DOI: 10.14529/jsfi230405

Analysis and Optimization of Output Operations in the INM RAS Earth System Model

Maria A. Tarasevich[1,2,3], *Ivan V. Tsybulin*[4], *Evgeny M. Volodin*[1], *Andrey S. Gritsun*[1]

The modern development of complex Earth system models forces developers to take into account not only the computational efficiency, but also the performance of the data input and output. This work evaluates the data output performance of the INM RAS Earth system model and optimizes its weak points. The output operations were found to be surprisingly slow on the Cray XC40-LC supercomputer compared to the results obtained on the INM RAS cluster. To identify the bottleneck, the computational time, the distributed data gathering time, and the file system output time were measured separately. The distributed data gathering time was the cause of the slowdown on the Cray XC40-LC, so optimizations were made to the gathering routines without any additional rework of the existing output code. The optimizations resulted in a significant reduction in the overall model running time on the Cray XC40-LC, while the gathering time itself was reduced by a factor of 10^2–10^3. The results highlight the importance of optimizing the output performance in Earth system models.

Keywords: Earth system model, INMCM6, MPI, data gathering, derived types, manual packing, Cray.

Introduction

The horizontal resolution in Earth system models (ESMs) has been increasing for the past two decades. In Coupled Model Intercomparison Project Phase 3 (CMIP3), the typical horizontal resolution was 250 km in the atmosphere and 150 km in the ocean models, while in CMIP6 this increased to 125 km and 75 km respectively [7]. In HighResMIP horizontal resolution is 50 km in the atmosphere model and 25 km in the ocean model, reaching eddy-permitting scale [11]. Between CMIP5 and CMIP6, including HighResMIP, the number of vertical levels in the atmosphere and ocean models nearly doubled [7]. Global models with finer grids capture more aspects of the circulation of the atmosphere with upper stratosphere and ocean, resolving more physical processes explicitly instead of parameterizing them.

Not only the spatial resolution of ESMs increases, but also does the complexity and range of described processes. For the last 4 decades aerosols, carbon cycle, dynamical vegetation, atmospheric chemistry and land ice have extended [16] the typical components of ESMs, which previously were only atmosphere, land, ocean and sea ice. Each ESM component produces more diagnostic information that can be used for applications.

Another modern trend is seamless weather-climate prediction approach [12], which means using one ESM to forecast on different timescales. This seamless approach forces the ESMs to support different frequency of output ranging from daily and monthly (climate scale) to hourly, allowing to capture diurnal cycle and predict extreme precipitation events well.

An increase in grid resolution and complexity of ESMs together with modern application ESMs for weather prediction leads to increase in amount of output data, including model results,

[1]Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences, Moscow, Russian Federation
[2]Hydrometeorological Research Center of Russian Federation, Moscow, Russian Federation
[3]Moscow Institute of Physics and Technology, Dolgoprudny, Russian Federation
[4]Yandex.Technologies, Moscow, Russian Federation

diagnostics, and intermediate variables, which need to be stored. Writing and saving this massive amount of data can degrade the overall performance and scalability of the model [1, 2].

INM RAS Earth system model (INMCM) is a global climate model. Two versions of the INM RAS Earth system model – INMCM5 [27, 28] and INMCM48 [29] – participate in the CMIP6 [9] and show good results [6]. INMCM5 is capable of simulating the present-day climate [27, 28], as well as its changes in 1850–2014 [23]. Moreover, INMCM5 simulates extreme climate and weather phenomena well [14, 19, 30]. Following the seamless approach, new systems of long-range [8, 15, 20, 33–35] and decadal [13, 31, 32] weather forecast based on the INMCM5 have been developed and have successfully completed operational testing at the Hydrometcenter of Russia.

The INM RAS Earth system model is constantly evolving, and the first version of the new model generation called INMCM6 has recently been released [22]. Following the world trends in climate modelling mentioned above, during the further INMCM6 development [18] we intend to supplement the model with new components, such as atmosphere chemistry, land nitrogen cycle, dynamical vegetation, methane cycle, ocean biochemistry. We also consider the version with spatial resolution of 1° in the atmosphere and 0.25° in the ocean as the primary candidate to participate in CMIP7 and a basis for the next version of long range weather forecast system. That means that output operations can become a bottleneck for INMCM too.

In this paper we evaluate the INMCM data output performance on two HPC systems and determine the weak points. We optimize the weak points without redesigning the output system from scratch. After that we give some insights on future evolution of INMCM output subsystem.

The organization of this paper is as follows. Section 1 provides an overview of the INM RAS climate model and the HPC systems used for numerical experiments. Section 2 describes the methodology of measurements and presents the output performance and its scalability. Section 3 explains the changes we made to optimize the output. Section 4 presents the effect of optimizations applied to the INMCM output. Finally, the conclusions summarize the results and highlight the future steps.

1. Materials and Methods

1.1. INM RAS Earth System Model

There is a family of INM RAS Earth system model versions with different spatial and temporal resolutions and different sets of included modules. Basically, INMCM consists of two models: the atmosphere model with interactive aerosol [24] and land surface [25, 26] modules and the ocean model [21] with sea ice dynamics and thermodynamics module [36, 37]. The aerosol module describes the evolution of the 10 substances concentrations. The sea ice dynamics and thermodynamics module describes sea ice with the elastic-viscous-plastic rheology with a single gradation of thickness.

The atmosphere model is based on the system of the hydrothermodynamic equations with hydrostatic approximation in advective form. The atmosphere general circulation model uses a semi-implicit integration scheme that requires solving an auxiliary Helmholtz-type equation at each dynamical step. The current version uses a fast Fourier transform based algorithm and requires global data transposing for its parallel implementation [10].

The ocean model solves a set of large-scale hydrothermodynamic equations with hydrostatic and Boussinesq approximations. The ocean model step consists of several stages. Two most com-

putationally demanding – an isopycnal diffusion and dissipation of the horizontal components of velocity – have recently been optimized [4, 5]. The other one is the barotropic adaptation because it requires solving a system of three implicitly discretized equations for the velocity components and the sea level. This system is solved [21] iteratively using GMRES with the block ILU(0) preconditioner from the PETSc [3] package.

The atmosphere and the ocean general circulation models and the aerosol module are implemented as coupled distributed applications that exchange data using MPI library. The aerosol module works on the same grid as the atmosphere model and uses the same size of MPI communicator.

In the study we use two versions from the up-to-date INM RAS Earth system model generation INMCM6 [22] – INMCM6LM and INMCM6M.

The horizontal resolution of the INMCM6LM atmosphere model is $2^\circ \times 1.5^\circ$ in longitude and latitude. The time step in the atmosphere dynamic core is 3 minutes for this spatial resolution. INMCM6M atmosphere model has the horizontal resolution of $1.25^\circ \times 1^\circ$ in longitude and latitude, and its atmosphere dynamic core does 32 steps per hour resulting in 1.88 minute time step.

The vertical resolution of the atmoshpere model is the same for INMCM6LM and INMCM6M. There are 73 vertical σ-levels with the resolution in the stratosphere about 500 m. The ocean general circulation model is also the same and has a horizontal resolution of $0.5^\circ \times 0.25^\circ$ in longitude and latitude and 40 vertical σ-levels. The time step in the ocean model is 12 minutes.

In the current INMCM implementation the most output is produced by the atmosphere model and is written by the atmosphere root process sequentially. In this paper we focus on atmosphere model output because ocean model output is limited only to monthly averaged fields. The set of atmosphere output fields includes the prognostic variables, the land surface parameters such as soil water and temperature, snow water equivalent, surface and radiative flux components. The output is grouped by both type (2D or 3D) and periodicity of writing. The atmosphere component output properties are summarized in Tab. 1. For INMCM6LM $nlat = 120$, $nlon = 180$, for INMCM6M $nlat = 180$, $nlon = 288$. For both model versions $nplevs = 26$, $nslevs = 73$.

Table 1. Output of INMCM atmosphere component

Group name	Number of fields	Periodicity	Size of one field	Type
dyz	17	1 per 24h	$nplevs \times nlat$	2D
dxy	71	1 per 24h	$nlat \times nlon$	2D
dxyz	7	1 per 24h	$nplevs \times nlat \times nlon$	3D
dxys	10	1 per 24h	$nslevs \times nlat \times nlon$	3D
6h	44	4 per 24h	$nlat \times nlon$	2D
3h	37	8 per 24h	$nlat \times nlon$	2D
1h	2	24 per 24h	$nlat \times nlon$	2D

1.2. HPC Systems

The INMCM output performance was studied using two following high performance computing systems.

The first is a Cray XC40-LC massively parallel supercomputer installed at the Main Computer Center of Federal Service for Hydrometeorology and Environmental Monitoring. The Cray XC40-LC consists of 976 compute nodes interconnected via the Cray's proprietary Aries network. Each node has 128 Gb of RAM and two Intel Xeon E5-2697v4 processors with 18 CPU cores and 45 Mb of Intel Smart Cache per processor. The total number of computational cores is 35136. This supercomputer uses Lustre 3.2 distributed file system.

The second is the INM RAS cluster installed at the Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences. This HPC system has 36 compute nodes connected by a single Mellanox EDR MSB7800 InfiniBand with up to 100 Gb/s bandwidth. Each node has two 20-core Intel Xeon Gold 6230v2 processors with 27.5 Mb of Intel Smart Cache per processor. The total number of available CPU cores is 1440. Each compute node has 256 Gb of RAM. The main storage is a RAID6 drive shared with computing nodes using NFS4 via 1Gb/s network.

The main differences between the HPC systems used are the number and performance of CPU cores (newer cores on the INM RAS cluster), the interconnect used and the file system type. We use slightly different versions of Intel compilers to compile and build INMCM: 19.1.2.254 on the Cray XC40-LC and 19.1.3.304 on the INM RAS cluster. The compilation optimization flags are the same on both HPC systems, `-O3` for the atmosphere model and aerosol module and `-O2` for the ocean model. MPI library implementations were different across the systems: Intel MPI 2019.9.304 was used on the INM RAS cluster, and Cray MPICH 7.7.16 was used on the Cray XC40-LC.

2. Analysis of INM RAS Earth System Model Output Performance

In [17] we evaluated the INMCM6M performance on the same two HPC systems and found its scalability as satisfying. However, these measurements were carried out with only monthly data output enabled. When the model was operated on Cray XC40-LC in production mode we observed huge slowdown unless the output is disabled. This fact drew our attention to the INM RAS Earth system model output performance.

Table 2. List of used INMCM configurations

MPI processes	INMCM6M		MPI processes	INMCM6LM
ATM	Cray XC40-LC	INM RAS	ATM	Cray XC40-LC
84	✓	✓	36	✓
120	✓	✓	60	✓
144	✓	✓	72	✓
180	✓	✓	84	✓
240	✓	✓	120	✓
288	✓	✓	240	✓
312		✓	270	✓
360	✓	✓		
432	✓			
540	✓			
720	✓			

We carried out a range of time measuring experiments for INMCM6M and INMCM6LM with full output of atmosphere component enabled. All INMCM simulations are performed under the conditions of the CMIP6 piControl scenario [9]. All forcings are fixed at conditions of the year of 1850. All simulations last for 1 model month. The configurations of the experiments are summarized in Tab. 2. All configurations are chosen so that the running time is determined by the atmosphere model [17].

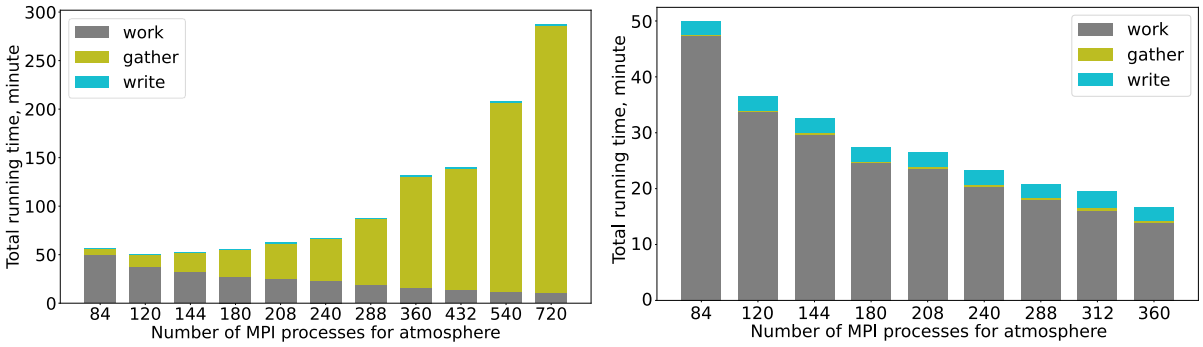
To analyze the output performance we added special tracing calls to all output subroutines. Each call appends current `MPI_Wtime` along with a small text message to an in-memory log. Since all output is done on the root MPI process, all time measurements were also done only on the root process.

At the end of the program the tracing log is flushed to a file, minimizing runtime tracing overhead. In any case, the overhead is not significant since we add tracing calls manually and only for those subroutines that are involved in producing output. When the tracing log file is produced by the program, an extra postprocessing step is necessary to extract all timings from the log and aggregate them by specific code regions.

After the initial code analysis we have divided the whole atmosphere code flow into three major parts: `work` – the computation process itself; `gather` – collecting the output data on the root process; `write` – writing the collected data to the file system.

2.1. INMCM6M

The results shown in Fig. 1 for INM RAS cluster and Cray XC40-LC are quite different. On the INM RAS cluster the running time is determined by the `work` and `write` part with the `gather` part being negligible. The `work` time reduces when the number of used MPI processes increases. The `write` part remains almost constant, which agrees with the fact that all output is done by a single process. The `gather` time increases, but remains under 3% of the total running time.



(a) On the Cray XC40-LC (b) On the INM RAS cluster

Figure 1. INMCM6M running wall time against the number of MPI processes for the atmosphere component

In contrast, on the Cray XC40-LC the `gather` time grows very quickly, preventing any speedup when the number of processes increases. Starting from 180 atmosphere processes configuration, the model spends most of its running time gathering output data.

The computational times (`work`) on both systems are in good agreement. The output writing on the INM RAS cluster is slower by 2.5–3 times, which can be explained by using a high performance distributed file system on Cray XC40-LC.

Figure 2 shows the distribution of time spent in gathering different output field types on a timeline. Most of the time is consumed by gathering daily 3D output on σ-levels. All gathering operations except for the `dyz` and `1h` output data are taking considerable amount of time compared to the actual computation time.

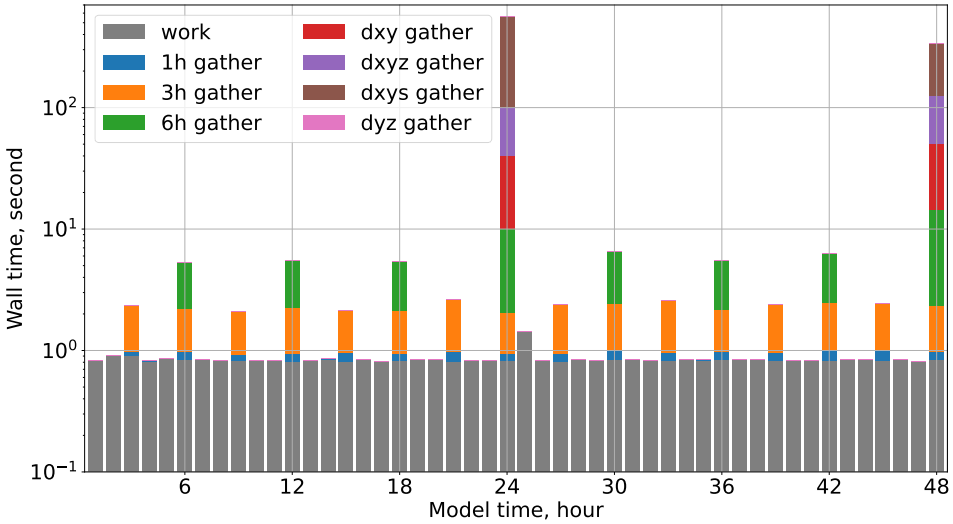
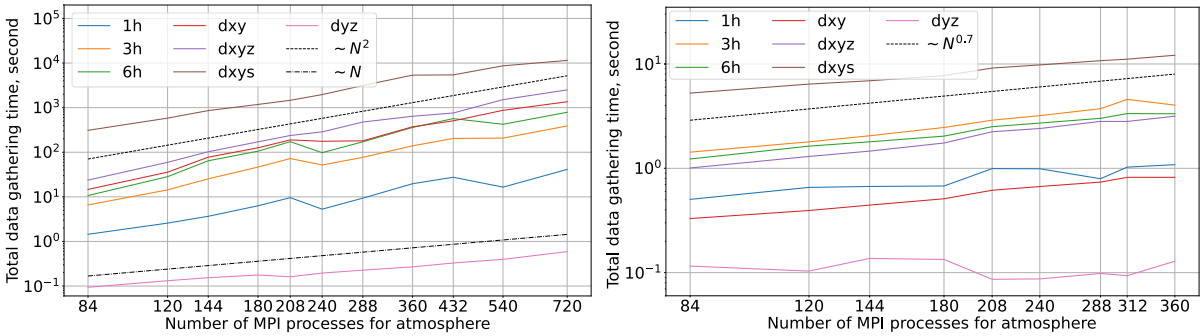


Figure 2. INMCM6M wall time distribution on the Cray XC40-LC for each model hour using 720 MPI processes for the atmosphere component. The wall time axis is in logarithmic scale

Figure 3 demonstrates that for the Cray XC40-LC the gathering time scales as the square of the number of involved MPI processes, while on the INM RAS cluster it remains sublinear. For all output except for the `dyz` the gathering is done from all of the processes, but for `dyz` output it is done only along one meridian. Gathering of `dyz` is fast enough, so we will not consider it further.

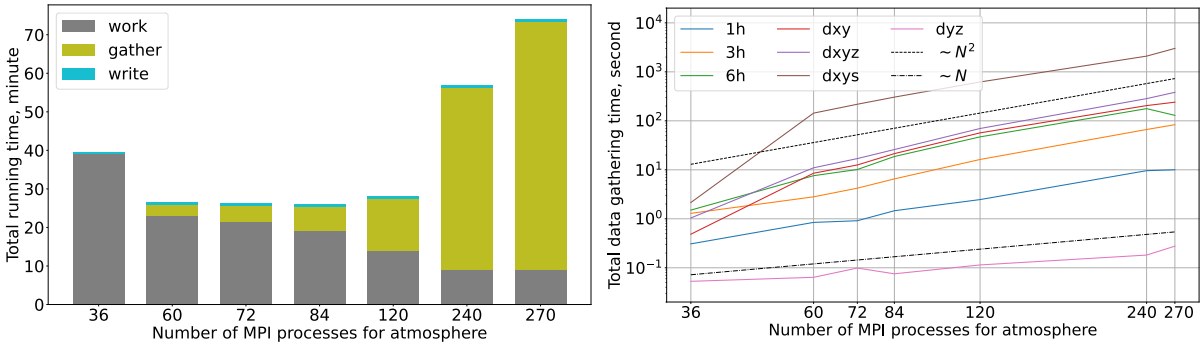


(a) On the Cray XC40-LC (b) On the INM RAS cluster

Figure 3. INMCM6M gathering time for different types of fields against the number of MPI processes for the atmosphere component

2.2. INMCM6LM

We have also studied how output affects the running time of the INMCM6LM, which has coarser horizontal resolution than INMCM6M. Its performance on Cray XC40-LC is similar to INMCM6M and is presented in Fig. 4. The gathering time is negligible when all processes share the same computational node (the first configuration) and grows quickly when the atmosphere component uses several nodes.



(a) Total running time

(b) Total gathering time

Figure 4. INMCM6LM running wall time and gathering time against the number of MPI processes for the atmosphere component

2.3. Details of Model Output

INM RAS Earth system model uses arrays that are distributed along the longitude and the latitude directions. Each process allocates a part of the array including halo zone for exchanges with neighbors. The root process, which has rank 0, allocates the array as full, but uses only a part of it, except for the output. When an output of a distributed array is needed, the array is gathered on the root process.

All output routines consist of gathering data and writing it to the file system. The outline of their implementation is given in Listing 1. For 3D arrays the subroutine Gather2D is called inside a loop over vertical coordinate.

Algorithm 1 Output a distributed field

```
procedure Output2D(A, record)
    Gather2D(A)
    if myrank = 0 then
        f ← OpenFile(...)
        WriteFile(f, record, A)
        CloseFile(f)
    end if
end procedure
```

```
procedure Output3D(A, record)
    for k = 1, ..., n_k do
        Gather2D(A(:, :, k))
    end for
    if myrank = 0 then
        f ← OpenFile(...)
        WriteFile(f, record, A)
        CloseFile(f)
    end if
end procedure
```

Gathering of a 2D distributed array is done in a straightforward linear algorithm given in Listing 2. For clarity the MPI_Type_commit and MPI_Type_free calls are omitted.

There is no clear understanding why such implementation of Gather2D leads to $O(nproc^2)$ time complexity observed on Cray XC40-LC. One possible explanation is: the 2D arrays are quite small, even for high resolution version the whole 2D array is only 200 kB and each process owns and sends less than 1 kB. All these messages are sent without blocking the sending part of the communication and are quickly delivered to the root process receiving queue, flooding it with $O(nproc)$ messages. After that the root process needs to repeatedly scan the queue looking for a message from a certain rank, resulting in $O(nproc^2)$ complexity.

Algorithm 2 Gathering of a distributed 2D array

```
procedure GATHER2D(A)                                   ▷ A – a distributed 2D array
Require: A(i_b : i_e, j_b : j_e) – elements owned by the current process
    if myrank = 0 then
        for rank = 1, ..., nprocs − 1 do
            i'_b, i'_e, j'_b, j'_e ← DOMAIN(rank)                ▷ elements owned by rank
            type ←MPI_TYPE_VECTOR(A(i'_b : i'_e, j'_b : j'_e))
            MPI_RECV(A(i'_b, j'_b), 1, type, rank, tag, comm)
        end for
    else
        type ←MPI_TYPE_VECTOR(A(i_b : i_e, j_b : j_e))
        MPI_SEND(A(i_b, j_b), 1, type, 0, tag, comm)
    end if
end procedure
```

3. Output Optimization

Previously we have shown that output time is dominated by distributed arrays gathering time. Therefore it is sufficient to optimize the gathering routines without any additional rework of the existing output code.

3.1. Gathering 3D Fields

The original implementation of 3D output is inefficient: it repeatedly gathers rather small 2D arrays. So we made a special subroutine for gathering 3D arrays that transfers whole 3D parts instead of slicing them. We introduced an extra 3D derived MPI type that combines all 2D slices into a single datatype. The implementation is given in Listing 3.

Algorithm 3 Gathering of a distributed 3D array

```
procedure GATHER3D(A)                                   ▷ A – a distributed 3D array
Require: A(i_b : i_e, j_b : j_e, 1 : n_k) – elements owned by the current process
    if myrank = 0 then
        for rank = 1, ..., nprocs − 1 do
            i'_b, i'_e, j'_b, j'_e ← DOMAIN(rank)                ▷ elements owned by rank
            slice ←MPI_TYPE_VECTOR(A(i'_b : i'_e, j'_b : j'_e))
            type ←MPI_TYPE_HVECTOR(n_k, 1, sizeof(A(:, :, 1)), slice)
            MPI_RECV(A(i'_b, j'_b, 1), 1, type, 0, tag, comm)
        end for
    else
        slice ←MPI_TYPE_VECTOR(A(i_b : i_e, j_b : j_e))
        type ←MPI_TYPE_HVECTOR(n_k, 1, sizeof(A(:, :, 1)), slice)
        MPI_SEND(A(i_b, j_b), 1, type, 0, tag, comm)
    end if
end procedure
```

3.2. Gathering 2D Fields

The MPI library has builtin collective call MPI_GATHERV that is capable of gathering a 1D array from blocks of unequal size. However, one cannot directly use it with partitioned 2D array, since its blocks are non-contiguous in memory.

To make use of MPI_GATHERV, we need to reshape 2D subarrays into a contiguous memory block, gather them into an auxiliary buffer and then unpack it into the 2D array on the root process. This buffer and extra arguments for MPI_GATHERV can be initialized once and reused on invocations. The implementation is outlined in Listing 4.

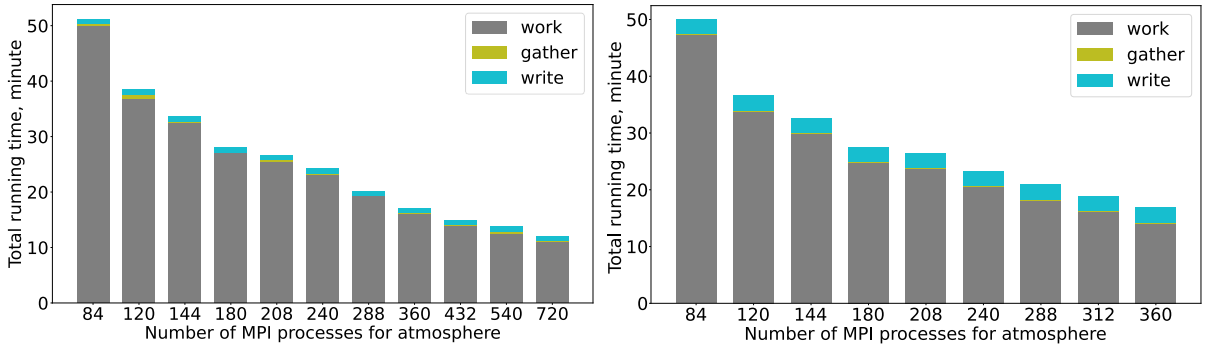
Algorithm 4 Optimized gathering of a distributed 2D array

procedure GATHER2D_OPT(A) ▷ A — a distributed 2D array
Require: $A(i_b : i_e, j_b : j_e)$ — elements owned by the current process
Require: allocated $recvbuf, sendbuf$, filled $recvcnts, displs$
  if $myrank = 0$ **then**
    MPI_GATHERV(MPI_IN_PLACE, 0, $recvbuf, recvcnts, displs, tag, comm$)
    for $rank = 1, \ldots, nprocs - 1$ **do**
      $i'_b, i'_e, j'_b, j'_e \leftarrow$ DOMAIN($rank$) ▷ elements owned by $rank$
      $A(i'_b : i'_e, j'_b : j'_e) \leftarrow$ UNPACK2D($rank, recvbuf$)
    end for
  else
    $sendbuf \leftarrow$ PACK2D($myrank$, $A(i_b : i_e, j_b : j_e)$)
    MPI_GATHERV($sendbuf, sendcount, \cdots, tag, comm$)
  end if
end procedure

The PACK2D and UNPACK2D are auxiliary routines that convert between 2D and 1D in native Fortran column-major order.

4. Results

4.1. INMCM6M



(a) On the Cray XC40-LC (b) On the INM RAS cluster

Figure 5. INMCM6M running wall time against the number of MPI processes for the atmosphere component after output data gathering was optimized

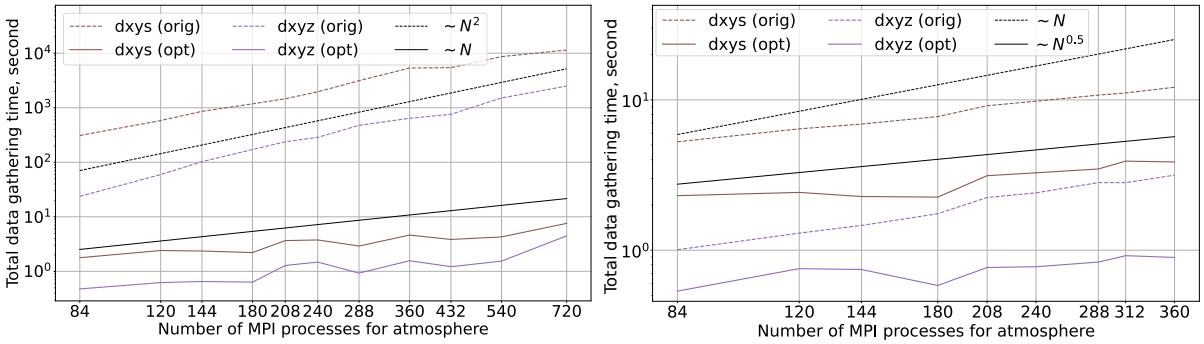
After optimizing gathering of both 3D and 2D arrays the total model running time on Cray XC40-LC reduced significantly, see Fig. 5. Gathering data is not a bottleneck anymore and output is bound by writing time now.

Let us discuss how different optimizations affect the gathering time for each type of the output.

4.1.1. Optimization of 3D output gathering

Optimization of 3D output was done by replacing looped Gather2D by Gather3D version, as shown in Listing 3. After this optimization the total gathering time of 3D fields reduced. Figure 6 demonstrates that on Cray XC40-LC the reduction was nearly 10^3 times, while on the INM RAS cluster it was only about 3–4 times. Total 3D gathering time was reduced below 10 seconds on both systems and does not exceed 2 % of the total running time.

It is worth noting that not only the gathering time was reduced, but also its asymptotic behavior changed: for Cray XC40-LC it dropped from $O(nproc^2)$ to sublinear and for INM RAS cluster it dropped from approximately linear to approximately $O(\sqrt{nproc})$.



(a) On the Cray XC40-LC (b) On the INM RAS cluster

Figure 6. INMCM6M optimized 3D output gathering time against the number of MPI processes for the atmosphere component

4.1.2. Optimization of 2D output gathering

The timings for 2D output gathering are presented in Fig. 7 and Fig. 8. For Cray XC40-LC the Gather2D_Opt which is based on MPI_Gatherv demonstrates outstanding gathering time drop for more than two orders of magnitude, reducing total 2D gathering time below 10 seconds. For INM RAS cluster there also is some improvement, but not so remarkable.

The optimized version of 2D gathering code has approximately equal performance both on Cray XC40-LC and INM RAS cluster, with the latter being a little bit faster, probably due to its smaller size and simpler interconnect topology.

The MPI_Gatherv based optimization of Gather2D was the first we have tried. Since MPI_Gatherv cannot be used with heterogeneous derived types, this implementation internally relies on manual packing. To study which part of the implementation is responsible for the speedup, we considered two additional implementations:

- Gather2D nonblocking – similar to Gather2D, but using MPI_Irecv and MPI_Waitall instead of sequential MPI_Recv;

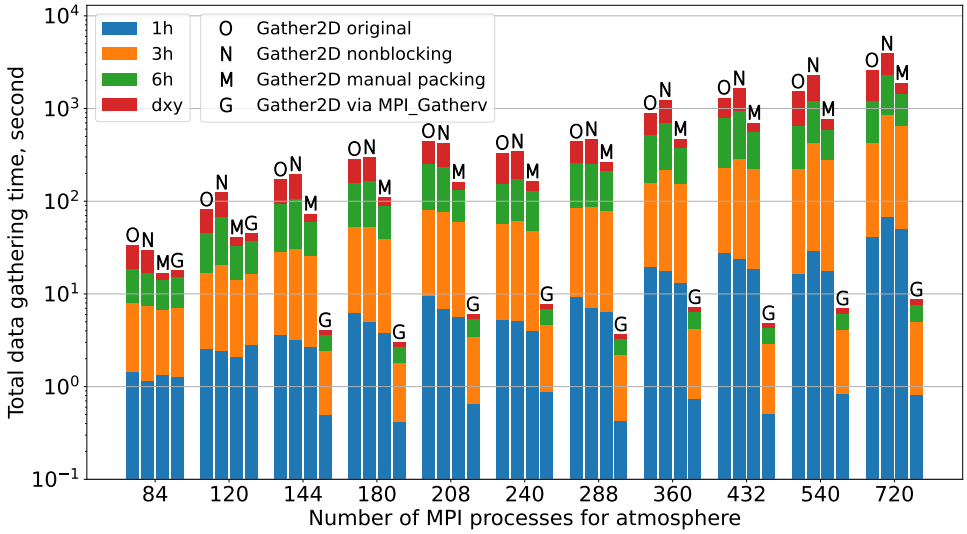


Figure 7. INMCM6M 2D output gathering time against the number of MPI processes for the atmosphere component on Cray XC40-LC. Note, that total time axis is in logarithmic scale

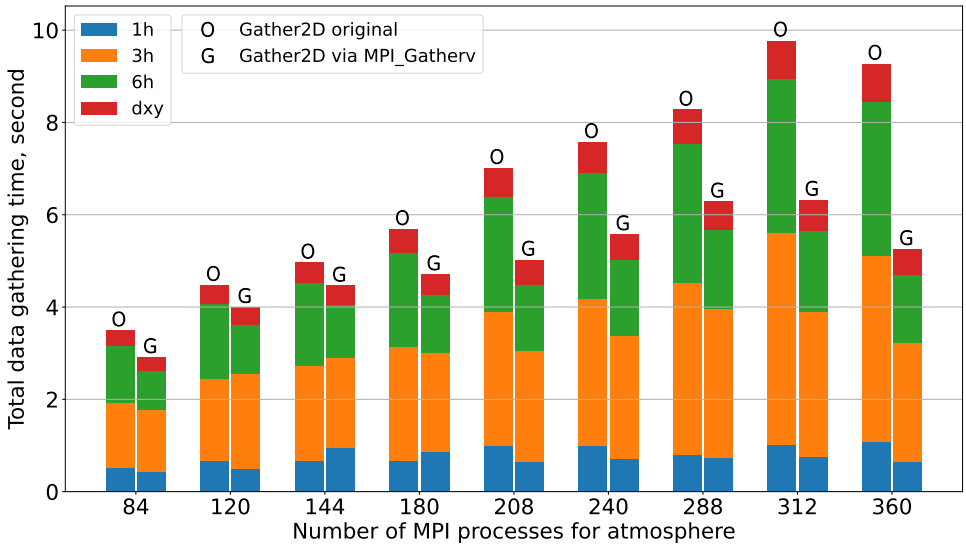


Figure 8. INMCM6M 2D output gathering time against the number of MPI processes for the atmosphere component on INM RAS cluster

- Gather2D manual packing – similar to Gather2D_Opt except that it uses MPI_Send, MPI_Irecv and MPI_Waitall instead of MPI_Gatherv and uses the same manual packing and unpacking routines Pack2D, Unpack2D instead of derived MPI types.

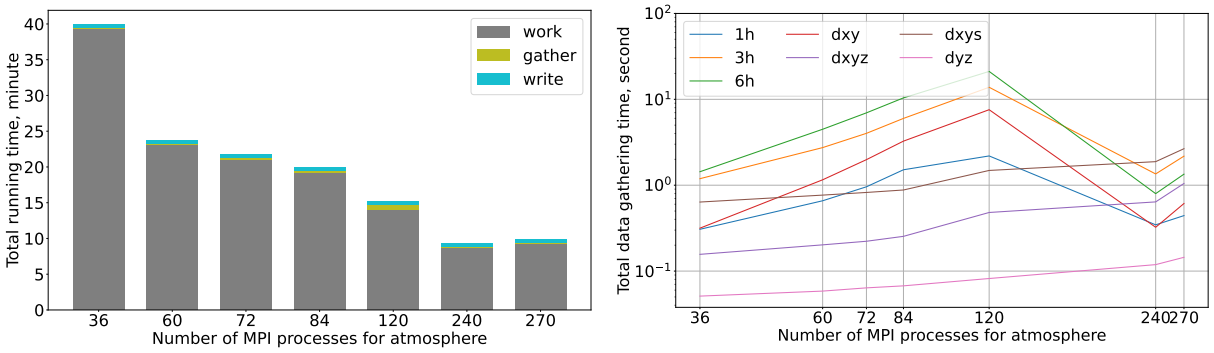
The idea behind Gather2D nonblocking is that messages could be processed in arriving order and not in rank order as in Gather2D. However, this implementation showed itself even worse when the number of processes was large and gave only minuscule improvement for small number of processes.

The Gather2D manual packing allows to eliminate any possible overhead associated with derived MPI types, for example, committing and freeing them each time we send or receive a message, or reallocating additional buffers in MPI library implementation. This version indeed showed some speedup, especially for `dxy` output, but still was two orders of magnitude slower than MPI_Gatherv.

We believe that outstanding MPI_Gatherv performance is explained by using topology-aware gathering strategy. While in INM RAS cluster all nodes are connected via a single InfiniBand commutator, the Cray XC40-LC has complex interconnect architecture arising from its size. Hence, topology-aware algorithm does not significantly improve gathering performance on INM RAS cluster, but is crucial for large HPC systems like Cray XC40-LC.

4.2. INMCM6LM

The results for INMCM6LM are consistent with INMCM6M except for the relative fractions of time spent for `work`, `gather` and `write` parts. Figure 9 shows the effect of output gathering optimization in INMCM6LM.



(a) Total running time

(b) Total gathering time

Figure 9. INMCM6LM running wall time and gathering time against the number of MPI processes for the atmosphere component after optimization

For the INMCM6LM version the fraction of time spent in `write` is greater that for the INMCM6M version. For example, when using 240 MPI processes in atmosphere with optimized gathers on Cray XC40-LC, INMCM6M spends 3.8 % of wall time in `write` while INMCM6LM spends 5.2 %. And for INM RAS cluster `write` takes three times longer due to less efficient file system. Therefore optimizing `write` performance is more important for INMCM6LM than for INMCM6M and for INM RAS cluster than for Cray XC40-LC.

Conclusion

Investigating the reasons behind INMCM6M slowdown on Cray XC40-LC with enabled daily data output we found a problem in distributed array gathering implementation. The naive algorithm that worked almost flawlessly for years on different HPC systems became a serious bottleneck when the model was ported to Cray XC40-LC.

For 3D data it was sufficient to replace repeated 2D gatherings with an implementation that sends all vertical levels at once. For 2D data the only way to overcome the bottleneck was to adapt an appropriate collective routine MPI_Gatherv from Cray's MPI library. We believe the huge performance difference is explained by a topology-aware algorithm used in the MPI library implementation. This leads to a conclusion that existing library collective operations should be preferred over the custom ones. The library versions guarantee some uniform performance across different HPC systems, while custom implementations might suddenly break after porting to a new system.

After the problem with data gathering is fixed, the next output bottleneck is serial data writing itself. Our future work is focused on switching to parallel data input and output. And though this is a reasonable thing to do on a distributed file system like Lustre on Cray XC40-LC, it is hard to tell in advance whether it would speed up output on HPC systems with a shared network drive, like INM RAS cluster.

Acknowledgements

The research was carried out at the Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences. The analysis of INMCM6M output performance and output optimization presented in Sections 2.1, 3 and 4.1 was supported by the Russian Federation research and technical development program in ecological strategy and climate change through grant FFMG-2023-0001 "Development of an extended version of the INM RAS Earth system model within a new computational framework". All results for the INMCM6LM described in Sections 2.2 and 4.2 were obtained with the support of the Russian Science Foundation, Project 20-17-00190. All computations were performed using the HPC system of the Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences and Cray XC40-LC HPC system at the MCC of Roshydromet. Preliminary analysis of INMCM output performance was carried out using MVS1Q1 at the Joint SuperComputer Center of the Russian Academy of Sciences.

References

1. Acosta, M.C., Palomas, S., Paronuzzi Ticco, S.V., *et al.*: The computational and energy cost of simulation and storage for climate science: lessons from CMIP6. Geoscientific Model Development Discussions 2023, 1–21 (2023). https://doi.org/10.5194/gmd-2023-188

2. Balaji, V., Maisonnave, E., Zadeh, N., *et al.*: CPMIP: measurements of real computational performance of Earth system models in CMIP6. Geoscientific Model Development 10(1), 19–34 (2017). https://doi.org/10.5194/gmd-10-19-2017

3. Balay, S., Gropp, W.D., McInnes, L.C., *et al.*: Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries. In: Arge, E., Bruaset, A.M., Langtangen, H.P. (eds.) Modern Software Tools for Scientific Computing, pp. 163–202. Birkhäuser Boston, Boston, MA (1997). https://doi.org/10.1007/978-1-4612-1986-6_8

4. Blagodatskikh, D.V.: Comparison of computational efficiency of two versions of a terrain-following ocean climate model. Numerical Methods and Programming 24, 440–449 (2023). https://doi.org/10.26089/NumMet.v24r430

5. Blagodatskikh, D.V., Iakovlev, N.G., Volodin, E.M., *et al.*: Non-local discretization of the isoneutral diffusion operator in a terrain-following climate ocean model. Russian Journal of Numerical Analysis and Mathematical Modelling 38(6), 1–8 (2023). https://doi.org/10.1515/rnam-2023-0026

6. Bock, L., Lauer, A., Schlund, M., *et al.*: Quantifying Progress Across Different CMIP Phases With the ESMValTool. Journal of Geophysical Research: Atmospheres 125(21), e2019JD032321 (2020). https://doi.org/10.1029/2019JD032321

7. Chen, D., Rojas, M., Samset, B.H., *et al.*: Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change. In: Masson-Delmotte, V., Zhai, P., Pirani, A., *et al.* (eds.) Climate Change 2021: The Physical Science Basis, pp. 147–286. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA (2021). https://doi.org/10.1017/9781009157896.003

8. Emelina, S.V., Khan, V.M., Semenov, V.A., *et al.*: Seasonal Hydrodynamic Forecasts Using the INM-CM5 Model for Estimating the Beginning of Birch Pollen Dispersion. Izv. Atmos. Ocean. Phys. 59, 351359 (2023). https://doi.org/10.1134/S0001433823040059

9. Eyring, V., Bony, S., Meehl, G.A., *et al.*: Overview of the Coupled Model Intercomparison Project Phase 6 (CMIP6) experimental design and organization. Geoscientific Model Development 9(5), 1937–1958 (2016). https://doi.org/10.5194/gmd-9-1937-2016

10. Gloukhov, V.: Parallel implementation of the INM atmospheric general circulation model on distributed memory multiprocessors. In: Sloot, P.M.A., Hoekstra, A.G., Tan, C.J.K., Dongarra, J.J. (eds.) Computational Science – ICCS 2002. Lecture Notes in Computer Science, vol. 2329, pp. 753–762. Springer Berlin, Heidelberg (2002). https://doi.org/10.1007/3-540-46043-8

11. Haarsma, R.J., Roberts, M.J., Vidale, P.L., *et al.*: High Resolution Model Intercomparison Project (HighResMIP v1.0) for CMIP6. Geoscientific Model Development 9(11), 4185–4208 (2016). https://doi.org/10.5194/gmd-9-4185-2016

12. Hoskins, B.: The potential for skill across the range of the seamless weather-climate prediction problem: a stimulus for our science. Quarterly Journal of the Royal Meteorological Society 139(672), 573–584 (2013). https://doi.org/10.1002/qj.1991

13. Khan, V.M., Vilfand, R., Tishchenko, V., *et al.*: Assessment of Changes in the Temperature Regime of Northern Eurasia for the Next Five Years According to the INM RAS Earth System Model Forecasts and Their Possible Consequences for Agriculture. Russ. Meteorol. Hydrol. 48, 745754 (2023). https://doi.org/10.3103/S1068373923090029

14. Kim, Y.H., Min, S.K., Zhang, X., *et al.*: Evaluation of the CMIP6 multi-model ensemble for climate extreme indices. Weather and Climate Extremes 29, 100269 (2020). https://doi.org/10.1016/j.wace.2020.100269

15. Kulikova, I.A., Nabokova, E.V., Khan, V.M., *et al.*: MaddenJulian Oscillation in the Context of Subseasonal Variability, Teleconnections, and Predictability. Russ. Meteorol. Hydrol. 48, 645657 (2023). https://doi.org/10.3103/S1068373923080010

16. Stocker, T.F., Qin, D., Plattner, G.K., *et al.* (eds.): IPCC, 2013: Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, p. 1535. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA (2013), https://www.ipcc.ch/site/assets/uploads/2018/02/WG1AR5_all_final.pdf

17. Tarasevich, M., Sakhno, A., Blagodatskikh, D., *et al.*: Scalability oftheINM RAS Earth System Model. In: Voevodin, V., Sobolev, S., Yakobovskiy, M., *et al.* (eds.) Supercomputing. pp. 202–216. Lecture Notes in Computer Science, Springer Nature Switzerland (2023). https://doi.org/10.1007/978-3-031-49432-1_16

18. Tarasevich, M.A., Tsybulin, I.V., Onoprienko, V.A., *et al.*: Ensemble-based statistical verification of INM RAS Earth system model. Russian Journal of Numerical Analysis and Mathematical Modelling 38(3), 173–186 (2023). https://doi.org/10.1515/rnam-2023-0014

19. Tarasevich, M.A., Volodin, E.M.: Influence of various parameters of INM RAS climate model on the results of extreme precipitation simulation. In: International Young Scientists School and Conference on Computational Information Technologies for Environmental Sciences, May 27 – June 6, 2019. IOP Conference Series: Earth and Environmental Science, vol. 386, p. 012012. IOP Publishing (2019). https://doi.org/10.1088/1755-1315/386/1/012012

20. Tarasevich, M.A., Volodin, E.M.: The Influence of Autumn Eurasian Snow Cover on the Atmospheric Dynamics Anomalies during the Next Winter in INMCM5 Model Data. Supercomputing Frontiers and Innovations 8(4), 24–39 (2021). https://doi.org/10.14529/jsfi210403

21. Terekhov, K.M., Volodin, E.M., Gusev, A.V.: Methods and efficiency estimation of parallel implementation of the σ-model of general ocean circulation. Russ. J. Numer. Anal. Math. Modelling 26(2), 189–208 (2011). https://doi.org/10.1515/rjnamm.2011.011

22. Volodin, E.M.: Simulation of Present-Day Climate with the INMCM60 Model. Izvestiya, Atmospheric and Oceanic Physics 59(1), 16–22 (2023). https://doi.org/10.1134/S0001433823010139

23. Volodin, E.M., Gritsun, A.S.: Simulation of observed climate changes in 1850–2014 with climate model INM-CM5. Earth System Dynamics 9(4), 1235–1242 (2018). https://doi.org/10.5194/esd-9-1235-2018

24. Volodin, E.M., Kostrykin, S.V.: The aerosol module in the INM RAS climate model. Russian Meteorology and Hydrology 41(8), 519–528 (2016). https://doi.org/10.3103/S106837391608001X

25. Volodin, E.M., Lykosov, 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(4), 405–416 (1998), https://www.researchgate.net/publication/270586916_Parameterization_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

26. Volodin, E.M., Lykosov, V.N.: Parametrization of Heat and Moisture Transfer in the Soil-Vegetation System for Use in Atmospheric General Circulation Models: 2. Numerical Experiments in Climate Modeling. Izvestiya, Atmospheric and Oceanic Physics 34(5), 559–569 (1998), https://www.researchgate.net/publication/270586932_Parameterization_of_Heat_and_Moisture_Transfer_in_the_Soil-Vegetation_System_for_Use_in_

Atmospheric_General_Circulation_Models_2_Numerical_Experiments_in_Climate_Modeling

27. Volodin, E.M., Mortikov, E.V., Kostrykin, S.V., *et al.*: Simulation of modern climate with the new version of the INM RAS climate model. Izvestiya, Atmospheric and Oceanic Physics 53(2), 142–155 (2017). https://doi.org/10.1134/S0001433817020128

28. Volodin, E.M., Mortikov, E.V., Kostrykin, S.V., *et al.*: Simulation of the present-day climate with the climate model INMCM5. Climate Dynamics 49(11), 3715–3734 (2017). https://doi.org/10.1007/s00382-017-3539-7

29. Volodin, E.M., Mortikov, E.V., Kostrykin, S.V., *et al.*: Simulation of the modern climate using the INM-CM48 climate model. Russian Journal of Numerical Analysis and Mathematical Modelling 33(6), 367–374 (2018). https://doi.org/10.1515/rnam-2018-0032

30. Volodin, E.M., Tarasevich, M.A.: Simulation of Climate and Weather Extreme Indices with the INM-CM5 Climate Model. Russian Meteorology and Hydrology 43(11), 756–762 (2018). https://doi.org/10.3103/S1068373918110067

31. Volodin, E.M., Vorobyeva, V.V.: On the multi-annual potential predictability of the Arctic Ocean climate state in the INM RAS climate model. Russian Journal of Numerical Analysis and Mathematical Modelling 37(2), 119–129 (2022). https://doi.org/10.1515/rnam-2022-0010

32. Vorobeva, V.V., Volodin, E.M., Gritsun, A.S., *et al.*: Analysis of the Atmosphere and the Ocean Upper Layer State Predictability for up to 5 Years Ahead Using the INMCM5 Climate Model Hindcasts. Russian Meteorology and Hydrology 48(7), 581–589 (2023). https://doi.org/10.3103/S106837392307004X

33. Vorobyeva, V., Volodin, E.: Analysis of the predictability of stratospheric variability and climate indices based on seasonal retrospective forecasts of the INM RAS climate model. Russian Journal of Numerical Analysis and Mathematical Modelling 36(2), 117–126 (2021). https://doi.org/10.1515/rnam-2021-0010

34. Vorobyeva, V., Volodin, E.: Evaluation of the INM RAS climate model skill in climate indices and stratospheric anomalies on seasonal timescale. Tellus A: Dynamic Meteorology and Oceanography 73(1), 1–12 (2021). https://doi.org/10.1080/16000870.2021.1892435

35. Vorobyeva, V.V., Volodin, E.M.: Experimental Studies of Seasonal Weather Predictability Based on the INM RAS Climate Model. Mathematical Models and Computer Simulations 13(4), 571–578 (2021). https://doi.org/10.1134/S2070048221040232

36. Yakovlev, N.G.: Reproduction of the large-scale state of water and sea ice in the Arctic Ocean from 1948 to 2002: Part II. The state of ice and snow cover. Izvestiya, Atmospheric and Oceanic Physics 45(4), 478–494 (2009). https://doi.org/10.1134/S0001433809040082

37. Yakovlev, N.G.: Reproduction of the large-scale state of water and sea ice in the Arctic Ocean in 1948–2002: Part I. Numerical model. Izvestiya, Atmospheric and Oceanic Physics 45(3), 357–371 (2009). https://doi.org/10.1134/S0001433809030098