



Different Solvers Evaluation for a Bucking Problem

Chau-Yi Chou, Jiunn-Horng Lee, Yu-Fen Cheng, Chih-Wei Hsieh, Weichung Wang

► **To cite this version:**

Chau-Yi Chou, Jiunn-Horng Lee, Yu-Fen Cheng, Chih-Wei Hsieh, Weichung Wang. Different Solvers Evaluation for a Bucking Problem. Ching-Hsien Hsu; Xuanhua Shi; Valentina Salapura. 11th IFIP International Conference on Network and Parallel Computing (NPC), Sep 2014, Ilan, Taiwan. Springer, Lecture Notes in Computer Science, LNCS-8707, pp.583-586, 2014, Network and Parallel Computing. <10.1007/978-3-662-44917-2_58>. <hal-01403151>

HAL Id: hal-01403151

<https://hal.inria.fr/hal-01403151>

Submitted on 25 Nov 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Different Solvers Evaluation for a Bucking Problem

Chau-Yi Chou¹, Jiunn-Horng Lee¹, Yu-Fen Cheng¹, Chih-Wei Hsieh¹, Weichung Wang²

¹ National Center for High-performance computing, Taiwan

² Institute of Applied Mathematical Sciences, National Taiwan University
cychou@nchc.narl.org.tw

Abstract. The linear system solver plays a key role in scientific computing. This paper evaluates the performances of PETSc (Portable, Extensible Toolkit for Scientific Computation) and HIPS (Portable, Extensible Toolkit for Scientific Computation) solving a singular matrix arising from a bucking problem from UF matrix collection. We employ preconditioned (Level-based ILU) iterative methods (GMRES) for PETSc while HIPS adopts the Schur complement method, named iterative or hybrid mode. Moreover, HIPS proposed HID (Hierarchical interface decomposition) to improve the parallel efficiency. We hope to transfer these results into industrial applications.

Results

These results were performed on ALPS in National Center for High-performance Computing (NCHC). The hardware of computing nodes on NCHC ALPS consists of 600 of Acer AR585 and are connected together with Qlogic InfiniBand in 4x QDR (40Gb). Each node contains 48 cores sharing 128 GB RAM in 4-memory-controller non-uniform memory access architecture.

The singular matrix was proposed by Arthur Raefsky for a buckling problem for container models in 1993. The id number in UF matrix collection [1] is 817. Table 1 presents the matrix characteristic. From Table 1, we know that this matrix is a singular matrix because of her dimension unequal to her rank.

Table 1. Matrix Characteristic

Name	N	NNZ	Condition Number	rank
raefsky4	19779	1316789	3.13E13	19771

The stopping criteria were used the scaled residual $< 1E-7$ [2]. This study first evaluated the performance of different k levels of ILU(k) preconditioned GMRES iterative method via PETSc. Table 2 depicted the results (the number of iterations, elapsed time in second, scaled residual, peak fill-in ratio) of different levels via PETSc. The “peak fill-in ratio” denoted the ratio of the memory to the original one in order to show the memory increment. It had to pay the penalties of the elapsed time and the memory for increasing levels. Because that the infinity norm [3] of the

computed solutions between 7 and 100 levels was near $7.55E-7$, we adopted 7 levels for PETSc hereafter.

Table 2. Results of ILU(k) preconditioned GMRES iterative method via PETSc

ILU(k)	No. iter.	Elapsed time (Sec.)	Scaled residual	Peak fill-in ratio
7	37	2.75E1	8.30E-6	11.70
10	24	4.88E1	3.63E-6	15.89
100	2	1.80E2	4.38E-6	25.19

HIPS used the Schur complement method, HID, threshold ILUT preconditioned GMRES iterative method, named iterative or hybrid mode [4] to solve the linear system. Iterative mode used the ILUT preconditioned GMRES iterative method; however, hybrid mode first divided the linear system into the interior system and the Schur complement. Then, the direct method dealt with the interior system while the ILUT preconditioned GMRES iterative method dealt with the Schur complement. Therefore, the hybrid mode used both the direct and iterative methods to solve this linear system.

Take 100 Krylov subspaces and let the maximal number of iterations be 100. Use the scaled residual to check the accuracy of the computed solutions. Fig. 1 shows the breakdown of the iterative mode compared with PETSc_seq. HIPS outperforms because of HID. Fig. 2 illustrates the breakdown of the hybrid mode. From Fig. 1 and Fig. 2, the hybrid mode lightly outperforms iterative mode when MPI jobs are less than 16; however, the iterative mode shows lightly faster than the hybrid mode on 16 MPI jobs.

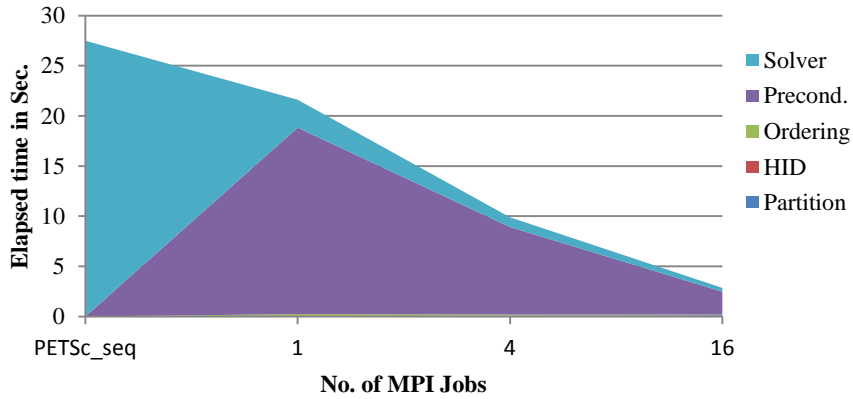


Fig. 1. Breakdown of PETSc and HIPS (iterative mode)

Fig. 3 shows the speedup of the iterative and hybrid modes via HIPS. Both modes via HIPS have near speedup because of problem size limit. Fig. 4 shows the number of iterations via HIPS. Parallel affects the accuracy in the iterative mode; however, the hybrid mode doesn't.

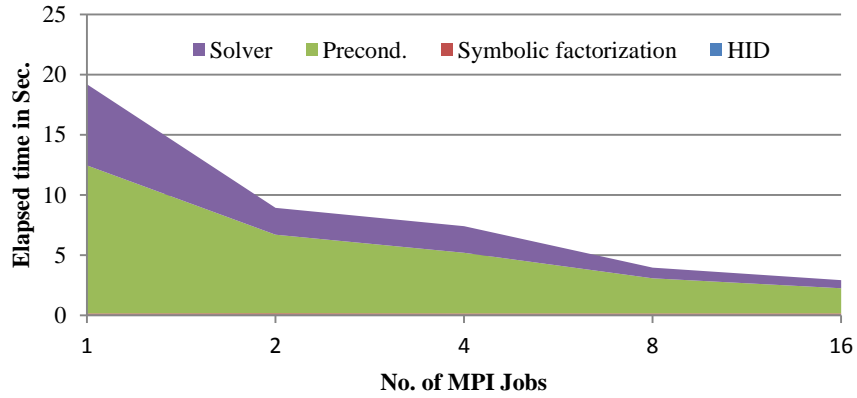


Fig. 2. Breakdown of HIPS hybrid mode

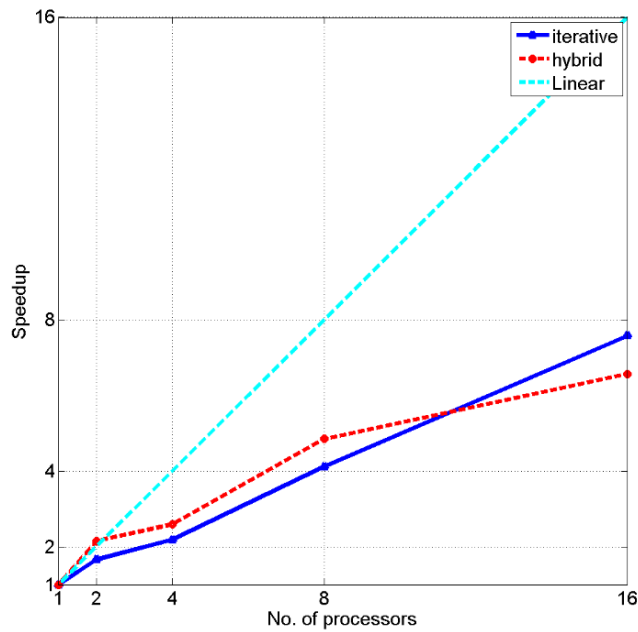


Fig. 3. HIPS Speedup

Conclusion

HIPS outperforms because of HID. HIPS shows good parallel performance, too. Dealing with this problem via HIPS, the hybrid mode shows lightly faster than the iterative mode when MPI jobs are less than 16 while the iterative mode shows lightly faster than the hybrid mode on 16 MPI jobs. Our future work will focus on auto

tuning via PETSc to suggest the best solution among these packages. Moreover, we will systematically analyze the linear solver performances for an application field, for example, solid and structural mechanics.

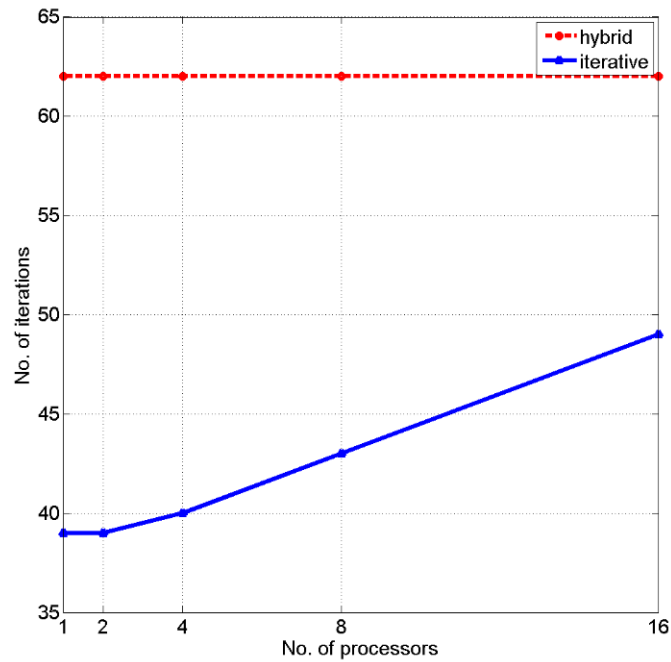


Fig. 4. The number of iterations via HIPS

Acknowledgments. We are grateful to the National Center for High-Performance Computing for computer time and facilities.

References

1. Davis, T. A., Hu, Y.: The university of Florida sparse matrix collection. *ACM Transactions on Mathematical Software*, vol.38 no.1 (2011) 1-25
2. Gould, N. I. M., Scott, J. A., Hu, Y.: A numerical evaluation of sparse direct solvers for the solution of large sparse symmetric linear systems of equations. *ACM Transactions on Mathematical Software*, vol.33 no.2 (2007) 10.
3. Golub, G. H., Loan, C. L.: *Matrix computations*. 3rd edn. Johns Hopkins University Press Baltimore, MD, USA (1996)
4. Gaidamour, J., Hénon, P.: A parallel direct/iterative solver based on a Schur complement approach. in: *Proceedings of the IEEE International Conference on Computational Science and Engineering* (2008) 98–105