SL-AV Model: Numerical Weather Prediction at Extra-Massively Parallel Supercomputer

Mikhail Tolstykh✉[1,2,3], Gordey Goyman[1,2], Rostislav Fadeev[1,2,3],

Vladimir Shashkin[1,2,3] and Sergei Lubov[4]

[1] Marchuk Institute of Numerical Mathematics Russian Academy of Sciences, Moscow, Russia
[2] Hydrometcentre of Russia, Moscow, Russia
[3] Moscow Institute of Physics and Technology, Dolgorpudny, Russia
[4] Main Computer Center of Federal Service for Hydrometeorology and Environmental Monitoring, Moscow, Russia

mtolstykh@mail.ru,
gordeygoyman@gmail.com,
rost.fadeev@gmail.com,
vvshashkin@gmail.com,
s.lubov@meteorf.ru

Abstract. The SL-AV global atmosphere model is used for operational medium-range and long-range forecasts at Hydrometcentre of Russia. The program complex uses the combination of MPI and OpenMP technologies. Currently, a new version of the model with the horizontal resolution about 10 km is being developed. In 2017, preliminary experiments have shown the scalability of the SL-AV model program complex up to 9000 processor cores with the efficiency of about 45 % for grid dimensions of 3024x1513x51. The profiling analysis for these experiments revealed bottlenecks of the code: non-optimal memory access in OpenMP threads in some parts of the code, time losses in the MPI data exchanges in the dynamical core, and the necessity to replace some numerical algorithms. The review of model code improvements targeting the increase of its parallel efficiency is presented. The new code is tested at the new Cray XC40 supercomputer installed at Roshydromet Main Computer Center.

Keywords: Global atmosphere model · Numerical weather prediction· Interannual predictability of atmosphere · Massively parallel computations · Combination of MPI and OpenMP technologies.

1 Introduction

The common ways to improve the quality of numerical weather prediction and fidelity of the atmosphere model 'climate' are the increase of the atmospheric model resolution and advancements in parameterized description of unresolved subgrid-scale processes. Both ways imply the increase in computational complexity of the atmospheric models. Operational numerical weather prediction requires the forecast to be

computed rapidly, usually in less than 10 minutes per forecast day, while the climate modelling requires many multi-year runs to be completed in reasonable time. The resolution of the atmospheric models grows permanently, so these models should be able to use tens of thousands processor cores efficiently. Currently, the typical horizontal resolution of the global medium-range numerical weather prediction models is 9-25 km with about 100 vertical levels [1]. Thus, the approximate number of grid points for these models is about 10^8-10^9. Most numerical weather prediction centers plan to increase the resolution of their models [1]. Many supercomputers of weather services and climate research centers have peak performance about 5-10 Pflops [1] and they are in the first hundred of Top500 list [2]. For example, UK MetOffice and Hadley Climate Centre supercomputer currently has the peak performance of 8.1 Pflops and is at 15th place of Top500 list (as of November 2017).

It is essential that parallel efficiency of an atmospheric model be considered together with its computational efficiency, i.e. ability of the model to compute the forecast of a given accuracy combined with a minimum wall-clock time for a given number of processors. Sophisticated numerical methods in the dynamical core of atmospheric models usually allow longer time steps but scale worse than simple explicit time-stepping algorithms so the balance between complexity of the applied numerical methods and their scalability should be found. Computational efficiency is under permanent evaluation in many world leading weather prediction centers and is an important criterion in selecting their development strategies [3, 4].

SL-AV is the global atmosphere model applied for the operational medium-range weather forecast at Hydrometeorological center of Russia and as a component of the long-range probabilistic forecast system. It is also used in experiments on interannual predictability and is an atmospheric component of the coupled atmosphere-ocean-sea-ice model [5]. SL-AV [6] is the model acronym (semi-Lagrangian, based on Absolute-Vorticity equation). It is developed at Marchuk Institute of Numerical Mathematics, Russian Academy of Sciences (INM RAS) in cooperation with the Hydrometeorological centre of Russia (HMCR). The dynamical core of this model uses the semi-implicit semi-Lagrangian time-integration algorithm [7]. The most part of subgrid-scale processes parameterizations algorithms are developed by ALADIN/LACE consortium [8, 9]; however, the model includes CLIRAD SW [10] and RRTMG LW [11] for parameterization of shorwave and longwave radiation respectively. The multilayer soil model developed at INM RAS [12] is also included. The parallel implementation of SL-AV model uses the combination of one-dimensional MPI decomposition and OpenMP loop parallelization [7]. The model code is also adapted to run at Intel Xeon Phi processors [13]. The code is written in Fortran language and consists of several hundred thousands lines.

The parallel structure of the model code is as follows: one-dimensional MPI decomposition is used along latitude or Fourier-space wave number. MPI-processes perform computations in the bands of grid latitudes during the first phase of the time-step, while OpenMP threads are used to parallelize loops along longitude or vertical coordinate. In the second phase of SL-AV time-step, each MPI-process performs computations for the set of longitude Fourier coefficients from pole to pole, and OpenMP parallelization for loops in vertical is applied.

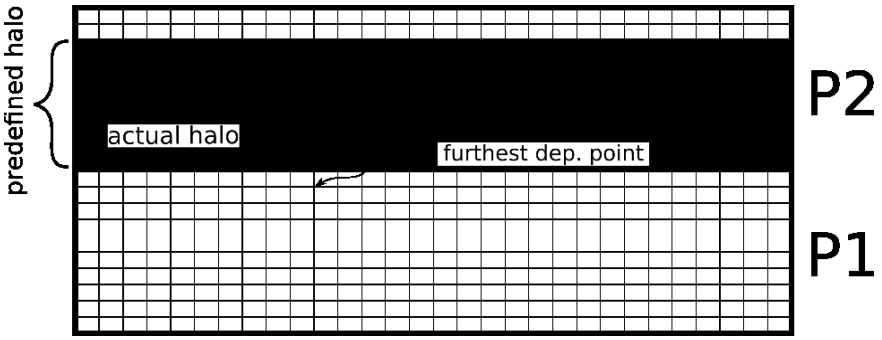
The first tests of the model with grid dimensions 3024x1513x51 showed that it scales up to 9072 cores with an efficiency of about 45%. However, these results were obtained without possibility of profiling, tuning and running the model on a system with such a number of processor cores. In this paper, we present recent works on improving scalability of the SL-AV program complex. Section 2 describes the changes in interprocessor MPI communications, and Section 3 gives an overview of OpenMP optimizations in the model code. We then have tested the modified code at Cray XC40 system using up to 27208 processor cores. The results of these works are presented in Section 4.

2 Parallel Communications Optimization

2.1 Semi-Lagrangian Algorithm Optimization

The semi-Lagrangian advection algorithm [14] consists of two main blocks: the calculation of the backward in time trajectories (position of air particles that arrive to the grid points at the next time step) and the spatial interpolation of advected variables to the departure points of these trajectories. Parallel implementation of this algorithm requires halo exchanges (exchanges of latitudinal bands adjacent to the process boundaries) with the width determined by the position of the furthest departure point and interpolation stencil. The width of the exchanges (in terms of grid point distance) in the original version of the model is calculated using predefined wind speed (estimate of the global maximum wind speed) and the model time step. This a priori estimate is the upper limit for data amount that may be required for calculations. Thus, the semi-Lagrangian advection block in the standard version of the model requires the exchange of values for wind speed components and advected variables of a fixed predefined width. Such an estimate for the region of parallel dependence is rough, and the actual one can be significantly lower, especially in regions with a small wind speed. When using one-dimensional MPI decomposition, the size of messages does not decrease with the increase in the number of computational cores, moreover, the number of neighbor processes increases, which negatively affects the parallel efficiency of the model. To reduce the volume of halo exchanges, another approach has been implemented in this block. First, the calculation of backward trajectories with predefined exchanges size of wind speed components is performed. Knowing the coordinates of the trajectories departure points, the width of exchanges necessary for each processor is computed and data exchanges for the values of advected variables are carried out for their further interpolation. Application of this approach allows to reduce significantly the average size of messages and the number of MPI-processes involved in these exchanges. The schematic of this algorithm is shown in Fig. 1.

4



Pic. 1. Schematic of the semi-Lagrangian advection halo optimization.

2.2 Reducing Number of Global Communications

In the SL-AV model, the fast Fourier transforms are used to convert systems of linear algebraic equations arising from discretization of elliptic problems (Helmholtz equation, hyper diffusion equation, wind velocity reconstruction, see [7] for details) allowing to reduce two-dimensional problems to a set of one-dimensional ones, which are then solved by a direct algorithm. The parallel implementation of this method includes data transpositions, i.e. global redistribution of data between processes. The use of data transpositions limits parallel efficiency and should be avoided whenever possible. Initially, the SL-AV model used four transpositions per time step. The model code modifications and rearrangements have been implemented to reduce this number to two per time step. One can note that the work is underway to introduce new solvers that allows abandoning the use of data transpositions [13].

3 OpenMP Optimizations

3.1 OpenMP Loop Parallelization

Initially, OpenMP technology was used in the SL-AV model to parallelize loops along the same direction as MPI decomposition. This approach limited the maximum number of processor cores used, so most of the code in the model was modified in a way to parallelize the loops along the additional direction, thereby forming a quasi-two-dimensional domain decomposition. However, due to the low computational cost and the laboriousness of code modification, some of the code parts remained unchanged. The available MPI-parallelism is exhausted when the number of processor cores is higher than 1512 (for horizontal grid dimensions of 3024×1513), and then these code sections become sequential in terms of OpenMP parallelization. This loss of parallel efficiency can be noticeable at extra-parallel scales. So the abovementioned modifications of the remaining code sections have been implemented.

3.2 Memory Access Optimization

The block computing right-hand sides of prognostic equations describing parameterized subgrid-scale processes is a significant time-consuming part of the model. Computations in this block are generally carried out in the vertical direction only allowing to perform them independently for different vertical columns of the model. Thus, optimal arrays indices arrangement for this block in terms of loop vectorization and memory access is *(horizontal dimension, vertical dimension)*. At the same time, longitudinal OpenMP parallelization is used in this part of the model. However, the use of OpenMP parallelization along 'fast' first Fortran array index is likely to be inefficient, due to false sharing and bad data localization. To increase the efficiency of OpenMP parallelization and cache memory access, local temporary arrays containing copies of variables necessary for calculations have been introduced for each thread. This modification leads to additional overheads related to data copying but allows to combine effective loop vectorization and OpenMP parallelization.

4 Numerical Experiments

4.1 Model Setup and System Configuration

We have tested two versions of the SL-AV model [6, 7]. Both have the same horizontal resolution of 0.119 degrees (approximately 13 km at the equator), the first one has 51 vertical levels, and the second one has 126 vertical levels. The grid dimensions are 3024×1513×51 and 3024×1513×126 respectively. The version with 51 vertical levels has the same resolution as was used for preliminary tests of the code before the modifications described above.

All the experiments were carried out at the Cray XC40 system [15] installed at Roshydromet. This system consists of 936 nodes having two Intel Xeon E2697v4 18-core CPUs and 128 GB memory. All the nodes are connected with Cray ARIES interconnect. The peak performance is 1.2 PFlops.

4.2 Results

First, we compare the parallel efficiency of the modified SL-AV model code with respect to the previous version using up to 9072 processor cores. The results are shown in Fig.2. Note that the parallel efficiency of the code (the ration of achieved speedup to linear one) has increased by approximately 15 % while using 9072 cores, from 45.5 to 60 %.

6

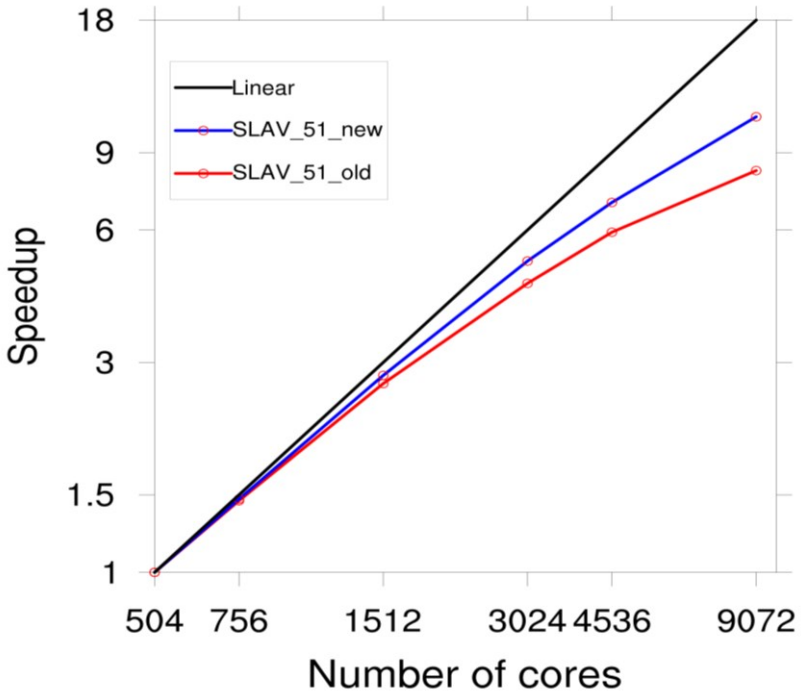


Fig. 2. Strong scalability of the SL-AV code (the version with 51 vertical levels): the version before modifications (red curve), the new version (blue curve), linear speedup (black curve).

We have studied strong scalability of the same code but having 126 vertical levels. The results are presented in Fig.3. We are able to launch the code at 27216 cores, however, there is just 20 % acceleration with respect to 13608 cores. The SL-AV code has the parallel efficiency of about 53 % while running at 13608 processor cores.

We also analyzed the parallel efficiency of different parts of the model with 126 vertical levels as a function of processor cores number; the results are depicted in Fig.4. Fig. 5 demonstrates similar dependencies for percentage of the time step spent in different parts of the model. Here ‘‘dynamics’’ stands for all the computations in the dynamical core except for semi-Lagrangian advection and solvers for Hemholtz equation, wind speed reconstruction and horizontal hyper diffusion [7], ‘‘sub-grid_param’’ denotes parameterizations for all subgrid-scale processes (shortwave and longwave radiation, deep and shallow convection, planetary boundary layer, gravity wave drag, microphysics et al), ‘‘SL_advection’’ refers to the block of semi-Lagrangian advection described in Subsection 2.1., and ‘‘elliptic solver’’ corresponds to the abovementioned solvers in Fourier longitudinal space.

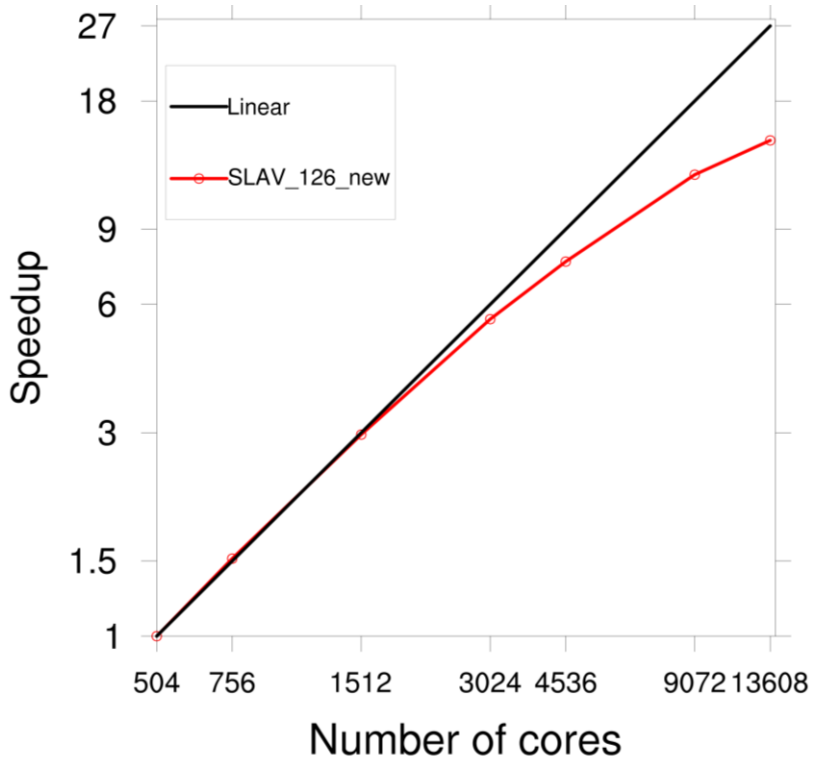


Fig. 3. Strong scalability of the SL-AV code with 126 vertical levels (red curve); linear speedup (black curve).

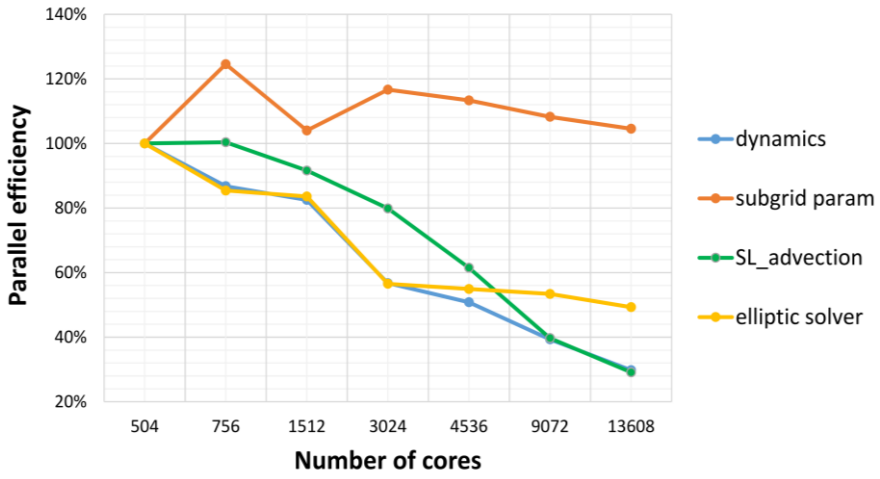


Fig. 4. Parallel efficiency for different parts of the model code as a function of processor core number. See text for details.

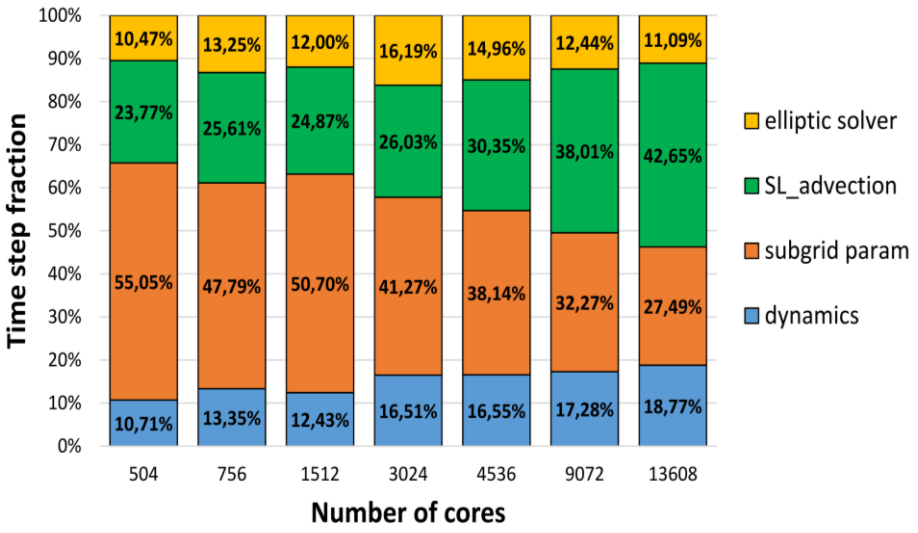


Fig. 5. Percentage of the time step occupied by different parts of the model code as a function of processor cores number.

One can see that parameterizations computations are well parallelized. That could be expected, as most of these computations involve independent computations in vertical columns only. For most of processor cores numbers, the parallel efficiency of this part is higher than for 504 cores. This can be explained by better use of cache memory in these cases. For other parts of the model, the parallel efficiency gradually decays.

The "SL_advection" and "dynamics" blocks seem to be bottlenecks of current model implementation. This is, on one hand, related to the nature of the one-dimensional MPI decomposition as the size of halo exchanges does not decrease when increasing the number of processes. Another reason is limited scalability of OpenMP parallelization. Thus, the best runtime configuration for 504 cores is 6 OpenMP threads and 84 MPI ranks, while the use of 18 threads is optimal when using more than 1512 cores. However, because different data layout in the "dynamics" and semi-Lagrangian advection blocks, the increase of OpenMP threads number from 6 to 18 does not give linear acceleration.

It is shown in [16] that the semi-Lagrangian advection algorithm can be implemented in a way that allows scaling at $O(10^4)$ processor cores. We further plan to implement the algorithm [16].

It also can be seen from the experiments that the data transposition which is the part of ''elliptic solver'' is not the main reason for the decrease in the parallel efficiency of the model for the studied configuration. However, current numerical algorithms used in this part are intrinsically bounded by 1D decomposition that limits

scalability. These algorithms will likely be replaced in future version of the model following [13].

4.3 Conclusions

A modern global atmospheric model should be able to run efficiently at parallel computer systems with tens of thousands processor cores. We have modified MPI exchanges and optimized OpenMP implementation in the program complex of the SL-AV global atmosphere model. These modifications allow to increase parallel efficiency of this code by approximately 15 %, reaching 63 % while using 9072 processor cores. Currently, the code is able to use 13608 cores with the efficiency slightly higher than 50 %, for grid dimensions of 3024×1513×126. Now the model with these dimensions and the time-step of 4 minutes running at 9072 cores would fit the operational time limit of 10 minutes per forecast day. It is also important that the low-resolution version of the model (0.9×0.72 degrees in longitude and latitude respectively, 85 vertical levels) used for interannual predictability experiments computes the atmosphere circulation for 3 model years in less than 20 hours while using 180 processor cores.

The profiling analysis has revealed the parts of the model code that need further improvements in parallel implementation or replacing numerical algorithms. Our next target is the efficient use of 25000 – 35000 cores for the future model version with the horizontal resolution about 10 km (grid dimensions of about 4500×2250×126).

Acknowledgements. This study was carried out at Marchuk Institute of Numerical Mathematics, Russian Academy of Sciences. The study presented in Sections 2 and 3 was supported with the Russian Science Foundation grant No. 14-27-00126P, the work described in Section 4 was supported with the Russian Academy of Sciences Program for Basic Researches No. I.26P.

References

1. WGNE Overview of plans at NWP Centres with Global Forecasting Systems. http://wgne.meteoinfo.ru/nwp-systems-wgne-table/
2. TOP500 Supercomputer sites. https://www.top500.org/
3. Wedi, N.P., Bauer, P., Deconinck, W., Diamantakis, M., Hamrud, M., Kuehnlein, C., Malardel, S., Mogensen, K., Mozdzynski, G., Smolarkiewicz, P.K.: The modelling infrastructure of the Integrated Forecasting System: Recent advances and future challenges. Technical Memorandum 760, ECMWF (2015).
4. Dynamical Core Evaluation Test Report for NOAA's Next Generation Global Prediction System (NGGPS). https://www.weather.gov/media/sti/nggps/NGGPS%20Dycore%20Phase%202%20Test%20Report%20website.pdf
5. Tolstykh, M.A., Geleyn, J.-F., Volodin, E.M., Kostrykin, S.V., Fadeev, R.Y., Shashkin, V.V., Bogoslovskii, N.N., Vilfand, R.M., Kiktev, D.B., Krasjuk, T.V., Mizyak, V.G., Shlyaeva, A.V., Ezau, I.N., Yurova, A.Y.: Development of the multiscale version of the

SL-AV global atmosphere model. Russ. Meteor. and Hydrol. **40**, 374-382 (2015). doi: 10.3103/S1068373915060035

6. Fadeev, R.Yu., Ushakov, K.V., Kalmykov, V.V., Tolstykh, M.A., Ibrayev, R.A.: Coupled atmosphere–ocean model SLAV–INMIO: implementation and first results. Russian J. Num. An. and Math. Mod. **31**, 329-337 (2016), doi: 10.1515/rnam-2016-0031
7. Tolstykh, M., Shashkin, V., Fadeev, R., Goyman, G.: Vorticity-divergence semi-Lagrangian global atmospheric model SL-AV20: dynamical core. Geosci. Model Dev. **10**, 1961-1983 (2017), doi:10.5194/gmd-10-1961-2017.
8. Geleyn, J.-F., Bazile, E., Bougeault, P., Deque, M., Ivanovici, V., Joly, A., Labbe, L., Piedelievre, J.-P., Piriou, J.-M., Royer, J.-F.: Atmospheric parameterization schemes in Meteo-France's ARPEGE N.W.P. model. In: Parameterization of subgrid-scale physical processes, ECMWF Seminar proceedings, pp. 385–402, ECMWF, Reading, UK (1994).
9. Gerard, L., Piriou, J.-M., Brožková, R., Geleyn, J.-F., and Banciu, D.: Cloud and Precipitation Parameterization in a Meso-Gamma-Scale Operational Weather Prediction Model, Mon. Weather Rev. **137**, 3960–3977 (2009). doi:10.1175/2009MWR2750
10. Tarasova, T., Fomin, B.: The Use of New Parameterizations for Gaseous Absorption in the CLIRAD-SW Solar Radiation Code for Models. J. Atmos. and Oceanic Technology **24**, 1157-1162 (2007). doi:10.1175/JTECH2023.1
11. Mlawer, E. J., Taubman, S. J., Brown, P. D.: RRTM, a validated correlated-k model for the longwave. — J. Geophys. Res. **102**, 16663-16682. (1997). doi:10.1029/97JD00237
12. Volodin, E.M., Lykossov, V.N.: Parametrization of Heat and Moisture Transfer in the Soil–Vegetation System for Use in Atmospheric General Circulation Models: 1. Formulation and Simulations Based on Local Observational Data // Izvestiya, Atmospheric and Oceanic Physics **34,** 402-416 (1998).
13. Tolstykh, M., Fadeev, R., Goyman, G., Shashkin, V.: Further Development of the Parallel Program Complex of SL-AV Atmosphere Model. In: Voevodin V., Sobolev S. (eds) Supercomputing. RuSCDays 2017. Communications in Computer and Information Science, vol. 793, pp. 290-298. Springer, Cham (2017). doi: 10.1007/978-3-319-71255-0_23
14. Staniforth, A., Cote, J.: Semi-Lagrangian integration schemes for atmospheric models -- A review, Mon. Weather Rev. **119**, 2206-2233 (1991).
15. Cray XC40 specifications. https://www.cray.com/sites/default/files/resources/cray_xc40_specifications.pdf
16. White III, J., Dongarra, J.: High-performance high-resolution tracer transport on a sphere. J. Comput. Phys. **230**, 6778 – 6799 (2011). doi:10.1016/j.jcp.2011.05.008