Parallel linear systems solution for multiphase flow problems in the INMOST framework*

Igor Konshin[1,2], Igor Kaporin[1], Kirill Nikitin[2] and Yuri Vassilevski[2]

Dorodnicyn Computing Centre of RAS, Moscow, Russia[1],
Institute of Numerical Mathematics of RAS, Moscow, Russia[2]

The solution on parallel computers of multiphase flow problems is considered. To store and operate with distributed mesh data on the problem discretization stage the INMOST program platform was exploited. The resulting linear systems were solved by the overlapping additive Schwarz method from PETSc package as well as by the developed linear solver on the base of BIILU2 method. The results of numerical experiments for different parallel computers were presented and analyzed. The efficiency of the INMOST platform and the involved linear solvers was demonstrated.

The main purpose of the paper is to demonstrate applicability of the INMOST program platform to the solution of complex geophysical problems such as multiphase flow model.

1. Three-phase black oil model

We consider the three-phase black oil model in a porous medium [2]. Two-phase flow model is introduced in a similar way and is presented in [1]. Subscripts denote to the three phases – water, oil (the liquid phase) and gas (the gaseous phase) – and the three components – water, oil and gas, respectively.

The basic equations for the black oil model are the following:

1. Mass conservation equations:

$$\frac{\partial}{\partial t}\Big(\phi\rho_w S_w\Big) = -\mathrm{div}\Big(\rho_w \mathbf{u}_w\Big) + \rho_w q_{W_s} \tag{1}$$

for the water component,

$$\frac{\partial}{\partial t}\Big(\phi\rho_{O_o} S_o\Big) = -\mathrm{div}\Big(\rho_{O_o} \mathbf{u}_o\Big) + \rho_{O_o} q_{O_s} \tag{2}$$

for the oil component, and

$$\frac{\partial}{\partial t}\Big[\phi\Big(\rho_g S_g + \rho_{G_o} S_o\Big)\Big] = -\mathrm{div}\Big(\rho_g \mathbf{u}_g + \rho_{G_o}\mathbf{u}_o\Big) + \rho_g q_{G_s} + \rho_{G_o} q_{O_s} \tag{3}$$

for the gas component.

2. Darcy's law for each phase:

$$\mathbf{u}_\alpha = -\frac{k_{r\alpha}}{\mu_\alpha}\mathbb{K}\Big(\nabla p_\alpha - \rho_\alpha \mathbf{g}\nabla z\Big), \quad \alpha = w, o, g. \tag{4}$$

3. The saturation constraint:

$$S_w + S_o + S_g = 1. \tag{5}$$

4. Pressure difference between phases is given by capillary pressure:

$$p_o - p_w = p_{cow}, \qquad p_g - p_o = p_{cgo}. \tag{6}$$

*This work has been supported in part by RFBR grants 14-01-00830, 15-35-20991 and ExxonMobil Upstream Research Company.

Here $\mathbb{K}$ is an absolute permeability tensor, ϕ is a porosity, p_α, S_α, $\mathbf{u}_\alpha$ are *unknown* pressure, saturation, volumetric velocity, B_α, μ_α and $k_{r\alpha}$ are the formation volume factor, viscosity and relative phase permeability for the phase $\alpha = w, o, g$. Also ρ_w, ρ_{O_o}, ρ_{G_o} and ρ_g are the densities at current conditions, z is the depth, $\mathbf{g}$ is a gravity term, q_{β_s} is a source/sink well term of the component $\beta = W, O, G$ at standard conditions.

We consider no-flow (homogeneous Neumann) boundary condition on the reservoir boundary and wells with a given bottom pressure p_{bh}, component flux q_{β_s} or total flux $q_T = q_{W_s} + q_{O_s} + q_{G_s}$.

In our simplified model each well is assumed to be vertical and connected to the center of a cell. The formula for the well term was suggested by Peaceman [3]:

$$q_{\beta_s} = \frac{k_{r\alpha}}{\mu_\alpha} WI\Big(p_{bh} - p_\alpha - \rho_\alpha \mathbf{g}(z_{bh} - z)\Big), \tag{7}$$

where WI is the *well index*, which doesn't depend on the properties of fluids, but depends on properties of the media.

In the discrete counterparts of (1)-(4) the mobilities $\lambda_\alpha = k_{r\alpha}(S_\alpha)/\mu_\alpha(p_\alpha)$ on the face f_{ij} are taken upwinded:

$$\lambda_\alpha(S_\alpha) = \begin{cases} \lambda_\alpha(S_{\alpha,i}, p_{\alpha,i}) & \text{if flow is directed from cell } i \text{ to cell } j, \\ \lambda_\alpha(S_{\alpha,j}, p_{\alpha,j}) & \text{if flow is directed from cell } j \text{ to cell } i. \end{cases}$$

The phase mobilities for well-producer are taken upwinded from the cell. For well-injector we have only water injected and thus take the downstream mobility from the cell with the well: $\lambda_{\text{inj}} = (k_{rw}/\mu_w + k_{ro}/\mu_o + k_{rg}/\mu_g)_{\text{cell}}$. We also assume that there is no capillary pressure in wells, so all the well fluxes depend on the same (oil) pressure.

Both two- and three-phase flow models are discretized using fully implicit schemes for time discretization and monotone nonlinear finite volume method for spatial discretization of the fluxes in Jacobian matrix [1].

2. Parallel multiphase flow experiments

The first numerical test for a parallel version of three-phase black-oil model was performed on the BlueGene/P cluster located in the Moscow State University and two parts of the INM cluster [12].

- BG/P system consists of relatively slow PowerPC 450 (850 MHz) cores with 2 GB RAM each.
- The first part of the INM cluster (INM-1) consists of nodes with two quad-core Intel Xeon X5355 (2.66 GHz) or Intel Xeon E5462 (2.80 GHz) processors and 8 GB RAM per node.
- The second part of the INM cluster (INM-2) consists of nodes with two six-core Intel Xeon X5650 (2.67 GHz) and 24 GB RAM per node.

The problem set-up is the following. The square region contains two wells in the opposite corners: one injector and one producer with given bottom hole pressures.

In our parallel simulation, we use parallel grid generation. At the first stage the computational domain is split into subdomains which are distributed between available cores. At the second stage each core constructs a local grid inside the associated subdomain and exchanges ghost cells with neighbours. Only one layer of ghost cells is sufficient due to the compact stencil of discrete operators. Grid partitioning example is shown on Fig. 1.

The total grid dimensions are 128 × 128 × 16 which gives us total of 304 192 nodes (cells + boundary entities).

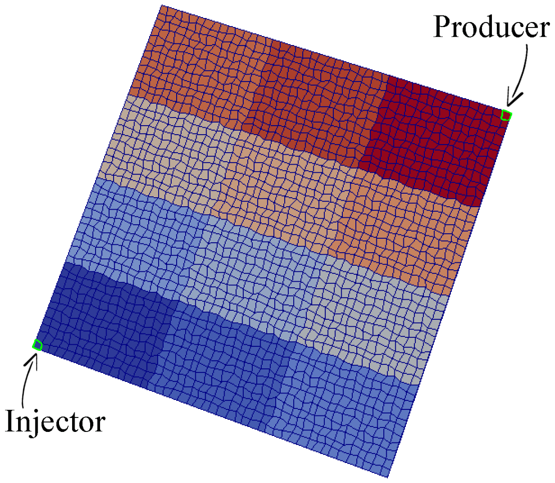

Figure 1. Grid partitioning example.

Table 1. Parallel speedup of simulation, BG/P.

#cores	Nodes/core	#lit	t_{init}	t_{sol}	speedup
8	38 024	71 024	68.8s	28 549s	1x
16	19 012	71 042	37.2s	14 471s	1.97x
32	9 506	71 648	19.6s	7 464s	3.82x
64	4 753	72 174	10.5s	3 874s	7.36x
128	2 377	73 806	5.9s	2 059s	13.86x

Table 2. Parallel speedup of simulation, INM-1 and INM-2 clusters.

	INM-1			INM-2		
#cores	t_{init}	t_{sol}	speedup	t_{init}	t_{sol}	speedup
8	9.9s	12 506s	1x	6.2s	4 909s	1x
16	5.2s	6 182s	2.02x	3.8s	2 980s	1.65x
32	3.0s	3 756s	3.33x	2.4s	1 957s	2.51x
64	1.7s	1 926s	6.49x	2.0s	1 092s	4.50x
128	1.0s	1 131s	11.06x	–	–	–

Linear systems were solved with the PETSc package. The chosen solver is BCG iterations combined with the additive Schwarz preconditioner and ILU0 preconditioners in subdomains.

Table 1 shows the results of the parallel experiment on BG/P for 200 days simulation. One can see good speedup for up to 128 cores (2.4k nodes per core). The number of total nonlinear iterations is 648 and does not depend on the number of cores. The number of linear iterations (#lit) increases slightly as #cores grows, while the initialization (t_{init}) and computation (t_{sol})

times decrease almost linearly. We note that the BG/P system has fast connection with relatively slow computational cores.

Table 2 shows the results for INM-1 and INM-2 which have much faster cores than BG/P. As expected, the speedup is lower albeit satisfactory: up to 11x for 8-to-128 cores on INM-1 and 4.5x for 8-to-64 cores on INM-2.

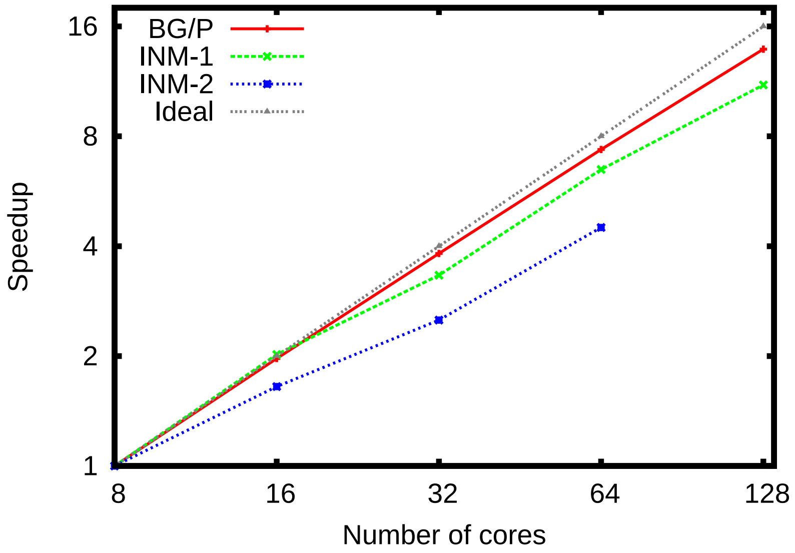

Figure 2. Parallel speedup, BG/P, INM-1 and INM-2.

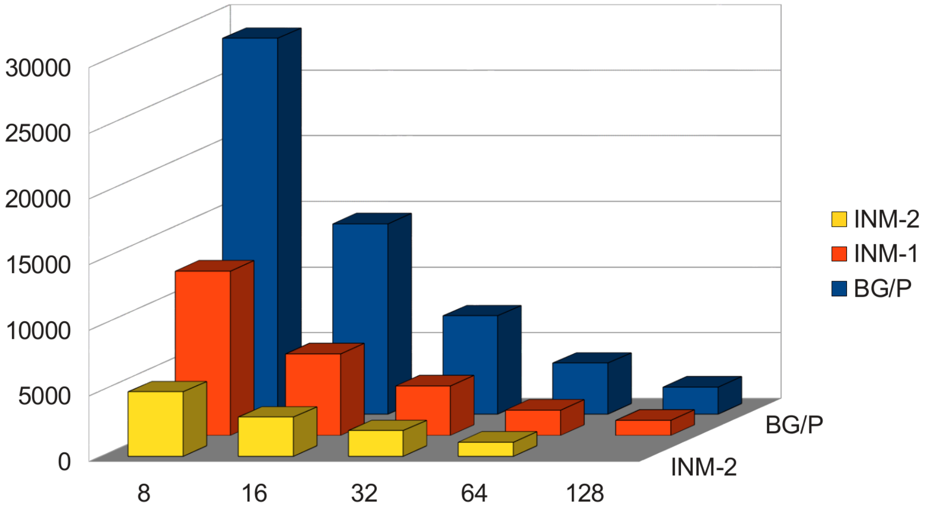

Figure 3. Solution times for parallel computation, BG/P, INM-1 and INM-2.

Figure 2 presents the speedup of the parallel computation which in case of BG/P cluster is close to the ideal linear speedup. Figure 3 shows the diagram with computational times on three clusters.

The presented results demonstrate good quality of the developed parallel data structure and algorithms [5,6], although we use the third-party PETSc linear solver that also can be improved.

The second experiment deals with two-phase flow model on a massively parallel BG/P system with up to 8k cores. Problem setup and grid construction method is similar to the first test case. We consider 50 days simulation on 0.9 million cells nonorthogonal hexahedral grid (1.8 million unknowns).

Table 3 presents number of linear (#lit) and nonlinear (#nonlit) iteration, initialization (t_{init}), grid generation (t_{grid}) and total simulation (t_{sol}) times of the parallel experiment for two-phase flow model. The reference results are taken for 512 cores run. One can see that the total simulation time decreases, yet there is almost no speedup for 1024-to-2048 and 4096-to-8192 pairs (see Fig. 4, left). This is explained by the sharp increase of the total number of the PETSc linear iterations in these pairs (see Fig. 4, right). As an additional argument, it should be paid attention to the fact that number of degrees of freedom per core (#DoF/core) for 8192 cores is extremally small (219).

Table 3. Parallel speedup of the two-phase flow simulation, BG/P.

#cores	#DoF/core	#nonlit	#lit	t_{init}	t_{grid}	t_{sol}	speedup
512	3515	151	247 919	2.69	1.19	1478.7	1x
1024	1757	151	190 099	1.80	0.70	559.0	2.64x
2048	878	150	333 369	1.34	0.49	536.2	2.76x
4096	439	148	291 533	1.53	0.41	296.7	4.98x
8192	219	147	402 742	1.91	0.46	296.0	5.0x

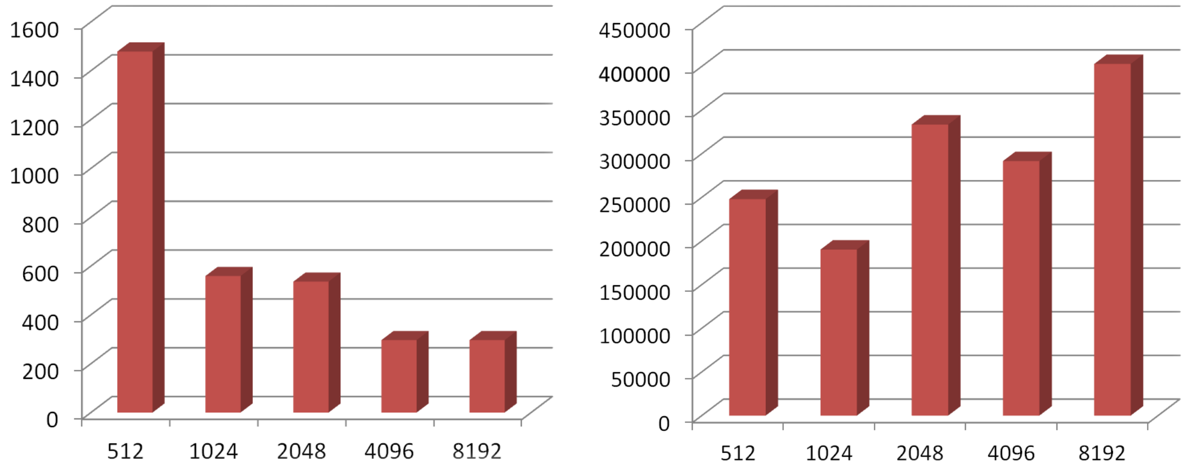

Figure 4. Left: reduction of computation time compared to 512-cores experiment. Right: total number of linear iterations for simulation.

3. Parallel solution of linear systems

In the INMOST framework it is possible to exploit some set of the inner parallel linear solvers (such as based on BiCGStab(L) solvers with the second order ILU factorization [8] or with second order Crout-ILU with inversed-based condition estimation and unsymmetric reordering for diagonal dominance used as a preconditioners), external linear solvers from PETSc and Trilinos, as well as user devepoled linear solvers.

In the present section we consider the solution of sample linear systems come from the described INM black oil simulator for the problem SPE-10 (see [4]). In the first test case we solve the linear system with the Jacobian matrix for a refined well model for 50-th day of simulation with 1.2434 days timestep (see Table 4). In this table we use the following notation: n - the size of the matrix A, nz(A) - the total number of nonzero entries in A, eqzd - the number of zero entries on the main diagonal of A, ltzd - the number of negative entries on the main diagonal of A, gtza - the number of positive entries in the off-diagonal part of A, $\mathrm{nzr}_{\mathrm{min}}$ and $\mathrm{nzr}_{\mathrm{max}}$ - minimum and maximum numbers of nonzeroes in a row of A, respectively. The matrix row with 253 nonzero elements cooresponds to one of the wells.

Table 4. INM simulator matrix properties.

name	n	nz(A)	eqzd	ltzd	gtza	$\mathrm{nzr}_{\mathrm{min}}$	$\mathrm{nzr}_{\mathrm{max}}$
N50	3896016	43925904	28	2737291	30234207	1	253

Table 5. N50 problem: PETSc AS(ILU(1);$q = 1$)-BiCGStab and BIILU2($q = 2; \tau = 0.005$)-BiCGStab solution.

Solver	p	#mvm	T_{prec}	T_{iter}	T_{total}	speedup
PETSc	1	163	2.35	128.83	131.18	1.00
	2	450	1.89	173.47	175.37	0.74
	4	356	0.99	122.76	123.75	1.06
	8	513	0.56	129.87	130.43	1.00
	16	281	0.30	59.67	59.97	2.18
	32	641	0.15	60.38	60.53	2.16
	64	734	0.10	46.61	46.71	2.80
BIILU2	1	139	54.50	50.91	105.41	1.00
	2	209	30.72	37.52	68.24	1.54
	4	185	18.38	21.11	39.49	2.66
	8	187	13.94	15.67	29.61	3.55
	16	228	7.01	14.98	21.99	4.79
	32	215	3.49	7.00	10.49	10.04
	64	321	4.82	5.40	10.22	10.31

In Table 5 we present the comparisson of parallel solution for the above linear system N50 for PETSc [7] and our research linear solver based on BIILU2 preconditioning [9,10]. The number of processors was $p = 1, 2, 4, ..., 64$ for the described above INM-2 cluster. For both solvers the optimal parameters were chosen: for PETSc package we have used BiCGStab iteration with overlapped (overlap size is $q = 1$) additive Swartz and structural ILU(1) factorization as a preconditioner, for BIILU2 solver we have used BiCGStab iteration with the second order threshold ILU2($\tau = 0.005$) factorization and $q = 1$ as a preconditioner. In this table T_{prec},

Table 6. Poisson equation solved by BIILU2($q=1; \tau=0.1$)-BiCGStab on INM-2 cluster in 8 nodes by 8 cores configuration.

mesh	p	n/p	#mvm	T_{prec}	T_{iter}	T_{total}	speedup
16x16x32	1	8192	33	0.0091	0.0116	0.0207	1.00
8x8x16	1	1024	15	0.0012	0.0007	0.0019	10.89
32x32x64	8	8192	71	0.0120	0.0331	0.0452	1.00
16x16x32	8	1024	37	0.0016	0.0028	0.0044	10.27
64x64x128	64	8192	129	0.0157	0.0727	0.0885	1.00
32x32x64	64	1024	71	0.0020	0.0133	0.0153	5.78

Table 7. Poisson equation solved by BIILU2($q=1; \tau=0.1$)-BiCGStab on INM-2 cluster in 8 nodes by 8 cores configuration.

Matrix	Method	p	dens	#mvm	T_{prec}	T_{iter}	T_{total}	S
N50	BiCGStab	64	1.89	241	4.78	3.87	8.65	1.00
N50	BiCGStab	128	1.94	173	1.99	2.18	4.17	2.07
N50	BiCGStab	256	2.02	217	2.47	2.10	4.58	1.88
N50	GMR[10]	64	1.89	217	5.88	4.84	10.73	1.00
N50	GMR[10]	128	1.94	197	1.99	2.45	4.44	2.41
N50	GMR[10]	256	2.02	217	2.53	**1.91**	4.45	2.40

T_{iter} and T_{total} denote the times for preconditioner construction, iterations and total solution, respectively. One can see that the grows of the iterations number #mvm for PETSc is much more than for BIILU2 solver. PETSc demonstrates much more grows of the iterations number #mvm in comparisson with BIILU2 solver. This is why PETSc provides very small speedup, while speedup for BIILU2 is quite reasonable.

In Table 6 the results for the solution of a sample Poisson equation is presented. We examine the performance of the solver for different number of processors as well as the different sizes of the linear systems reducing down to 1024 rows per processor. One can see only 4.27 grows of solution time for the same subproblem size equal to 8192 for 1 and 64 number of processors.

In addition, for N50 problem we apply the polynomial preconditioning [11] aimed to reduce the number of global syncronizations. It was assumed that it may have a reason for a large number of processors. In Table 7 the results of parallel runs on p = 64, 128, 256 processors are presented. The experiments were performed on “Lomonosov” computer cluster [13]. One can see, that for p = 256 the polynomial method GMR[10] work faster (T_{total} = 4.45) that regular BIILU2 (T_{total} = 4.58) for the same number of iterations (#mvm=217).

4. Conclusions

We considered the parallel solution of the multiphase flow problems using the INMOST program platform. The platform demonstrated the user friendly interface, flexibility and efficiency of the mesh related operations. Linear systems originated from the problems were solved using both the inner BIILU2 solver and external PETSc package. BIILU2 showed better scalability than the commonly used PETSc.

References

1. Nikitin K.D., Terekhov K.M., Vassilevski Y.V. A monotone nonlinear finite volume method for diffusion equations and multiphase flows // *Computational Geosciences* Vol. 18, No 3 (2014), pp 311-324, DOI: 10.1007/s10596-013-9387-6

2. Chen Z., Huan G., Ma Y. *Computational Methods for Multiphase Flows in Porous Media.* SIAM. 2006, 549p.

3. Peaceman D.W. *Interpretation of Well-Block Pressures in Numerical Reservoir Simulation.* SPEJ, June (1978), 183–194.

4. SPE Comparative Solution Project: description of model 2. URL (2015): http://www.spe.org/web/csp/datasets/set02.htm

5. Vassilevski Yu, Konshin I., Kopytov G., Terekhov K. INMOST – a software platform and a graphical environment for development of parallel numerical models on general meshes. Moscow State Univ. Publ., Moscow, 2013, 144 p. (in Russian).

6. INMOST: A toolkit for distributed mathematical modeling.
   URL (2015): http://www.inmost.org/

7. PETSc: Portable, Extensible Toolkit for Scientific Computation.
   URL (2015): http://www.mcs.anl.gov/petsc/

8. Kaporin I.E. High quality preconditioning of a general symmetric positive definite matrix based on its $U^TU + U^TR + R^TU$-decomposition. *Numer. Linear Algebra Appls.* **5**, 483–509, 1998.

9. Kaporin I.E., Konshin I.N. A parallel block overlap preconditioning with inexact submatrix inversion for linear elasticity problems. Numer. Linear Algebra Appl. 2002, Vol. 9, 141-–162.

10. Kaporin,I.E., Milyukova O.Yu. Massively parallel algorithm of the preconditioned conjugate gradient method for the numerical solution of systems of linear algebraic equations. In: Optimization and Applications, Issue 2 (V.G.Zhadan, ed.). A.A.Dorodnicyn Comput. Center of Russian Academy of Sciences Publ. 2011, pp.132-157 (in Russian).

11. Kaporin I.E. Discrete least squares method for polynomial preconditioning of unsymmetric matrices. In: Optimization and Applications, Issue 3 (V.G.Zhadan, ed.). Comput. Center of Russian Academy of Sciences Publ. 2013, pp.120-139 (in Russian).

12. INM RAS cluster description (in Russian). URL (2015): http://cluster2.inm.ras.ru/

13. Lomonosov cluster description (in Russian).
    URL (2015): http://www.parallel.ru/cluster/lomonosov.html