed by request.

See the manual for a complete list of features

Monitor the SIESTA web page for updates

We welcome feedback

THANK YOU !