CMAP-LAP

CMAP-LAP: Configurable Massively Parallel Solver for Lattice Problems

***Yuji Shinano**, Zuse Institute Berlin*

In Short

- CMAP-LAP is a massively parallel solver for Shortest Vector Problem (SVP) and related lattice problems.
- CMAP-LAP is the world first practical asynchronous distributed-memory solver, which is built on a new multi-algorithm paradigm.

A (full-rank) *lattice* of dimension n is the set of all integral linear combinations

$$L = \mathcal{L}(\mathbf{b}_1, \ldots, \mathbf{b}_n) := \left\{ \sum_{i=1}^{n} x_i \mathbf{b}_i : x_1, \ldots, x_n \in \mathbb{Z} \right\}, \tag{0.1}$$

where $\mathbf{b}_1, \ldots, \mathbf{b}_n$ are n linearly independent vectors in $\mathbb{R}^n$ for a positive integer n. The set of the n vectors $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ is called a *basis* of L. When another set of vectors $\{\mathbf{c}_1, \ldots, \mathbf{c}_n\}$ spans the same lattice L, it is also called a basis of L. Furthermore, $\mathcal{L}(\mathbf{B})$ denotes the lattice spanned by the row vectors of an invertible matrix $\mathbf{B}$. The $n \times n$ matrix $\mathbf{B}$ is called a *basis matrix* of L.

Lattice problems are algorithmic problems that involve lattices. Among lattice problems, the following is of fundamental importance:

Definition 1 (Shortest Vector Problem (SVP))**.** Find the shortest non-zero vector with respect to the ℓ_2-norm in the lattice $\mathcal{L}(\mathbf{B})$, given a basis matrix $\mathbf{B}$.

SVP is a discrete optimization problem for finding x_i's in (0.1) and is shown to be NP-hard under randomized reductions [1]. The length of the shortest non-zero vector in L is denoted by $\lambda_1(L)$. SVP is the problem of finding a lattice vector $\mathbf{s} \in L$ with $\|\mathbf{s}\| = \lambda_1(L)$. It should be emphasized that there is no known NP algorithm to check if $\|\mathbf{v}\| = \lambda_1(L)$ for a given $\mathbf{v} \in L$. Therefore, we rely on *Gaussian Heuristic*, which assumes that the number of vectors in $L \cap S$ is roughly equal to $\mathrm{vol}(S)/\mathrm{vol}(L)$ for a measurable set S in $\mathbb{R}^n$. By taking S to be the ball of radius $\lambda_1(L)$ centered at the origin $\mathbf{0}$ in $\mathbb{R}^n$, the Gaussian Heuristic leads to an estimation of $\lambda_1(L)$ as

$$\lambda_1(L) \approx \left(\frac{\mathrm{vol}(L)}{\omega_n} \right)^{1/n},$$

where ω_n denotes the volume of the n-dimensional unit ball. By Stirling's formula, we have $\omega_n \approx \left(\frac{2\pi e}{n}\right)^{n/2}$ as $n \to \infty$, and define

$$\mathrm{GH}(L) := \sqrt{\frac{n}{2\pi e}} \mathrm{vol}(L)^{1/n}. \tag{0.2}$$

Then, $\lambda_1(L) \approx \mathrm{GH}(L)$ holds for random lattices L in high dimensions $n \geq 40$. (Gaussian Heuristic does not hold in low dimensions.) For a vector $\mathbf{v} \in L$, the value $\|\mathbf{v}\|/\mathrm{GH}(L)$ is called the *approximation factor* of $\mathbf{v}$. Similarly, for a basis matrix $\mathbf{B}$, the value $\min_{1 \leq i \leq n} \|\mathbf{b}_i\|/\mathrm{GH}(L)$ is called the approximation factor of $\mathbf{B}$. They are evaluation metrics for the lattice vector and the basis. Based on this observation, an approximate variant of SVP is defined:

Definition 2 (Hermite Shortest Vector Problem (HSVP))**.** Given a basis matrix $\mathbf{B}$ and an approximation factor $\gamma > 0$, find a non-zero vector $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ such that $\|\mathbf{v}\| \leq \gamma \cdot \mathrm{vol}(\mathcal{L}(\mathbf{B}))^{1/n}$.

Another important lattice problem is:

Definition 3 (Closest Vector Problem (CVP))**.** Given a basis of a lattice L and a target vector $\mathbf{t}$, find a vector in L that is closest to $\mathbf{t}$.

Lattice problems are believed to be computationally hard with both classical and quantum algorithms [3] and have been used to construct various cryptosystems [5], including post-quantum cryptography. Therefore, developing a framework for lattice problems is an important task both in large-scale optimization and cryptanalysis. More specifically, the security of many cryptosystems is based on the hardness of an approximate variant of SVP. Lattice problem solvers have been extensively tested at the Darmstadt SVP challenge [6], which asks to find a lattice vector shorter than 1.05 times the expected length of a non-zero shortest lattice vector.

There are three basic families of lattice algorithms that have been developed to solve practical lattice

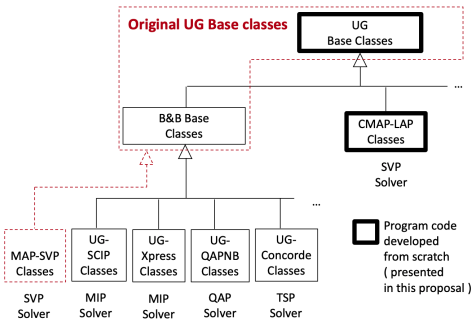

***Figure 1:** Refactoring of the UG framework*

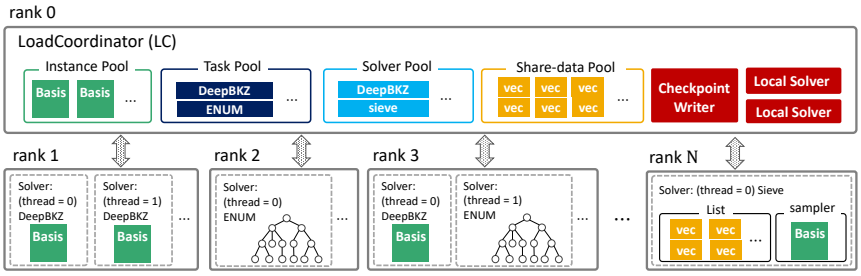

Figure 2: *System overview of CMAP-LAP for SVP*

problems: basis reduction, enumeration (ENUM), and sieve. These algorithms have advantages and disadvantages, and there is no single definite algorithm for lattice problems. Therefore, practical lattice-problem solvers generally rely on two or more algorithms. G6K [2] implements a variety of basis reduction and sieve algorithms, and it is considered the state-of-the-art SVP solver. G6K is equipped with both CPU and GPU highly parallelized implementations, but it runs only on a single machine. Furthermore, the memory requirement is exponential with respect to the dimension of the lattice, which is inevitable for sieve algorithms. On the other hand, MAP-SVP [7] is based on basis reduction and ENUM, which showed efficient scalability above $100,000$ MPI processes. We designed *Configurable Massively Parallel Solver for Lattice Problems (CMAP-LAP)* so that all features needed to investigate massive parallelization for solving lattice problems could be realized.

Existing solvers are limited to a fixed set of algorithms and lack in flexibility. There are two main obstacles in developing a large-scale multi-paradigm solver: the need for an efficient high-level information-sharing scheme across different algorithms, and an adaptive task selection and distribution strategy for hundreds of thousands of processes. This project is to provide solutions to overcome these obstacles and develop a flexible framework to make various algorithms work cooperatively on a large-scale distributed computing platform. By exploiting the mathematical properties of lattice, a clever vector pooling scheme is introduced to minimize the amount of information communicated among processes. The original UG codes [8] have been refactored into the *Generalized Ubiquity Generator framework* (Generalized UG[1]) to allow more flexibility necessary for lattice algorithms (see Figure 1). Particular emphasis is put on the efficient and versatile message-sharing mechanics. Based on the Generalized UG framework, we will develop the CMAP-LAP, whose architecture is shown in Figure 2.

[1]This is bem00052 HLRN project

WWW

https://www.zib.de/members/shinano

More Information

[1] M. Ajtai, *Generating hard instances of lattice problems*, in Symposium on Theory of Computing (STOC 1996), ACM, 1996, pp. 99–108.

[2] M. Albrecht, L. Ducas, G. Herold, E. Kirshanova, E. W. Postlethwaite, and M. Stevens, *The general sieve kernel and new records in lattice reduction*, in Advances in Cryptology–EUROCRYPT 2019, vol. 11477 of Lecture Notes in Computer Science, Springer, 2019, pp. 717–746.

[3] J.-Y. Cai, *The complexity of some lattice problems*, in Algorithmic Number Theory, W. Bosma, ed., Berlin, Heidelberg, 2000, Springer Berlin Heidelberg, pp. 1–32.

[4] A. Joux, *A tutorial on high performance computing applied to cryptanalysis (invited talk)*, in Advances in Cryptology–EUROCRYPT 2012, vol. 7237 of Lecture Notes in Computer Science, Springer, 2012, pp. 1–7.

[5] C. Peikert, *A decade of lattice cryptography*, Foundations and Trends in Theoretical Computer Science, 10 (2016), pp. 283–424.

[6] M. Schneider, N. Gama, P. Baumann, and L. Nobach, *SVP challenge (2010)*, URL: http://latticechallenge.org/svp-challenge.

[7] N. Tateiwa, Y. Shinano, S. Nakamura, A. Yoshida, S. Kaji, M. Yasuda, and K. Fujisawa, *Massive parallelization for finding shortest lattice vectors based on ubiquity generator framework*, in SC20: International Conference for High Performance Computing, Networking, Storage and Analysis, IEEE, 2020, pp. 1–15.

[8] *UG: Ubiquity Generator framework*. http://ug.zib.de/.

Project Partners

IMI(Kyushu University), RIKEN R-CCS

Funding

Forschungscampus Modal