# Peta-scale General Solver for Semidefinite Programming – Extremely Large-scale Parallel Cholesky Solver –

Kyushu University & JST CREST *FUJISAWA Katsuki
Tokyo Institute of Technology & JST CREST ENDO Toshio

## 1. Introduction

In the last two decades, semidefinite programming (SDP) problems have been intensively studied both in theoretical and practical aspects in a wide range of fields such as combinatorial optimization, structural optimization, control theory, economics, quantum chemistry, sensor network location, data mining, and machine learning. The SDPA algorithm (SDPA) is one of the most famous software packages of the primal-dual interior-point method (PDIPM) for solving the standard-form SDP problem (formulation (1)). software packages including the SDPA to be solvable on a single node. There exist two well-known major bottleneck parts in the algorithmic framework of PDIPM. The first part is the generation of the so-called Schur complement matrix (SCM). The second part is the Cholesky factorization of SCM. These two parts are called ELEMENTS and CHOLESKY, respectively.

The standard-form SDP has the following primal-dual form.

$$\begin{array}{rcll} \mathcal{P} & : & \text{minimize} & \sum_{k=1}^{m} c_k x_k \\ & & \text{subject to} & \boldsymbol{X} = \sum_{k=1}^{m} \boldsymbol{F}_k x_k - \boldsymbol{F}_0, \\ & & & \boldsymbol{X} \succeq \boldsymbol{O}. \\ \mathcal{D} & : & \text{maximize} & \boldsymbol{F}_0 \bullet \boldsymbol{Y} \\ & & \text{subject to} & \boldsymbol{F}_k \bullet \boldsymbol{Y} = c_k \ (k = 1, \ldots, m), \\ & & & \boldsymbol{Y} \succeq \boldsymbol{O}. \end{array} \tag{1}$$

We denote by $\mathbb{S}^n$ the space of $n \times n$ symmetric matrices. The notation $\boldsymbol{X} \succeq \boldsymbol{O}$ ($\boldsymbol{X} \succ \boldsymbol{O}$) indicates that $\boldsymbol{X} \in \mathbb{S}^n$ is a positive semidefinite (positive definite) matrix. The inner-product between $\boldsymbol{U} \in \mathbb{S}^n$ and $\boldsymbol{V} \in \mathbb{S}^n$ is defined by $\boldsymbol{U} \bullet \boldsymbol{V} = \sum_{i=1}^{n} \sum_{j=1}^{n} U_{ij} V_{ij}$.

The size of a given SDP problem can be roughly measured in terms of five metrics:

1. $m$: the number of equality constraints in the dual form $\mathcal{D}$ (which equals the size of the SCM)
2. $n$: the size of the variable matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$

We denote the time complexities of ELEMENTS and CHOLESKY by $O(mn^3 + m^2n^2)$ and $O(m^3)$, respectively.

We developed a new version of semidefinite programming algorithm parallel version (SDPARA) 7.6.0-G, which is a parallel implementation on multiples CPUs and GPUs for solving extremely large-scale SDP problems. SDPARA is designed to execute PDIPM on parallel computers with distributed memory space. The speed-up achieved by SDPARA is essentially attributable to its use of parallel computation to overcome the computational bottlenecks of ELEMENTS and CHOLESKY. Each process reads the input data and stores them and all variables in the process memory space, while the SCM data are divided between processes. We previously reported that SDPARA can compute each row of the SCM in parallel, and applied the parallel Cholesky factorization provided by ScaLAPACK to the SCM. In our previous work [1], we developed SDPARA 7.5.0-G on TSUBAME 2.0 [1], which is a high-performance GPU-accelerated supercomputer at the Tokyo Institute of Technology. We solved the largest SDP problem (which has over 1.48 million constraints), and created a new world record in 2012. In the same year, our implementation also achieved 533 TFlops in double precision for large-scale Cholesky factorization using 4,080 GPUs.

[1] http://www.gsic.titech.ac.jp/en/tsubame

## 2. Extremely Large-scale Parallel Cholesky Solver

As mentioned in the previous Section, SDP has many applications that involve SDP problems with special structures. We initiated the SDPA project [2], which aims to develop high-performance software packages for SDP, and we have solved a large number of SDP problems since 1995; therefore, we can classify the various types of SDP problems into the following three cases:

1. Case 1: SDP problems are sparse and satisfy the property of correlative sparsity; therefore, SCM tends to become sparse (e.g., the sensor network location problem and the polynomial optimization problem). In this case, CHOLESKY is the bottleneck part of PDIPM.
2. Case 2: $m$ is less or not considerably greater than $n$ and SCM is fully dense (e.g., the quantum chemistry problem and the truss topology problem). In this case, ELEMENTS is the bottleneck part of PDIPM, so we can decrease the time complexity of ELEMENTS $O(mn^3 + m^2n^2)$ to $O(m^2)$ by exploiting the sparsity of the data matrix.
3. Case 3: $m$ is considerably greater than $n$ and SCM is fully dense (e.g., the combinatorial optimization problem and quadratic assignment problem(QAP) [1]). In this case, CHOLESKY is the bottleneck part of PDIPM.

Table 1: Performance (teraflops) of GPU CHOLESKY obtained by using up to 1360 nodes (4080 GPUs) on TSUBAME 2.0 [1] and 2.5 [2].

(a) 400 nodes (1200 GPUs)

| Name | $m$ | org(2.0) | new(2.0) | new(2.5) |
|---|---|---|---|---|
| QAP6 | 709,275 | 223.0 | 233.0 | 314.5 |
| QAP7 | 1,218,400 | 248.8 | 306.2 | 505.8 |

(b) 700 nodes (2100 GPUs)

| Name | $m$ | org(2.0) | new(2.0) | new(2.5) |
|---|---|---|---|---|
| QAP6 | 709,275 | 309.5 | 329.0 | 387.5 |
| QAP7 | 1,218,400 | 440.0 | 470.0 | 707.1 |
| QAP8 | 1,484,406 | 463.8 | 512.9 | 825.1 |

(c) 1360 nodes (4080 GPUs)

| Name | $m$ | org(2.0) | new(2.0) | new(2.5) |
|---|---|---|---|---|
| QAP6 | 709,275 | 439.6 | 437.8 | 508.7 |
| QAP7 | 1,218,400 | 695.2 | 718.8 | 952.0 |
| QAP8 | 1,484,406 | 779.3 | 825.6 | 1186.4 |
| QAP9 | 1,962,225 | – | 964.4 | 1526.5 |
| QAP10 | 2,339,331 | – | 1018.5 | 1713.0 |

We previously reported that SDPARA can certainly determine whether the SCM of an input SDP problem becomes sparse (Case 1) or not (Cases 2 and 3). In the present study, we mainly focused on parallel computation of ELEMENTS and CHOLESKY in Cases 2 and 3, respectively. We also demonstrated that SDPARA is a high-performance general solver for SDPs in various application fields through numerical experiments on the TSUBAME 2.5 supercomputer and solved the largest SDP problem (QAP10), which has over 2.33 million constraints [2]; and we created a new world record. Fig 1 and Table 1 show the speed of the CHOLESKY component in teraflops. " New " corresponds to the latest algorithm [2], while " org " denotes the original algorithm in our previous paper [1]. Our implementation also achieved 1.713 PFlops in double precision for large-scale Cholesky factorization using 2,720 CPUs and 4,080 GPUs. Table 2 shows that the speed of the CHOLESKY

[2] http://sdpa.sourceforge.net/

Table 2: Performance (teraflops) of GPU CHOLESKY obtained by using up to 384 nodes (384 GPUs) on CX400

(a) 128 nodes (128 GPUs)

| Name | $m$ | org(2.0) | new(2.0) | new(2.5) |
|---|---|---|---|---|
| QAP6 | 709,275 | – | – | 90.1 |
| QAP7 | 1,218,400 | – | – | 98.7 |

(b) 384 nodes (384 GPUs)

| Name | $m$ | org(2.0) | new(2.0) | new(2.5) |
|---|---|---|---|---|
| QAP8 | 1,484,406 | – | – | 288.0 |
| QAP8-1 | 1,495,602 | – | – | 294.2 |

component in teraflops on the CX400 supercomputer at Kyushu University. We have achieved 294.2 TFlops when using 384 GPUs.

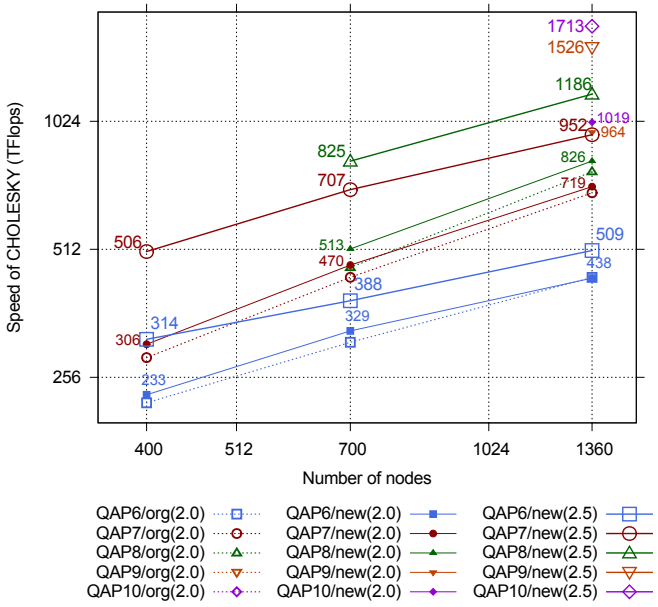


Figure 1: Performance of GPU CHOLESKY obtained by using up to 1360 nodes (4080 GPUs) on TSUBAME 2.0 and 2.5.

## Acknowledgment

This research was supported by " Advanced Computational Scientific Program " of Research Institute for Information Technology, Kyushu University, the Japan Science and Technology Agency (JST), the Core Research of Evolutionary Science, and Technology(CREST) research project, and the TSUBAME 2.5 Supercomputer Grand Challenge Program at the Tokyo Institute of Technology.

## REFERENCES

[1] K. Fujisawa et al. : High-Performance General Solver for Extremely Large-scale Semidefinite Programming Problems, The 2012 ACM/IEEE Conference on Supercomputing, SC'12, (2012)

[2] K. Fujisawa et al. : Peta-scale General Solver for Semidefinite Programming Problems with over Two Million Constraints, The 28th IEEE International Parallel & Distributed Processing Symposium (IPDPS2014), (2014)