Communication Papers of the of the 17th Conference on Computer Science and Intelligence Systems pp. 81–84
DOI: 10.15439/2022F224
ISSN 2300-5963 ACSIS, Vol. 32

Performance and Scalability Experiments with a Large-scale Air Pollution Model on the EuroHPC Petascale Supercomputer DISCOVERER

Tzvetan Ostromsky
Institute of Information and Communication Technologies (IICT),
Bulgarian Academy of Sciences (BAS), Sofia, Bulgaria
Email: ceco@parallel.bas.bg

***Abstract*—The basic parallel versions of the Danish Eulerian Model (UNI-DEM) has been implemented on the new petascale supercomputer DISCOVERER, installed last year in Sofia, Bulgaria by the company Atos. DISCOVERER is part of the European High Performance Computing Joint Undertaking (EuroHPC), which is building a network of 8 powerful supercomputers across the European Union (3 pre-exascale and 5 petascale).**

The results of some scalability experiments with the basic MPI and a hybrid MPI-OpenMP parallel implementations of UNI-DEM on the new Bulgarian petascale supercomputer DISCOVERER (in EuroHPC network) are presented here. They are compared with similar earlier experiments performed on the Mare Nostrum III supercomputer (petascale too) at Barcelona Supercomputing Centre – the most powerful supercomputer in Spain by that time, upgraded currently to the pre-exascale Mare Nostrum V, also part of the EuroHPC JU infrastructure.

***Index Terms*—air pollution, numerical model, supercomputer, parallel computing, algorithm, scalability, MPI, OpenMP**

I. Introduction

THE environmental modelling and air pollution modelling in particular is one of the toughest problems of computational mathematics (together with the meteorological modelling). All relevant physical and chemical processes in the atmosphere should be taken into account, which are mathematically represented by a complex PDE system. To simplify it a proper splitting procedure is applied. As a result the initial system is replaced by several simpler systems (submodels), connected with the main physical and chemical processes. These systems should be calculated in a large spatial domain, as the pollutants migrate quickly on long distances, driven by the atmosphere dynamics, especially on high altitude. Here they are exposed to temperature, light and other condition changes in extremely wide range, so does the speed of most chemical reactions. One of the major sources of difficulty is the dynamics of the atmospheric processes, which require small time-step to be used (at least, for the chemistry submodel) in order to get a stable numerical solution of the corresponding system. All this makes the treatment of large-scale air pollution models a tuff and heavy computational task. It has always been a serious challenge, even for the fastest and most powerful state-of-the-art supercomputers.

The first crucial point on the way to this goal is domain decomposition technique. This is a natural way to achieve distributed memory parallelization of any numerical problem over a large spatial domain. For some of them however, like the advection-diffusion equations in our case, there is always certain overhead due to the boundary conditions. Minimizing this overhead is a key point towards efficient optimization. On the other hand, optimization should not restrict the portability of the parallel implementation, as the intensive development in the computer technology inevitably leads to regular updates or complete replacement of the outdated hardware. Standard parallel programming tools as MPI and OpenMP (for distributed / shared memory models) are used in order to preserve portability of the code. Another important parallel optimization issue is the load-balance. MPI barriers, used to force synchronization between the processes in data transfer commands, often do not allow good load-balance. This obstacle can be avoided to some extent by using non-blocking communication routines from the MPI standard library.

II. Description and parallel implementations of the Danish Eulerian Model

DEM is a powerful and sophisticated large scale air pollution model, with some 30-year development history [8], [5], [6], [9]. Over the years it was successfully applied in different long-term environmental studies in various areas (including environment protection, human health care, agricultural production, forestry, wildlife, culture heritage protection, etc.). By processing a huge amount of data (most of it - for the quickly changing meteorological conditions), the model is able to calculate the variable concentrations of a number of pollutants and other chemically active species interacting with them (precursors), over a long time period. Moreover, various accumulative quantities (AOT40, AOT60, etc.) are calculated on yearly basis.

A. Mathematical representation of UNI-DEM

An air pollution model can be described by a system of partial differential equations for calculating the concentrations of a number of pollutants (and other chemical species that interact with the pollutants) in the atmosphere above the

studied geographical region. The main physical and chemical processes (horizontal and vertical wind, diffusion, chemical reactions, emissions and deposition) should be adequately represented in the system. In particular, the Danish Eulerian Model (DEM) [7], [8] is mathematically represented by the following system of partial differential equations:

$$\frac{\partial c_s}{\partial t} = -\frac{\partial(uc_s)}{\partial x} - \frac{\partial(vc_s)}{\partial y} - \frac{\partial(wc_s)}{\partial z} + \quad (1)$$

$$+\frac{\partial}{\partial x}\left(K_x\frac{\partial c_s}{\partial x}\right) + \frac{\partial}{\partial y}\left(K_y\frac{\partial c_s}{\partial y}\right) + \quad (2)$$

$$+\frac{\partial}{\partial z}\left(K_z\frac{\partial c_s}{\partial z}\right) + \quad (3)$$

$$+E_s + Q_s(c_1, c_2, \ldots, c_q) - (k_{1s} + k_{2s})c_s \; ; \quad (4)$$

$$s = 1, 2, \ldots, q \; ; \quad (5)$$

where the following notation is used:

q - number of equations = number of chemical species,
c_s - concentrations of the chemical species considered,
u, v, w - components of the wind along the coordinate axes,
K_x, K_y, K_z - diffusion coefficients,
E_s - emissions in the space domain,
k_{1s}, k_{2s} - coefficients of dry and wet deposition respectively ($s = 1, \ldots, q$),
$Q_s(c_1, c_2, \ldots, c_q)$ - non-linear functions that describe the chemical reactions between the species.

B. Splitting into submodels

The above rather complex system is split into three subsystems (submodels), according to the major physical and chemical processes as well as the numerical methods applied in their solution (marked by different colors in the right-hand-side of the system). These are horizontal advection and diffusion chemistry, emissions and deposition and vertical exchange submodels, respectively. The discretization of the spatial derivatives in these sub-models results in forming three large systems of ordinary differential equations. More details about the numerical methods, applied to solve these systems, can be found in [1], [3], [8].

C. Parallelization strategy

The MPI standard library is used as a main parallelization tool. The MPI (Message Passing Interface) was initially developed as a standard communication library for distributed memory computers. Later, proving to be efficient, portable and easy to use, it became one of the most popular parallelization tools for application programming. Now it can be used on much wider class of parallel systems, including shared-memory computers and clustered systems (each node of the cluster being a separate shared-memory machine). Thus it provides high level of portability of the code.

In the case of DEM, MPI parallelization is based on the space domain partitioning [5], [6]. The space domain is divided into sub-domains (the number of the sub-domains is equal to the number of MPI tasks). Each MPI task works on its own sub-domain. On each time step there is no data dependency between the MPI tasks on both the chemistry and the vertical exchange stages. This is not so with the advection-diffusion stage. Spatial grid partitioning between the MPI tasks requires overlapping of the inner boundaries and exchange of certain boundary values on the neighboring subgrids for proper treatment of the boundary conditions. The subdomains are usually too large to fit into the fastest cache memory of the corresponding CPU. In order to achieve good data locality, the smaller calculation tasks are grouped in chunks (if appropriate) for more efficient cache utilization. An input parameter CHUNKSIZE is provided, which controls the amount of short-term reusable data in order to reduce the transfer between the cache and the main (slower access) memory. It should be tuned with respect to the cache size of the target machine.

More detailed description of the main computational stages of DEM and the parallelization techniques used in each of them can be found in [1], [5], [6], [8].

III. Numerical experiments with UNI-DEM on two petascale supercomputers (IBM MareNostrum III in Barcelona, Spain and Atos DISCOVERER EuroHPC in Sofia, Bulgaria)

Results of scalability experiments with the 2-D fine-resolution grid version of UNI-DEM on two of the most powerful supercomputers in Europe will be shown in Tables I and II in this section.

A. Numerical experiments on the IBM MareNostrum III supercomputer at BSC - Barcelona, Spain

Characteristics of the system IBM MareNostrum III

- 3028 nodes IBM dx360 M4, 16-core, 32 GB RAM per node;
- 48488 cores in total (Intel SandyBridge-EP E5-2670, 2600 MHz);
- Total RAM $>$ 94 TB; Disk storage 1,9 PB;
- Interconnection networks: Infiniband / Gigabit Ethernet;
- Theoretical peak performance $\sim$ 1 PFLOPS.

Some values of the user-defined parameters of UNI-DEM used in the experiments are given below:

- Grid-version: $(480 \times 480 \times 1)$;
- Time period of modelling: 1 year;
- Time step: 90 sec. (both in advection and chemistry stages);
- Cache utilization parameter: `NSIZE` = 32 .

B. Numerical experiments on the EuroHPC JU supercomputer DISCOVERER in Bulgaria

Characteristics of the DISCOVERER supercomputer

- Based on the Atos' platform BullSequana XH2000

TABLE I
TIME (T) IN SECONDS AND SPEED-UP (*Sp*) OF UNI-DEM ON IBM MARENOSTRUM III AT BSC - BARCELONA, SPAIN

Time (T) in seconds and speed-up (*Sp*) of UNI-DEM (MPI basic version) on IBM MareNostrum III								
NP	#	Advection		Chemistry		TOTAL		
(MPI)	NODES	T [s]	(*Sp*)	T [s]	(*Sp*)	T [s]	(*Sp*)	*E* [%]
4	1	33340	(*4.0*)	29050	(*4.0*)	69378	(*4.0*)	100 %
8	1	16927	(*7.9*)	15508	(*7.5*)	38162	(*7.3*)	91 %
16	1	9000	(*14.8*)	7739	(*15.0*)	22072	(*12.6*)	79 %
32	2	4929	(*27.1*)	4120	(*28.2*)	14113	(*19.7*)	61 %
64	4	2569	(*51.9*)	2176	(*53.4*)	9235	(*30.1*)	47 %
96	6	2067	(*64.5*)	1409	(*82.5*)	6985	(*39.7*)	41 %
160	10	1471	(*90.6*)	845	(*137.5*)	5638	(*49.2*)	31 %

TABLE II
TIME (T) IN SECONDS AND SPEED-UP (*Sp*) OF UNI-DEM ON THE EUROHPC JU SUPERCOMPUTER DISCOVERER IN SOFIA, BULGARIA

Time (T) in seconds and speed-up (*Sp*) of UNI-DEM (MPI basic version) on DISCOVERER								
NP	#	Advection		Chemistry		TOTAL		
(MPI)	NODES	T [s]	(*Sp*)	T [s]	(*Sp*)	T [s]	(*Sp*)	*E* [%]
4	1	23408	(*4.0*)	21615	(*4.0*)	48604	(*4.0*)	100 %
8	1	11830	(*7.9*)	11072	(*7.8*)	25045	(*7.8*)	97 %
12	1	7785	(*12.0*)	7112	(*12.2*)	17036	(*11.4*)	95 %
16	1	6023	(*15.5*)	5438	(*15.3*)	13061	(*14.9*)	93 %
24	2	4075	(*23.0*)	3630	(*23.8*)	9148	(*21.3*)	89 %
36	3	2786	(*33.6*)	2314	(*37.4*)	6248	(*31.1*)	86 %
48	4	2216	(*42.3*)	1845	(*46.9*)	4805	(*40.5*)	84 %
60	4	1790	(*52.3*)	1358	(*61.3*)	3638	(*53.4*)	89 %
80	5	1420	(*65.9*)	978	(*85.2*)	3050	(*63.7*)	80 %
96	6	1243	(*75.3*)	824	(*101.1*)	2701	(*72.0*)	75 %
120	8	1072	(*87.3*)	662	(*125.8*)	2394	(*81.2*)	68 %
160	10	895	(*104.6*)	498	(*167.3*)	2052	(*94.7*)	59 %

- CPU type: AMD EPYC 7H12 (code name Rome), frequency 2.6GHz, power consumption 280W
- 12 racks, 376 blades, 1128 nodes, 144384 cores in total
- 128 cores per node, RAM: 128 GB per node
- Total RAM 300 TB; Disk storage 2 PB
- Interconnection: Dragonfly+ with 200Gbps (IB HDR) bandwidth per link
- Sustained performance: 4.518 petaflops
- Theoretical peak performance: 6 petaflops
- TOP500 ranking: # 91 in the world; # 27 in EU (June 2021)

DEM code parallelization and optimization details

- AMD compilers for MPI code (mpifort, mpicc)
- Automatic compiler optimization level: -O3 (mostly)
- Additional parallelism by OpenMP directives of the performance-critical parts of the code, if appropriate (with -fopemp argument for invoking the proper version (extension) of the compiler and linking the necessary libraries)

IV. CONCLUSIONS

- The parallel MPI implementation of DEM is well balanced, portable and proved to run efficiently on some of the most powerful parallel supercomputers in Europe, including the Bulgarian Petascale supercomputer DISCOVERER, part of the EuroHPC JU network.
- The efficiency and speed-up is higher in the computationally-intensive stages. In particular, the chemistry stage (which does not need any communication between the tasks) has almost linear overall speed-up. The advection stage scales pretty well too, taking into account that there is some unavoidable computational overhead due to overlapping boundaries of the partitioning.
- The time for the computationally-intensive stages is additionally reduced in relation with the number of threads in the hybrid MPI-OpenMP code with the OpenMP lower level of parallelism switched on, which can be exploited on core level within a node.
- For the sake of comparability the size of chunks at the chemistry-deposition stage `NSIZE`=32 has been set in all experiments. This might not be the optimal value in all cases. When the number of MPI tasks is large, better results can be expected with a lower value of `NSIZE`, especially when the OpenMP level of parallelism is switched on.

V. PLANS FOR FUTURE WORK

For some environmental applications it is highly desirable to simplify the model as much as possible, keeping the reliability of its output results. A careful sensitivity analysis is needed in order to decide how to do such simplifications. On the other hand, it is important to analyze the influence of variations of

the initial conditions, the boundary conditions, the rates of some chemical reactions, etc. on the model results in order to make right assumptions about the possible simplifications, which could be done. Such analysis can give valuable information about the performance of reliable and reasonable simplifications. It can also help in identifying the accuracy-critical parameters. Thus the sensitivity analysis version of DEM SA-DEM was created [2], [4]. Its complexity is of higher order, a real challenge for the top performance supercomputers nowadays. Its implementation on DISCOVERER and obtaining results - data for further research in various important scientific, social and economic areas is our goal for the future.

ACKNOWLEDGMENTS

This research was supported in part by the Bulgarian NSF project "Efficient methods for modelling, optimization and decision making" (contract # KP-06-N52/5), and by the European High-Performance Computing Joint Undertaking via the Discoverer Fast Track & Benchmark Access project Installing, benchmarking, performance and scalability analysis of the UNI-DEM package for air pollution modelling.

REFERENCES

[1] V. Alexandrov, A. Sameh, Y. Siddique and Z. Zlatev, Numerical integration of chemical ODE problems arising in air pollution models, *Env. Modeling and Assessment*, 2 (1997) 365–377.

[2] I. Dimov, R. Georgieva, S. Ivanovska, Tz. Ostromsky, Z. Zlatev, Studying the sensitivity of pollutants' concentrations caused by variations of chemical rates, *J. Comput. Appl. Math., Vol. 235 (2010), pp. 391–402.*

[3] Ø. Hov, Z. Zlatev, R. Berkowicz, A. Eliassen and L. P. Prahm, Comparison of numerical techniques for use in air pollution models with non-linear chemical reactions, Atmospheric Environment, Vol. 23 (1988), pp. 967–983.

[4] Tz. Ostromsky, I. Dimov, R. Georgieva, Z. Zlatev, Air pollution modelling, sensitivity analysis and parallel implementation, Int. Journal of Environment and Pollution, Vol. 46 (1-2), (2011), pp. 83–96.

[5] Tz. Ostromsky, Z. Zlatev, Parallel Implementation of a Large-scale 3-D Air Pollution Model, in: Large Scale Scientific Computing (S. Margenov, J. Wasniewski, P. Yalamov, Eds.), LNCS-2179, Springer, 2001, pp. 309–316.

[6] Tz. Ostromsky, Z. Zlatev, Flexible Two-level Parallel Implementations of a Large Air Pollution Model, in: Numerical Methods and Applications (I.Dimov, I.Lirkov, S. Margenov, Z. Zlatev - eds.), LNCS-2542, Springer (2002), pp. 545–554.

[7] WEB-site of the Danish Eulerian Model, available at: http://www.dmu.dk/AtmosphericEnvironment/DEM

[8] Z. Zlatev, Computer treatment of large air pollution models, Kluwer (1995).

[9] Z. Zlatev, I. Dimov, Computational and Numerical Challenges in Environmental Modelling, Elsevier, Amsterdam (2006).