A Parallel Hybrid Method of GMRES on GRID System

Ye ZHANG, Guy BERGERE and Serge PETITON

Laboratoire d'Informatique Fondamentale de Lille, USTL
Villeneuve D'Ascq, 59650, France
{Ye.Zhang, bergereg, petiton}@lifl.fr

Abstract

Grid computing focuses on making use of a very large amount of resources from a large-scale computing environment. It intends to deliver high-performance computing over distributed platforms for computation and data-intensive applications. In this paper, we will present an effective parallel hybrid asynchronous method to solve large sparse linear systems by the use of a Grid Computing platform Grid5000. This hybrid method combines a parallel GMRES(m) (Generalized Minimum RESidual) algorithm with the Least Square method that needs some eigenvalues obtained from a parallel Arnoldi algorithm. All of these algorithms run on the different processors of the platform Grid5000. Grid5000, a 5000 CPUs nation-wide infrastructure for research in Grid computing, is designed to provide a scientific tool for computing. We discuss the performances of this hybrid method deployed on Grid5000, and compare these performances with those on the IBM SP series supercomputers.

1. Introduction

Many scientific applications can be transformed to the problem of solving the linear system of the form Ax=b. To solve this nonsymmetric linear system, the GMRES algorithm is a classic iterative method. Saad and Schultz introduced this popular GMRES method in [2]. It is based on the Arnoldi process and allows computing sparse matrices in compressed formats, without loading zeros which are useless for the computing. It has been implemented on parallel systems [1], but this method does not always converge very fast. There are some ways to accelerate the convergence of GMRES. One of those is to calculate in parallel some eigenvalues by the Arnoldi method [3,4]. As soon as they are approximated with a sufficient accuracy, the eigenvalues are used to perform some iterations of the Least Squares method [6] for getting a better initial vector for the next GMRES restarts.

We perform our experiments on two platforms: one is the Grid system, and the other is with the SMP architecture. The Grid is well established as a research domain and proposes technologies that are mature enough to be used for real-life applications. Projects like e-Science, TeraGrid, DEISA and NAREGI, to cite a few, demonstrate that a large scale infrastructure can be deployed to provide scientists fairly easy access to geographically distributed resources belonging to different administration domains[13]. A high performance of large scale computing can be achieved by using this large amount of unoccupied computing resources.

The Grid5000 project has been launched to provide the community of Grid researchers with an unprecedented large-scale infrastructure to study Grid issues under real experimental conditions [14]. It aims at providing a strong reconfiguration, control and monitoring infrastructure, transforming the full system into a scientific instrument. It is implemented as a nation wide cluster of clusters over 9 sites in France. Every site is equipped with a cluster ranging from 100 to 1000 CPUs and all sites are connected by the high speed network RENATER.

The IBM RS6000 SP series supercomputers with the SMP architecture are widely used in the world for various scientific and commercial applications. Equipped with high speed processors and high bandwidth interconnections between the nodes, it provides an excellent scientific calculation environment.

In this paper, we present the distributed hybrid method GMRES(m)/LS-Arnoldi which is well implemented on the GRID system Grid'5000. We will furthermore compare the performances on Grid'5000 with those on IBM SP series supercomputers.

This paper is organized as follows. The numerical methods used in our hybrid method will be present in section 2. In section 3, we introduce the implementation on Grid'5000. In section 4, we present the results obtained on the platform Grid'5000 and IBM SP series supercomputers. At the same time we analyze and compare the performance of computing from the

1-4244-0910-1/07/$20.00 ©2007 IEEE

viewpoint of the method and the viewpoint of the hardware platform. Finally, in section 5, we present a summary and discuss directions for future research.

2. The GMRES(m)/LS-Arnoldi hybrid parallel method

2.1. GMRES method

The GMRES (Generalized Minimum RESidual) method was proposed by Saad and Schultz[2] in 1986. It is a Krylov method for solving non-symmetric linear systems. The m^{th} iterate x_m of GMRES is the solution of the least squares problem: $\text{minimize}_{x \in x_0 + K_m(A,r_0)} \| b - Ax \|_2$, where $r_0 = b - Ax_0$ is the residual of the initial solution. The Arnoldi process applied to $K_m(A, r_0)$ builds $V_{m+1} = [V_m, v_{m+1}]$, an orthonormal basis of $K_m(A, r_0)$, the m+1 by m matrix $\overline{H}_m$ and $\beta = \|r_0\|_2$. These matrices satisfy the relation $AV_m = V_{m+1}\overline{H}_m$. The iterate x_m can be written as $x_m = x_0 + V_m y_m$, where $y_m \in \Re^m$ is the solution of the least squares problem: $\text{minimize}_{y \in \Re^m} \left\| \beta e_1 - \overline{H}_m y \right\|_2$

In the GMRES algorithm the number of vectors requiring storage increases with m. One way to address this problem is using the algorithm iteratively, by finding the iterate x_m, and restarting the algorithm with the initial guess $x_0 = x_m$, until convergence. Thus, we obtain the restarted GMRES(m) after iteration of GMRES.

Algorithm GMRES(m):

1. Start: choose x_0 an initial guess of the solution,
   m is the solution of Krylov subspaces, and ε the tolerance,
   compute $r_0 = b - Ax_0$
2. Apply Arnoldi process to $K_m(A, r_0)$
3. Compute $y_m = \arg\min_{y \in \Re^m} \| \beta e_1 - \overline{H}_m y \|_2$ with QR factorization,
   and set $x_m = x_0 + V_m y_m$, $r_m = b - Ax_m$
4. Restart: if $\| r_m \|_2 \le \varepsilon$
   else set $x_0 = x_m$, $r_0 = r_m$, and goto 2.

2.2. The hybrid algorithm GMRES(m)/LS(k,l)

The hybrid algorithm applies the GMRES(m) algorithm as the basic part to solve the linear system. In addition, it also integrates two methods: Arnoldi and Least Square. The idea of the whole process is to calculate in parallel some eigenvalues on other machines by the Arnoldi method [5]. As they will be approximated with a sufficient accuracy, eigenvalues are used to perform some iterations of the Least Squares method [6] in order to obtain a better initial vector for the next GMRES iterations.

At first, we describe the Arnoldi method and the method Least Square.

Algorithm: Arnoldi's method

1. Start: choose v an initial vector, m the dimension of krylov subspaces,
   and d, the number of desired dominant eigenvalues, with the threshold ε.
2. Apply Arnoldi process to $K_m(A, v)$.
3. Compute the eigenvalues(λ_i, $1 \le i \le d$) and the associated
   Eigenvectors (y_i, $1 \le i \le d$) of H_m.
4. Set $u_i = V_m y_i$, for i=1, ···, d, the Ritz vectors.
5, Compute $\rho_i = \left\| \lambda_i u_i - A u_i \right\|_2$, $1 \le i \le d$.
6. Restart: if $\max_{i=1}^{d} |\rho_i| < \varepsilon$ stop else set $v = \sum_{i=1}^{d} \text{Re}(u_i)$, and goto 2.

The Least Square method can be written as follows: $\tilde{x} = x_0 + P_k(A) r_0$ where x_0 an initial approximation, r_0 its residual, and P_k is a polynomial of degree k-1. Let $\mathcal{P}_k^1$ be the set of the real polynomials p of degree k, such that p(0)=1, and define the polynomial $R_k \in \mathcal{P}_k^1$ by $R_k(z) = 1 - zP_k(z)$. Then the residual of the iterate $\tilde{x}$ is $\tilde{r} = R_k(A) r_0$.

In general, we do not have the whole spectrum of A, but only some eigenvalue estimates contained in a convex hull H. H is constructed such as it does not contain the origin. Smolarski and Saylor [11] proposed to find R_k minimizing a weighted L_2-norm on the space of real polynomials, with a suitable weight function w, defined on the boundary of H. We obtain the following least squares problem $\min_{R_k \in P_k^1} \left\| R_k \right\|_w$.

The obtained polynomial $P_k = \sum_{i=0}^{k-1} \eta_i t_i$ is expressed in the scaled and shifted Chebyshev basis defined by $t_j(\lambda) = T_j\left(\frac{\lambda - c}{d}\right) / T_j\left(\frac{a}{d}\right)$ j=0,1,… This is the best basis of polynomials on the ellipse $\varepsilon(c, d, a)$ of smallest area enclosing H (see [10] and [5] for an algorithm computing this optimal ellipse). For more details, see [6].

The hybrid algorithm GMRES(m)/LS(k,l) can be given as follows:

Algorithm: GMRES(m)/LS(k, l)

1. Start: Choose x_0, m, m′ the dimension of Krylov subspaces. k is the degree of the least squares polynomial, ε the threshold and l the number of the successive applications of the Least Squares method.

2. Compute x_m, the m^{th} iterate of GMRES starting with x_0,

if $\| b - Ax_m \|_2 < \varepsilon$ Stop else set $x_0 = x_m$, $r_0 = b - Ax_0$.

3. Perform m′ iterations of the Arnoldi process starting with r_0,

And compute the eigenvalues of $H_{m'}$.

4. Compute the least squares polynomial P_k on the boundary of H, the hull convex enclosing all computed eigenvalues.

5. For $j = 1, \cdots, l$ do

Compute $\tilde{x} = x_0 + P_k(A)r_0$, and set $x_0 = \tilde{x}$, $r_0 = b - Ax_0$.

end do

6. Restart: if $\| r_0 \|_2 < \varepsilon$ Stop, else goto 2.

3. Implementation on Grid'5000

Grid'5000 is not a Grid, but a highly configurable, controllable and monitorable instrument that can be configured to work as a real Grid. There are 5000 CPUs distributed over 9 sites in France. Every site hosts a cluster and all sites are connected by RENATER through VLANS implemented by MPLS at level 2 [14]. Besides, on account of the security, the sites of Grid'5000 are not directly connected to the Internet, but all communication packets fly without limitation between Grid'5000 sites.

Firstly, we customize our environment MPI on Grid'5000 for our experiments. Next, we distribute our algorithm on the processors reserved in one or some sites of Grid'5000.

We use primarily the clusters of the site Orsay and the site Rennes. These two sites are interconnected by the high speed network RENATER. After deploying our environment, we have the copy of the program and the data needed for calculation on all the processors reserved. All the sparse matrices are stored in the compressed format CSR (Compress Sparse Row) for saving the memory and reducing the communication on the network.

The most processors reserved are used to run the algorithm GMRES(m) by the way of the SPMD model with an administrative process and p identical calculation processes. The calculation processors read directly their own data and execute the method GMRES(m), communicating with their brother processes.

The processors dedicated to the parallel package "PARPACK" are in charge of the reception of residuals, the projection of Arnoldi and the calculation of eigenvalues, independently of the processes GMRES.

Only one processor is in charge of the sequential part (the LS method and the sorting of eigenvalues) because of the small set of data for calculation. The parameters "Least Square" obtained are then sent to the processors executing the algorithm GMRES(m) later.

The whole process and the relationship of the communication between the three parts are presented in Figure 1. We also do our implementations according to this schema.

4. Numeric Results and Analysis

The detailed configuration of the sites Orsay and Rennes are showed in Table 1. The bandwidth and the average latency inside the cluster of Orsay and those between these two sites are present in Table 2. Table 3 shows the configuration of the IBM SP series supercomputers.

Number of PCs	CPU	Memory
the site Orsay		
216	Dual AMD Opteron 246, 1.95GHz	1.96G
121	Dual AMD Opteron 250, 1.95GHz	1.96G
the site Rennes		
99	Dual AMD Opteron 246, 1.96GHz	1.96G
64	Dual AMD Opteron 248, 2.14GHz	1.96G
64	Intel Xeon IA32, 2.33GHz	0.99G
32	PowerPC, 1.95GHz	1.5G

Table 1. Resources of Orsay and Rennes

	bandwidth	latency
Orsay	941 Mb/s	0.09ms
Orsay – Rennes	27.25 Mb/s	9ms

Table 2. The bandwidth and the average latency between the nodes

Number of node	CPUs of node	Memory /node
SP3		
4	16 Power 3 NH2, 375MHz	16G
SP4		
2	32 Power 4, 1.3GHz	64G
2	32 Power 4, 1.7GHz	64G
5	32 Power 4, 1.7GHz	32G

Table 3. Configuration of the SP3 and SP4

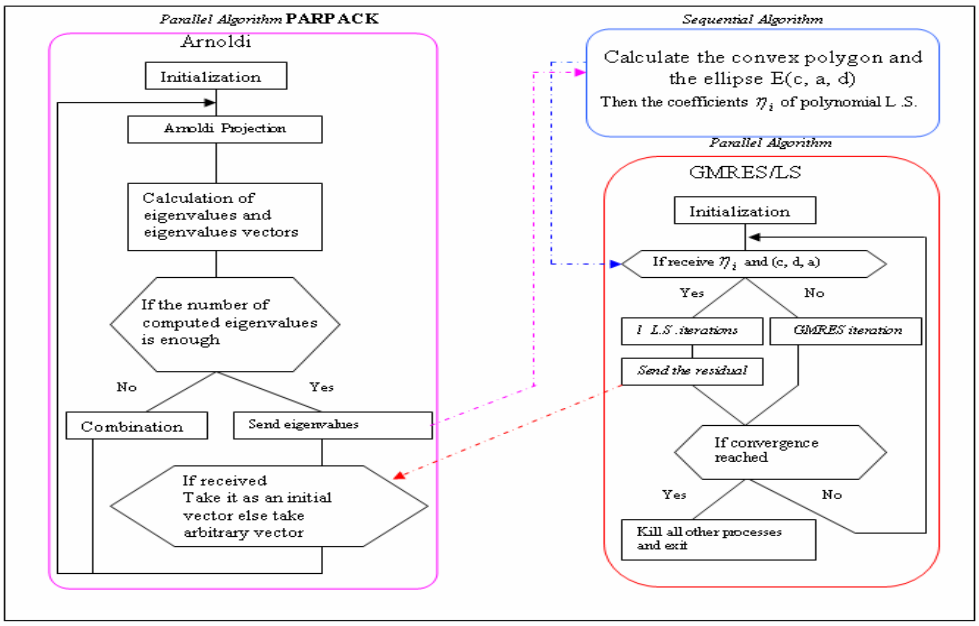

Figure 1. General scheme of asynchronous hybrid GMRES/LS-Arnoldi process

We will analyze the numeric results obtained from the viewpoint of the algorithm itself and the hardware platform. In our experiments, we use the sparse matrices derived from the site "MatrixMarket". And the comparisons are made in the same condition of the parameters of the method.

4.1. Analysis and comparisons between the hybrid method and the classic method GMRES(m)

According to the experiments results (see figure 2), it is clear that the global computation time of the hybrid parallel method is better than the parallel GMRES(m) itself. And the speed up can even be spectacular when the convergence of GMRES itself is difficult. It is because we use continually the new better approximate initial vector obtained by the method LS for the restart of the next GMRES iterations. After each iteration, the processors executing GMRES(m) always check if the parameters of the method LS arrive. In this case, the GMRES algorithm is then stopped, the processes perform the parallel part of the hybridizations, and GMRES(m) restart with the obtained iterate.

When the hybrid computation using LS parameters by the GMRES/LS processes occurs, we then notice an obvious decrease of the residual, usually after a temporary increase of the residual. Thereby, too frequent sending of LS parameters damage the efficiency, as each LS iterate influences many GMRES iterations. When the peaks are high and nearby, divergence may even occur.

The implementation of this hybrid method allows to take advantage of available parallelism. Both GMRES(m) and the hybrid method all have an optimal number of processors involved in the calculation. When we put more processors into the calculation after this threshold, that time we used to get the resolution increases contrarily. Because the GMRES processes require intensive communications and multiple synchronizations, the time of calculation gained by the acceleration with more processors involved is less than the time consumed by the communications among the processors. However, for the hybrid method, we can increase the used parallelism because of the join of the parallel processes Arnoldi. The parallel Arnoldi algorithm calculates the eigenvalues necessary for the hybridization independently of the processes GMRES(m). Additionally, although there is also the optimal number of processors for the Arnoldi algorithm, we can still increase moderately the number of processors beyond this optimal number. As mentioned above, sending eigenvalues too frequently is not desirable, so the Arnoldi computation needs to be very efficient.

4.2. Analysis and comparisons between the platform Grid'5000 and the supercomputer

In the Table 4, we easily remark that the global computation time on Grid'5000 is much shorter not only for the method GMRES(m) itself but also for our hybrid method. It is because the hardware configuration of Grid'5000 is obviously better than the supercomputer SP3 (see Table 1, Table 3)

We observe that on the SP3 supercomputer there are more peaks (see Figure 3). In other word, the hybrid computation using LS parameters on the SP3 occurs more times than that on Grid'5000. Because on the Grid5000 the GMRES(m) iterations are faster, there are more GMRES(m) iterations between two receipts of the LS parameters taken into account by the processes GMRES. We remark also that on SP3 when nG (number of processors for the running of the algorithm GMRES) is 2, the divergence occurs. However on Grid'5000 the convergence is realized. This result accords with the theory that the group of the processors running the algorithm GMRES function too slowly to have enough time to deal with the LS parameters before the reach of the new LS parameters.

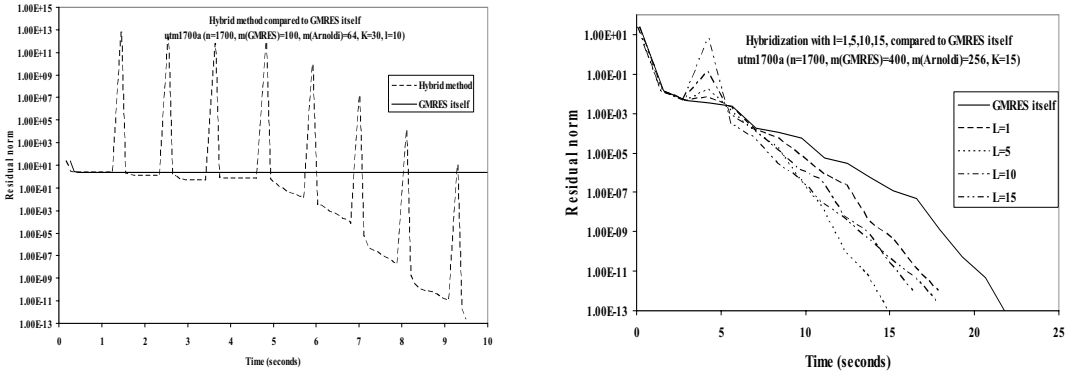

Figure 2. The comparison of hybrid method and GMRES itself of "utm1700a" on Grid'5000

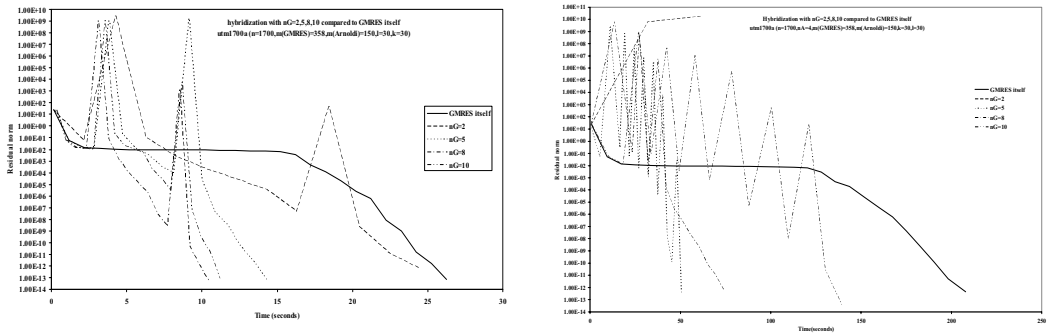

Figure 3. The comparison of hybrid method between two platforms Grid'5000 and SP3

	global computation time				
	GMRES itself	nG=2	nG=5	nG=8	nG=10
SP3	207.85s	∞	50.51s	74.2s	139.3s
Grid5000	26.25s	24.45s	14.3s	10.5s	11.2s

Table 4. The time comparison of hybrid method between two platforms Grid'5000 and SP3

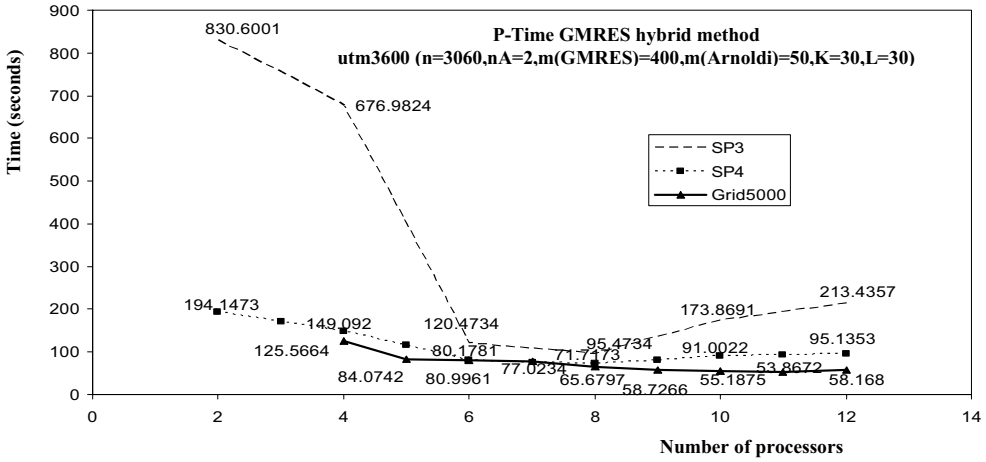

Figure 4. The influence of the number of processors for utm3600

	N=4	N=6	N=8	N=10	N=12
SP3	676.9824s	120.4734s	95.4734s	173.8691s	213.4357s
SP4	149.092s	80.1781s	71.7173s	91.0022s	95.1353s
Grid'5000	125.5664s	80.9961s	65.6797s	55.1875s	58.168s

Table 5. The influence of the number of processors for utm3600

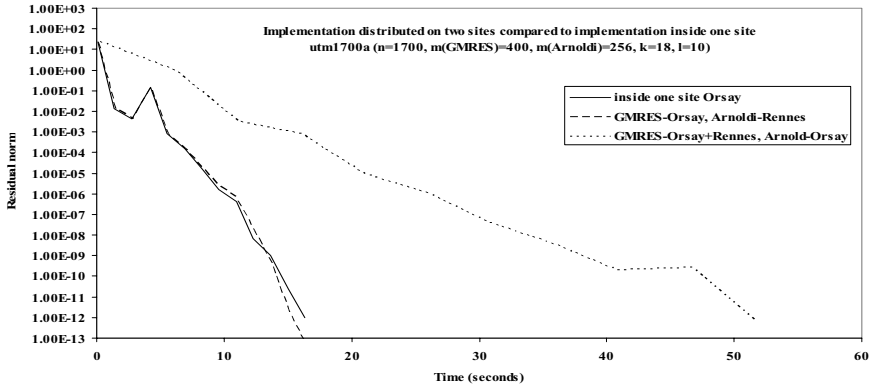

Figure 5. The comparison of the implementation on Orsay and on two sites (Orsay and Rennes)

Viewed from the parallelism granularity perspective, the results (In Figure 4 and Table 5) show that the global computation time on Grid'5000 is better than the IBM RS6000 SP4 too. We also notice that on Grid'5000 and SP4 the evolutions of the computing time along with the number of processors are much smoother than on SP3. It is due to their excellent hardware configurations. We can also notice this phenomenon in Table 4. Additionally, on Grid'5000 the optimal computation time is obtained when the number of processors is 10, and on the supercomputers SP3 and SP4 the optimal number of processors is 8. In other words, we can take advantage of available parallelism on Grid'5000. It can be explained by the fact that when we use more processors, the time consumed by the communication do not increase much more thanks to the high speed network of Grid'5000, and this increased time can be compensated by the diminution of the computing time.

Moreover, we tested our hybrid algorithm with two network configuration on Grid'5000: inside the cluster of

one site Orsay or Rennes and inter-sites Orsay and Rennes. We can notice that there is almost no difference between the implementation only inside one site Orsay and the implementation distributed inter-components on two sites (see Figure 5). However, when we distribute the part GMRES on two sites, the performance drops. The experiment further shows that the communications among the processors taking charge of the algorithm GMRES(m) are intense, and the communications between the components are relatively little. So in the future when we use the geographically distributed computing resources, the implementation distributed inter-components will have relatively small influence for the whole computing performance.

As mention above, it can be concluded that combining the advantage of the hybrid method GMRES(m)/LS-Arnoldi (as shown in Figure 2) and the excellent Grid platform Grid'5000 (as shown in Figure 3), we can achieve a higher performance of computing.

5. Conclusion

We implemented our algorithm for the hybrid method GMRES(m)/LS-Arnoldi in two computing platforms: Grid'5000 with the environment MPI and a supercomputer system IBM SP series.

Without any doubt, the experimental results show the interest of the hybrid method. We have obtained very important convergence accelerations. And thanks to the low amount of communications between its components, our hybrid method takes advantage of available parallelism unusable with the classic method.

According to the results of the comparison with the supercomputer system IBM SP series, the platform Grid'5000 shows its good performance thanks to its excellent hardware and network configuration.

In future, we will continue our experiments on Grid'5000, reconfiguring the others distribution computing environments. For the hybrid method, we will extend it to the scientific problems of very large size. In addition, we will do more tests on the other supercomputers or cluster (i.e. the supercomputer in Japan, IBM cell in France), and apply our experiments in an environment with a WAN based configuration.

Reference

[1] R.D.Da Cunha and T.Hopkins. A parallel implementation of the restarted GMRES iterative algorithm for nonsymmetric systems of linear equations. Advances in Computational Mathematics. 2(1994).pp261-277

[2] Y.Saad. M. H. Schultz. GMRES: A Generalized GMRES Algorithm for Solving Nonsymmetric Linear Systems, SIAM J. Sci. Statist. Compt., 7(1986) 856-869

[3] G.Edjlali, N.Emad, S .Petiton, Hybrid methods on network of heterogeneous computers, 14th IMACS World Congress, 1994.

[4] Y.Saad, Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices, Linear Algebra Appl., 34(1980), pp.269-295

[5] Y.Saad, Numerical methods for large eigenvalue problems, Manchester University Press, Manchester (1992)

[6] Y.Saad, Least Squares Polynomials in the Complex Plane and their Use for Solving Nonsymmetric Linear Systems, SIAM J. Sci. Statist. Comput., 7(1987),pp 155-169.

[7] Haiwu HE, Guy Bergere, Serge Petiton, A parallel asynchronous hybrid method to accelerate convergence of a linear system, DCABES 2004, Distributed Computing and Algorithms for Business, Engineering, and Sciences

[8] Haiwu He, Guy Bergere, Serge Petiton, GMRES Method on Lightweight GRID System, ISPDC 2005, the 4th International Symposium on Parallel and Distributed Computing

[9] A.Essai, G.Bergere, Serge Petiton. Heterogeneous parallel hybrid GMRES/LS-Arnold method. Ninth SIAM Conference on Parallel Processing for Scientific Computing, 1999

[10] T.A.Manteuffel, The Tchebychev iteration for nonsymmetric linear systems, Numer. Math., 28 (1997), pp.307-327

[11] D.C.Smolarski, P.E.Saylor, An optimum iterative method for solving any linear system with a square matrix, BIT, 28(1988), pp.163-178

[12] Y.Georgiou, O.Richard, P.Neyron, G.Huard, and C.Martin. A batch scheduler with high level components. In proceedings of CCGRID'2005. IEEE Computer Society, 2005

[13] C.Franck, C.Eddy, D.Michel, D.Frederic, J.Yvon, Grid'5000: a large scale and highly reconfigurable Grid experimental testbed, International Journal of High Performance Computing Applications, 20(4):481-494, 2006

[14] Q.Benjamin, C.Franck, A survey of Grid research tools: simulators, emulators and real life platform, in proceedings of the 17th IMACS World Congress (IMACS 2005), Paris, France, 2005