MHPC
Master in High Performance Computing

PARALLEL THINKING

SISSA
Scuola Internazionale Superiore
di Studi Avanzati

ICTP
The Abdus Salam
International Centre
for Theoretical Physics

Measure of Parallelism

- there are two main reasons to write a parallel program using the message passing paradigm approach (distributed memory):
  - access to larger amount of memory (aggregated)
  - reduce time to solution
- evaluation of the performance improvement is not always straightforward, specially considering at the increasing the problem complexity

How do we evaluate the improvement?

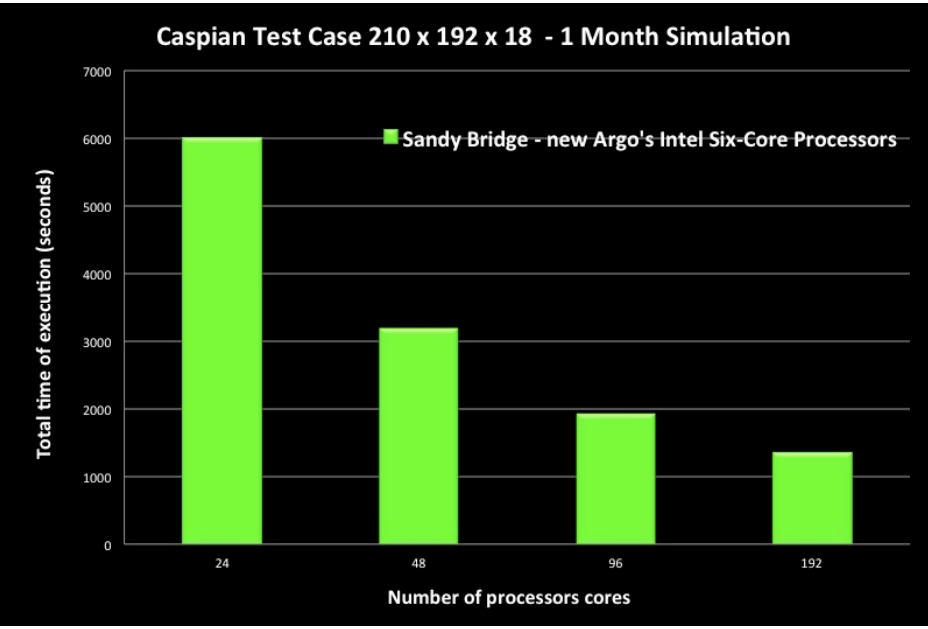
To quantify the improvement we use the term **Speedup**:

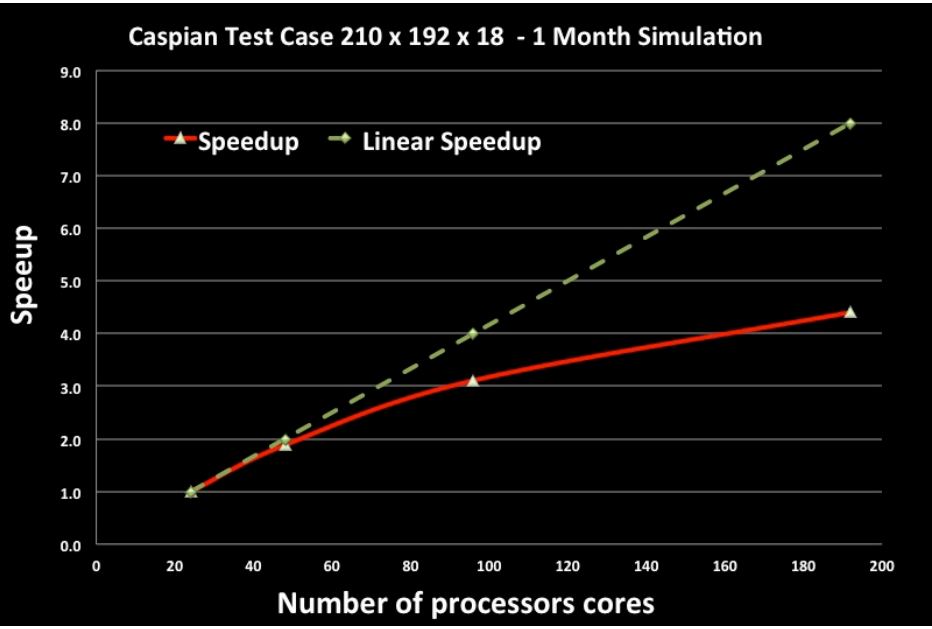
$$S_P = \frac{T_S}{T_P}$$

- In the ideal case Tp = Ts/npe.
- When Sp > npe it is named super-linear Speedup
- Speedup provides a measure of the strong scalability

MHPC
Master in High Performance Computing

Speedup





ma per seguir virtute e canoscenza
SISSA
Scuola Internazionale Superiore
di Studi Avanzati

The Abdus Salam
ICTP International Centre
for Theoretical Physics

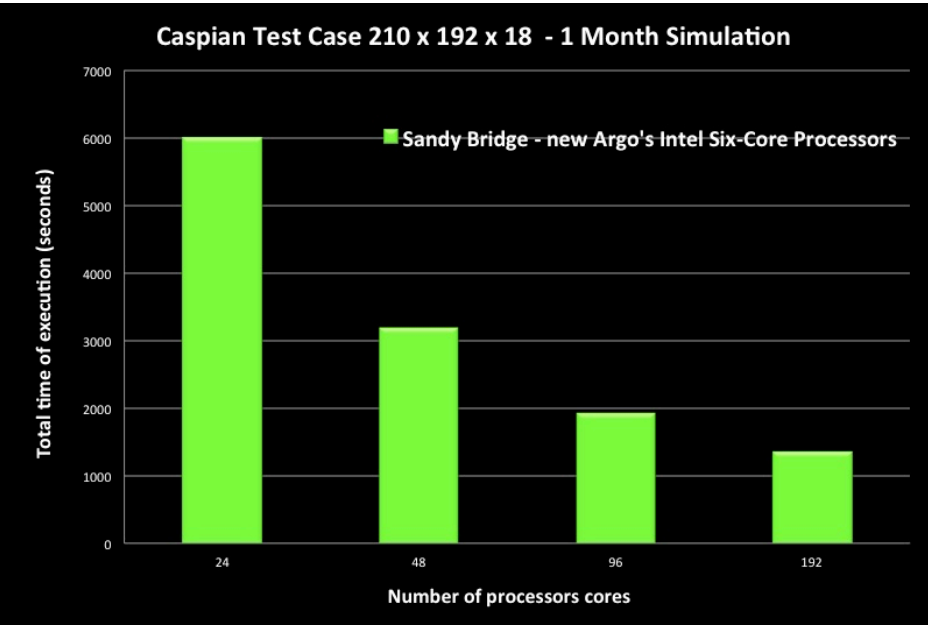
Efficiency

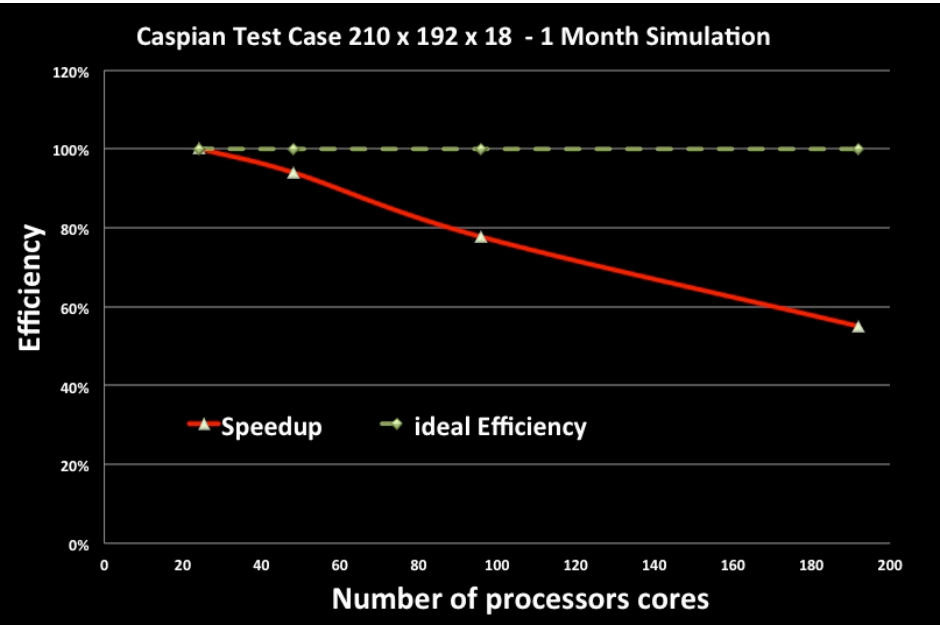
- Only embarrassing parallel algorithms can obtain and maintain an ideal Speedup with increasing concurrency
- The **Efficiency** is a measure of the fraction of time for which a processing element is usefully employed:

$$E_P = \frac{S_P}{p}$$

MHPC
Master in High Performance Computing

Efficiency





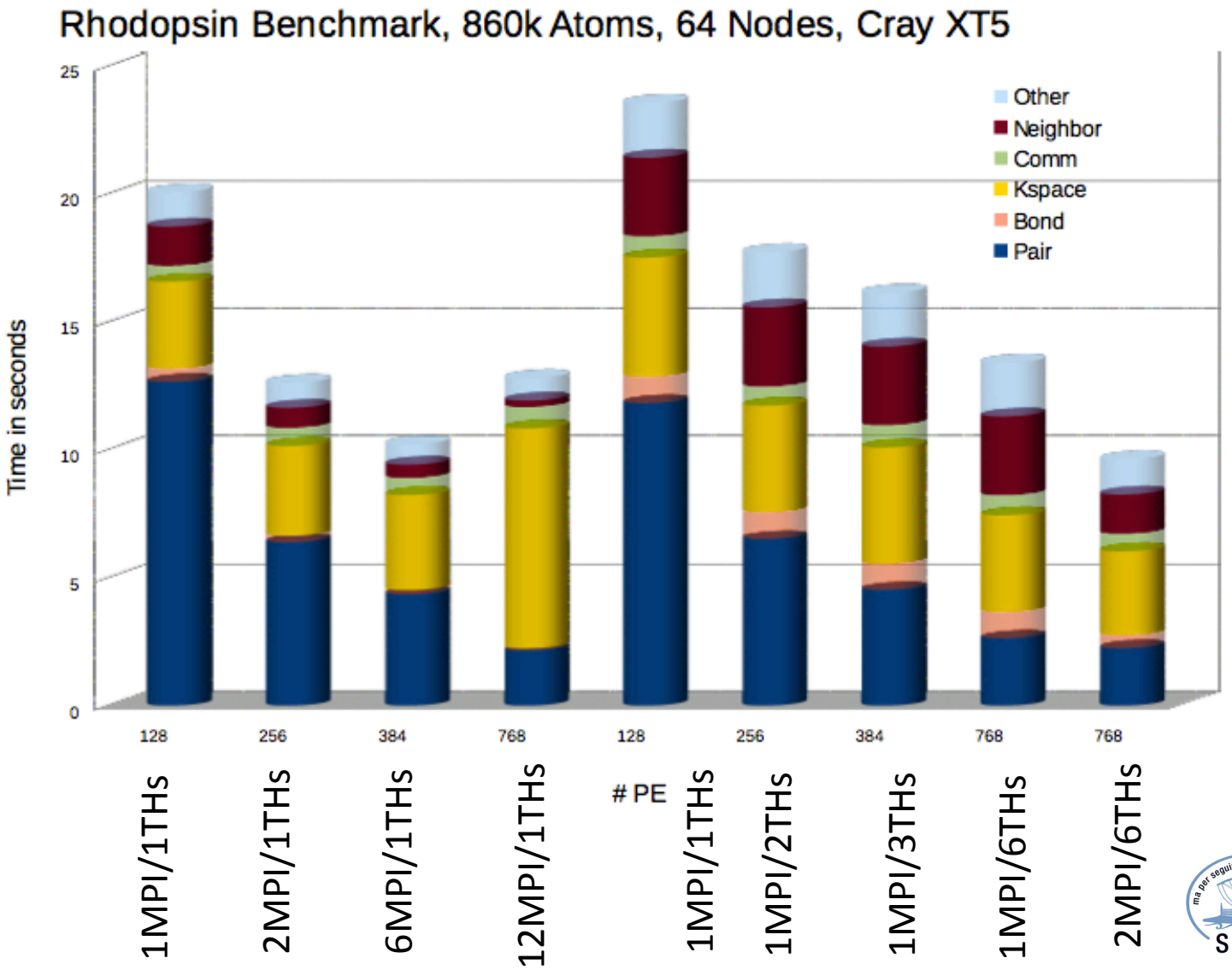
SISSA
ma per seguir virtute e canoscenza
Scuola Internazionale Superiore
di Studi Avanzati

ICTP
The Abdus Salam
International Centre
for Theoretical Physics

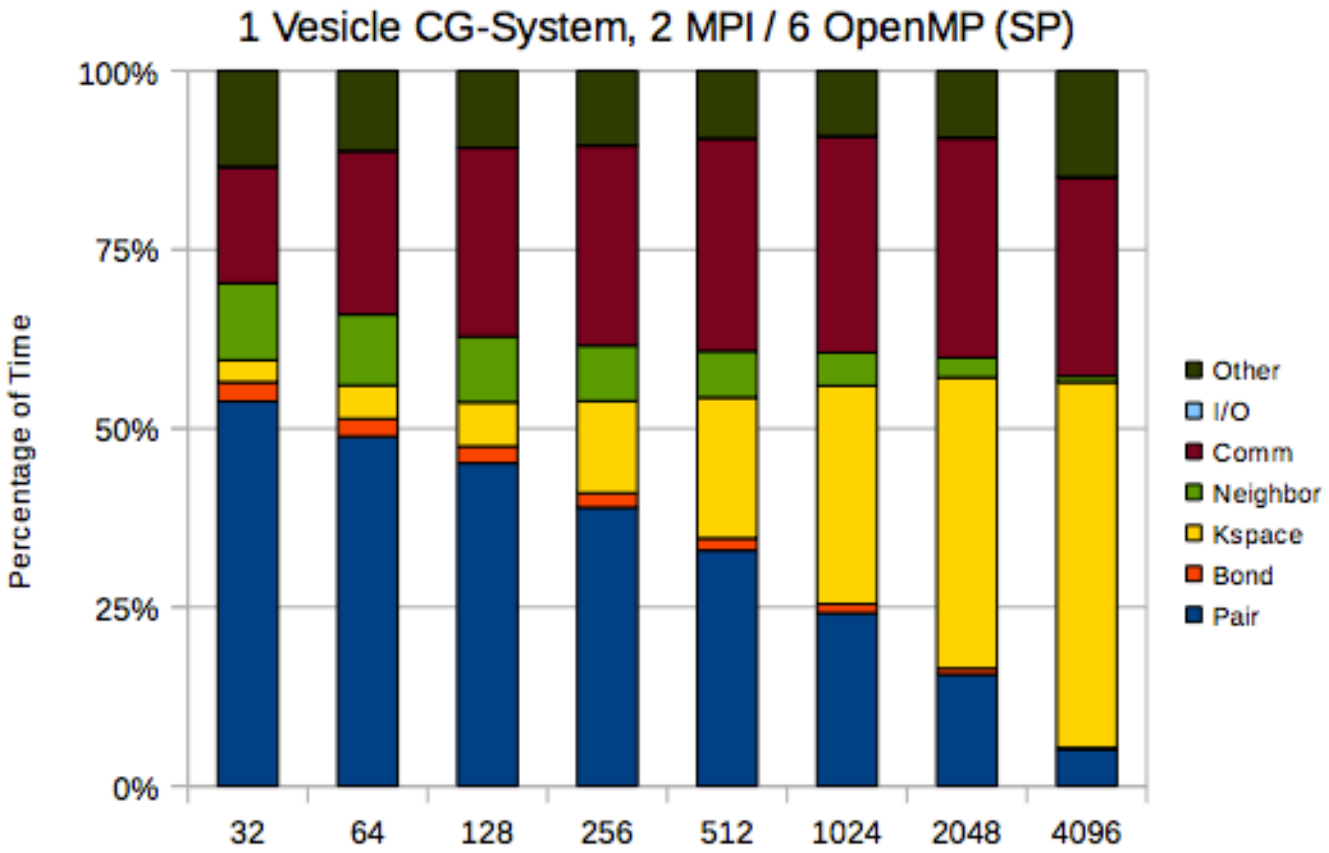
MHPC
Master in High Performance Computing

Scalability Limits

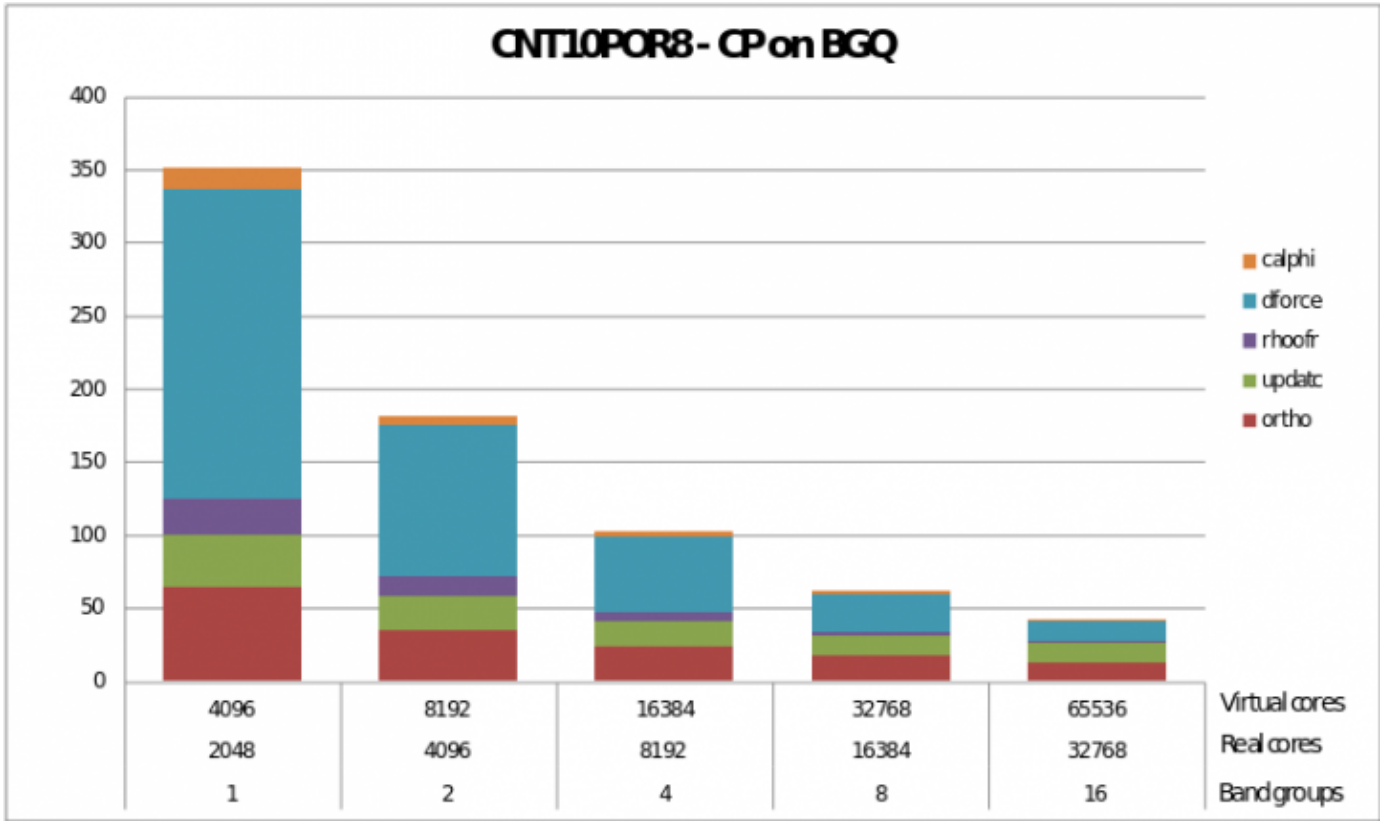
- Amdahl's law (there is a serial part which is not effected)
- Parallelism introduces overhead due to communication, synchronization, additional operations (I/O, memory copies, reordering, etc...)
- Sometimes the limit of scalability can be somehow predicted:
  - is there enough work for any process?


Rhodopsin Benchmark, 860k Atoms, 64 Nodes, Cray XT5
Time in seconds
0
5
10
15
20
25
Other
Neighbor
Comm
Kspace
Bond
Pair
128
256
384
768
128
256
384
768
768
PE
1MPI/1THs
2MPI/1THs
6MPI/1THs
12MPI/1THs
1MPI/1THs
1MPI/2THs
1MPI/3THs
1MPI/6THs
2MPI/6THs

Amdal's Law And Real Life



Scaling - QE-CP on Fermi BGQ @ CINECA



MHPC
Master in High Performance Computing

Timing

Time measurement becomes a relevant task:

MPI_WTIME()

Clock

Seconds()

```
 convergence NOT achieved after   5 iterations: stopping

 Writing output data file c8_atm213_k111.save

 init_run     :     93.79s CPU     93.79s WALL (       1 calls)
 electrons    :    961.37s CPU    961.37s WALL (       1 calls)

 Called by init_run:
 wfcinit      :     69.37s CPU     69.37s WALL (       1 calls)
 potinit      :      4.76s CPU      4.76s WALL (       1 calls)

 Called by electrons:
 c_bands      :    883.32s CPU    883.32s WALL (       5 calls)
 sum_band     :     40.30s CPU     40.30s WALL (       5 calls)
 v_of_rho     :      1.10s CPU      1.10s WALL (       6 calls)
 mix_rho      :      1.51s CPU      1.51s WALL (       5 calls)

 Called by c_bands:
 init_us_2    :      0.50s CPU      0.50s WALL (      11 calls)
 cegterg      :    882.01s CPU    882.01s WALL (       5 calls)

 Called by *egterg:
 h_psi        :    259.11s CPU    259.11s WALL (      17 calls)
 g_psi        :      9.02s CPU      9.02s WALL (      11 calls)
 cdiaghg      :    401.37s CPU    401.37s WALL (      16 calls)

 Called by h_psi:
 add_vuspsi   :     22.44s CPU     22.44s WALL (      17 calls)

 General routines
 calbec       :     17.25s CPU     17.25s WALL (      17 calls)
 fft          :      0.52s CPU      0.52s WALL (      66 calls)
 ffts         :      0.63s CPU      0.63s WALL (     117 calls)
 fftw         :    231.61s CPU    231.61s WALL (   10260 calls)
 davcio       :      4.72s CPU      4.72s WALL (       5 calls)

 Parallel routines
 fft_scatter  :     63.50s CPU     63.51s WALL (   10443 calls)
 ALLTOALL     :     10.66s CPU     10.67s WALL (   10252 calls)
 EXX routines

 PWSCF        : 17m42.94s CPU    17m42.94s WALL
```

```
 convergence NOT achieved after   5 iterations: stopping

 Writing output data file c8_atm213_k111.save

 init_run     :    119.48s CPU    119.48s WALL (       1 calls)
 electrons    :   1369.53s CPU   1369.53s WALL (       1 calls)

 Called by init_run:
 wfcinit      :     98.55s CPU     98.55s WALL (       1 calls)
 potinit      :      2.15s CPU      2.15s WALL (       1 calls)

 Called by electrons:
 c_bands      :   1289.41s CPU   1289.41s WALL (       5 calls)
 sum_band     :     56.06s CPU     56.06s WALL (       5 calls)
 v_of_rho     :      1.39s CPU      1.39s WALL (       6 calls)
 mix_rho      :      1.23s CPU      1.23s WALL (       5 calls)

 Called by c_bands:
 init_us_2    :      0.13s CPU      0.13s WALL (      11 calls)
 cegterg      :   1288.89s CPU   1288.89s WALL (       5 calls)

 Called by *egterg:
 h_psi        :    409.59s CPU    409.59s WALL (      17 calls)
 g_psi        :      2.35s CPU      2.35s WALL (      11 calls)
 cdiaghg      :    528.61s CPU    528.61s WALL (      16 calls)

 Called by h_psi:
 add_vuspsi   :     32.96s CPU     32.96s WALL (      17 calls)

 General routines
 calbec       :     31.22s CPU     31.22s WALL (      17 calls)
 fft          :      0.62s CPU      0.62s WALL (      66 calls)
 ffts         :      0.86s CPU      0.86s WALL (     117 calls)
 fftw         :    376.02s CPU    376.04s WALL (   82004 calls)
 davcio       :      6.38s CPU      6.38s WALL (       5 calls)

 Parallel routines
 fft_scatter  :     81.64s CPU     81.65s WALL (   82187 calls)

 PWSCF        : 24m57.48s CPU    24m57.48s WALL


 This run was terminated on:  12:25:36  12Oct2012
```

Profiling

Profiling usually means:

- Instrumentation of code (e.g. during compilation)
- Automated collection of timing data during execution
- Analysis of collected data, breakdown by function

Example: gcc -o some_exe.x -pg some_code.c

- ./some_exe.x
- gprof some_exe.x gmon.out

Profiling is often incompatible with code optimization or can be misleading (inlining)

MHPC
Master in High Performance Computing

A Non-Trivial Parallel Algorithm

- Identify portions of the work that can be performed concurrently
- Mapping the concurrent pieces of work onto multiple processes running in parallel
- Distributing the input, output and intermediate data associated within the program
- Managing accesses to data shared by multiple processors
- Synchronizing the processors at various stages of the parallel program execution

Granularity

- Granularity is determined by the decomposition level (number of tasks) on which the problem is divided (grain)
- In general, if the granularity of a decomposition is finer, the associated overhead (as a ratio of useful work associated with a task) increases.
- The degree to which task/data can be subdivided is limit to concurrency and parallel execution
- Parallelization has to become “topology aware”
  - coarse grain and fine grained parallelization has to be mapped to the topology to reduce memory and I/O contention

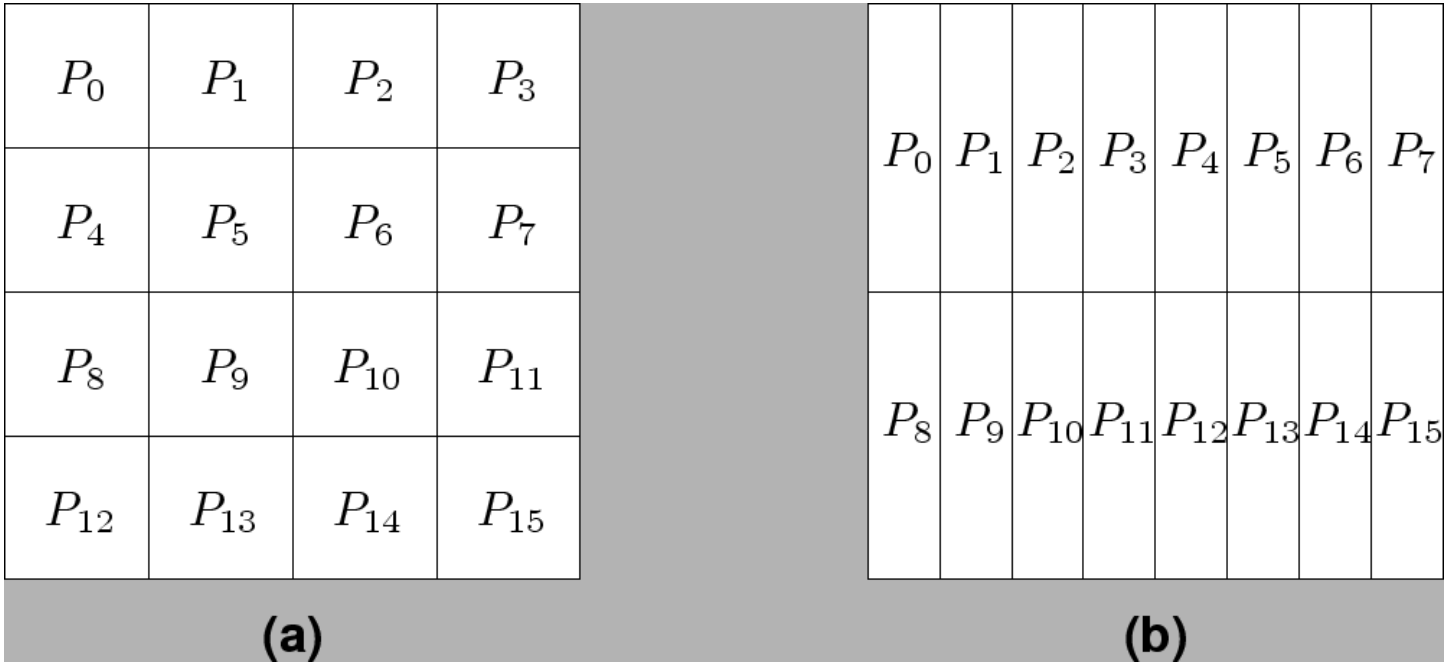
Process Mapping

Dynamic Mapping: processes workload is defined at runtime and changes during program execution. Usually it requires a relevant implementation effort but it might deliver better load-balance in case of dynamic models and unbalanced problem at large

Static mapping: mostly used in scientific computing where a fixed domain is equally distributed among processes

Block Array Distribution Schemes

Block distribution schemes can be generalized to higher dimensions as well.

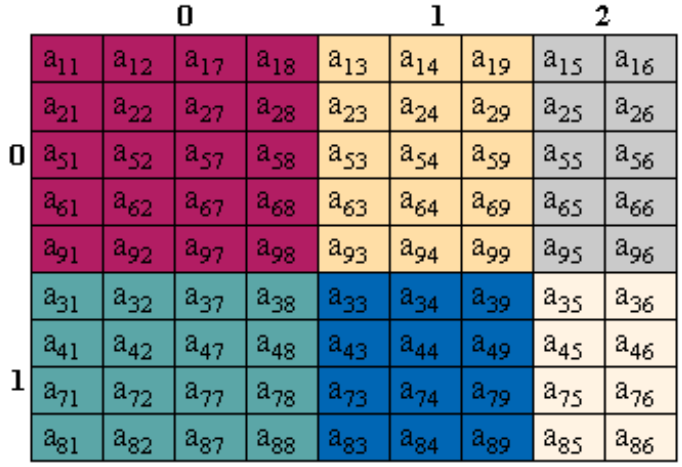


Degree to which tasks/data can be subdivided is limit to concurrency and parallel execution!!

Master in High Performance Computing

block cyclic distribution

Global View



Local (distributed) View

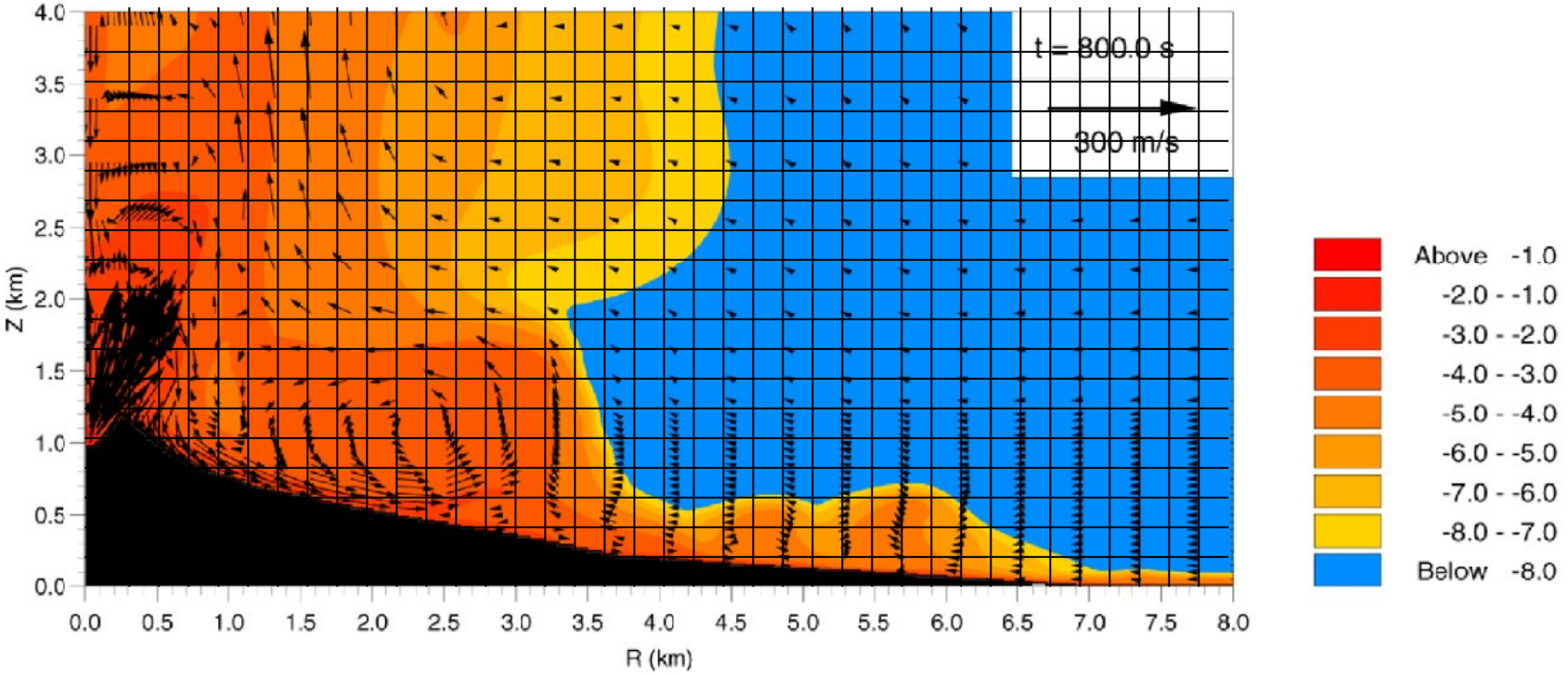
SISSA
Scuola Internazionale Superiore di Studi Avanzati

ICTP The Abdus Salam International Centre for Theoretical Physics

MHPC
Master in High Performance Computing

Parallelization Strategies

In a simulation the space and time are discretized, and the transport equations are solved numerically on a discrete grid

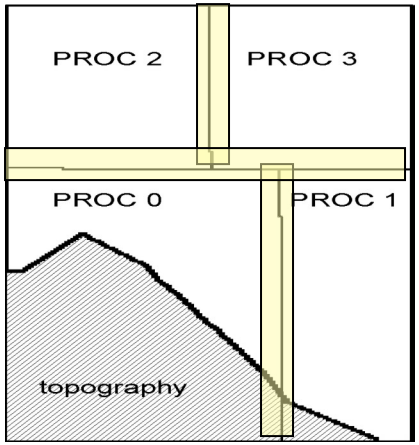


Computational Grid (Nx * Nz), topography, gas and particle fields

SISSA ICTP

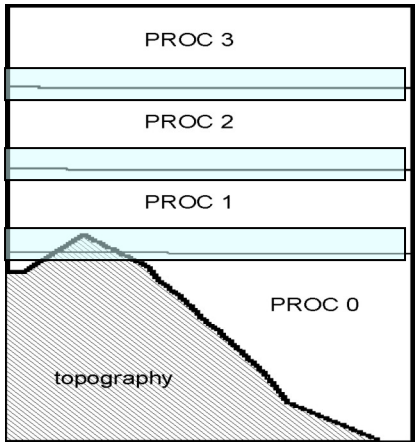
Parallelization Strategies

The main strategy is to define sub grids and distribute them to the available processors. Intermediate results on local data need to be exchanged across domain boundaries the solve the equations defined in the global domain.



Block like distribution

less overall communication



layer like distribution,

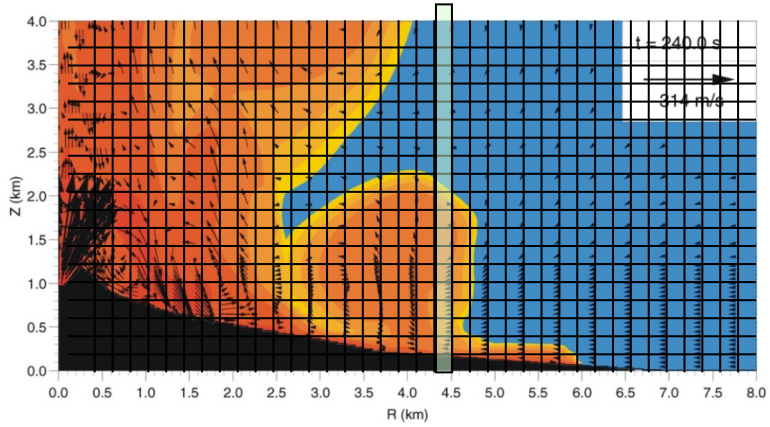
comunication only with nearest neighbours

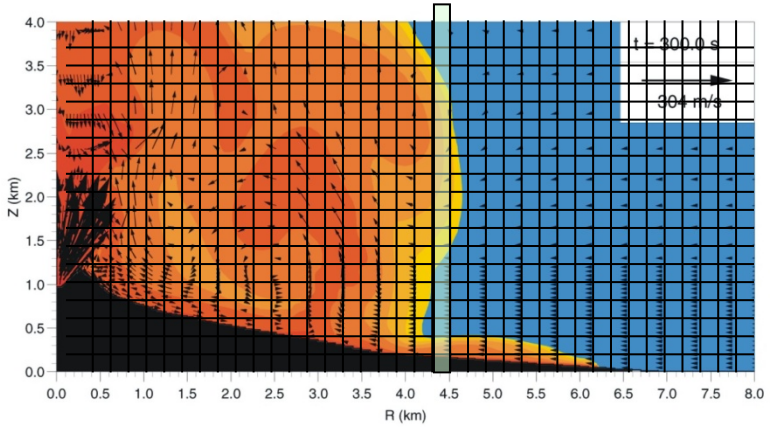
MHPC
Master in High Performance Computing

Time evolution

$$s(t+\Delta t) = F(s(t), s(t-\Delta t), ..)$$

System status at time $t+\Delta t$ is a Function of the System at previous time values t, $t-\Delta t$, ecc…
Usually we need only information on system from t.

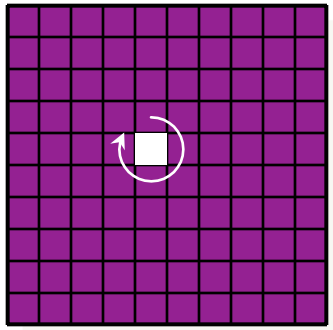




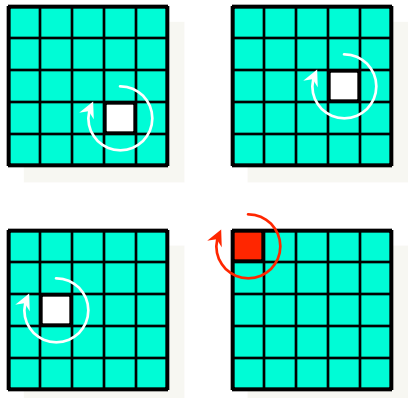
ma per seguir virtute e canoscenza
SISSA
ICTP

Ghost Cells

- In many algorithms used to solve partial differential equations, the vaule of a given field in a given point/ cell is updated using the values of the neighbouring cells.



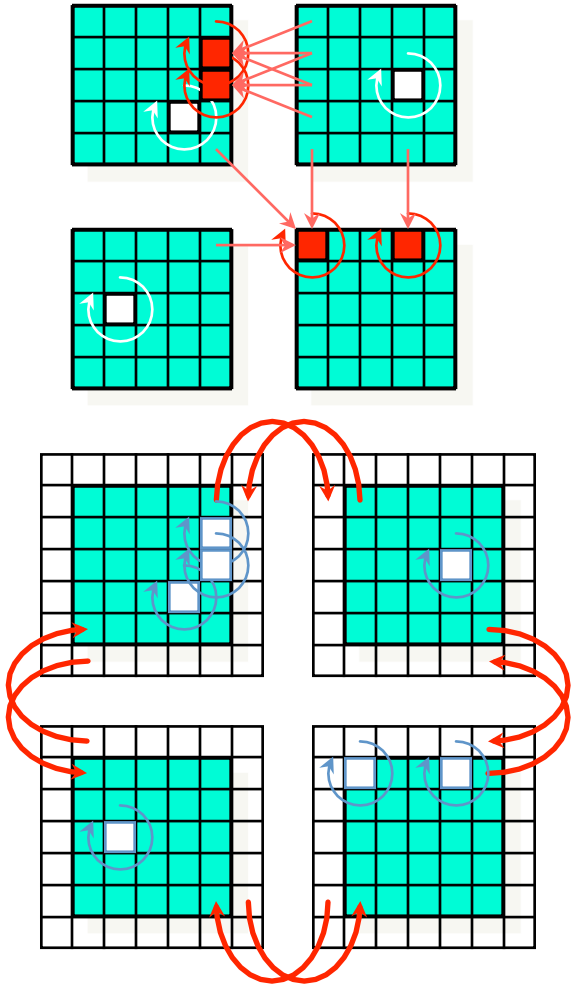
- Distributing the field to different PE implies that some point lays close to a boundary between two task.



- The boundary elements require values stored in the memory of another task
- At first approach a developer could be tended to perform a communication for every element!!! => No!!!

Ghost Cells /2

- To solve the problem, each PE extend its domain to contain a copy of the value of its neighbouring PEs
- The “ghost cells” are updated with a single communication before the updating cycle
- The locality of the computation is restored.

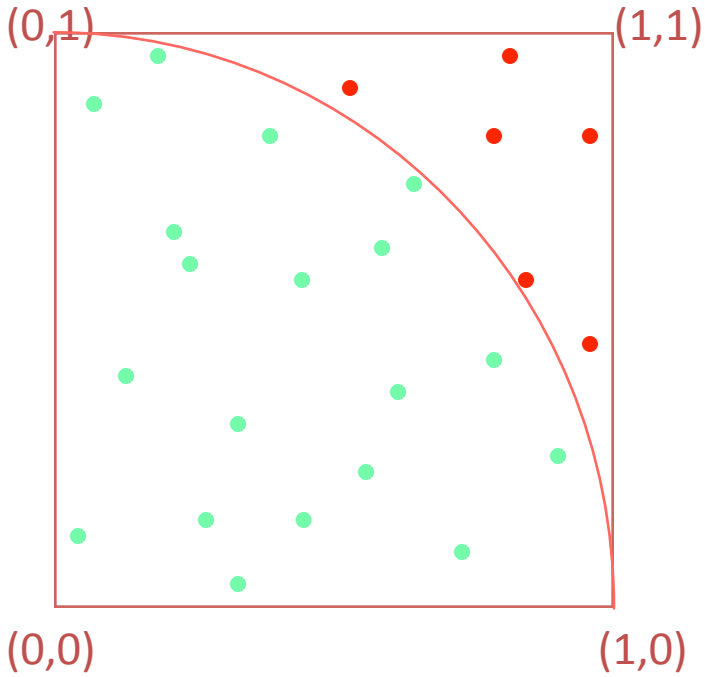


MHPC
Master in High Performance Computing

Monte-Carlo Integrals

- Compute π measuring probability of finding a random number [0..1]×[0..1] within a distance 1 from the origin
- The accurancy is proportional to the number of points
- Each point is independent from the others.
- Synchronization only at the end

Beware! Random sequence on different pe!!



- **Different sequence on different processor**
- **The sequences should be non overlapping**
- **The sequances should be non-correlated**
- **SPRNG library (sprng.cs.fsu.edu)**

SISSA
ma per seguir virtute e canoscenza
Scuola Internazionale Superiore di Studi Avanzati

ICTP The Abdus Salam International Centre for Theoretical Physics