Scalable Computations of GeRa Code on the Base of Software Platform INMOST

Igor Konshin[1,2](✉) and Ivan Kapyrin[1,2]

[1] Institute of Numerical Mathematics of the Russian Academy of Sciences,
Moscow 119333, Russia
igor.konshin@gmail.com

[2] Nuclear Safety Institute of the Russian Academy of Sciences,
Moscow 115191, Russia

Abstract. The hydrogeological modeling code GeRa is based on INMOST software platform, which operates with distributed mesh data and allows to assemble and solve the system of linear equations. The set of groundwater flow models with filtration, transport, and chemical processes are considered. The comparison of parallel efficiency for different linear solvers in the INMOST framework is performed. The analysis of scalability of GeRa code on different computer platforms from multicore laptop to Lomonosov supercomputer is presented.

Keywords: Numerical modelling · Software platform · Distributed meshes · Subsurface flow and transport

1 Introduction

At present the problem of safe radioactive waste (RW) disposal is of great interest for the countries utilizing nuclear energy and radionuclides in their national economy. Along with relatively successful practice of low level waste disposal in surface repositories the advances in creation of national high-level waste (HLW) disposals are moderate. No country except Finland issued a license for the construction of such an object while the national programs for the creation of HLW deep geological disposals had been conducted for several decades. The reason for that is the complexity of safety assessment problem stemming from extremely large time and space scales, large variety of coupled processes and uncertainties.

Hydrogeological modeling codes able to model the groundwater flow and transport processes are the basis for disposal safety assessment. By order of the State Atomic Energy Corporation ROSATOM Nuclear Safety Institute and Institute of Numerical Mathematics of the Russian Academy of Sciences develop the GeRa numerical code designed for the solution of a broad class of surface and deep geological RW disposals safety assessment problems. GeRa features the application of 3D unstructured adaptive grids, initially established means of parallelization and integral modeling approach. The latter means that the code shall allow to solve the problem as a whole, starting from geological model

V. Malyshkin (Ed.): PaCT 2017, LNCS 10421, pp. 433–445, 2017.
DOI: 10.1007/978-3-319-62932-2_42

generation and ending with doses for the population calculations with the proper uncertainty analysis. At present the following major processes can be modeled in GeRa:

- groundwater flow in confined, unconfined and unsaturated conditions;
- transport in uniform and dual-porosity media (advection, dispersion, diffusion);
- equilibrium chemical reactions either governed by sorption isotherms or with real chemical calculations;
- radioactive decay chains;
- heat generation caused by radioactive decay;
- density and temperature driven convection.

The discretizations of GeRa are based on finite volume (FV) method. Besides the conventional two-point (TPFA) and multi-point flux approximation (MPFA) schemes [1] a nonlinear monotone FV method [2,3] is applied for the diffusion operator approximation. The advection operator may be discretized either using TVD-schemes with limiters or an upwind first order accurate scheme. The discretizations are aimed at use on polyhedral conformal grids. Two grid generators were implemented in GeRa [4]. The first one is the generator of triangular-prismatic grids, the second is a hexahedral grid generator based on octree structures with the ability of cell cutting. The geochemical module iPhreeqc [5] is used for chemical reactions calculation (see [6] for an example).

The INMOST [7,8] software platform is used in GeRa to support the distributed mesh and data storage and operations as well as assembling and solution of linear systems. In this work we analyze for the first time the parallel efficiency of GeRa on several computer architectures from multicore laptop to clusters and supercomputers.

The article is organized as follows. In Sect. 1 a brief overview of INMOST platform based GeRa code is given; in Sect. 2 the test problems are defined; the available linear solvers are described in Sect. 3; the results of numerical experiments are presented in Sect. 4; while the conclusions are given in Sect. 5.

2 Model Problems Description

A set of model problems common for hydrogeological modeling was chosen for numerical experiments. Different physical and chemical processes are taken into account and meshes of different sizes are generated in these tests. In the following when solving problems on a series of refined grids the following notation is used: the letter is the first letter in the name of the test ("g" – "geos", "c" – "chemistry", "t" – "transport") and the following it digits denote the number of mesh cells measured in thousands. For example model "t5740" denotes the "transport" model with 5740 thousand cells in the computational grid.

The model "f262" is a steady groundwater flow problem in a rectangular domain $[0;1]\times[0;1]\times[0;0.1]$. A regular rectangular grid containing $128\times128\times16$ cells (approximately 262 thousand cells) were used.

The "geos" set contain the groundwater flow models that are solved in a real-life domain with heterogeneous parameters. Three geological layers are present in the model. The top and bottom layers are aquitards (hydraulic conductivity $K = 0.001$ m/day), the middle layer is an aquifer with hydraulic conductivity $K = 1$ m/day. The coarsest grid contains approximately 28 thousand cells, the major part of these are triangular prisms, but also there are 102 tetrahedra and 108 pyramids caused by the top layer pinch-out. A stationary saturated groundwater flow problem is solved using the MPFA scheme. The series contains "g28", "g185", "g402", "g1425" and "g5740" models.

In the "chemistry" set of tests a reactive transport advection problem is solved. A full description of the problem is available in [6]. Five wells with balanced rates are working in a uniform layer 200 × 200 × 10 meters in size: four of them being injection wells located in the corners of the domain; the fifth being a production well in the middle. The chemical calculations are done using iPhreeqc [5]. Hexahedral octree-based grids with local refinement to well screens are used. The series contains five tests: "c18", "c60", "c254", "c700", "c1120".

In the "transport" set of tests one dimensional advective transport along the X-axis is modeled in a rectangular domain $[0; 1000] \times [0; 100] \times [0; 1]$. The discretization is done using the TVD scheme which implies local optimization problem solution on each mesh cell in the process of concentration gradient cellwise reconstruction. Triangular prismatic meshes are used. Five models with different grid sized are in the set: "t20", "t70", "t377", "t1215", "t5740".

3 Linear System Solvers Available in INMOST

Except for distributed mesh operations, INMOST software platform provides a user the interface to collect the coefficient matrix and the right-hand side of the discretization linear system and than to solve it. The main feature of this interface is the handling to the matrix row/column indices i and j by its global value just as for the dense one. It gives an opportunity for the problem discretization to simplify the collection of the coefficient matrix.

INMOST provides a common interface to linear solvers: both set of inner solvers and the third party (PETSc, Trilinos, SuperLU, Ani3D) ones. The most of inner solvers are based on advanced second order incomplete triangular factorization ILU2(τ) or in other words two-threshold ILU2(τ_1, τ_2) factorization [9]. In our experiments we used the theoretically approved values $\tau_1 = \tau$ and $\tau_2 = \tau^2$. It should be noted that the partial case $\tau_1 = \tau_2 = \tau$ is reduced to the conventional one-threshold ILU(τ) factorization.

The preconditioner parallelization is based on either Additive Schwarz AS(q) or BIILU(q) scheme with the overlap size parameter q. The geometrical interpretation of this parameter is the number of layers in the subdomains overlap.

In the present paper we consider two inner linear solvers InnerILU2 and BIILU2 ones based on ILU2(τ) with AS(q) and BIILU(q) preconditioning, respectively, as well as the conventional linear solver from PETSc package [10] based on structural factorization ILU(k) and AS(q) preconditioning. All the considered linear solvers are accelerated by BiCGStab iterations.

The BIILU2 linear solver is of the special care in our team: the first time it was presented [11] namely at the PaCT conference in 1999. It was a symmetric version with the second order incomplete Cholesky factorization IC2(τ) and block incomplete inverse Cholesky BIIC(q) as a parallel scheme with Preconditioned Conjugate Gradient (PCG) iterations. Next, at PaCT-2009 [12] the version with post filtration of triangular factors were presented.

4 Numerical Experiments

4.1 The Parallel Computer Platforms Available

In the present paper we performed the comparative analysis of parallel run properties for the developed GeRa code. The following parallel computer platforms were used:

- quad-core laptop Intel i7-4810MQ (2.80 GHz) with 16 GB RAM under Ubuntu 16.04.1 using compiler gcc v.5.4.0 and mpicc for MPICH v.3.2.
- INM RAS cluster [13] consisting of nodes with two six-core Intel Xeon X5650 (2.67 GHz) and 24 GB RAM per node under SUSE Linux Enterprise Server 11 SP1 (x86_64) using compiler Intel C v.4.0.1 and Intel MPI v.5.0.3.
- "test" and "regular4" partitions of "Lomonosov" supercomputer [14] located in the Moscow State University consisting of nodes with quad-core Intel Xeon E5-2697 v3 (2.60 GHz) and 12 GB RAM per node.

When performing numerical experiments, we analyzed the discretization stage time $T = T(p)$ obtained on one of the mentioned platform using p cores. The relative speedup $S = T(1)/T(p)$ and the resulting computational efficiency $E = S/p$ were calculated as well. The PETSc linear solver BiCGStab+AS(q)+ILU(k) with the parameters $q = 1$ and $k = 1$ was used as a default one.

4.2 Numerical Experiment on a Multicore Laptop

Table 1 presents the computation time $T(p)$ using p cores for some described above models. Numerical experiments performed on quad-core laptop showed a fairly good monotonic acceleration of computation time (up to about 3-fold) with increasing number of used cores to 4. This may allow to a GeRa user to accelerate the calculations carried out even on the local PC without exploiting an external computing cluster or in the case when the cluster is unavailable.

The data for calculation for "t70" model on 8 threads are missing in the table due to RAM restrictions. It should be noted that the use of hyper-threading technology with 8 threads allows to actually reduce the computation time for the models considered.

Table 1. Computation times (in sec.) for some models on quad-core laptop.

p	"c60"	"d224"	"g1425"	"t70"
1	320.485	4.339460	59.388	11.406
2	242.605	3.217543	35.801	7.762
4	150.606	2.927468	30.258	4.455
8	119.705	2.586953	28.005	—

4.3 Preliminary Experiments on INM RAS Cluster

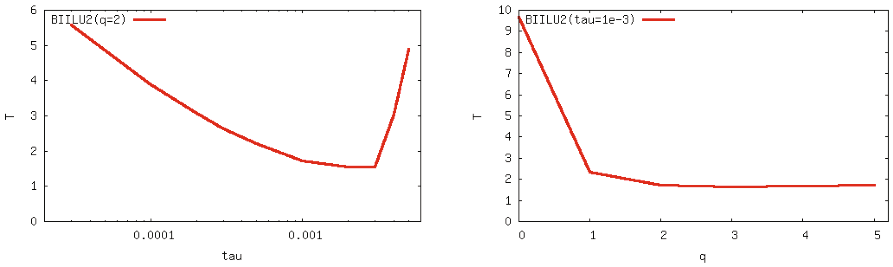
In Table 2 we compare the performance of different linear solvers of INMOST software platform when solving "f262" model using $p = 64$ cores of INM RAS cluster. Besides the already mentioned default linear solver PETSc (with parameters $q = 1$, $k = 1$) we used the inner INMOST solvers InnerILU2 (with parameters $q = 1$, $\tau = 0.005$) and solver BIILU2 (with parameters $q = 1$, $\tau = 0.03$). The default parameter settings were used in all linear solvers.

The analysis of the results in Table 2 shows the most efficient use of the linear solver BIILU2, which gives a reason for a more detailed study of its properties.

Table 2. Computation times for model "f262" on 64 cores of INM RAS cluster for different linear solvers.

	T_{discr}
InnerILU2 ($q = 1$, $\tau = 0.005$)	2.749
PETSc ($q = 1$, $k = 1$)	2.230
BIILU2 ($q = 1$, $\tau = 0.03$)	1.371

In the next experiment we analyze the influence of the parameters choice to the BIILU2 solver performance behavior. The experiment was performed on INM RAS cluster using 64 cores. The results of parameters tuning are presented on Fig. 1. The tuning of threshold parameter τ (for $q = 2$) is presented on Fig. 1a. One can observe the very smooth behavior of solution time in wide range of the parameter τ from $5 \cdot 10^{-3}$ to 10^{-6}. Next, on Fig. 1b the tuning of overlap parameter q ones again demonstrates the smooth behavior of solution time depending on q. The most important conclusion is the crucial importance of the overlap usage ($q > 0$) as well as very stable behavior up to overlap $q = 5$. The optimal values for the considered model "t1215" for $p = 64$ are $q = 3$ and $\tau = 0.001$, which are more strict ones than the usually exploited default set $q = 1$ and $\tau = 0.03$. It means that the considered model "t1215" is a more difficult to solve among the other ones.



(a)

(b)

Fig. 1. Tuning of (a) threshold parameter τ (for $q = 2$) and (b) overlap parameter q (for $\tau = 0.001$) of the BIILU2 linear solver for "t1215" model on INM RAS cluster.

4.4 Solution of Chemical Models

In the present subsection we consider the numerical results obtained on different computer platforms: INM RAS cluster and the "Lomonosov" supercomputer specified in Sect. 4.1. Both platform have about the same scalar performance, but communication rate of "Lomonosov" is appreciably higher to provide a possibility of efficient parallel computations on several thousand cores.

The number of used cores on both platforms varied from 1 to 128. Let us consider the solution of some of the model series on these two computer platforms in more detail.

Table 3 presents the speedup with respect to the serial run for the set "c" models on the INM RAS cluster using the solver PETSc($q = 3, k = 3$). The table above shows that with increasing number of cores there is a significant acceleration in computation time, and in most cases with an increase in the dimension of the problem the obtained speedup is growing. The latter can be seen most clearly in Fig. 2a. This effect is associated with a decrease in the portion of communications regarding that of calculations when the size of the local subproblem is increasing.

The maximum speedup obtained on 128 cores is 56.17, which means about 50% of parallel efficiency reached.

Table 4 presents the speedup with respect to the serial run for the set "c" models on "Lomonosov" supercomputer using the same solver PETSc($q = 3, k = 3$). The table above shows the acceleration in computation time is even better than for INM RAS (see Fig. 2b). This fact is in agreement with the above remarks on communication rate. The maximum speedup for "Lomonosov" supercomputer obtained on 128 cores is 80.28, which means more than 60% of parallel efficiency reached.

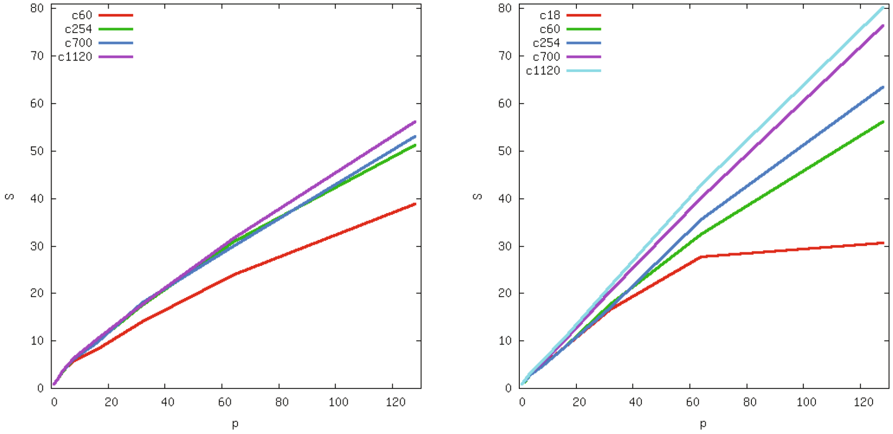
It should be noted the growth in speedup for the model "c18" on 128 cores, which means the acceleration even for about 150 unknowns per core. Effective functioning with such a small dimension subproblems on one core means high efficiency parallel implementation of INMOST platform as well as the GeRa code itself.

Table 3. Speedups for the set “c” models on INM RAS cluster with PETSc($q = 3, k = 3$) for $p = 1, ..., 128$ cores.

p	“c18”	“c60”	“c254”	“c700”	“c1120”
1	1.00	1.00	1.00	1.00	1.00
2	1.84	1.86	1.82	1.82	1.88
4	3.36	3.66	3.54	3.64	3.69
8	5.08	5.81	6.13	6.25	6.41
16	8.39	8.17	9.89	9.62	10.45
32	13.35	14.05	17.59	18.04	17.75
64	20.29	23.98	30.75	29.91	31.64
128	20.15	38.85	51.18	53.13	56.17

Table 4. Speedups for the set “c” models on “Lomonosov” supercomputer with PETSc($q = 3, k = 3$) for $p = 1, ..., 128$ cores.

p	“c18”	“c60”	“c254”	“c700”	“c1120”
1	1.00	1.00	1.00	1.00	1.00
2	1.54	1.52	1.66	1.79	1.78
4	3.14	2.87	2.93	3.27	3.28
8	4.80	4.59	4.59	5.27	5.57
16	8.75	8.79	8.69	10.19	10.84
32	16.59	17.63	17.04	20.40	21.54
64	27.82	32.48	35.60	40.19	42.84
128	30.65	56.20	63.57	76.35	80.28



(a)

(b)

Fig. 2. Speedups for the set “c” models on both (a) INM RAS cluster and (b) “Lomonosov” supercomputer with the PETSc($q = 3, k = 3$) linear solver.

4.5 Solution of "geos" Models

As shown in Table 1 the "geos" problem of dimension more than 1 million of cells can be solved on a regular laptop. In the present section we continue to analyze the parallel performance for this set of models.

Table 5 presents the speedups with respect to serial run for some models from set "g" both on INM RAS cluster and "Lomonosov" supercomputer, respectively. The default PETSc($q = 3, k = 3$) solver was used for $p = 1, ..., 128$ cores.

Table 5. Speedups for two models of the set "g" on both INM RAS cluster and "Lomonosov" supercomputer by PETSc($q = 3, k = 3$) solver for $p = 1, ..., 128$ cores.

	INM RAS cluster			"Lomonosov"	
p	"g402"	"g1425"	"g5740"	"g402"	"g1425"
1	1.00	1.00	—	1.00	1.00
2	1.85	1.97	—	1.90	2.09
4	3.18	3.49	—	3.14	3.60
8	4.64	4.81	—	4.59	3.49
16	4.50	4.28	1.00	9.06	9.78
32	10.84	8.97	1.64	16.68	19.12
64	13.63	21.03	3.16	28.46	36.07
128	16.47	22.81	5.86	48.56	62.32

From the above data, it can be seen that for "g1425" model the monotonous increase of the speedup can be obtained up to 128 cores, besides the maximal speedups for INM RAS cluster and "Lomonosov" supercomputer are 22.81 and 62.32, respectively (see also Fig. 3). From the analysis of the above data it should also be noted that for the largest model "g5740" RAM limit on "Lomonosov" supercomputer does not allow to obtain the problem solution, as well as when using from 1 to 8 cores of INM RAS cluster. However, when using from 16 to 128 cores it is possible to obtain a solution, and even with sufficiently high relative speedup 5.86 for 128 cores with respect to run on 16 cores. On one hand, the latter shows the ability to solve the problem of over 5 million of computational cells, and of the other hand, it indicates the existence of problems for which the resources of personal computer are insufficient and there is a necessity for a parallel version of the GeRa code. The latter is not only due to increasing of computation efficiency, but namely the opportunities to solve the problem itself.

4.6 Solution of Transport Models

In dealing with transport models it is required to solve the problem for groundwater flow and than the respective transport problem. In GeRa two separate default

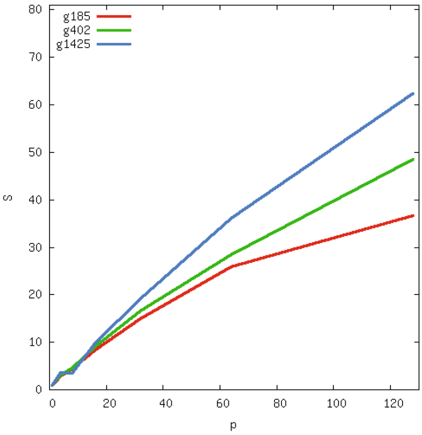


Fig. 3. Speedups for the medium size models from "g" set on "Lomonosov" supercomputer with the PETSc($q = 3, k = 3$) linear solver.

set of linear solver parameters used for these two problems: PETSc($q = 3, k = 3$) for flow problem and PETSc($q = 1,\ k = 1$) for the transport one.

When solving "t" set of models using the solver PETSc with the above default parameters, there was no convergence for the flow problem. By this reason, it was necessary to "enhance" the PETSc parameter for the flow equation up to values ($q = 7, k = 7$).

The comparative numerical results for two models from the set "t" on INM RAS cluster and "Lomonosov" supercomputer are shown in Table 6. As expected, the obtained results on "Lomonosov" supercomputer were much more scalable (as can be seen in Fig. 4).

Table 6. Speedups for two models of the set "t" on both INM RAS cluster and "Lomonosov" supercomputer by PETSc($q = 7, k = 7$) solver for $p = 1, ..., 128$ cores.

	INM RAS cluster		"Lomonosov"	
p	"t70"	"t377"	"t70"	"t377"
1	1.00	1.00	1.00	1.00
2	1.67	1.67	1.69	1.68
4	3.24	3.08	3.30	3.09
8	5.06	5.25	5.20	5.39
16	7.03	8.05	8.77	9.49
32	10.68	12.94	15.30	18.09
64	12.92	18.68	22.80	27.15
128	12.19	24.89	30.10	44.92

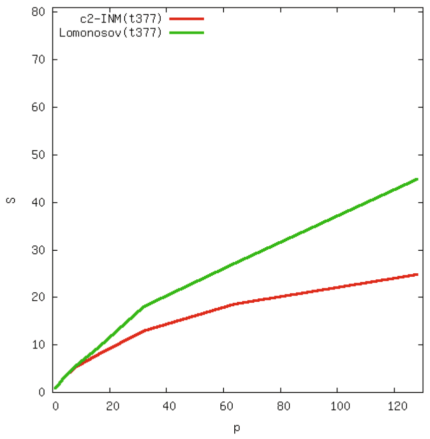


Fig. 4. Speedups for the model "t377" on INM RAS cluster and "Lomonosov" supercomputer with the PETSc($q = 7, k = 7$) linear solver.

At the next step of the linear solvers properties study we used the BIILU2 solver with default set of parameters $q = 5$ and $\tau = 0.003$. In this case all the problems at once have been successfully solved.

The calculation results are shown in Tables 7, 8 and Fig. 5 for INM RAS cluster and "Lomonosov" supercomputer, respectively. As expected, on "Lomonosov" supercomputer, it was able to achieve more significant speedup. When using 128 cores for the model "t1215" it was attained the value of 78.62, that means the parallel efficiency more than 60%. It should be noted, that the time reduction for the smallest model "t20" were observed until the use of 64 cores, which is about 300 computational cells per one core.

Table 7. Speedups for the set "t" models on INM RAS cluster by BIILU2($q = 5, \tau = 0.003$) for $p = 1, ..., 128$ cores.

p	"t20"	"t70"	"t377"	"t1215"
1	1.00	1.00	1.00	1.00
2	1.98	1.84	1.81	1.79
4	3.74	3.46	3.33	3.08
8	6.98	6.77	5.34	5.20
16	4.43	7.82	9.09	6.47
32	6.65	12.62	14.64	17.26
64	5.00	15.13	26.98	30.06
128	1.05	10.12	40.50	54.64

Table 8. Speedups for the set "t" models on "Lomonosov" supercomputer by BIILU2($q = 5, \tau = 0.003$) for $p = 1, ..., 128$ cores.

p	"t20"	"t70"	"t377"	"t1215"
1	1.00	1.00	1.00	1.00
2	1.90	1.79	1.75	1.80
4	3.69	3.31	3.29	3.23
8	7.01	6.65	5.75	5.23
16	13.02	12.91	11.64	10.18
32	22.65	24.35	24.17	24.11
64	27.35	42.70	45.41	45.78
128	20.13	50.21	76.63	78.62

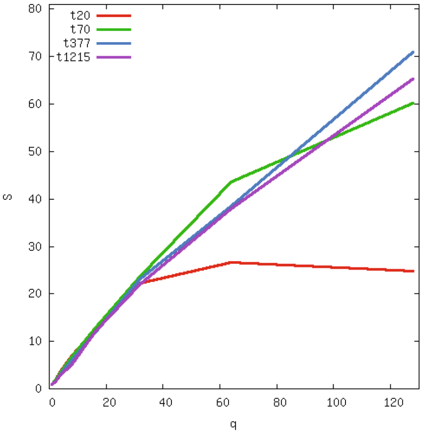


Fig. 5. Speedups for the set "t" models on "Lomonosov" supercomputer for the BIILU2($q = 3, \tau = 0.001$) linear solver.

4.7 The Results for the Largest Models

As a final illustration of the achieved parallel efficiency for the model problems, Fig. 6 shows the plots of the speedups of numerical experiments on "Lomonosov" supercomputer for the largest size models using a variety of INMOST linear solvers. From these plots, one can observe a sufficiently high parallel efficiency, which in most of the considered runs amounted more than 50 %.

To conclude this section, it should be noted the high reliability and efficiency of developed parallel linear solver BIILU2 from INMOST software platform. With BIILU2 it was able to solve all without an exception linear system for the considered models as well as for the above examples to get solutions in less time than a linear solver PETSc.

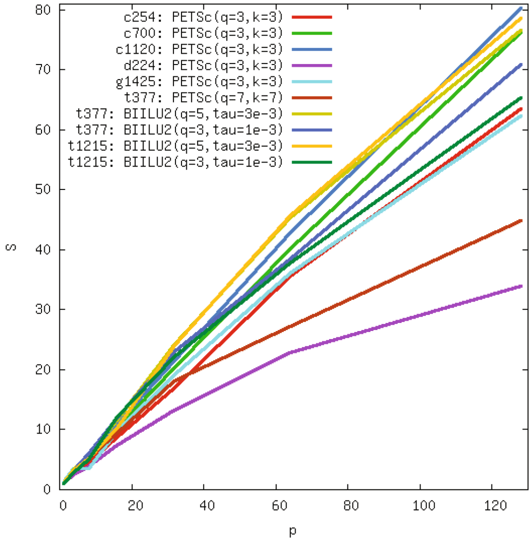


Fig. 6. Speedups for the large size models from all sets on "Lomonosov" supercomputer.

5 Conclusions

- A speedup up to 3 for GeRa code on regular quad-core laptop can be achieved for sufficiently large size models.
- The nonoverlapping block Jacobi preconditioning can be used for the most simple models only, the more complicated problems require the use of overlapping preconditioning such as AS(q) or BIILU(q).
- A conventional AS(q)+ILU(k) preconditioning from PETSc is unable to obtain the solution for the most hard-to-solve linear systems, the usage of advanced BIILU2(q, τ) preconditioning is required.
- The communication rate of the computer is of importance; Having about the same scalar performance as INM RAS cluster, the Lomonosov supercomputer takes less time for parallel runs.
- The speedup 80 on 128 cores can be achieved of Lomonosov supercomputer for sufficiently large size models.
- A speedup can be obtained even for sufficiently small size models with up to 150 unknowns per each of 128 cores.
- Ranging the models types by degree of scalability, it should be noted that the worse scalable physical process is diffusion, the transport is the sufficiently good one, while the chemical processes are the best scalable due the large amount of independent work in each mesh cell.
- Both the distributed mesh operations and linear solvers from INMOST demonstrate hight reliability and parallel efficiency in the framework of groundwater flow modelling by GeRa code.

Acknowledgements. The authors express their gratitude to V. Kramarenko for his permanent assistance with assembling the GeRa code, for installing GeRa on different computer platforms, and for generating the model sets from the GeRa verification tests.

References

1. Aavatsmark, I., Barkve, T., Boe, O., Mannseth, T.: Discretization on unstructured grids for inhomogeneous, anizotropic media. Part I: derivation of the methods. SIAM J. Sci. Comput. **19**(5), 1700–1716 (1998)
2. Danilov, A., Vassilevski, Y.: A monotone nonlinear finite volume method for diffusion equations on conformal polyhedral meshes. Russ. J. Numer. Anal. Math. Model. **24**(3), 207–227 (2009)
3. Kapyrin, I., Nikitin, K., Terekhov, K., Vassilevski, Y.: Nonlinear monotone FV schemes for radionuclide geomigration and multiphase flow models. In: Fuhrmann, J., Ohlberger, M., Rohde, C. (eds.) Finite Volumes for Complex Applications VII-Elliptic, Parabolic and Hyperbolic Problems. Springer Proceedings in Mathematics & Statistics, vol. 78, pp. 655–663. Springer, Berlin (2014)
4. Plenkin, A.V., Chernyshenko, A.Y., Chugunov, V.N., Kapyrin, I.V.: Adaptive unstructured mesh generation methods for hydrogeological problems. Vychisl. Metody Program. **16**(4), 518–533 (2015)
5. Charlton, S.R., Parkhurst, D.L.: Modules based on the geochemical model PHREEQC for use in scripting and programming languages. Comput. Geosci. **37**(10), 1653–1663 (2011)
6. Boldyrev, K.A., Kapyrin, I.V., Konstantinova, L.I., Zakharova, E.V.: Simulation of strontium sorption onto rocks at high concentrations of sodium nitrate in the solution. Radiochemistry **58**(3), 243–251 (2016)
7. INMOST: a toolkit for distributed mathematical modelling. URL: http://www.inmost.org
8. Vassilevski, Y.V., Konshin, I.N., Kopytov, G.V., Terekhov, K.M.: INMOST - a software platform and a graphical environment for development of parallel numerical models on general meshes, 144 pp. Moscow State University Publication, Moscow (2013) (in Russian)
9. Kaporin, I.E.: High quality preconditioning of a general symmetric positive definite matrix based on its $U^TU + U^TR + R^TU$-decomposition. Numer. Lin. Alg. Applic. **5**(6), 483–509 (1998)
10. PETSc - Portable, Extensible Toolkit for Scientific Computation. https://www.mcs.anl.gov/petsc
11. Kaporin, I.E., Konshin, I.N.: Parallel solution of large sparse SPD linear systems based on overlapping domain decomposition. In: Malyshkin, V. (ed.) PaCT 1999. LNCS, vol. 1662, pp. 436–446. Springer, Heidelberg (1999). doi:10.1007/3-540-48387-X_45
12. Kaporin, I.E., Konshin, I.N.: Load balancing of parallel block overlapped incomplete cholesky preconditioning. In: Malyshkin, V. (ed.) PaCT 2009. LNCS, vol. 5698, pp. 304–315. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03275-2_30
13. INM RAS cluster. http://cluster2.inm.ras.ru (in Russian)
14. ”Lomonosov” supercomputer. https://parallel.ru/cluster/lomonosov.html (in Russian)