Power Distribution Networks Simulation Using Parallel Linear Algebra Packages

Vasiliy Yu. Voronov

Department of Computational mathematics and cybernetics, Lomonosov Moscow State University,
Leninskie Gori, Moscow 119992, Russia

basrav@angel.cmc.msu.ru

Abstract

Power distribution networks simulation involves evaluation of very large ($10^7 - 10^9$ nodes) electrical circuits. Accurate simulation requires solving large ODE on parallel platform. In this work, software for parallel power grid simulation on multicore and distributed memory platforms is proposed. Parallel approach is implemented at the stage of solving sparse linear systems of equations which is obtained from numerical integration of the ODE. For these purposes, PARDISO direct solver and iterative solvers from the Intel MKL package are used for multicore platform, and iterative solvers from the PETSc package are used for cluster. Scalability and performance results are presented and discussed in this work.

This work was supported by Federal Program "Development of scientific and teaching staff in innovative Russia 2009–2013".

1. Introduction

POWER DISTRIBUTION NETWORK is multi-layered metal structure which is used to attach power source to power drains in electronic device. Typical regular structure is shown on fig. 1.

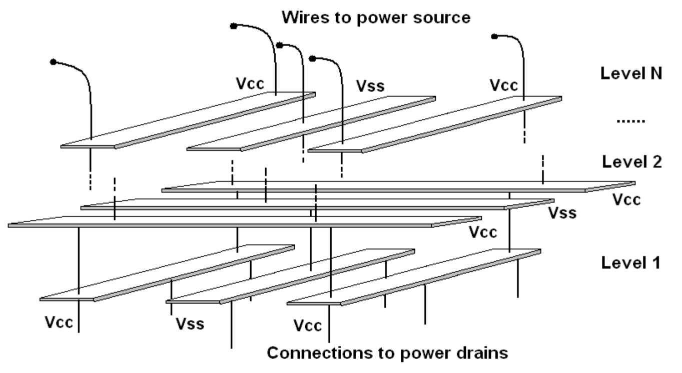


Figure 1: *Power distribution network structure*

Physical behaviour lies in IR-drop caused by resistance of metal lines and transient process caused by $L\frac{dI}{dt}$-drop at the inductance branches. Simulation problem statement is in finding node potentials at the time interval $t \in [0, T]$. It is done by considering equivalent electrical circuit and applying Kirchoff laws.

Existing simulation approaches include full simulation [Chen T., Chen C. 2001], model order reduction-based approaches [Kozhaya J., Nassif S., Najm F. 2002], [Nassif S., Kozhaya J., 2000]. However, existing studies deal with relatively small power grids ($10^6 - 10^7$ nodes).

2. Theory

Kirchhoff voltage and current laws gives ordinary differential equation (ODE):

$$C\dot{x} + Gx = F(t). \tag{1}$$

Eq. (1) in expanded form

$$\begin{bmatrix} A_C C A_C^T & 0 & 0 \\ 0 & L & 0 \\ 0 & 0 & 0 \end{bmatrix} \frac{d}{dt} \begin{bmatrix} e \\ i_L \\ i_V \end{bmatrix} + \begin{bmatrix} A_G G A_G^T & A_L & A_V \\ -A_L^T & 0 & 0 \\ A_V^T & 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ i_L \\ i_V \end{bmatrix} = \begin{bmatrix} -A_I I(t) \\ 0 \\ E \end{bmatrix}$$

Backward-Euler integration gives

$$(G + \frac{C}{h})x(t+h) = F(t+h) + \frac{C}{h}x(t),$$

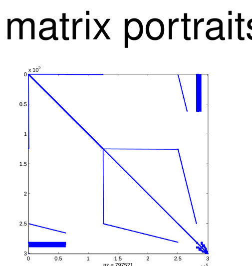
a series of linear systems of equations with time-independent matrix for each time step (*transient analysis*)

$$Ax^{(i)} = b^{(i)}, \text{ where } b^{(i)} = F(t+ih) + \frac{C}{h}x^{(i-1)} \tag{2}$$

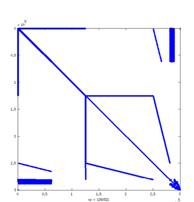
Initial value $x^{(0)}$ is found as solution of *direct current* problem

$$\begin{bmatrix} A_G G A G^T & A_V \\ A_V^T & 0 \end{bmatrix} \begin{bmatrix} e \\ i_V \end{bmatrix} = \begin{bmatrix} -A_I I(0) \\ E \end{bmatrix}$$

Examples of matrix portraits:



Direct current matrix



Transient matrix

3. Proposed software implementation

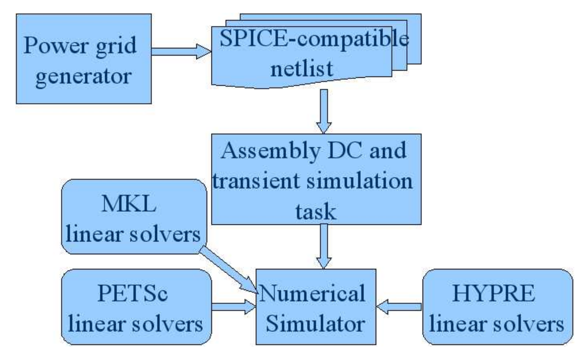


Figure 2: *Components of implemented parallel power grid simulator*

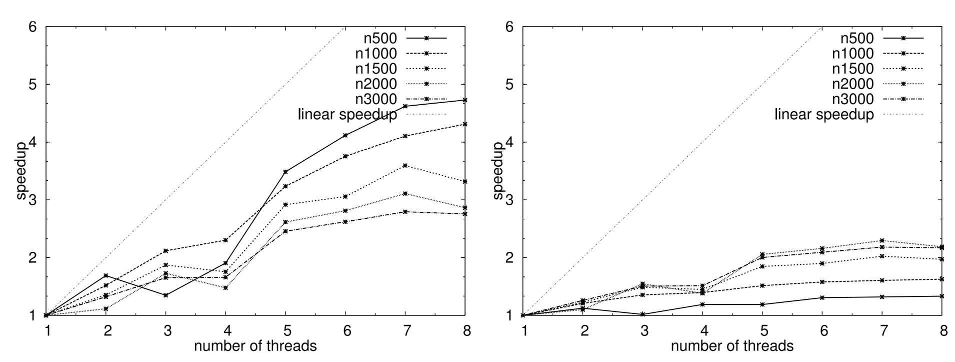
Extensive experiments were performed to study performance and scalability of proposed simulation scheme. Basic details on data sets are shown in the table 1. Here T = 10^{-9}s, h = 10^{-12}s are considered for simulations. In experiments, multicore platform is 8-cored 2 x Intel 5472 3.2GHz, 16 Gb RAM. Cluster of multicores is SKIF-MGU "Chebyshev" platform with 2 x Intel 5470 3.0GHz, 4–8 Gb RAM nodes and InfiniBand DDR2 interconnect.

For multicore, we study scalability of $n500 - n4000$ power grids simulation due to RAM limitations. For "Chebyshev" cluster, 3 nodes with total 24 cores are minimum for simulations because of the RAM constraints.

4. Experimental results for multicore

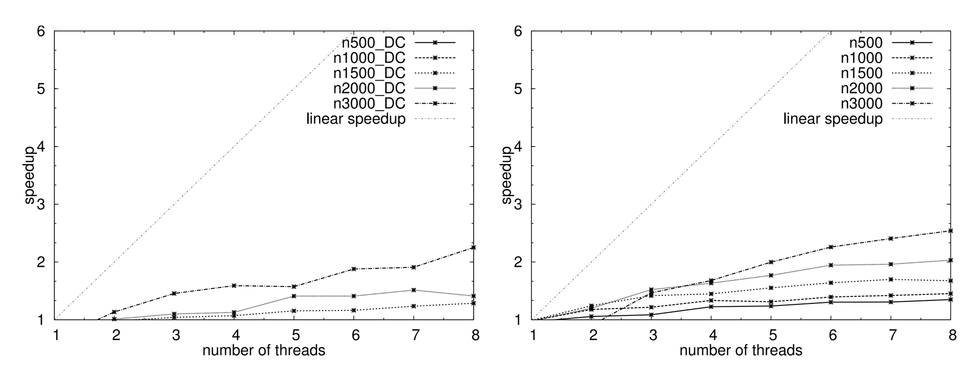
Table 1: *Properties of power grid models and corresponding matrices used in simulations. Here # C–number of capacitors, # R–number of resistors, # L–number of inductances, # I–number of current sources, n–matrix dimension, nnz–nonzeros in matrix. Matrices with _DC suffix denote initial value problem, _T are matrices from transient analysis task*

Matrix	# nodes	# C	# R	# I	# L	n	nnz
n500	300.6K	124.8K	466.2K	1.3K	11	300.6K	984.0K
n500_DC	300.6K	0	124.8K	0	0	300.7K	1.2M
n1000	1.2M	499.5K	867.9K	5.0K	33	1.2M	4.9M
n1000_DC	1.2M	0	867.8K	0	0	1.2M	3.9M
n1500	2.7M	1.1M	4.2M	11.3K	60	2.7M	11.1M
n1500_DC	2.7M	0	4.2M	0	0	2.7M	8.9M
n2000	4.8M	1.9M	7.5M	20.2K	127	4.8M	1.9M
n2000_DC	4.8M	0	7.5M	0	0	4.8M	16.6M
n3000	10.8M	4.5M	16.8M	45.5K	263	10.8M	44.4M
n3000_DC	10.8M	0	16.8M	0	0	10.8M	35.4M
n4000	19.2M	7.9M	32.6M	79.6K	536	19.2M	80.9M
n4000_DC	19.2M	0	32.9M	0	0	19.2M	68.9M
n7000	58.8M	24.5M	100.6M	244.8K	1558	58.8M	211.0M
n7000_DC	58.8M	0	100.6M	0	0	58.8M	173.2M
n9000	97.3M	40.4M	166.3M	405.9K	2543	97.3M	331.2M
n9000_DC	97.3M	0	166.3M	0	0	97.3M	296.4M



(a) (b)

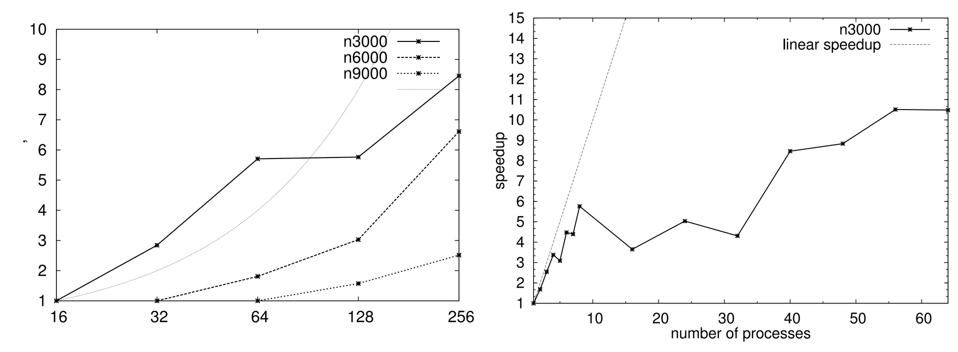
Figure 3: *Speedup of factorization sub-stage (a) and whole time step of transient simulation (b) using MKL PARDISO solver*



(a) (b)

Figure 4: *Speedup of DC-analysis stage (a) and transient analysis time step (b) using MKL's iterative GMRES+ILUT solver.*

5. Experimental results for cluster of multicores



(a) (b)

Figure 5: *Scalability results for solving DC-problems (a) and transient analysis problem (b) on "Chebyshev" cluster for different power grid size.*

Table 2: *Solution time of $n9000$ power grid for various simulation stages on cluster of 8-cores. Sparse linear solvers are used from PETSc. Solvers with the best speedup for each stage are marked with bold.*

Solver type	peak RAM	Number of cores				max. speedup
		24	48	64	128	
DC-task solution						
GMRES+Jacobi	14.3 Gb	841.4	562.4	409.4	287.6	2.9
GMRES+Bjacobi	17.6 Gb	439.2	297.5	214.8	133.6	3.3
GMRES+ASM	17.9 Gb	457.7	301.3	-	151.2	3.0
BiCGStab+Jacobi	15.9 Gb	971.9	-	514.5	367.2	2.6
BiCGStab+Bjacobi	18.4 Gb	614.8	-	309.6	171.3	**3.6**
BiCGStab+ASM	18.9 Gb	-	347.3	289.2	189.2	1.8
1 time step of transient analysis						
GMRES+Jacobi	19.2 Gb	1.2K	-	652.33	429.41	2.9
GMRES+Bjacobi	23.5 Gb	879.27	514.43	431.60	252.18	3.5
GMRES+ASM	22.4 Gb	853.76	590.26	423.59	269.10	3.2
BiCGStab+Jacobi	18.8 Gb	-	498.31	327.28	130.81	**3.8**
BiCGStab+Bjacobi	23.5 Gb	914.16	532.41	-	-	1.7
BiCGStab+ASM	23.1 Gb	936.44	586.31	-	309.18	3.0
DC-task + 1000 time steps of transient analysis						
GMRES+Jacobi	19.6 Gb	838.5K	-	-	409.2K	2.04
GMRES+Bjacobi	24.1 Gb	569.3K	-	-	208.4K	**2.72**
GMRES+ASM	23.4 Gb	558.8K	-	-	213.4K	2.61

6. Conclusions

1. For multicore platform, iterative solvers from MKL show better performance, scalability and RAM requirements compared with PARDISO direct solver and PETSc' iterative solvers. MKL iterative solver speeds up to 2.6 times on 8 cores when solving transient simulation problem and up to 2.1 times for the direct current problem.
2. For cluster of multicores, PETSc's GMRES+Block Jacobi solver displays best reliability and may be considered to be best choice of solver for the considered problem class.
3. For cluster of multicores, both "MPI on all cores" and "MPI+threads on node" parallel hybrid approaches were evaluated, "MPI on all cores" has displayed better performance for small clusters (10–20 nodes).
4. Linear solver type and its settings selection heavily depends on power grid geometry and physical properties. A careful study of numerical properties of problems should be done when switching to another power grid classes.
5. Problem's "complexity" grows faster than problem's size (specifically, the bigger the power network, the larger is *conditioning number* of the corresponding numerical problems). Sparse linear equations for larger power grids become ill-conditioned. It means that iterative solvers may be efficiently used for networks with up to ≈ 150M nodes.
6. Further studies include more complex power networks (add irregularities into network structure, add nonlinear inductances), propose domain-decomposition solvers for larger power grids ($10^8 - 10^9$ nodes) on massively parallel platforms.

International Supercomputing Conference 2010. May, 30 – June, 03 2010. Hamburg, Germany