Research Reports on Mathematical and Computing Sciences
Series B : Operations Research

Department of Mathematical and Computing Sciences
Tokyo Institute of Technology
2-12-1 Oh-Okayama, Meguro-ku, Tokyo 152-8552 Japan

High Performance Grid and Cluster Computing for Some Optimization Problems

Katsuki Fujisawa, Masakazu Kojima, Akiko Takeda and Makoto Yamashita

Research Report B-396, October 2003

Abstract.
The aim of this short article is to show that grid and cluster computing provides tremendous power to optimization methods. The methods that the article picks up are a successive convex relaxation method for quadratic optimization problems, a polyhedral homotopy method for polynomial systems of equations and a primal-dual interior-point method for semidefinite programming problems. Their parallel implementations on grids and clusters together with numerical results are reported.

Key words.
Grid, Cluster, Semidefinite Program, Semidefinite Program Relaxation, Polynomial Equations, Polyhedral Homotopy Method

High Performance Grid and Cluster Computing for Some Optimization Problems

Katsuki Fujisawa (fujisawa@r.dendai.ac.jp)
Tokyo Denki University, Department of Mathematical Sciences
National Institute of Advanced Industrial Science and Technology,
Grid Technology Research Center, Japan

Masakazu Kojima (kojima@is.titech.ac.jp)
Akiko Takeda (takeda@is.titech.ac.jp)
Makoto Yamashita (yamashi9@is.titech.ac.jp)
Department of Mathematical and Computing Sciences,
Tokyo Institute of Technology, Japan.

Abstract. The aim of this short article is to show that grid and cluster computing provides tremendous power to optimization methods. The methods that the article picks up are a successive convex relaxation method for quadratic optimization problems, a polyhedral homotopy method for polynomial systems of equations and a primal-dual interior-point method for semidefinite programming problems. Their parallel implementations on grids and clusters together with numerical results are reported.

1 Introduction

The main objective of optimization is to find the best solution among all feasible ones. Optimization has a variety of practical applications in various fields such as operations research, engineering, science, biology and economics. Many combinatorial or nonconvex optimization problems are known to be $\mathcal{NP}$-hard, which means that there exists no algorithm that finds an optimal solution in polynomial time unless $\mathcal{P} = \mathcal{NP}$. Hence, solving large scale combinatorial or nonconvex optimization problems requires a huge amount of computational time and resources, and the size of such problems that we can solve has been limited. As computing resources continue to improve, however, optimal solutions for larger size optimization problems become more achievable. In particular, grid computing technology has recently received much attention as a powerful and inexpensive methodology for solving large scale optimization problems that an existing single CPU cannot process. To solve large scale optimization problems, we need new computing infrastructure which enables us to easily access to computational resources including hardware and software library distributed across a wide area network.

In this paper, we present grid and cluster computing for some optimization problems with numerical results.

We focus our attention on GridRPC, which is an RPC system redesigned for the Grid. Some GridRPC systems such as Netsolve and Ninf are widely used. The Ninf system [4] developed by AIST (National Institute of Advanced Industrial Science and Technology, Japan) employs a client-server model, where server and client machines are connected via a local area network or the Internet. We implemented highly parallel algorithms for some optimization problems and polynomial systems of equations with the use of the Ninf system on several PC clusters connected via a high speed local area network and/or the Internet.

2 Parallel Successive Convex Relaxation Method for Nonconvex Quadratic Optimization Problems

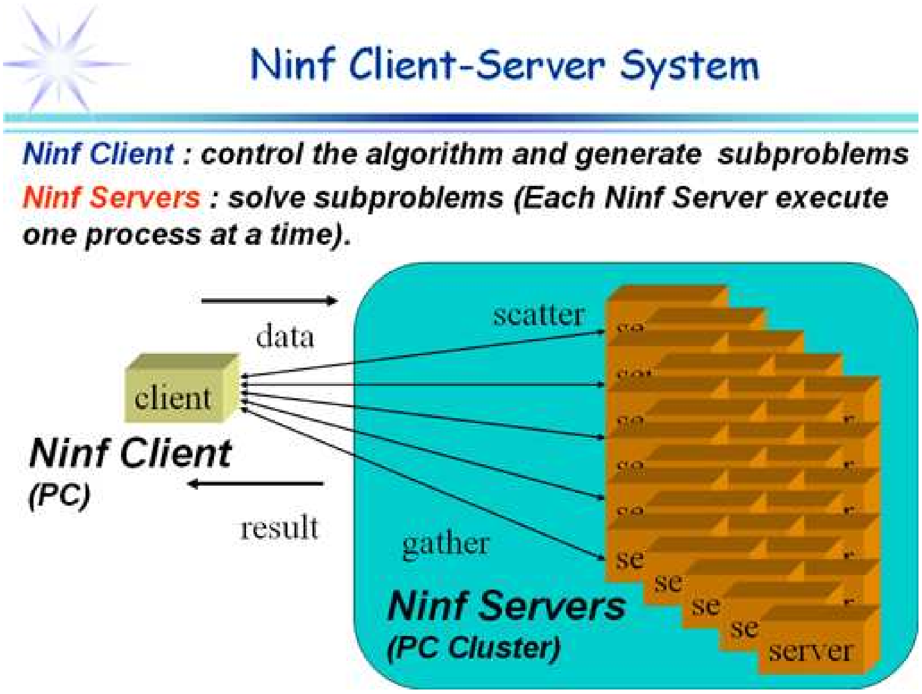

Figure 1: Ninf client-server model

Let $\mathbb{R}^n$ and $\mathcal{S}^n$ denote the n-dimensional Euclidean space and the space of $n \times n$ real symmetric matrices, respectively. A general QOP (Quadratic Optimization Problem) is described in the the following form:

$$\text{(QOP)} \quad \left| \begin{array}{ll} \max & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} & \gamma_i + 2\boldsymbol{q}_i^T\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{Q}_i\boldsymbol{x} \leq 0 \ (i = 1, \dots m), \end{array} \right. \tag{1}$$

where $\boldsymbol{c} \in \mathbb{R}^n$, $\gamma_i \in \mathbb{R}$, $\boldsymbol{q}_i \in \mathbb{R}^n$ and $\boldsymbol{Q}_i \in \mathcal{S}^n$ $(i = 1, \dots, m)$. When a given QOP has a quadratic objective function such as $\gamma_0 + 2\boldsymbol{q}_0^T\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{Q}_0\boldsymbol{x}$, we can transform it into QOP (1) by replacing the quadratic objective function with a new variable t and adding $-\gamma_0 - 2\boldsymbol{q}_0^T\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{Q}_0\boldsymbol{x} + t \leq 0$ to the set of inequality constraints. A general QOP is known as one of the most significant nonlinear programming problems. QOPs cover not only economics and engineering applications but also various important nonconvex mathematical programs. Because of their theoretical and computational difficulties, however, solvable QOPs had been limited to convex QOPs where all $\boldsymbol{Q}_i$s are assumed to be positive semidefinite or general QOPs with small size.

The SCRM (Successive Convex Relaxation Method) proposed by Kojima-Tuncel [3] is a powerful numerical method to compute upper bounds of general QOPs applying SDP (Semidefinite Programming) relaxations repeatedly. The SCRM generates and solves a large number of SDP problems at each iteration. Takeda et al. [5] reported that the SCRM can deal with some larger size QOPs through numerical experiments on a PC cluster. To get more accurate upper bounds and/or to process larger scale of QOPs, we need more and more computing resources. We implemented a highly parallel SCRM on the Ninf system.

Figure 1 shows that a Ninf client-server model. The Ninf client controls the SCRM applied to a QOP and generates a large number of subproblems each of which forms an SDP problem at each iteration. All generated SDP problems are sent to the Ninf server machines. Then each Ninf server solves the SDP using the software SDPA [8]. Note that each Ninf server solves only one SDP at a time. After finishing the execution of the SDPA, the result is sent back to the Ninf client machine.

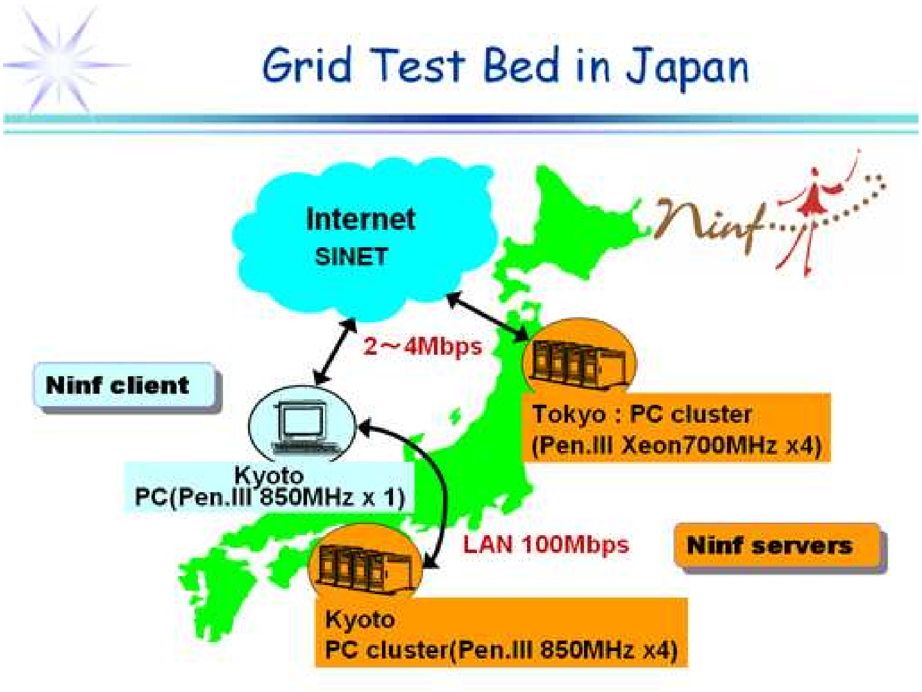

Figure 2: Grid Environment(Kyoto $\Longleftrightarrow$ Tokyo)

We now show some numerical experiments of the SCRM on a Grid environment. Figure 2 illustrates that we have two PC clusters which are located in Kyoto and Tokyo. The Kyoto cluster has 4 nodes which are connected to a Ninf client PC through LAN (100BASE-TX). The Tokyo cluster has 4 nodes which are connected to the Ninf client PC through the Internet called SINET. We estimate the average network speed of the SINET between Kyoto and Tokyo is about 2 to 4 Mbps.

Table 1 shows that numerical results of the SCRM on the grid environment illustrated in Figure 2. The first row "# subproblems" denotes that the total number of SDPs which the Ninf servers in the Kyoto cluster and the Tokyo cluster solved, respectively. The second row denotes that the total execution time of the SDPA in each cluster. The total execution time of the Kyoto cluster is almost the same as that of the Tokyo cluster. Note also that the total transmitting time between the Ninf client and Ninf servers is extremely small. ¿From the Table 1, we can say that the SCRM is well paralleled in the grid environment.

3 Parallel Polyhedral Homotopy Method

The software package "PHoM" [2] is designed to find all isolated (real and complex) solutions of polynomial system of equations (such as the cyclic n problem, see Table 2) using a polyhedral homotopy method. The package consists of three modules. The first module

Table 1: Numerical results on grid environments

Prob. name	LC80-144		BLevel20-3	
Ninf Server	Kyoto	Tokyo	Kyoto	Tokyo
# subproblems	1795	1333	883	597
total execution time(sec.)	16486.5	16395.6	1053.6	984.2
total trans. time C $\rightarrow$ S(sec.)	0.150	0.117	0.046	0.028
total trans. time C $\leftarrow$ S(sec.)	0.275	0.069	0.130	0.182

which we call "StartSystem" constructs a family of polyhedral-linear homotopy functions. Takeda, Kojima and Fujisawa [6] implemented a highly paralleled StartSystem using the Ninf. The second module traces the homotopy paths to compute all isolated solutions of a polynomial system of equations. The third module verifies whether all isolated solutions of the polynomial system of equations have been computed correctly. We present below some numerical results of a parallel implementation of the second module on PC clusters.

Table 2: cyclic n problem

$$
\begin{array}{l}
x_1 + x_2 + \cdots + x_n = 0 \\
x_1x_2 + x_2x_3 + \cdots + x_nx_1 = 0 \\
x_1x_2x_3 + \cdots + x_nx_1x_2 = 0 \\
\cdots \\
x_1x_2x_3 \cdots x_n - 1 = 0,
\end{array}
$$

Table 3: Numerical results of the homotopy method

	cyclic-11		cyclic-12		cyclic-13	
# CPUs	time(s.)	sp-up-raito	time(s.)	sp-up-ratio	time(s.)	sp-up-ratio
2	47,345	1.00				
4	23,674	2.00				
8	11,852	3.99				
16	5,927	7.99				
32	2,967	15.96				
64	1,487	31.84	2,592	1.00		
128			1,332	1.95	10,151	1.00
256			703	3.69	5,191	1.95
# paths traced	16,796		41,696		208,012	

We also employ the Ninf client-server system, illustrated in Figure 1, for the parallel implementation of the second module of the PHoM. The Ninf servers trace the homotopy paths by applying the predictor-corrector method. Table 3 shows numerical results of the second module for the cyclic problems with dimensions $n = 11$, 12 and 13. There are many homotopy paths, which can be traced independently in parallel; specifically the cyclic 13 problem has 208,012 homotopy paths. We use two PC clusters which are located in Tokyo Institute of Technology. All homotopy paths of the cyclic 11 problem were computed in the Presto I cluster which has 64 nodes (CPU : Celeron 500MHz), while all homotopy paths of the cyclic 12 and 13 problems were computed on the Presto III cluster which has 256 nodes and 512 CPUs (each CPU is Athlon 1900+). From Table 3, we observe that the second module of the PHoM is quite suitable for parallel computation on PC clusters. Note that more than 5,000 seconds were required to compute all homotopy paths of the cyclic 13 problem even when we used 256 CPUs simultaneously. Therefore parallel computing is indispensable to process larger size polynomial systems of equations.

4 Parallel Implementation of Semidefinite Programming

In the last decade, SDP (Semidefinite Programming) problems have been intensively studied both in theoretical and practical aspects in various fields such as interior-point methods, combinatorial optimization, control and systems, robust optimization and quantum chemistry. The SDPA (SemiDefinite Programming Algorithm) [8] is an optimization software, written by the C++ language, of a PDIPM (primal-dual interior-point method) for solving the standard form SDP problem. The SDPA incorporates a special data structure for handling block diagonal data matrices and an efficient method proposed by Fujisawa, Kojima and Nakata [1] for computing search directions when problems to be solved are large scale and sparse. In many applications, however, SDP problems become too large for SDP software packages including the SDPA to solve on a single processor. It is well-known that the PDIPM for SDP problems has two major time consuming parts at each iteration even if we exploit the sparsity of the date matrices. The first part is the computation of the so-called Schur complement matrix. The second part is the Cholesky factorization of the Schur complement matrix. These two parts are called ELEMENTS and CHOLESKY in Figure 3 below.

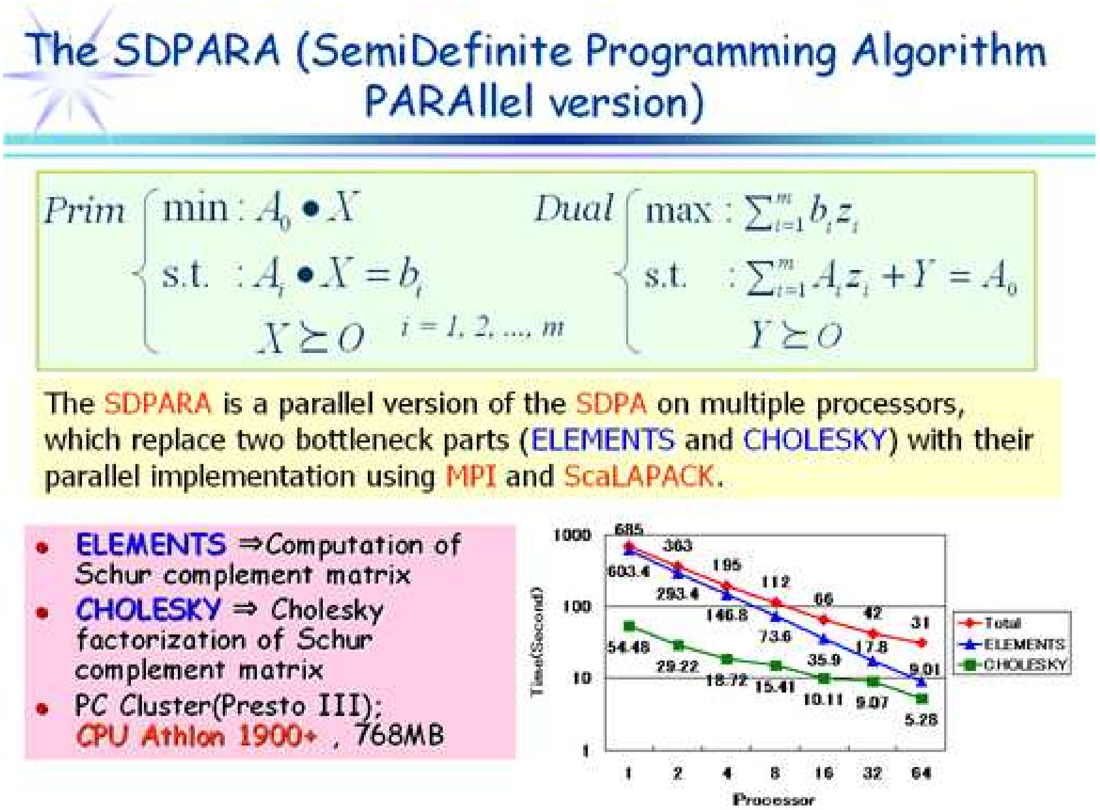

Figure 3: The SDPARA (SemiDefinite Programming Algorithm PARAlell Version)

The SDPARA (SemiDefinite Programming Algorithm PARAllel version) [9] is a parallel version of the SDPA on multiple processors and distributed memory, which replaces these two bottleneck parts mentioned above by their parallel implementation using MPI and ScaLAPACK; the SDPARA computes each row of the Schur complement matrix independently in parallel and applies a parallel Cholesky factorization provided by ScaLAPACK to the Schur complement matrix. Figure 3 explains the main features of the SDPARA. The SDPARA on a PC cluster attains high scalability for large scale SDPs. The largest problem which we could solve generates a $24,503 \times 24.503$ Schur complement matrix. For more details, see the Yamashita et al. [9].

5 Future Works

We are planning to apply the Grid and Cluster computing to some other optimization problems such as the vehicle routing problem with time windows and more general mixed integer programming problems. So far we have mainly employed the Ninf GridRPC system. Tanaka et al. [7] have recently redesigned the Ninf and implemented a GridRPC system called Ninf-G. The Ninf-G is a full re-implementation of Ninf using Globus Toolkit. This takes advantage of high interoperability with some other Globus-based Grid systems. We have plans to reconstruct our software packages such as the SCRM and the PHoM using the Ninf-G and Globus Toolkit.

References

[1] K. Fujisawa, M. Kojima and K. Nakata, "Exploiting sparsity in primal-dual interior-point methods for semidefinite programming," *Mathematical Programming* **79** (1977) pp. 235–253.

[2] T. Gunji, S. Kim, M. Kojima, A. Takeda, K. Fujisawa and T. Mizutani, "PHoM – a Polyhedral Homotopy Continuation Method." December 2002, revised January 2003. To appear in *Computing*.

[3] M. Kojima and L. Tuncel, "Cones of Matrices and Successive Convex Relaxations of Nonconvex Sets," *SIAM Journal on Optimization* **10** (2000) pp.750–778.

[4] M. Sato, H. Nakada, S. Sekiguchi, S. Matsuoka, U. Nagashima and H. Takagi, "Ninf: A Network based Information Library for a Global World-Wide Computing Infrastructure," *HPCN'97* **LNCS-1225** (1997) pp.491–502.

[5] A. Takeda, K. Fujisawa, Y. Fukaya and M. Kojima, "Parallel Implementation of Successive Convex Relaxation Methods for Quadratic Optimization Problems," *J. of Global Optimization* **24** (2002) pp237–260.

[6] A. Takeda, M. Kojima and K. Fujisawa, "Enumeration of All Solutions of a Combinatorial Linear Inequality System Arising from the Polyhedral Homotopy Continuation Method," *Journal of the Operations Research Society of Japan* **45** (2002) pp. 64–82.

[7] Y. Tanaka, H. Nakada, S. Sekiguchi, T. Suzumura and S. Matsuoka, "Ninf-G : A Reference Implementation of RPC-based Programming Middleware for Grid Computing," *J. of Grid Computing* **1** (2003) pp.41–51.

[8] M. Yamashita, K. Fujisawa and M. Kojima, "Implementation and Evaluation of SDPA 6.0 (SemiDefinite Programming Algorithm 6.0)," *Journal of Optimization Methods and Software* **18** (2003) pp.491–505.

[9] M. Yamashita, K. Fujisawa and M. Kojima, "SDPARA : SemiDefinite Programming Algorithm PARAllel Version," *Journal of Parallel Computing* **29** (2003) pp.1053–1067.