Calcul parallèle et simulation moléculaire d'adsorption dans les matériaux carbonés

"Parallel computing and adsorption simulation through the carbon-based materials"

H. Ramézani[a,b] , D. Nguemalieu Kouetcha[b] and N. Mathieu-Cohaut[b]

[a] École Polytechnique de l'Université d'Orléans, Université d'Orléans, 8 rue Léonrad de Vinci, 45072 Orléans, France

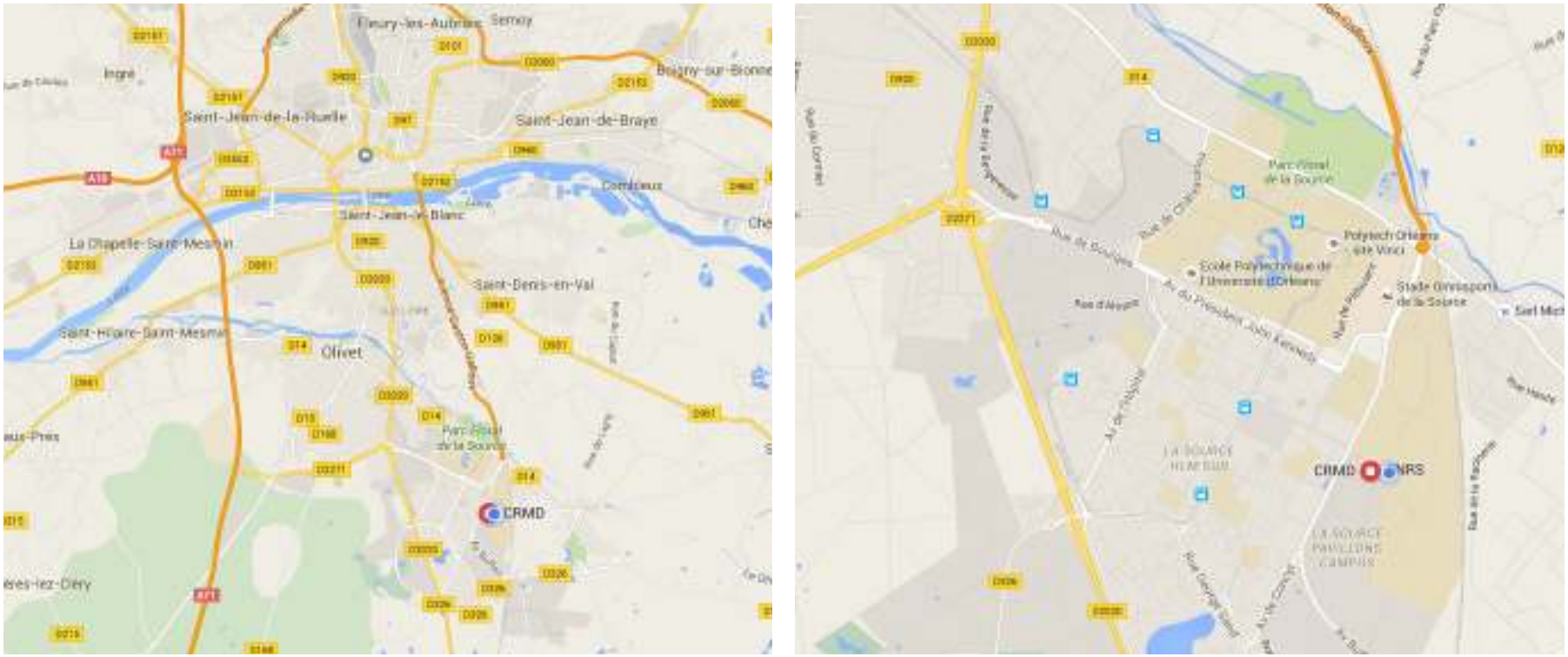
[b] Interfaces, Confinement, Matériaux et Nanostructures, ICMN, UMR CNRS 7374, CNRS-Université d'Orléans, 1b rue de la Férollerie, CS 40059 45071 Orléans cedex, France

27e journée du projet CASCIMODOT
IRD, 5, rue du carbone (2 sur le plan, case F2) Université d'Orléans

11 décembre 2017

Présentation de l'ICMN ex CRMD

- ICMN - UMR7374 (Institut de Physique)
- 27 permanents (PR, DR, MCF, MCF-HDR, IR, IE) et 10 non-permanents (doctorants et post-doctorants).



(a) **(b)**

Figure: Plan du campus CNRS et Polytech Orléans.

Présentation de l'ICMN ex CRMD

Équipes de recherche, http://www.icmn.cnrs-orleans.fr/

1. **Matériaux Nanostructurés et Confinés**
   - i. Thermodynamique et structure à l'échelle nanométrique,
   - ii. Auto-organisation ou structuration,
   - iii. Dynamique dans les systèmes confinés.

2. **Carbones fonctionnels-Environnement-Biomatériaux**
   - i. Élaboration du matériau carboné fonctionnalisé (Carbones fonctionnels),
   - ii. Intégration des carbones dans des systèmes ou procédés (Environnement),
   - iii. Étude de la réactivité interfaciale et des propriétés physico-chimiques (Biomatériaux).

Présentation de l'ICMN ex CRMD

Équipes de recherche, http://www.icmn.cnrs-orleans.fr/

1 Matériaux Nanostructurés et Confinés

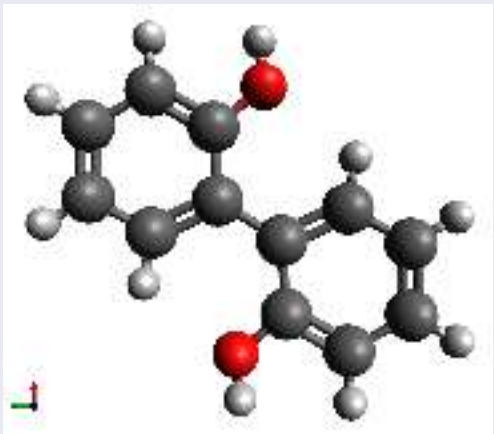
i. Thermodynamique et structure à l'échelle nanométrique,
ii. Auto-organisation ou structuration,
iii. Dynamique dans les systèmes confinés.

2 Carbones fonctionnels-Environnement-Biomatériaux

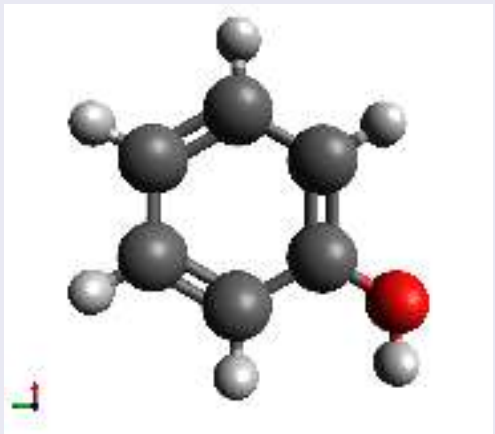
i. Élaboration du matériau carboné fonctionnalisé (Carbones fonctionnels),
ii. Intégration des carbones dans des systèmes ou procédés (Environnement),
iii. Étude de la réactivité interfaciale et des propriétés physico-chimiques (Biomatériaux).

What are the micropollutants?

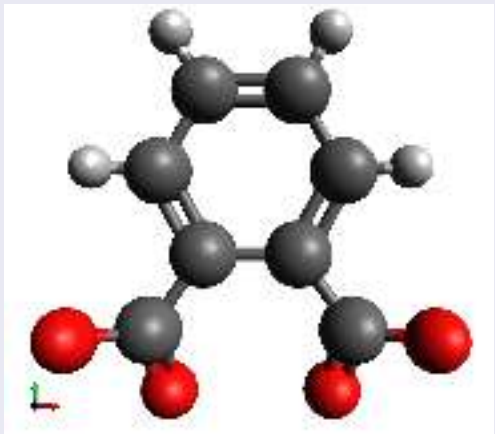
Main micropollutants: Nanoparticles, Biphenol, Phenol, Phthalate ...



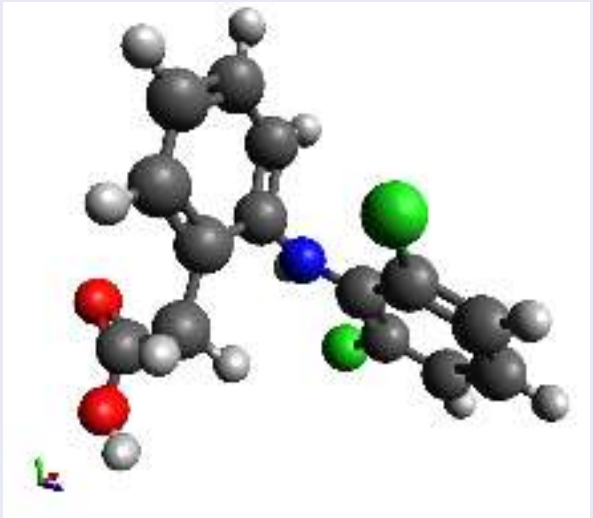
(a) Biphenol ($C_{12}H_{10}O_2$)



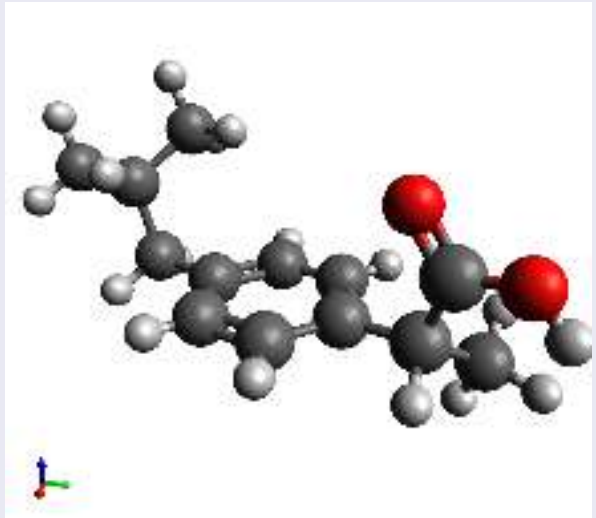
(b) Phenol (C_6H_6O)



(c) Phthalate ($C_8H_4O_4$)



(d) Diclofenac ($C_{14}H_{11}Cl_2NO_2$)



(e) Ibuprofen $C_{13}H_{18}O_2$

What are the micropollutants?

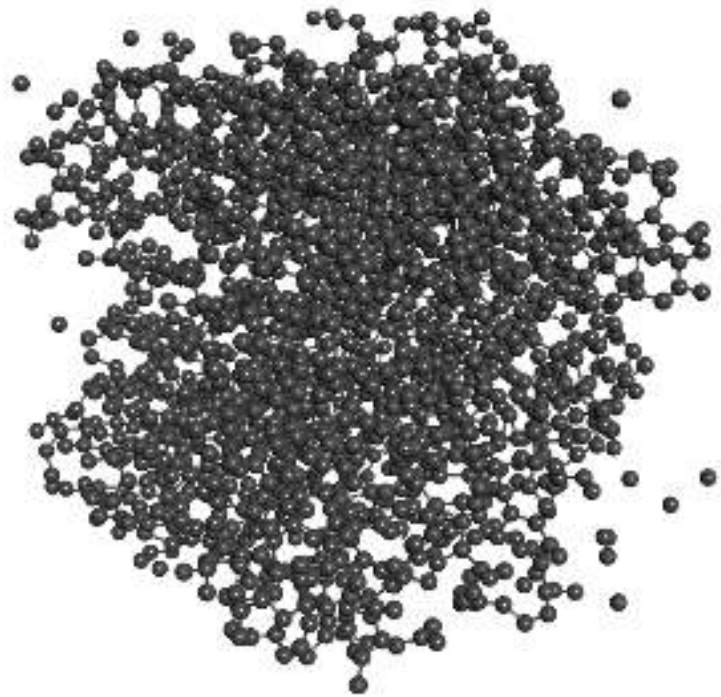
Problem statement

Major issues about the micropollutants

1. The micropollutants are the most challenge for the environment.
2. They can be readily found out in the drinking water.
3. The EU strengths the restrictions on the micropollutants concentration in the drinking water.

Remedies:

1. Optimizing existing procedures,
2. **Additional treatment procedures.**
   1. Ozonation: ozone is a powerful oxidizing agent that degrades chemical substances in gas and liquid mixtures,
   2. Advanced oxidation processes: adding hydrogen peroxide and ultraviolet light to ozone to intensify the oxidation process.
   3. **Adsorption on granular activated carbon**: the porous structure facilitates adsorption of a wide range of micropollutants.



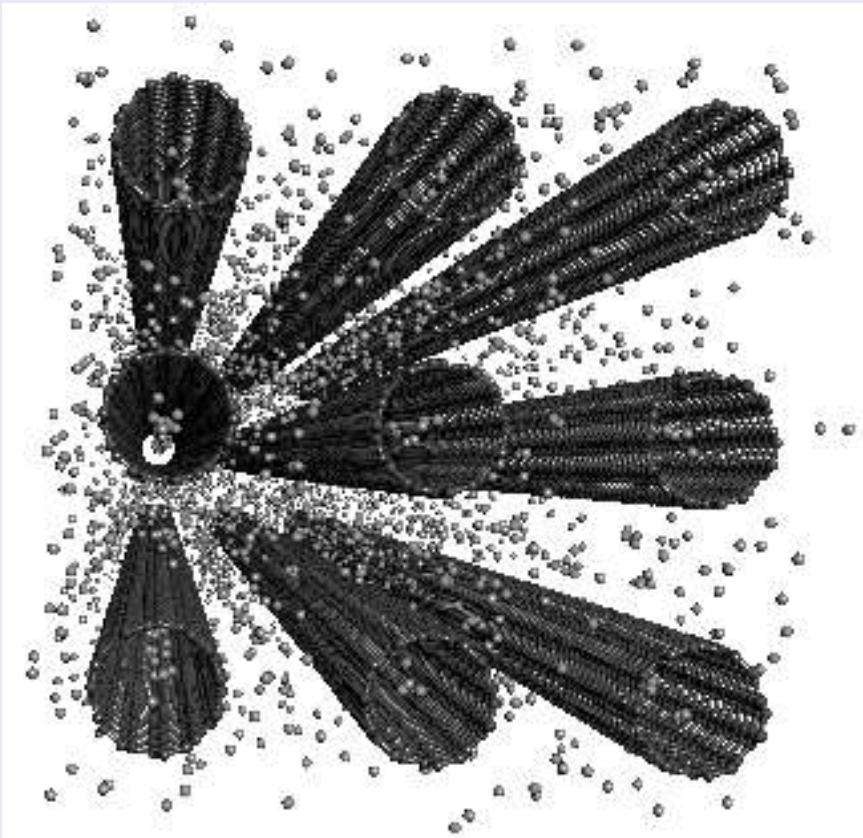
(a) Activated carbon [Pikunic et al., 2003]

What is the adsorption process of the micro-pollutants?

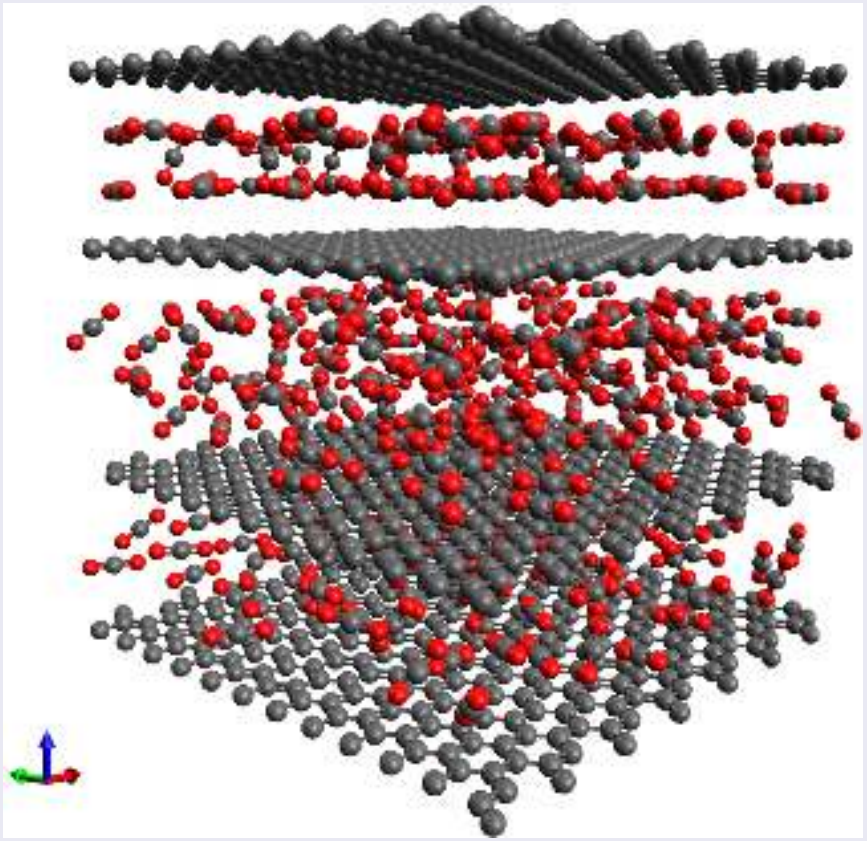
Main definition

Adsorption is the adhesion of atoms, ions, or molecules from a gas, liquid, or dissolved solid to a surface.

(a) H_2 adsorption in SWCNT [Kouetcha et al., 2015]



(b) CO_2 adsorption illustration



Physisorption (physical adsorption) and Chemisorption (chemical adsorption)

Adsorption types

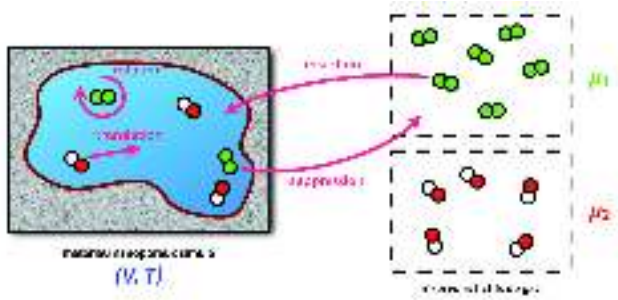
- **Physisorption** or so-called physical adsorption, is a process in which the electronic structure of the atom or molecule is mainly perturbed upon adsorption,
- **Chemisorption** or so-called chemical adsorption involves a chemical reaction between the surface and the adsorbate.

Physical adsorption modeling

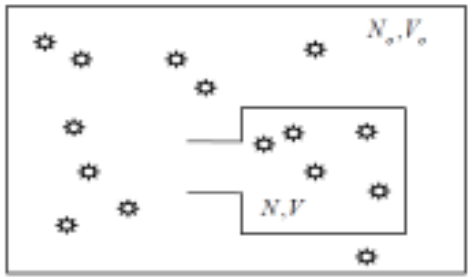
- The physical adsorption can be handled via the
  1. statistical mechanics,
  2. ensembles of the statistical mechanics (Grand Canonical ensemble)
- Physical adsorption using the Grand Canonical ensemble assumes given values for the **chemical potential**, **temperature** and **volume**.
- The **guest molecules density** and **their positions**, **adsorption heat** and **pressure** can be computed.

Molecular simulation : Grand Canonical Monte Carlo

Probability rules [Metropolis et al., 1953]



Feasible movements



Grand Canonical ensemble illustration (μ, V, T).

- Creation

$$P_{\text{acc}}(\mathrm{r}^N \to \mathrm{r}^{N+1}) = \min\left(1, \frac{V\exp(\beta\mu)}{(N+1)\Lambda^3} \times \exp\left[-\beta\left(U(\mathrm{r}^{N+1}) - U(\mathrm{r}^N)\right)\right] \times \frac{P_{\text{gen}}(\mathrm{r}^{N+1} \to \mathrm{r}^N)}{P_{\text{gen}}(\mathrm{r}^N \to \mathrm{r}^{N+1})}\right)$$

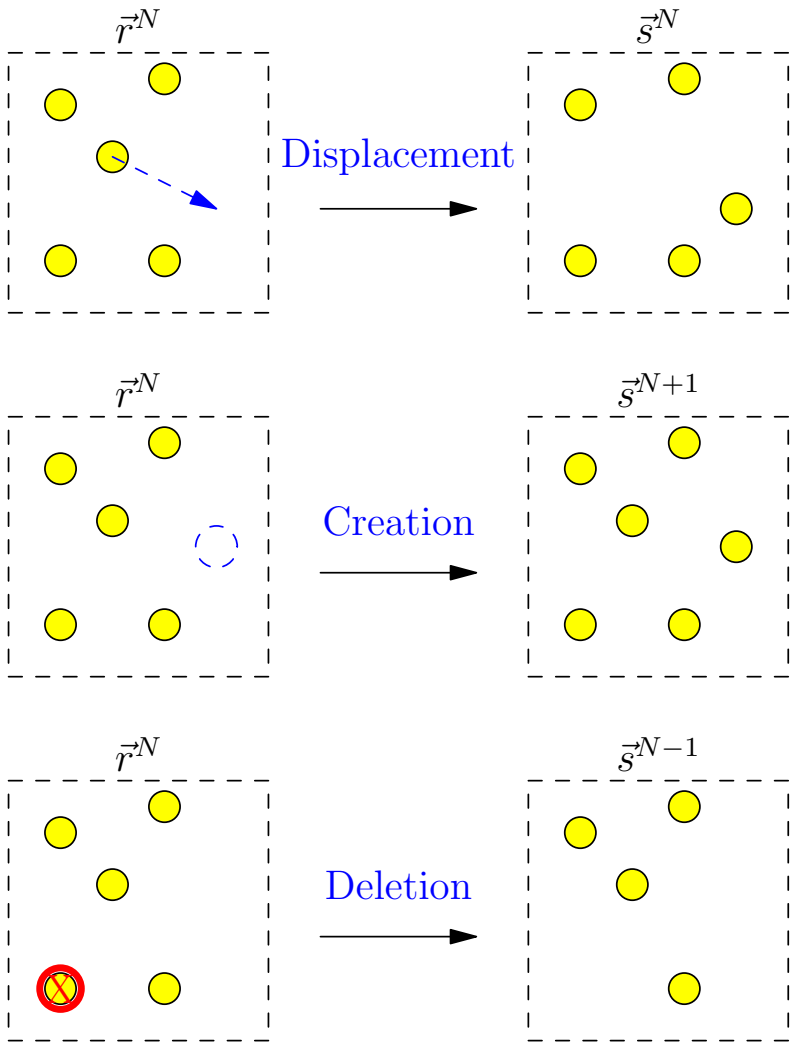
- Deletion

$$P_{\text{acc}}(\mathrm{r}^N \to \mathrm{r}^{N-1}) = \min\left(1, \frac{N\Lambda^3}{V\exp(\beta\mu)} \times \exp\left[-\beta\left(U(\mathrm{r}^{N-1}) - U(\mathrm{r}^N)\right)\right] \times \frac{P_{\text{gen}}(\mathrm{r}^{N-1} \to \mathrm{r}^N)}{P_{\text{gen}}(\mathrm{r}^N \to \mathrm{r}^{N-1})}\right)$$

- Displacement

$$P_{\text{acc}}(\mathrm{r}^N \to \mathrm{r}^{N-1}) = \min\left(1, \exp\left[-\beta\left(U(\mathrm{r}^{N-1}) - U(\mathrm{r}^N)\right)\right] \times \frac{P_{\text{gen}}(\mathrm{r}^{N-1} \to \mathrm{r}^N)}{P_{\text{gen}}(\mathrm{r}^N \to \mathrm{r}^{N-1})}\right)$$

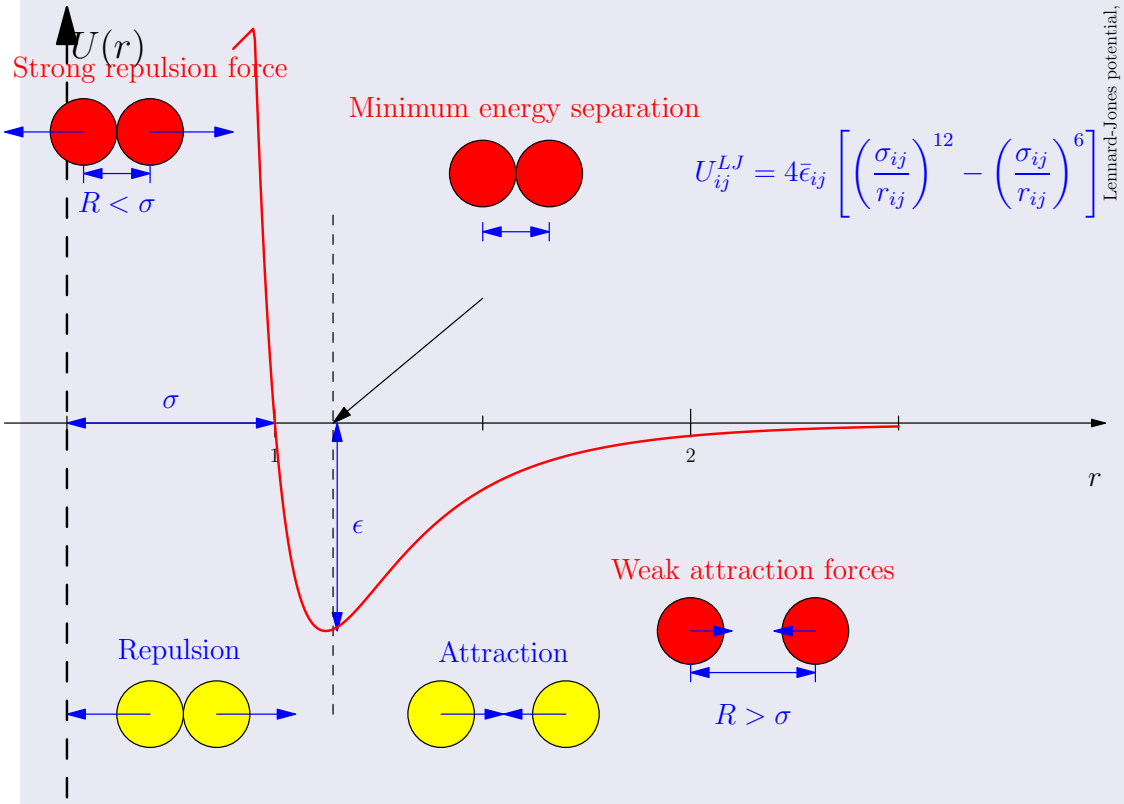
Molecular simulation : GCMC kernel



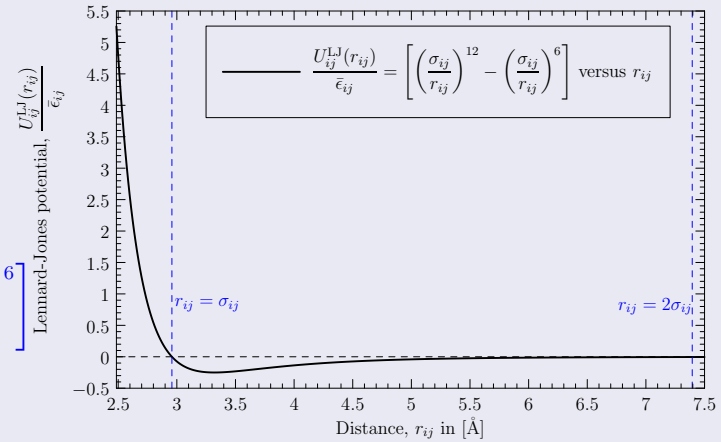
(a) GCMC statistical engine [Allen and Tildesley, 1989; Frenkel and Smit, 2002; Ramézani and Chuta, 2014]

Grand Canonical Monte Carlo (GCMC)

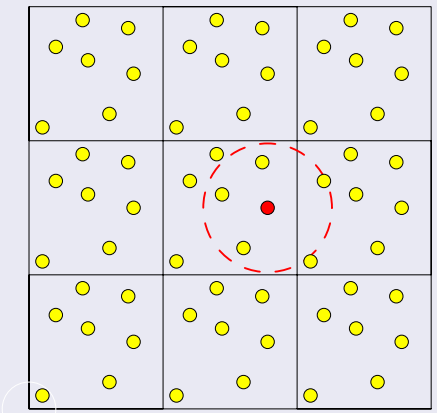
Van der Waals Potential energy illustration



(a) Lennard Jones curve illustration [Crowell and Brown, 1982; Wang and Johnson, 1998]

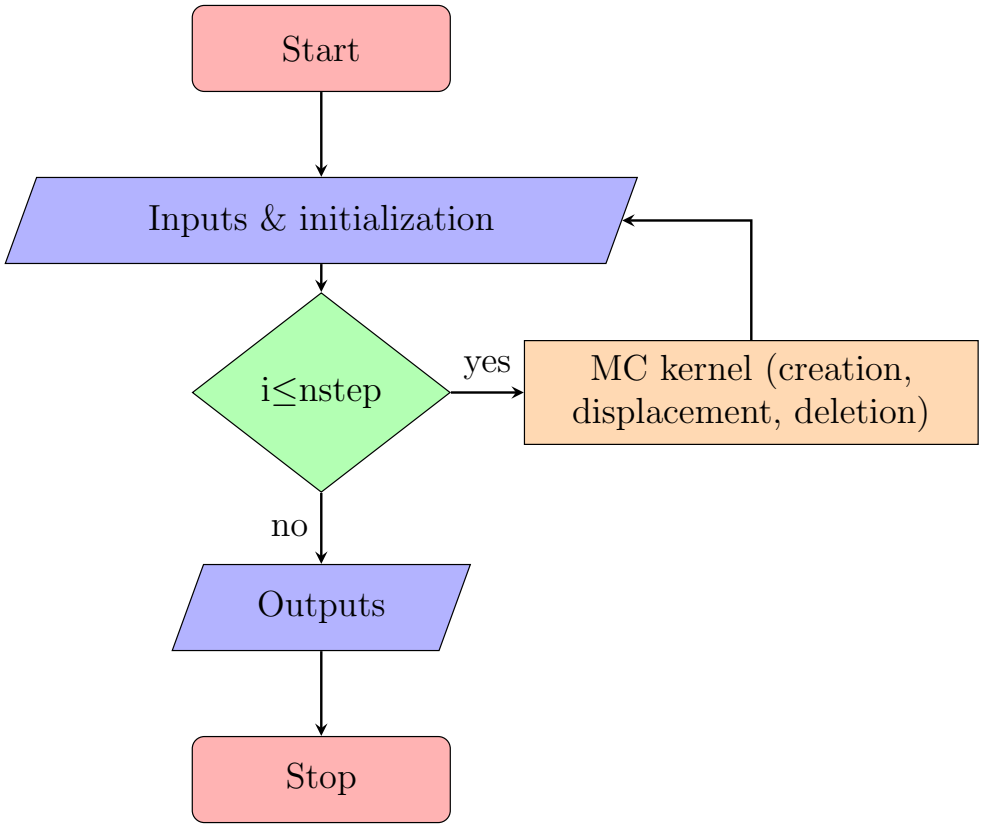


(a) LJ curve of H_2



(b) Cut-off sphere illustration

Grand Canonical Monte Carlo (GCMC): Basic flowchart

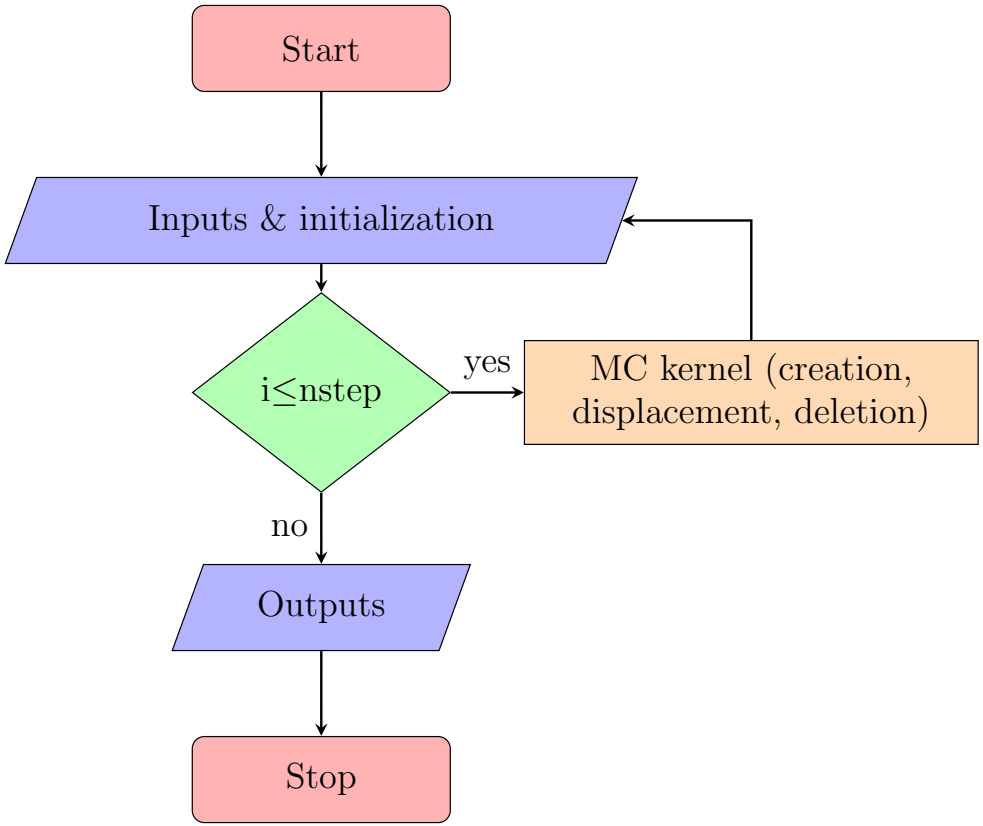


(a) Simple flowchart of GCMC [Chuta, 2013]

Why does one need the supercomputing?

1. Monte Carlo method needs at least 10^3 iterations per host atom for so-called rigid molecules (one-point molecule assumption),
2. This gets worse when the non-rigid molecules are considered and iteration no. must be greater than 10^6.
3. The adsorption curves will undertake various chemical potentials and it means that it is required to perform the simulations for every single point in the plot (iteration no. $=50 \times 10^6$ per host atom.

Grand Canonical Monte Carlo (GCMC): Basic flowchart

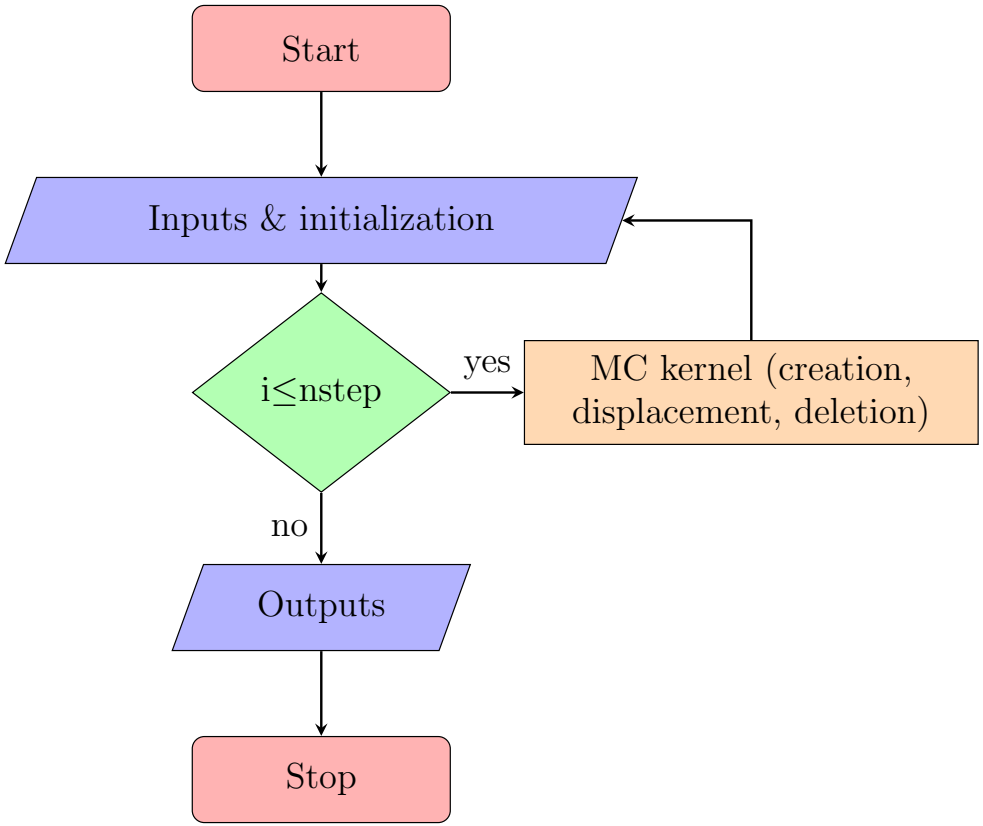


(a) Simple flowchart of GCMC [Chuta, 2013]

Why does one need the supercomputing?

1. Monte Carlo method needs at least 10^3 iterations per host atom for so-called rigid molecules (one-point molecule assumption),
2. This gets worse when the non-rigid molecules are considered and iteration no. must be greater than 10^6.
3. The adsorption curves will undertake various chemical potentials and it means that it is required to perform the simulations for every single point in the plot (iteration no. $=50 \times 10^6$ per host atom.

Grand Canonical Monte Carlo (GCMC): Basic flowchart



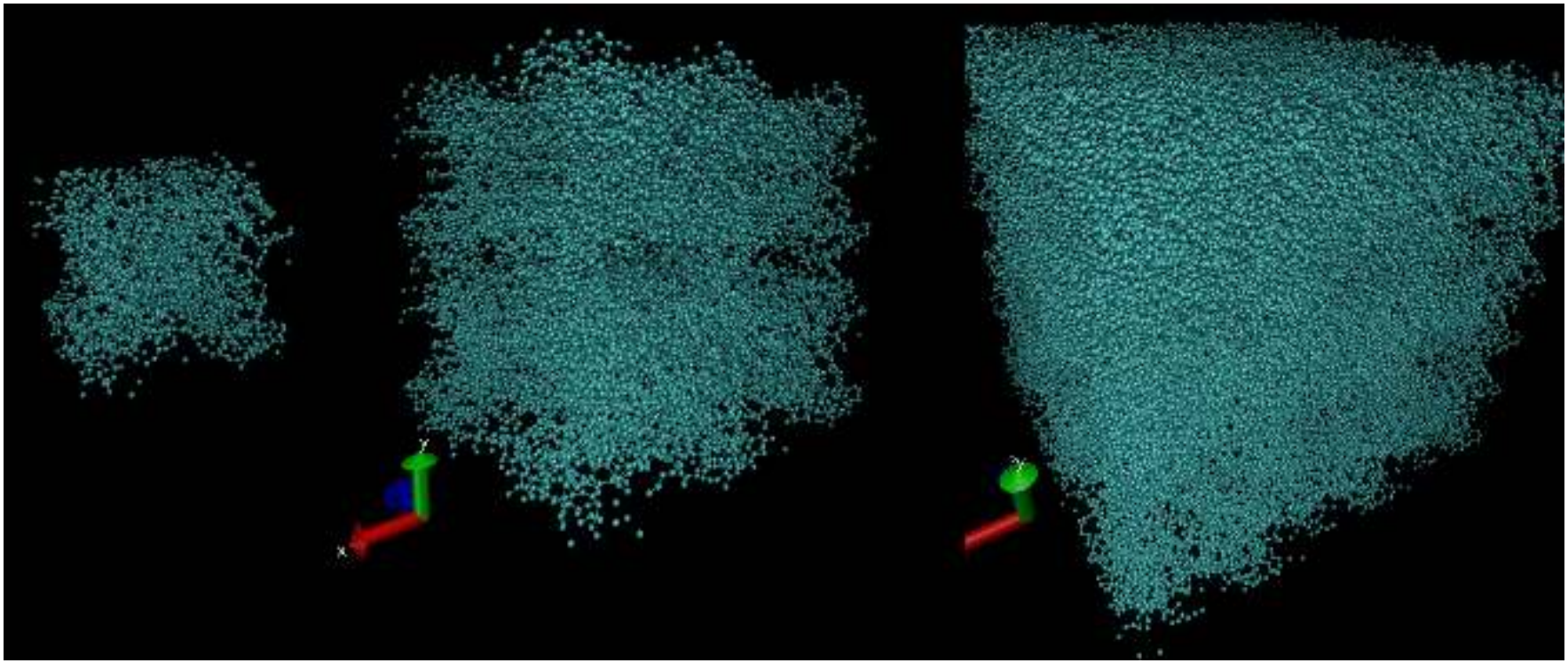
(a) Simple flowchart of GCMC [Chuta, 2013]

Why does one need the supercomputing?

1. Monte Carlo method needs at least 10^3 iterations per host atom for so-called rigid molecules (one-point molecule assumption),
2. This gets worse when the non-rigid molecules are considered and iteration no. must be greater than 10^6.
3. The adsorption curves will undertake various chemical potentials and it means that it is required to perform the simulations for every single point in the plot (iteration no. $=50 \times 10^6$ per host atom.

Grand Canonical Monte Carlo (GCMC)

Highly disordered structures



(a) Size effect (from 30×30×30 to 90×80×90 Å)

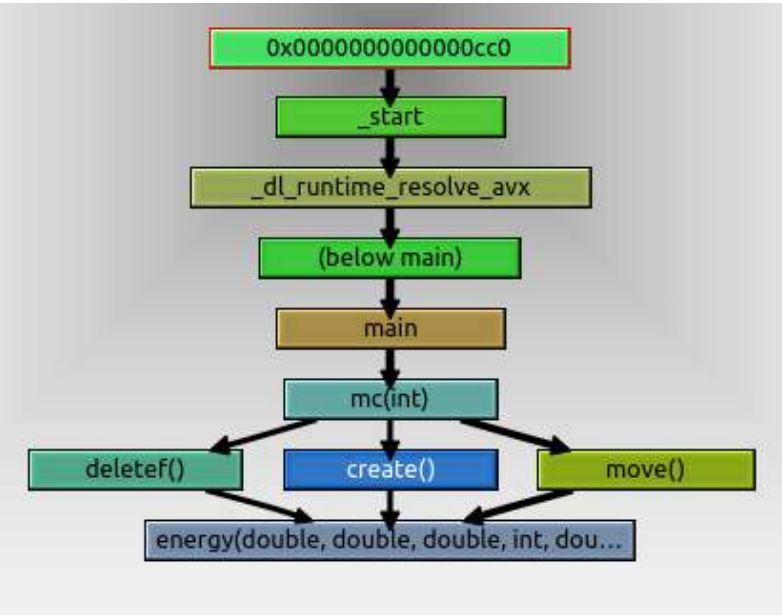
Size	30×30×30	60×60×60	90×90×90	120×120×120	240×240×240	360×360×360
No. of atoms	2130	17040	57510	136320	1090560	3680640

Grand Canonical Monte Carlo (GCMC) and its bottleneck

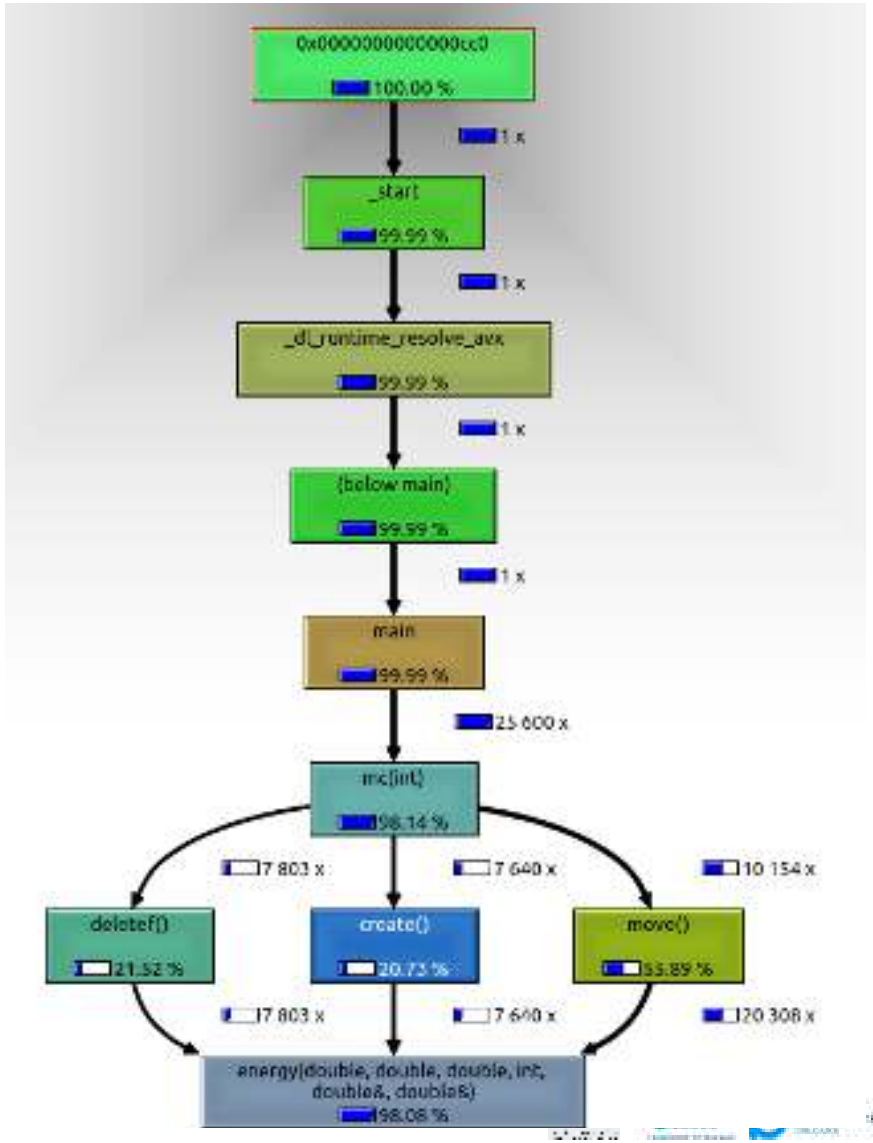
Main goals and single program profiling of the serial GCMC code (GNU compilers and valgrind profiler)

Main goals

- Code portability [Stroustrup, 2013],
- Code parallelism and scalability,
- Code flexibility.



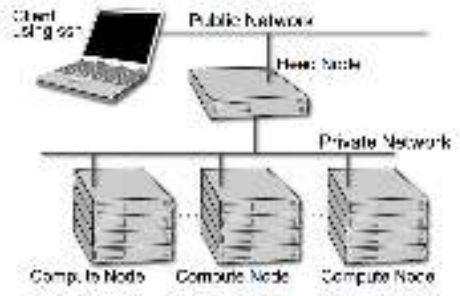
(a) Compact graph of GCMC code



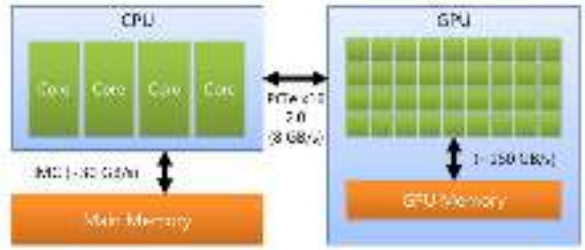
(b) Normal graph of GCMC code.

Parallel computing (OpenMP/MPI/CUDA)

Message Passing Interface (MPI-3)



(a) CPU computing

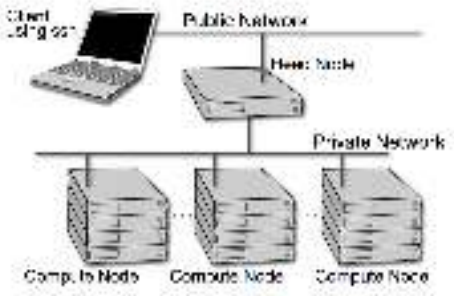


(b) GPU computing

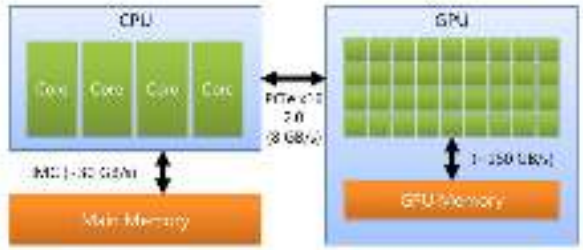
- MPI programming is applied for the parallel computations herein (MPI-3) [Gropp et al., 2014a,b; Karniadakis and Kirby, 2003; Snir, 1998],
- GNU GCC compilers (gcc 5.3.0) and open MPI 1.10.2 are mainly used to do the experiments,
- Parallel computations have been carried out in the CCSC cluster in France (`http://cascimodot.fdpoisson.fr/ccsc`).
- Two different Parallel MPI implementations have been taken into account:
  1. **Energy subroutine** is solely parallelized (**First paradigm**)
  2. Virtual volumes have been created and the averaging issues of GCMC have been achieved.**Monte Carlo iterations have been parallelized** as a whole (**Second paradigm**)

Parallel computing (OpenMP/MPI/CUDA)

Message Passing Interface (MPI-3)



(c) CPU computing



(d) GPU computing

- MPI programming is applied for the parallel computations herein (MPI-3) [Gropp et al., 2014a,b; Karniadakis and Kirby, 2003; Snir, 1998],
- GNU GCC compilers (gcc 5.3.0) and open MPI 1.10.2 are mainly used to do the experiments,
- Parallel computations have been carried out in the CCSC cluster in France (`http://cascimodot.fdpoisson.fr/ccsc`).
- Two different Parallel MPI implementations have been taken into account:
  1. **Energy subroutine** is solely parallelized (**First paradigm**)
  2. Virtual volumes have been created and the averaging issues of GCMC have been achieved.**Monte Carlo iterations have been parallelized** as a whole (**Second paradigm**).

Parallel computing

First paradigm: Energy subroutine is only parallelized

First paradigm

- The energy subroutine takes 98.08% of computation cost.
- The energy subroutine parallelization may reduce the computation runtime.

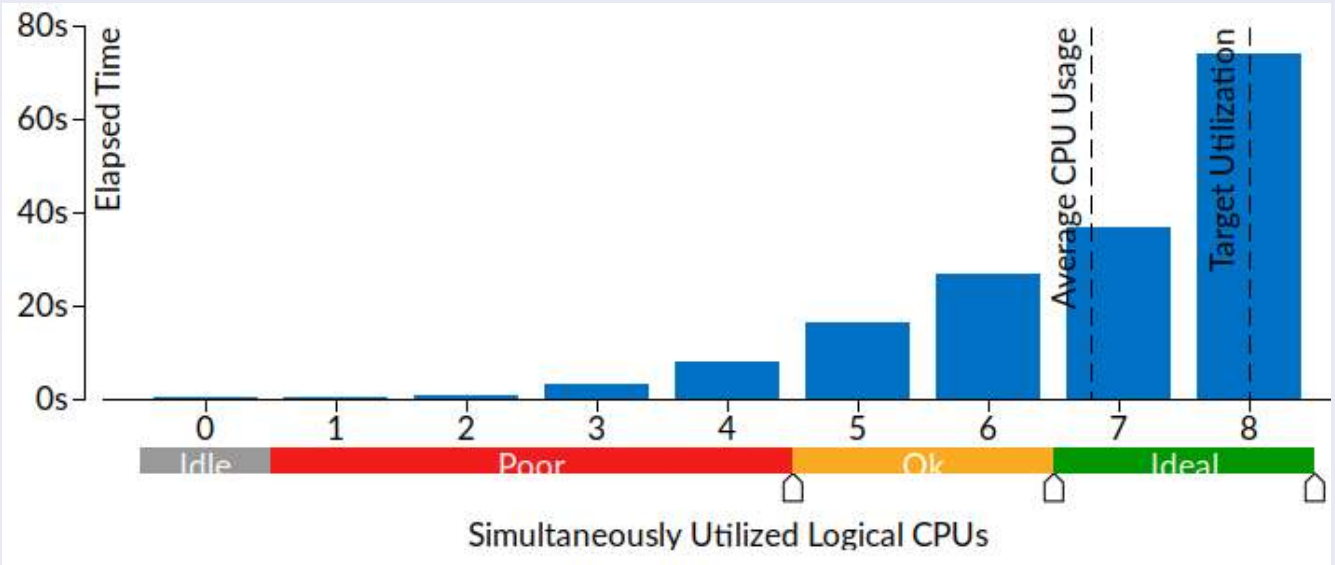
Parallelization of energy subroutine

- The energy subroutine main loop has been distributed along the CPUs and the outcomes have been collected via the MPI collective functions (MPI_Allreduce and MPI_Reduce).
- The following MPI functions have been globally applied to perform the parallel computing.
  1. **Basic MPI functions:** MPI_Int, MPI_Comm_size, MPI_Comm_rank and MPI_Finalize,
  2. **Additional MPI functions:** MPI_time, MPI_Barrier, MPI_Allreduce and MPI_Reduce.

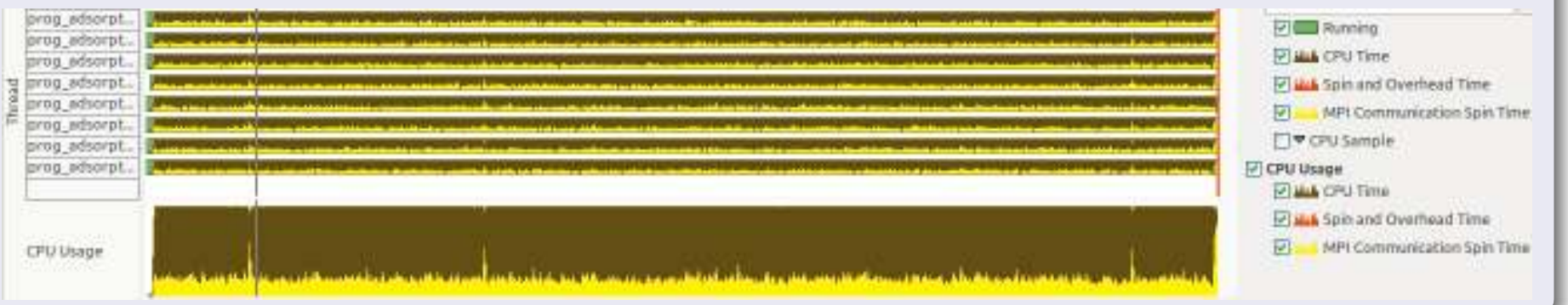
First paradigm

Parallel profiling and CPU usage

Parallel profiling



CPU usage



Second paradigm via virtual volumes [Deublein et al., 2011]

Parallel profiling and CPU usage

Second paradigm

- Monte Carlo computations would be performed in the virtual volumes in an independent manner,
- Every single processor achieves the Monte Carlo simulations in its corresponding virtual volume.

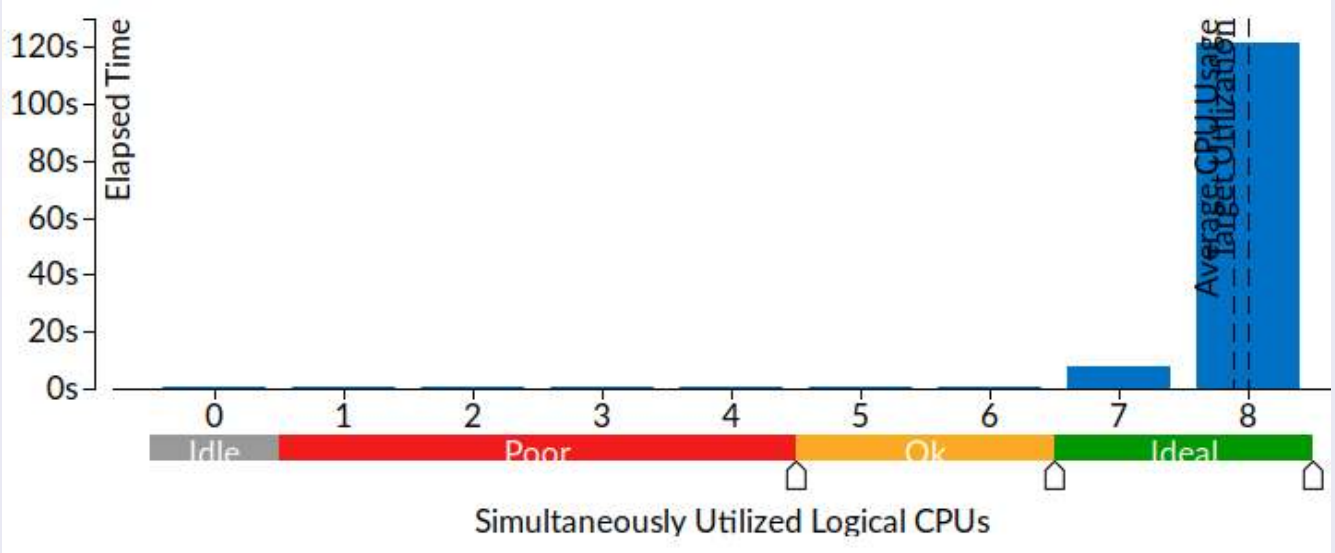
Parallelization of MC iterations

- The main MC loop has been distributed along the CPUs and the outcomes have been collected via the MPI collective functions (MPI_Reduce).
- The following MPI functions have been applied to perform the parallel computing.
  1. **Basic MPI functions:** MPI_Int, MPI_Comm_size, MPI_Comm_rank and MPI_Finalize,
  2. **Additional MPI functions:** MPI_time, MPI_Barrier, MPI_Get_processor_name and MPI_Reduce.

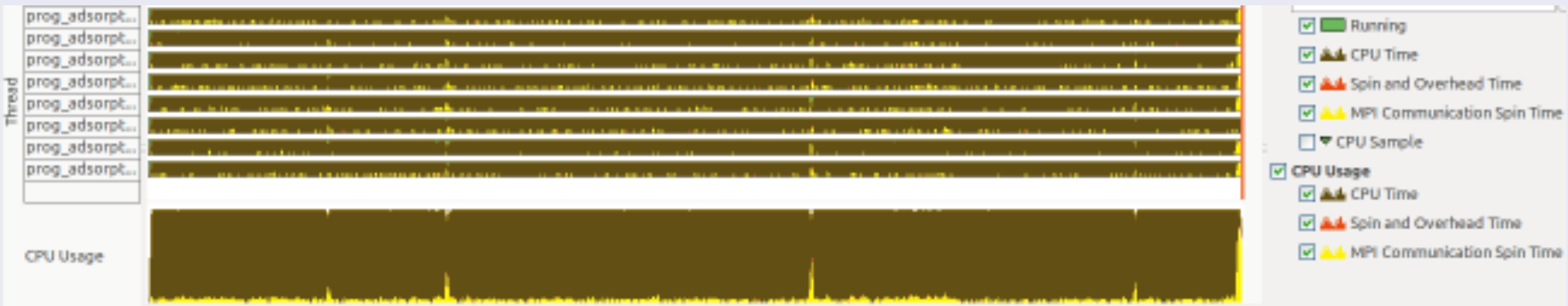
Second paradigm via virtual volumes [Deublein et al., 2011]

Parallel profiling and CPU usage

Parallel profiling

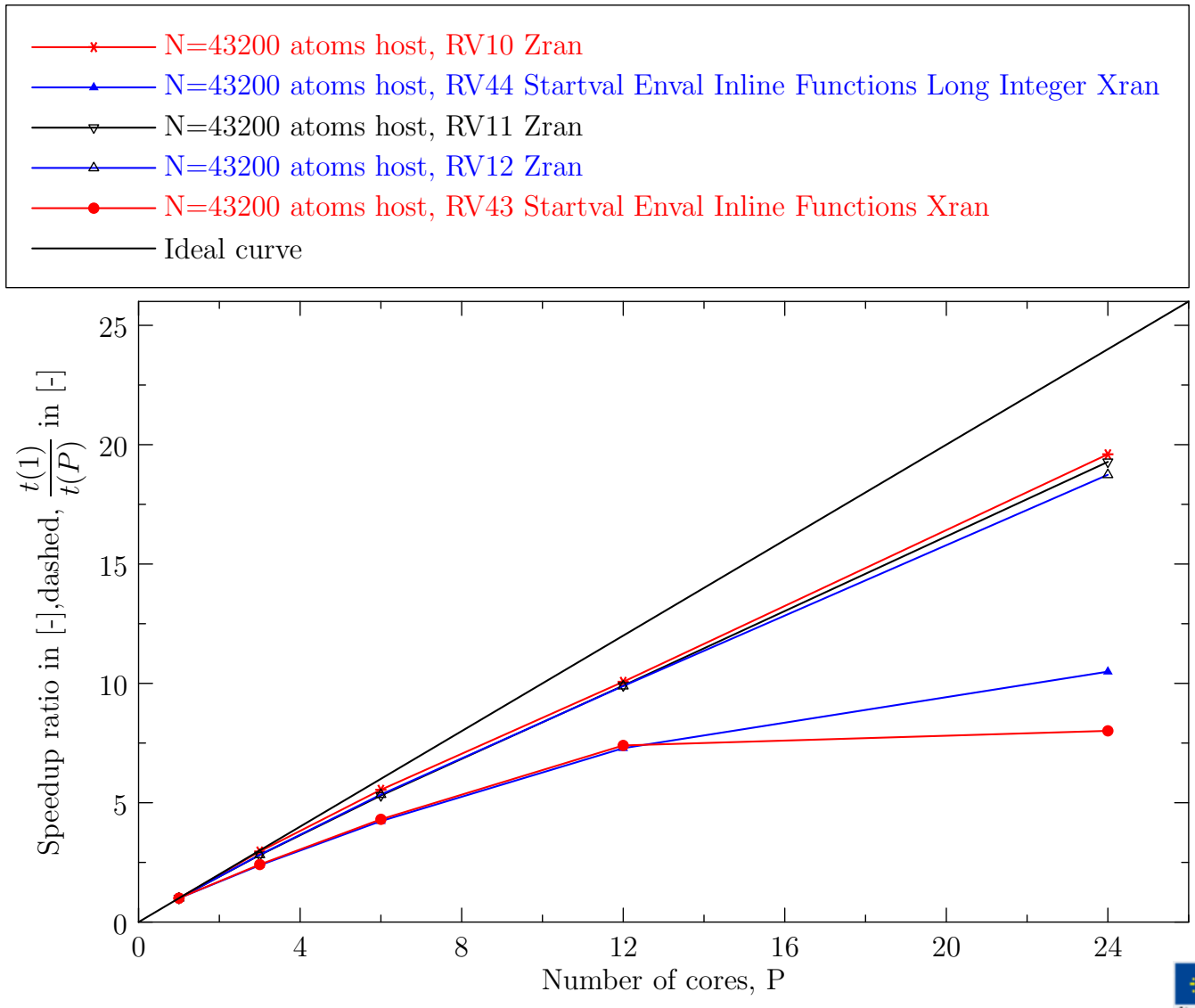


CPU usage



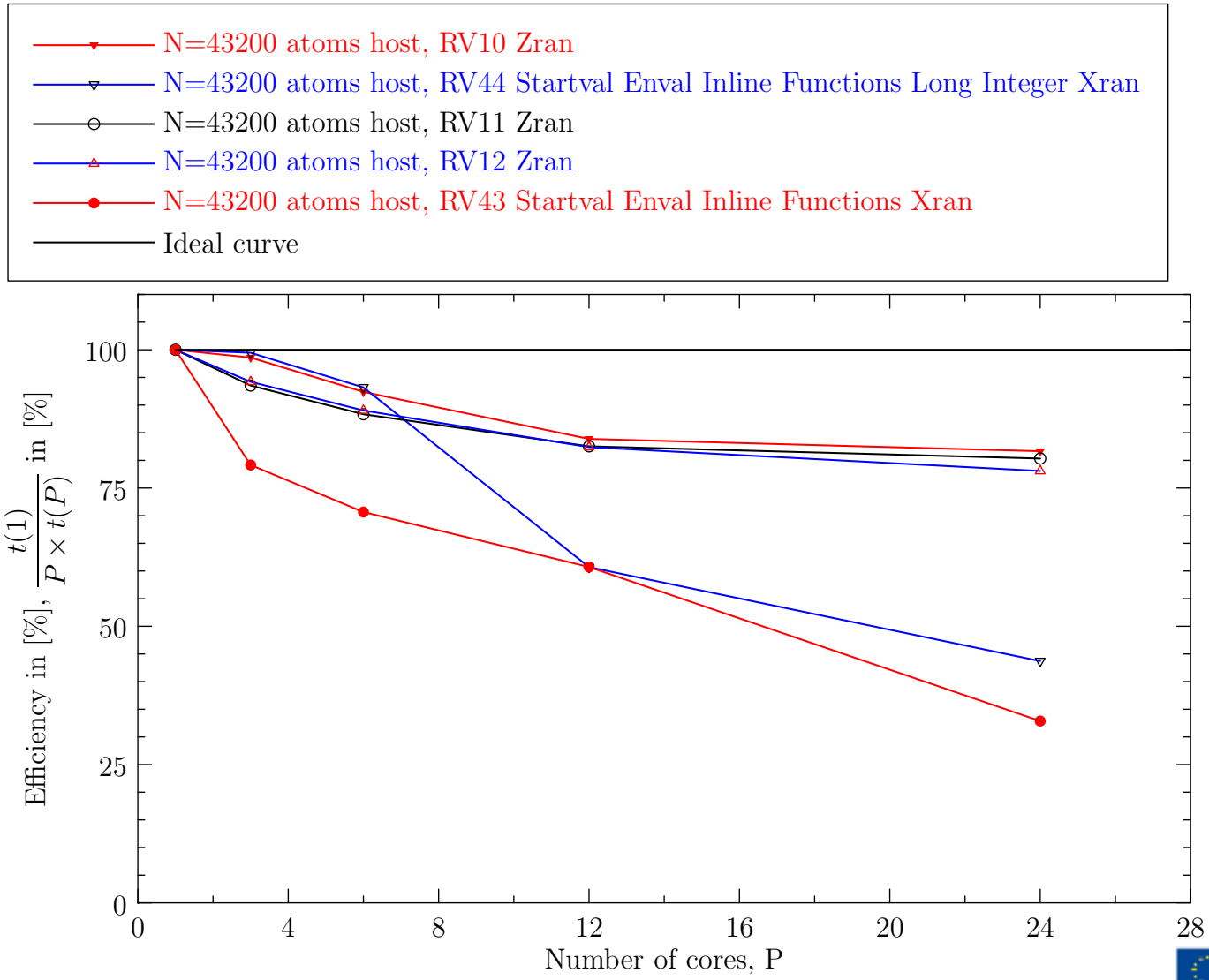
Speedup and efficiency on computing server

Speedup plot using Precision T7810-Intel Xeon E5-2670v3 (2×12C. 2.3GHz. 30Mo cache. 2133 Mhz)



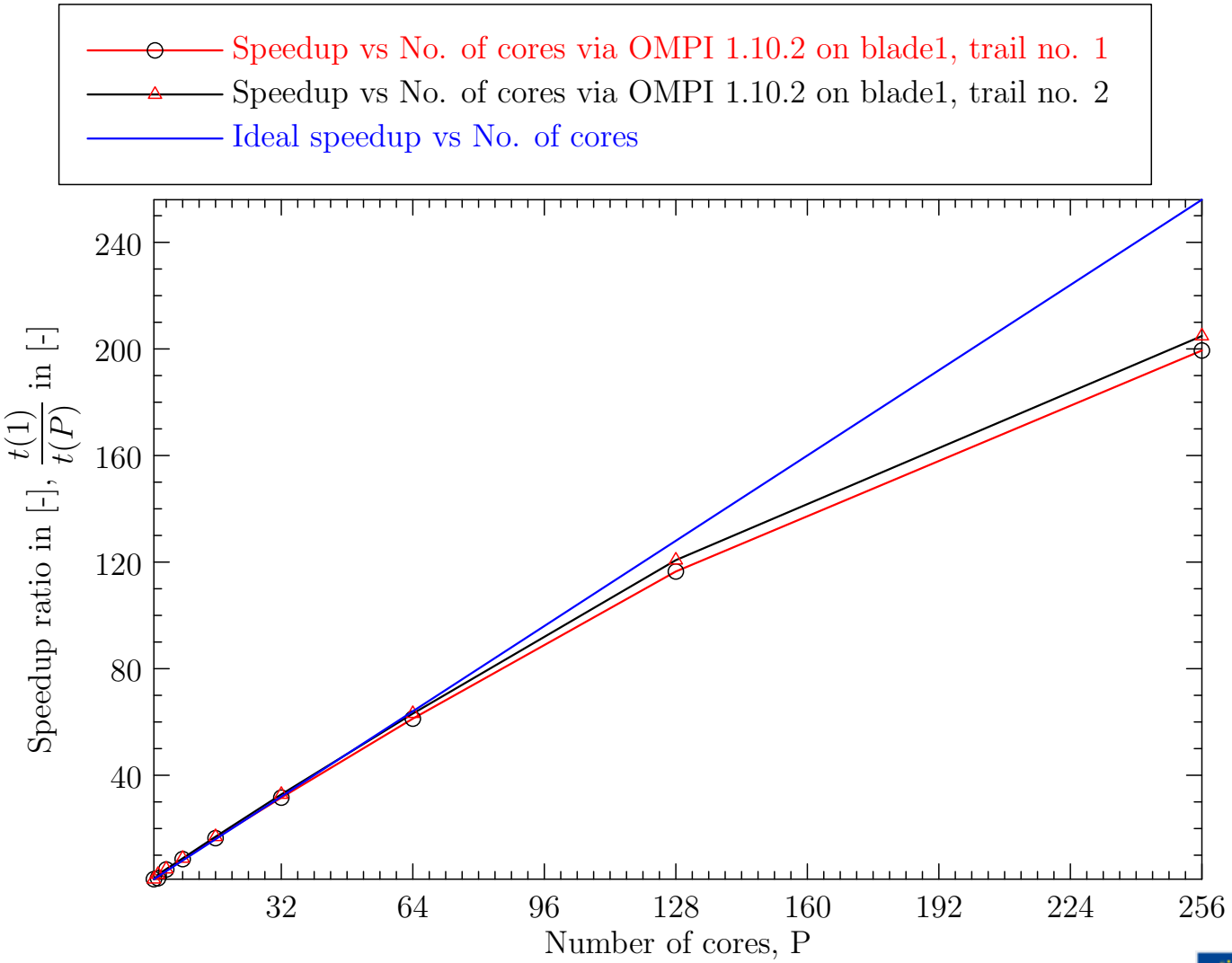
Speedup and efficiency on computing server

Efficiency plot using Precision T7810-Intel Xeon E5-2670v3 (2×12C. 2.3GHz. 30Mo cache. 2133 Mhz)



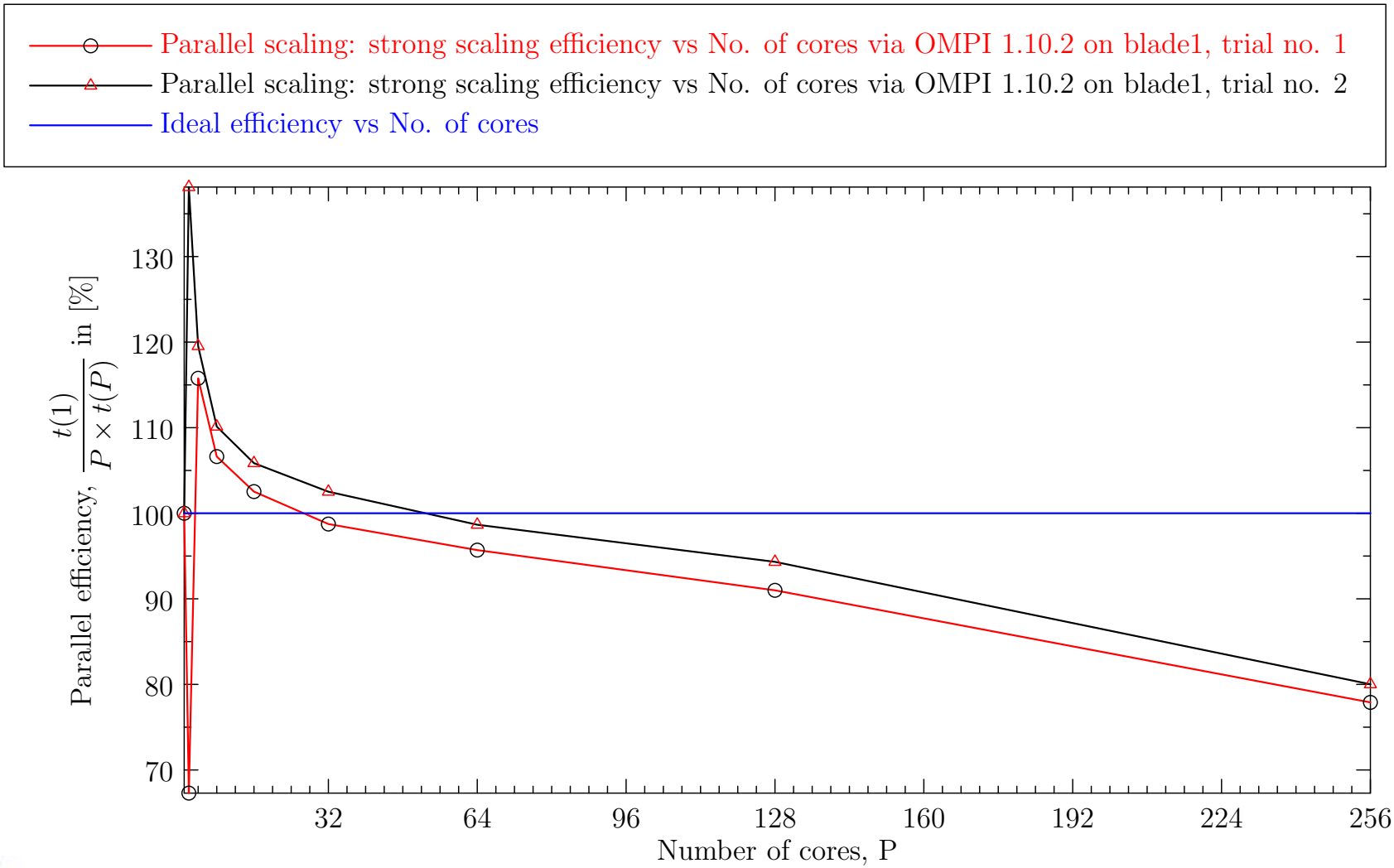
Speedup and efficiency on CCSC (Blade1, Blade2 & defq)

Speedup plot - Distributed Memory



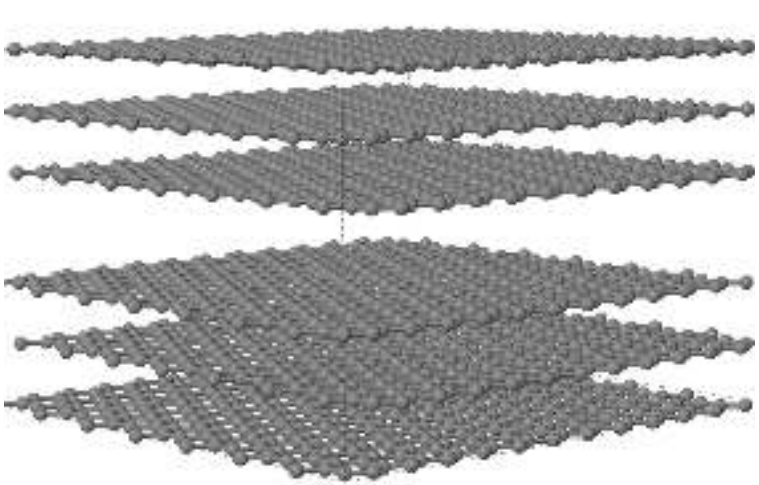
Speedup and efficiency on CCSC (Blade1, Blade2 & defq)

Efficiency plot - Distributed Memory

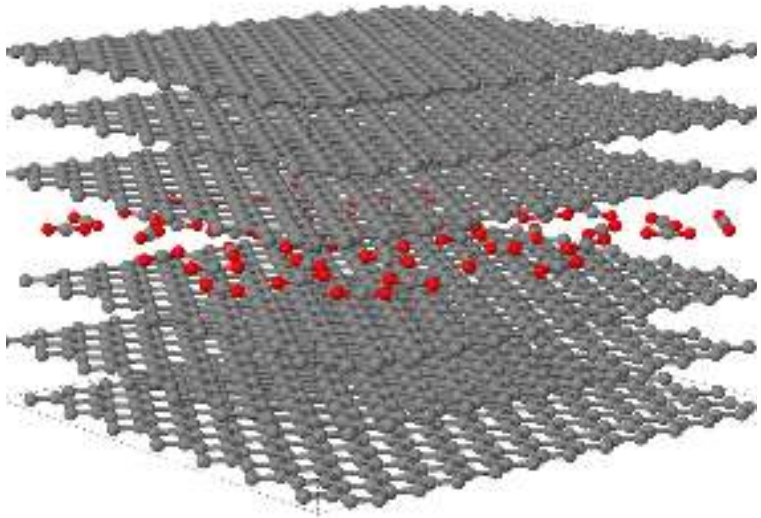


GCMC adsorption outcomes

Adsorption of CO_2 through slik-like carbon structures

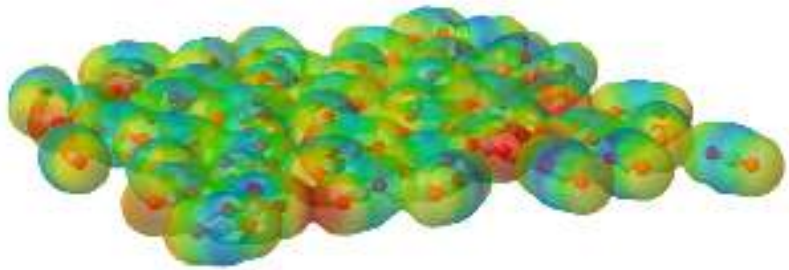


(a) Slit-like structure.



(b) CO_2 molecules adsorption snapshot.

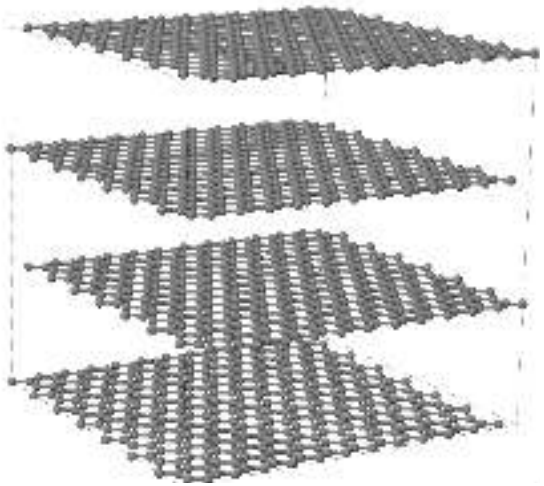
(c) CO_2 molecules excluding electrostatic chagres.



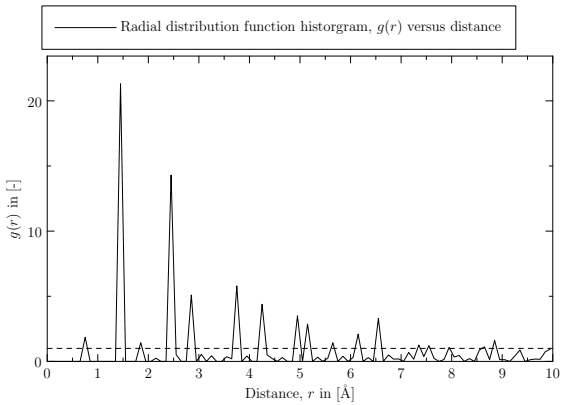
(d) CO_2 molecules including electrostatic chagres.

GCMC adsorption outcomes

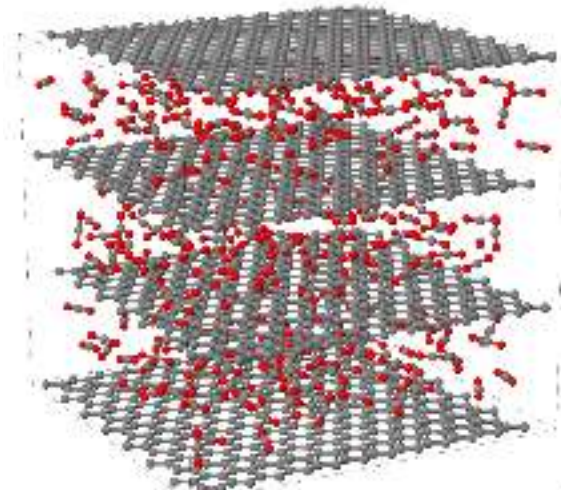
Adsorption of CO_2 through so-called turbostratic carbon structures



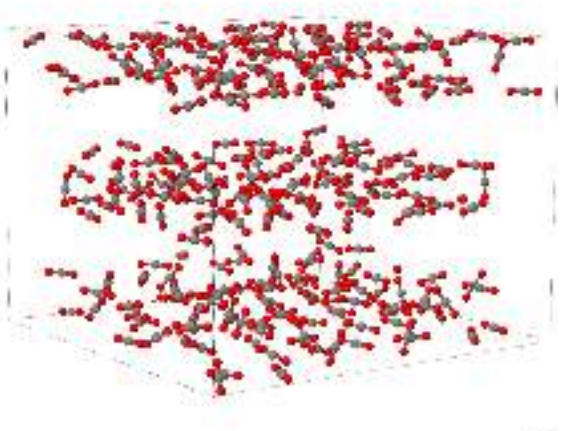
(a) Turbostratic structure.



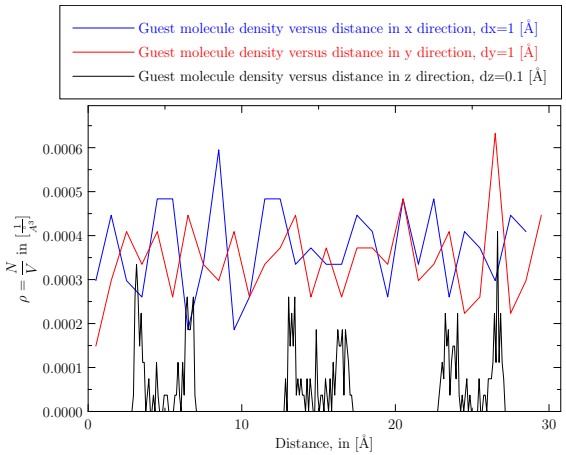
(b) Radial distribution function [Kouetcha et al., 2017].



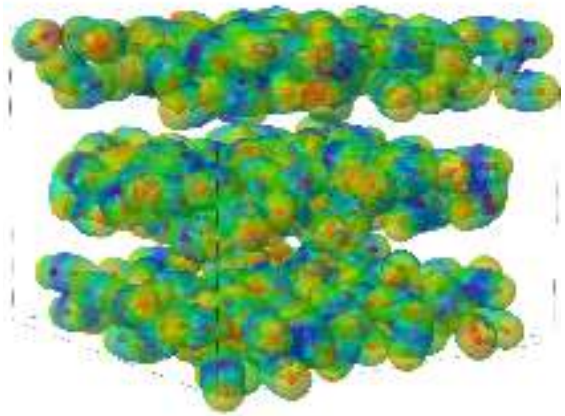
(c) CO_2 molecules adsorption.



(d) CO_2 molecules excluding electrostatic chagres.



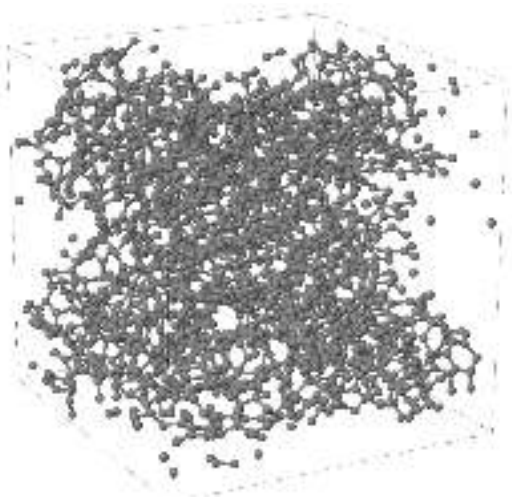
(e) Density profiler of CO_2 molecules.



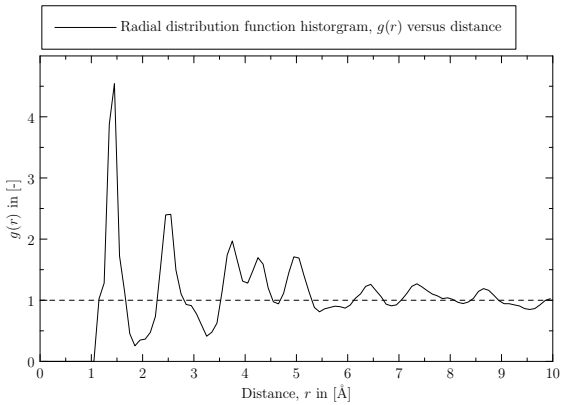
(f) CO_2 molecules including electrostatic chagres.

GCMC adsorption outcomes

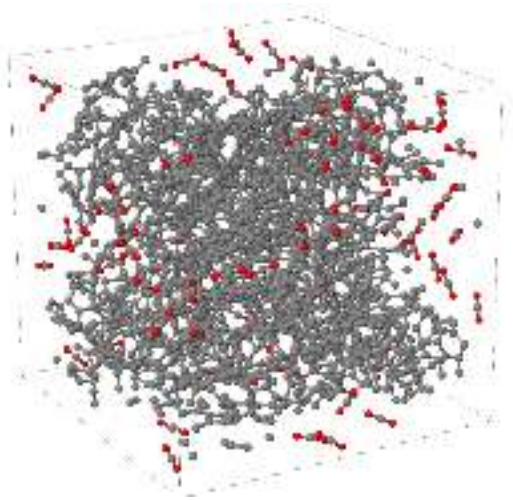
Adsorption of CO_2 through Activated carbon [Pikunic et al., 2003]



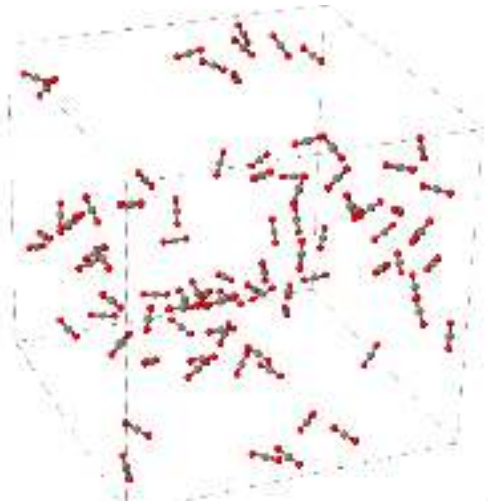
(a) Activated carbon.



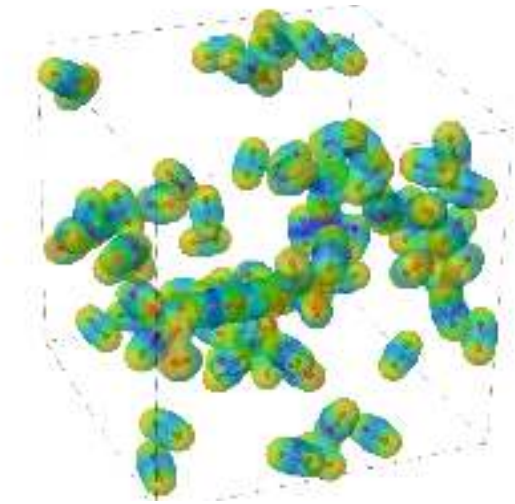
(b) Radial distribution function.



(c) CO_2 molecules adsorption.



(d) CO_2 molecules excluding electrostatic chagres.



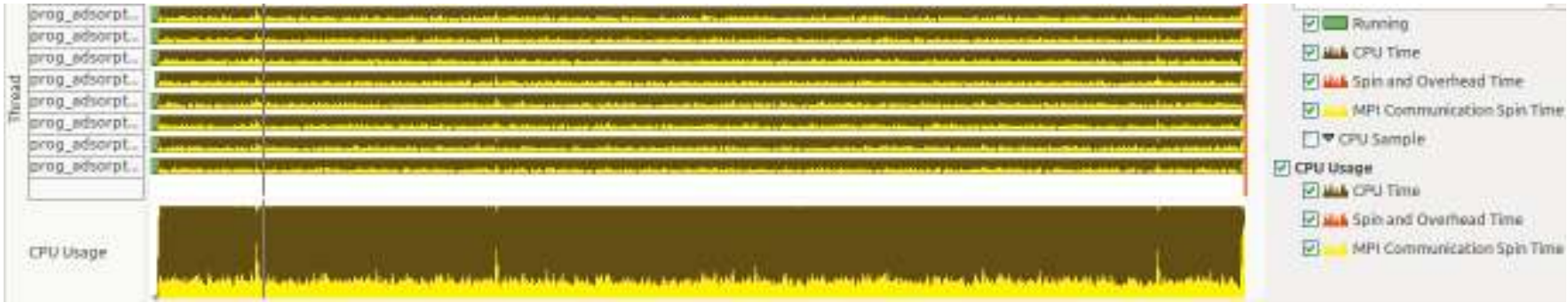
(e) CO_2 molecules including electrostatic chagres.

Comparison between two algorithms

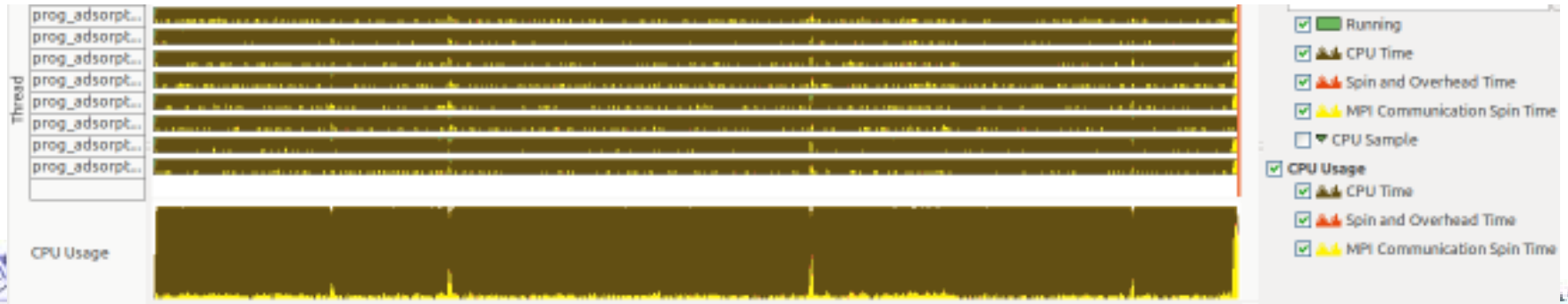
Parallel optimization

- The nearly linear scalable plots for the paradigm 2 rely on two facts:
  1. MPI communications are likely less than the first algorithm,
  2. Serial program runtime is less than the first algorithm using the virtual volumes or so-called copied volumes.
  3. As a result, the program runtime on single core over MPI communications ratio is fairly optimized.

Paradigm 1



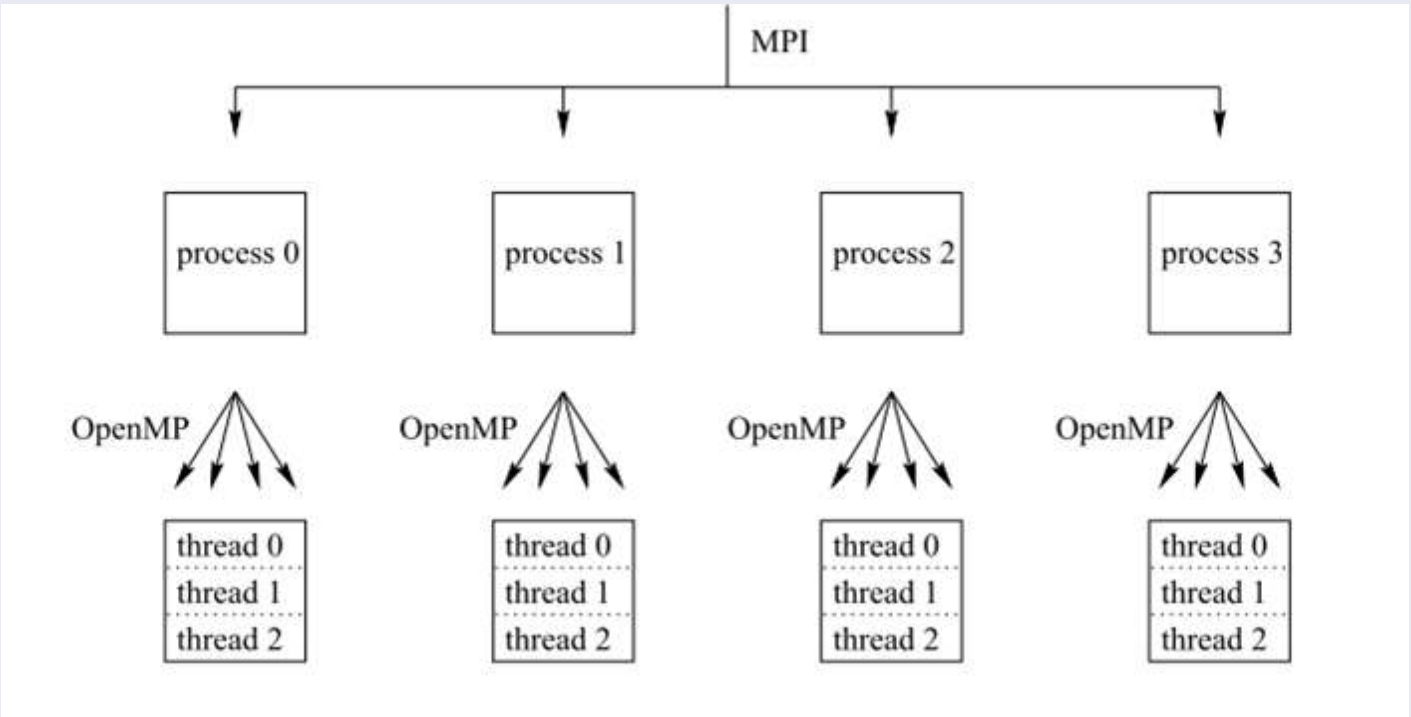
Paradigm 2



Further speedup

Hybrid MPI-OMP programming??

1. Hybrid MPI/OpenMP programming.
   - MPI does inter-node computations (distributed memory computation),
   - OpenMP performs the computations on the available threads (shared memory computation)



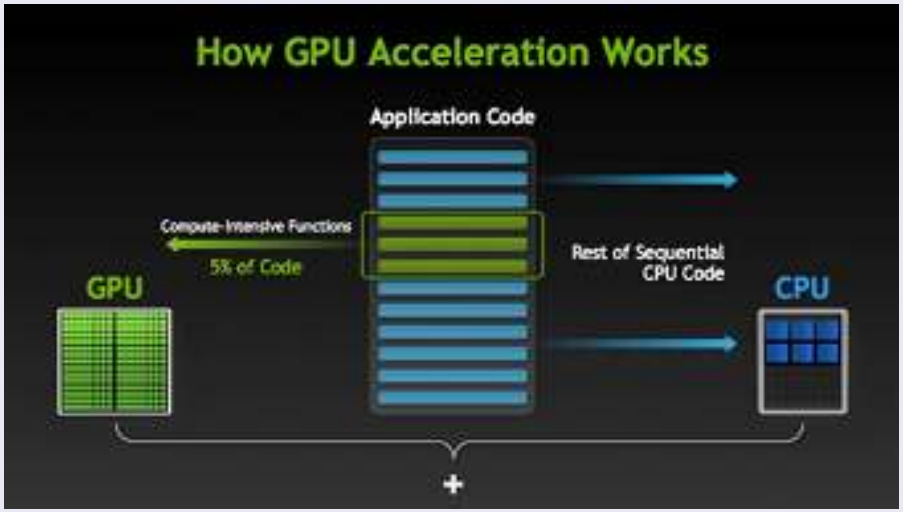
Further speedup

GPU computing

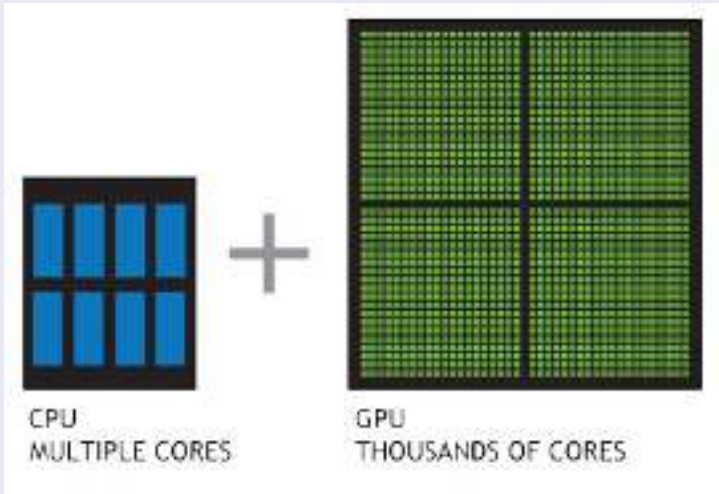
http://www.nvidia.com/object/what-is-gpu-computing.html

GPU computing

- Cuda C programming [Cheng et al., 2014; Sanders and Kandrot, 2010],
- OpenCL programming.



(a) How GPUs accelerate software applications.



(b) Thousands of cores.

Further speedup

GPU supercomputing data extracted from "https://www.top500.org/lists/2017/11/"

TOP 5 list of the fastest supercomputers in the world (November 2017)

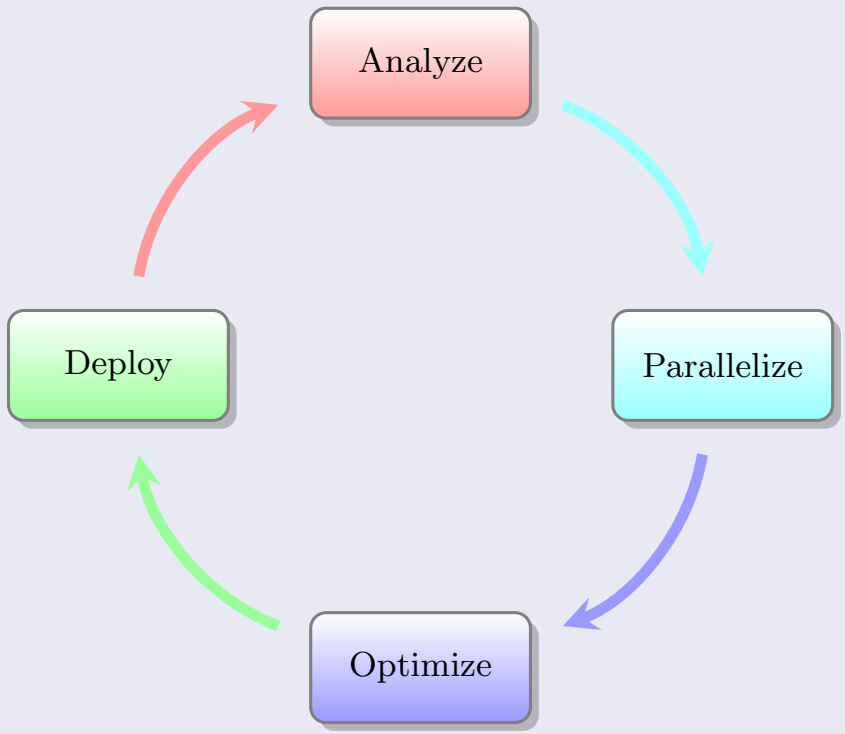
Rank	System	Cores	Rmax (TFlop/s)	Rpeak (TFlop/s)	Power (kW)
1	**Sunway TaihuLight** - Sunway MPP, Sunway SW26010 260C 1.45GHz, Sunway , NRCPC National Supercomputing Center in Wuxi China	10,649,600	93,014.6	125,435.9	15,371
2	**Tianhe-2 (MilkyWay-2)** - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P , NUDT National Super Computer Center in Guangzhou China	3,120,000	33,862.7	54,902.4	17,808
3	**Piz Daint** - Cray XC50, Xeon E5-2690v3 12C 2.6GHz, Aries interconnect , NVIDIA Tesla P100 , Cray Inc. Swiss National Supercomputing Centre (CSCS) Switzerland	361,760	19,590.0	25,326.3	2,272
4	**Gyoukou** - ZettaScaler-2.2 HPC system, Xeon D-1571 16C 1.3GHz, Infiniband EDR, PEZY-SC2 700Mhz , ExaScaler Japan Agency for Marine-Earth Science and Technology Japan	19,860,000	19,135.8	28,192.0	1,350
5	**Titan** - Cray XK7, Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x , Cray Inc. DOE/SC/Oak Ridge National Laboratory United States	560,640	17,590.0	27,112.5	8,209

REAL speedup

Basic rules for real speedup and APOD

1. Think parallel and design parallel algorithms (CB-GCMC, AI ...)
2. Use the basic rules of optimizations,
3. Profiling (serial code) and parallel code profiling,
4. Micro-optimizations,
5. Minimizing the amount of time spent on memory.

APOD



Acknowledgments

Acknowledgments

- We gratefully acknowledge the financial support provided to the PIVOTS project by the Région Centre - Val de Loire (ARD 2020 program and CPER 2015 -2020) and the French Ministry of Higher Education and Research (CPER 2015 -2020 and public service subsidy to CNRS and Université d'Orléans). Cette opération est cofinancée par l'Union européenne. L'Europe s'engage en région Centre-Val de Loire avec le Fonds Européen de Développement Régional.

- The authors thank **CCSC (Centre de Calcul Scientifique en région Centre - CaSciModOT)** computing center for the achieved computations in the current presentation (`http://cascimodot.fdpoisson.fr/ccsc`).

- The first author gratefully thanks Professor **Suresh Bhatia** form School of Chemical Engineering, University of Queensland, Australia (`http://www.chemeng.uq.edu.au/bhatia`).

Questions?

Bibliography

Allen, M., Tildesley, D., 1989. Computer simulation of liquids. Oxford science publications. Clarendon Press.
URL http://books.google.fr/books?id=O32VXB9e5P4C

Cheng, J., Grossman, M., McKercher, T., 2014. Professional CUDA C Programming. EBL-Schweitzer. Wiley.
URL https://books.google.fr/books?id=q3DvBQAAQBAJ

Chuta, E., September 2013. Modélisation numérique des interactions surface/molécules : Application aux carbones nanoporeux. Master's thesis, Université d'Orléans, CNRS, CRMD, FRE 3520, (in French).

Crowell, A., Brown, J., 1982. Laterally averaged interaction potentials for $^1h_2$ and $^2h_2$ on the (0001) graphite surface. Surface Science 123 (2), 296 – 304.
URL http://www.sciencedirect.com/science/article/pii/0039602882903302

Deublein, S., Eckl, B., Stoll, J., Lishchuk, S. V., Guevara-Carrion, G., Glass, C. W., Merker, T., Bernreuther, M., Hasse, H., Vrabec, J., 2011. ms2: A molecular simulation tool for thermodynamic properties. Computer Physics Communications 182 (11), 2350–2367.
URL https://www-sciencedirect-com.budistant.univ-orleans.fr/science/article/pii/S0010465511002025

Frenkel, D., Smit, B., 2002. Understanding molecular simulation: from algorithms to applications. Computational science. Academic Press.
URL http://books.google.fr/books?id=5qTzldS9ROIC

Gropp, W., Hoefler, T., Lusk, E., Thakur, R., 2014a. Using Advanced MPI: Modern Features of the Message-Passing Interface. Computer science & intelligent systems. MIT Press.
URL https://books.google.co.kr/books?id=Po5IBQAAQBAJ

Gropp, W., Lusk, E., Skjellum, A., 2014b. Using MPI: Portable Parallel Programming with the Message-Passing Interface. Scientific and Engineering Computation. MIT Press.
URL https://books.google.co.kr/books?id=bY9kBQAAQBAJ

Karniadakis, G., Kirby, R., 2003. Parallel scientific computing in C++ and MPI: a seamless approach to parallel algorithms and their implementation. Cambridge University Press.
URL http://books.google.fr/books?id=EnqgXI1AXAQC

Kouetcha, D., Ramézani, H., Cohaut, N., May 2015. Etude structurale et détermination de la fonction de corrélation de paire du graphène et du graphite. In: Colloque Francophone du Carbone GFEC - 2015. Karellis, Savoie, France, (in French).

Kouetcha, D. N., Ramézani, H., Cohaut, N., April 2017. Ultrafast scalable parallel algorithm for the radial distribution function histogramming using MPI maps. The Journal of Supercomputing 73 (4), 1629–1653.
URL http://dx.doi.org/10.1007/s11227-016-1854-0

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E., 1953. Equation of state calculations by fast computing machinesn. J. Chem. Phys 21.
URL https://ssl.cs.dartmouth.edu/~gevorg/89/13W/Metropolis_MC.pdf

Pikunic, J., Clinard, C., Cohaut, N., Gubbins, K. E., Guet, J.-M., Pellenq, R. J.-M., Rannou, I., Rouzaud, J.-N., 2003. Structural modeling of porous carbons:â constrained reverse monte carlo method. Langmuir 19 (20), 8565–8582.
URL http://dx.doi.org/10.1021/la034595y

Ramézani, H., Chuta, E., November 2014. Hydrogen adsorption simulation in the single-wall carbon nanotube (SWNT) network. In: Matériaux 2014. Fédération Française des Matériaux (FFM), Montpellier, France, (in English).

Sanders, J., Kandrot, E., 2010. CUDA by Example: An Introduction to General-Purpose GPU Programming, Portable Documents. Pearson Education.
URL https://books.google.fr/books?id=49OmnOmTEtQC

Snir, M., 1998. MPI the Complete Reference: The MPI Core. MPI the Complete Reference. MIT Press.
URL http://books.google.fr/books?id=x79puJ2YkroC

Stroustrup, B., 2013. The C++ Programming Language: 4th Edition. Always learning. Addison-Wesley.
URL https://books.google.fr/books?id=L6MPBQAAQBAJ

Wang, Q., Johnson, J. K., 1998. Hydrogen adsorption on graphite and in carbon slit pores from path integral simulations. Molecular physics.
URL http://www.physics.mun.ca/~court/kceamngwfgvfar2y.pdf