# Random Sampling for Short Lattice Vectors on Graphics Cards

TECHNISCHE UNIVERSITÄT DARMSTADT

**Michael Schneider, Norman Göttert**

**TU Darmstadt, Germany**

**mischnei@cdc.informatik.tu-darmstadt.de**


CHES 2011, Nara

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Introduction

Simple Sampling Reduction (SSR)

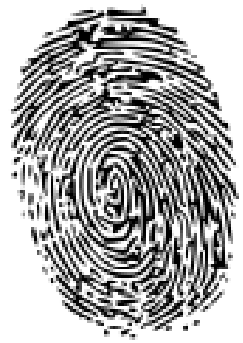
GPU-SSR

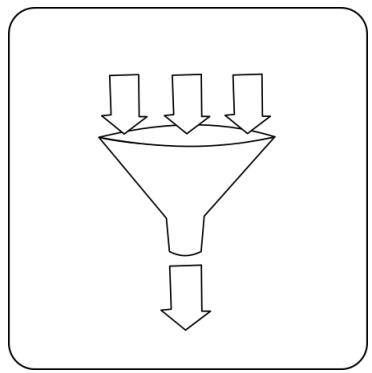
Results

Conclusion

# Lattice-Based Cryptography

TECHNISCHE UNIVERSITÄT DARMSTADT

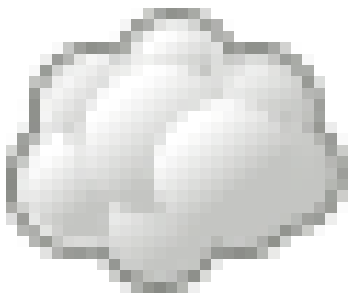





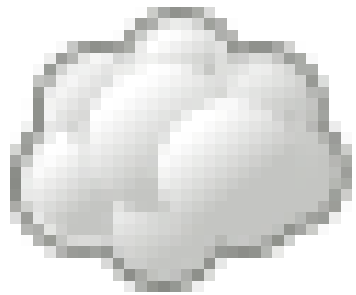
- Based on the approximate shortest vector problem (aSVP)

# Special Compute Hardware

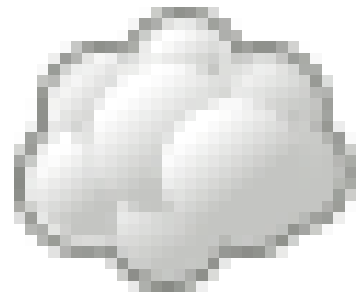
TECHNISCHE UNIVERSITÄT DARMSTADT



- Multicore-CPUs



- Graphics cards



- Compute clouds

# Goal

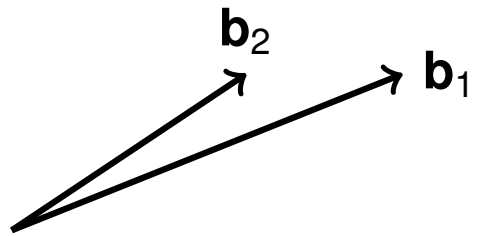
TECHNISCHE
UNIVERSITÄT
DARMSTADT

Assessing hardness of aSVP

using special hardware

# Lattices

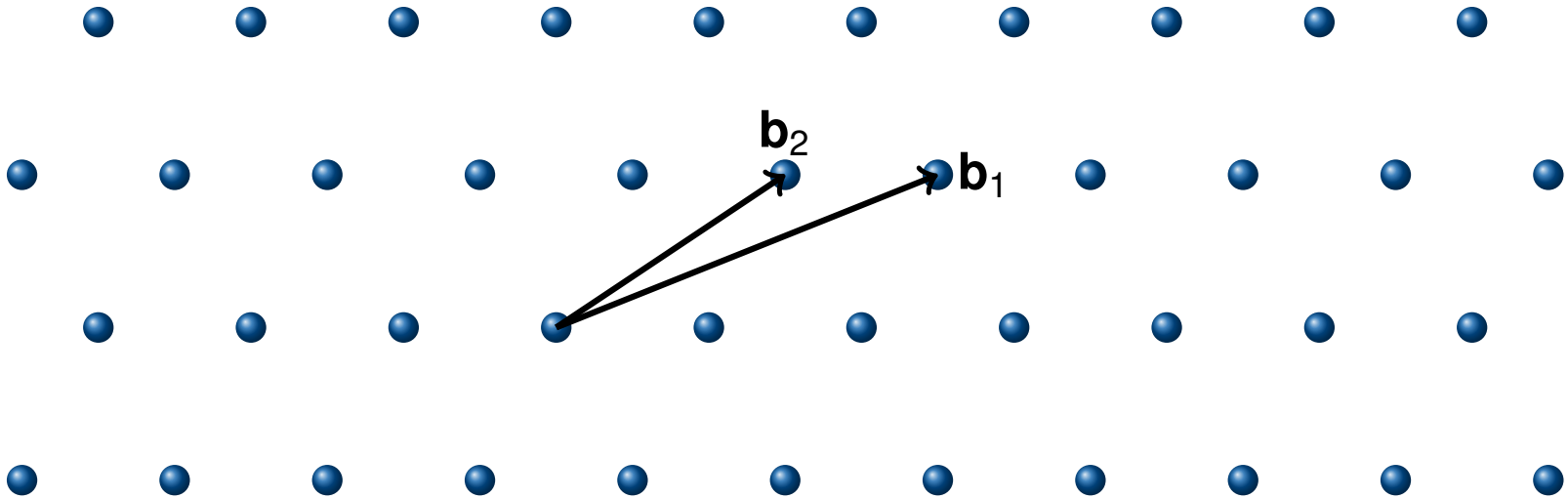
TECHNISCHE UNIVERSITÄT DARMSTADT




- Basis matrix $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ with $\mathbf{b}_i \in \mathbb{R}^d$
- Lattice: $\mathcal{L}(\mathbf{B}) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i, x_i \in \mathbb{Z}\}$

# Lattices

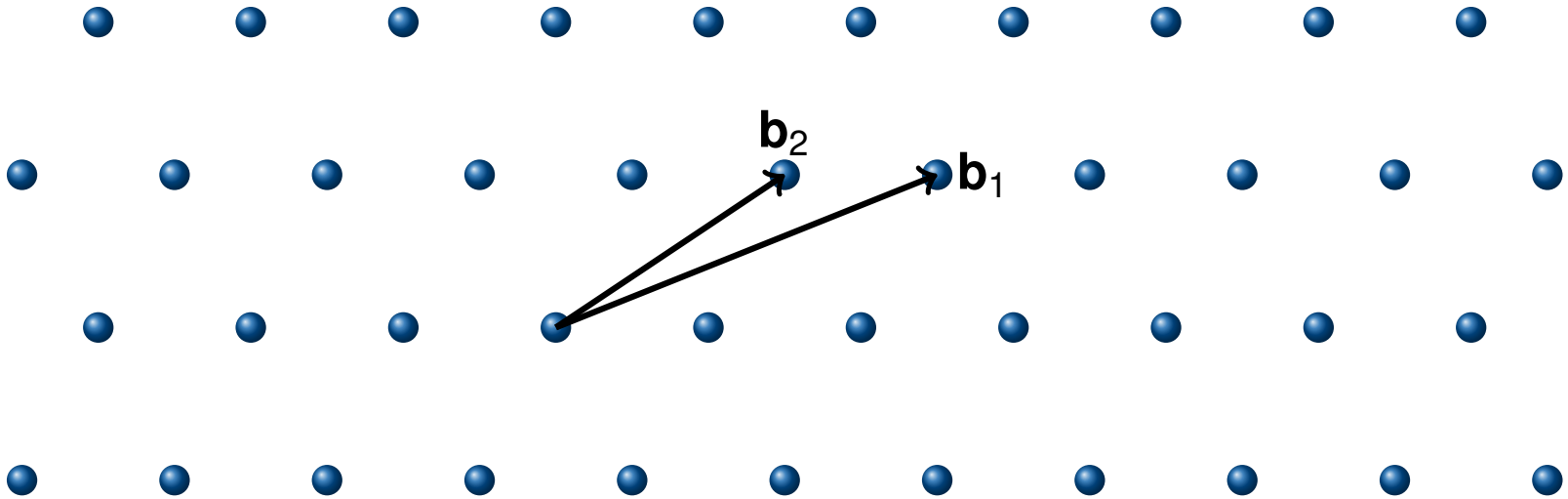
TECHNISCHE UNIVERSITÄT DARMSTADT




- Basis matrix $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$ with $\mathbf{b}_i \in \mathbb{R}^d$
- Lattice: $\mathcal{L}(\mathbf{B}) = \{\sum_{i=1}^{n} x_i \mathbf{b}_i, x_i \in \mathbb{Z}\}$

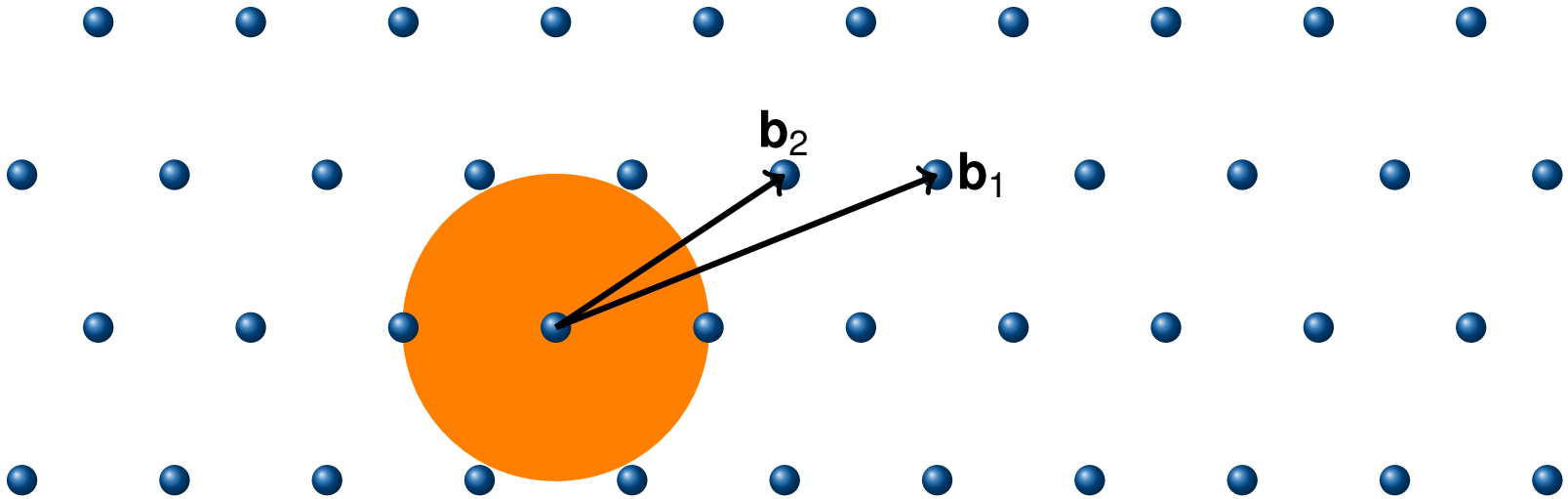
# approx. Shortest Vector Problem (aSVP)

TECHNISCHE UNIVERSITÄT DARMSTADT



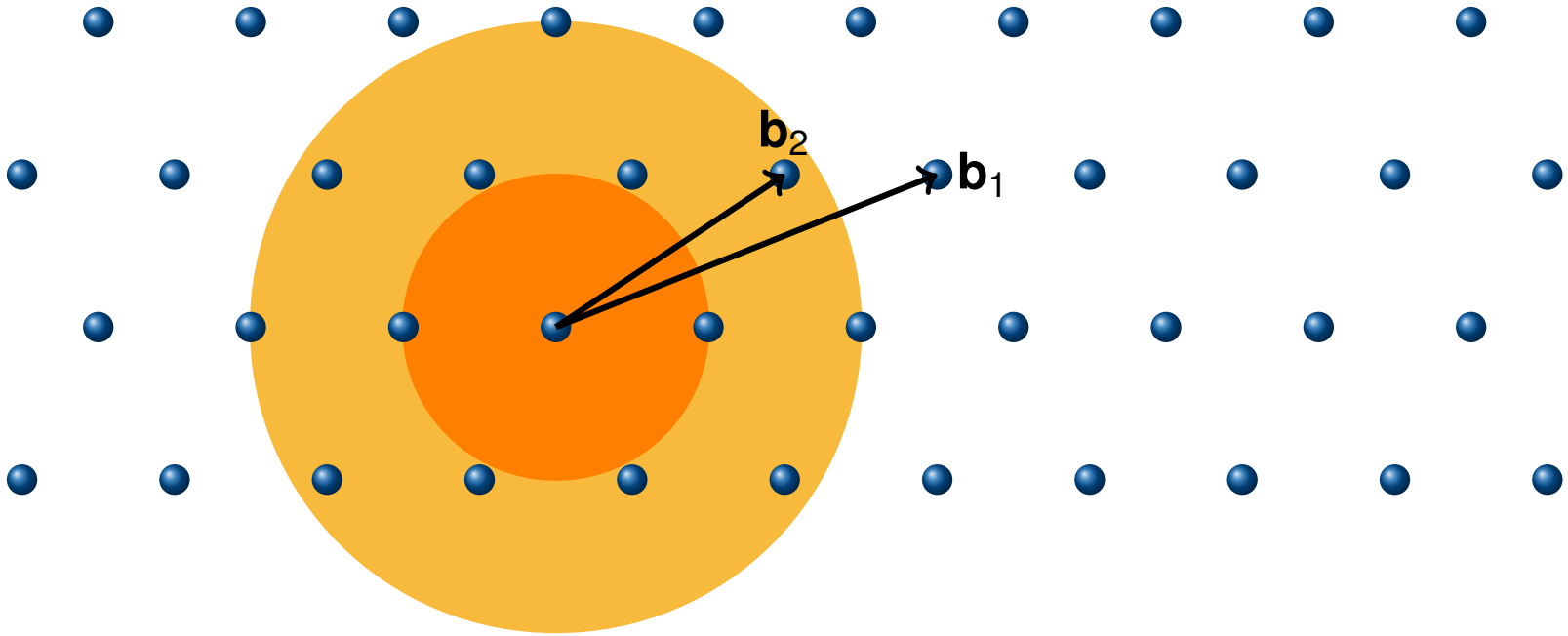
# approx. Shortest Vector Problem (aSVP)

TECHNISCHE UNIVERSITÄT DARMSTADT



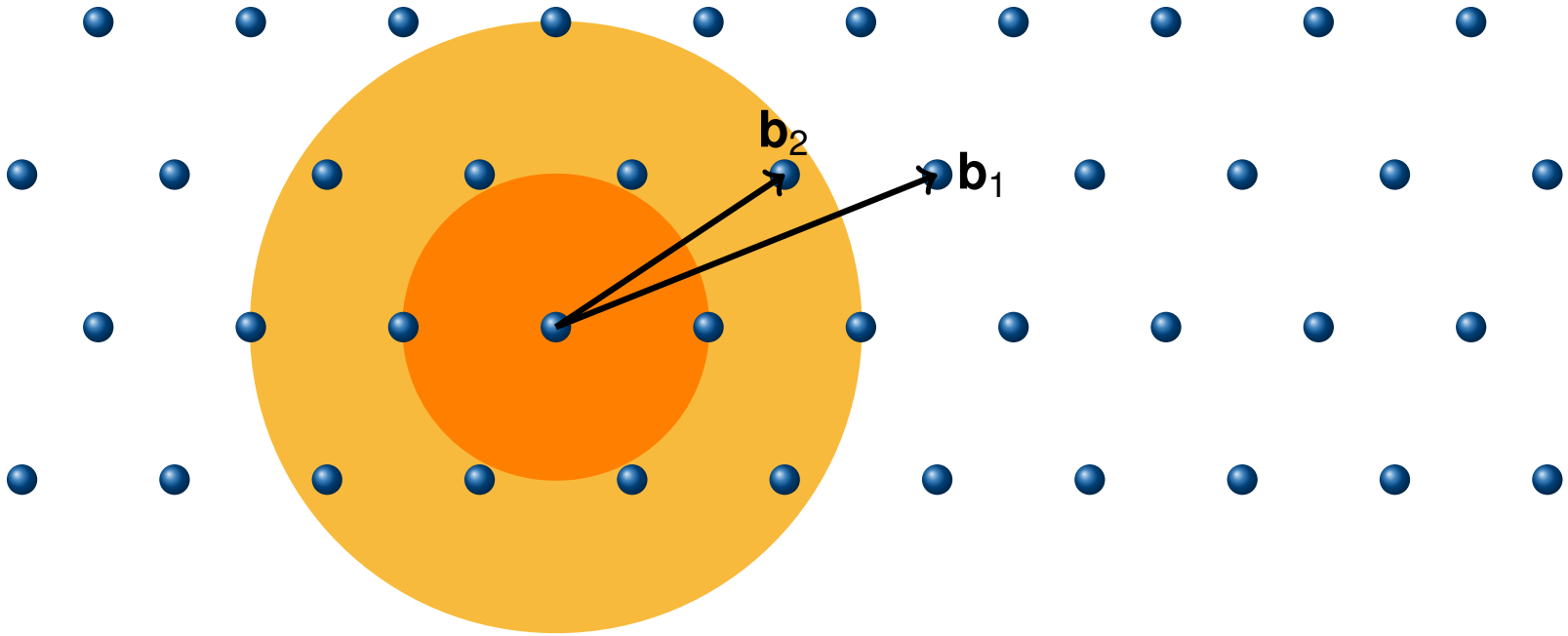
# approx. Shortest Vector Problem (aSVP)

TECHNISCHE UNIVERSITÄT DARMSTADT



# approx. Shortest Vector Problem (aSVP)

TECHNISCHE UNIVERSITÄT DARMSTADT




## Algorithms for aSVP

- BKZ [SE94]
- SSR [Lud05]

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Introduction

Simple Sampling Reduction (SSR)

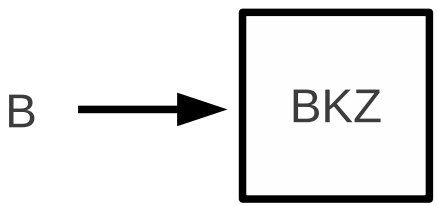
GPU-SSR

Results

Conclusion

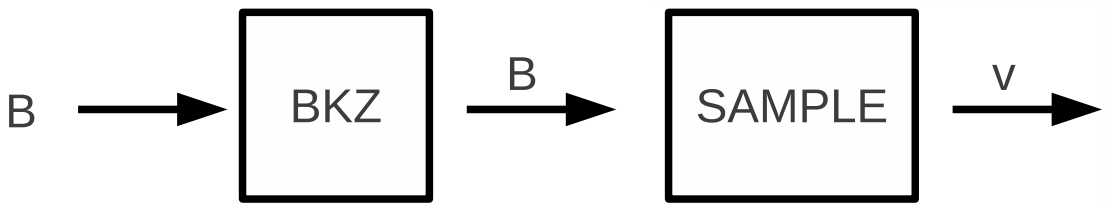
# SSR

TECHNISCHE UNIVERSITÄT DARMSTADT



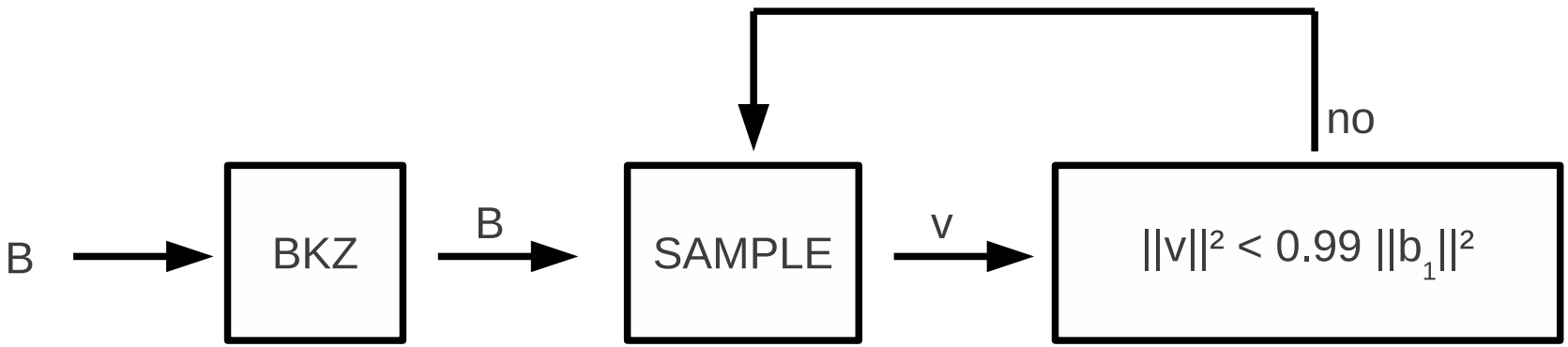
# SSR

TECHNISCHE UNIVERSITÄT DARMSTADT



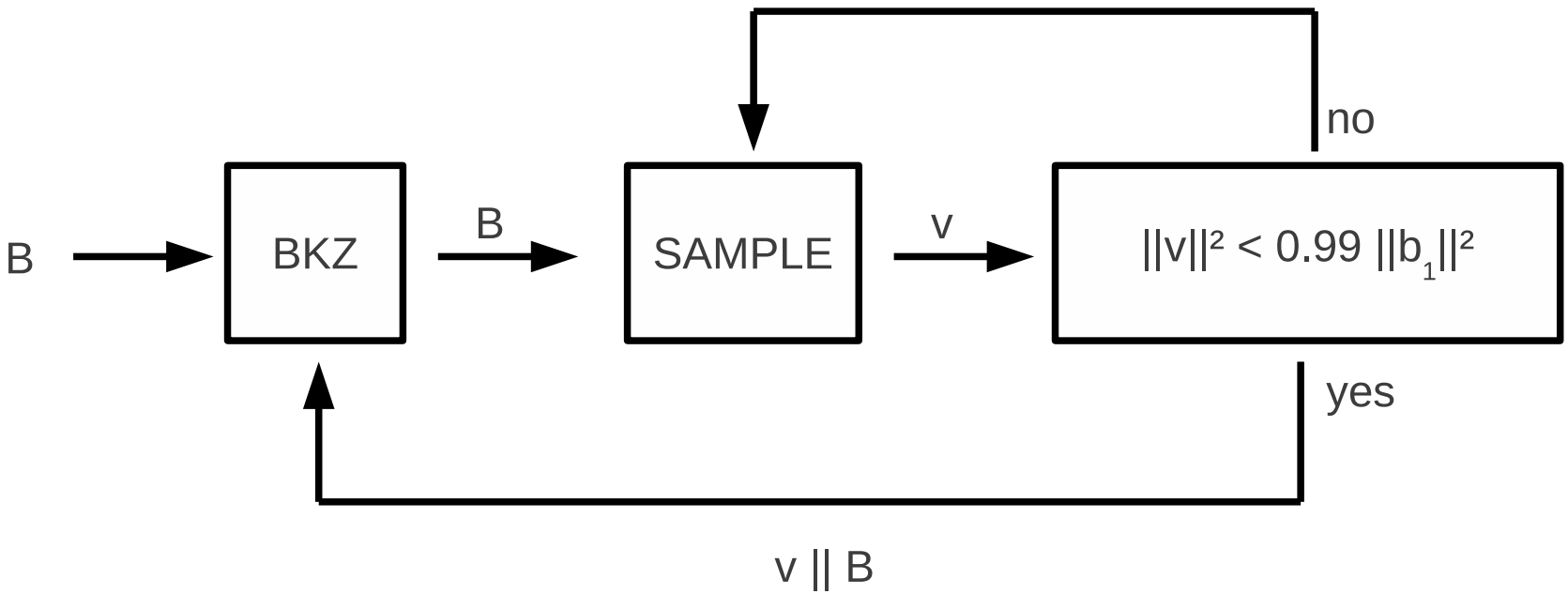
# SSR

TECHNISCHE UNIVERSITÄT DARMSTADT



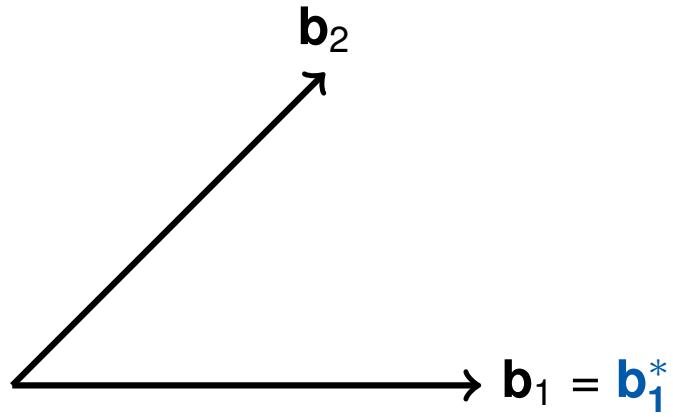
# SSR

TECHNISCHE UNIVERSITÄT DARMSTADT



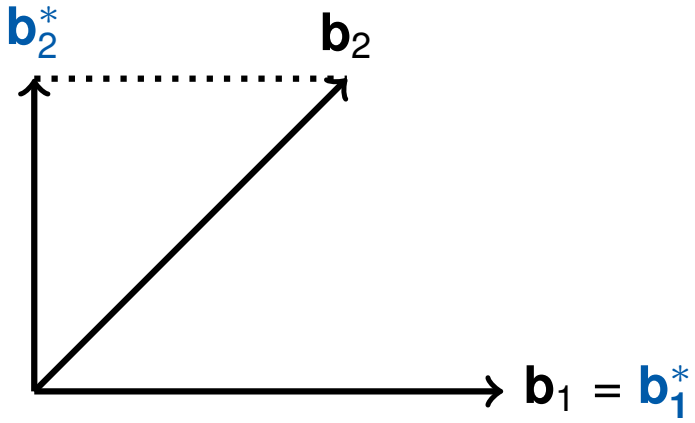
# Gram-Schmidt Orthogonalization

TECHNISCHE UNIVERSITÄT DARMSTADT



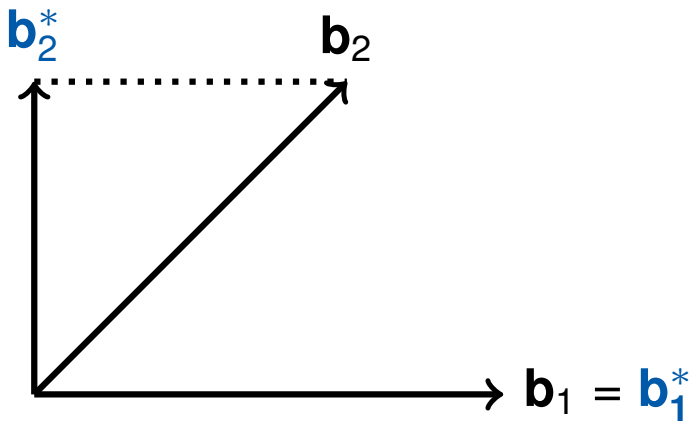
# Gram-Schmidt Orthogonalization

TECHNISCHE UNIVERSITÄT DARMSTADT



# Gram-Schmidt Orthogonalization

TECHNISCHE UNIVERSITÄT DARMSTADT




- $\mathbf{b}_1 + \mathbf{b}_2 = \nu_1 \mathbf{b}_1^* + \nu_2 \mathbf{b}_2^*$

# Random Sampling

TECHNISCHE UNIVERSITÄT DARMSTADT

$\mathbf{v} \in \mathcal{L} \quad , \quad \nu_i \in \mathbb{R}$

$$\mathbf{v} = \nu_1 \mathbf{b}_1^* + \nu_2 \mathbf{b}_2^* + \ldots + \nu_{n-1} \mathbf{b}_{n-1}^* + \nu_n \mathbf{b}_n^*$$

# Random Sampling

TECHNISCHE UNIVERSITÄT DARMSTADT

$\mathbf{v} \in \mathcal{L} \quad , \quad \nu_i \in \mathbb{R}$

$$\begin{aligned} \mathbf{v} &= \nu_1 \mathbf{b}_1^* + \nu_2 \mathbf{b}_2^* + \ldots + \nu_{n-1} \mathbf{b}_{n-1}^* + \nu_n \mathbf{b}_n^* \\ \|\mathbf{v}\|^2 &= \nu_1^2 \|\mathbf{b}_1^*\|^2 + \nu_2^2 \|\mathbf{b}_2^*\|^2 + \ldots + \nu_{n-1}^2 \left\|\mathbf{b}_{n-1}^*\right\|^2 + \nu_n^2 \|\mathbf{b}_n^*\|^2 \end{aligned}$$

# Random Sampling

TECHNISCHE UNIVERSITÄT DARMSTADT

$\mathbf{v} \in \mathcal{L} \quad , \quad \nu_i \in \mathbb{R}$

$$\begin{array}{rcccccccccc} \mathbf{v} & = & \nu_1 \mathbf{b}_1^* & + & \nu_2 \mathbf{b}_2^* & + \ldots + & \nu_{n-1} \mathbf{b}_{n-1}^* & + & \nu_n \mathbf{b}_n^* \\ \|\mathbf{v}\|^2 & = & \nu_1^2 \|\mathbf{b}_1^*\|^2 & + & \nu_2^2 \|\mathbf{b}_2^*\|^2 & + \ldots + & \nu_{n-1}^2 \left\|\mathbf{b}_{n-1}^*\right\|^2 & + & \nu_n^2 \|\mathbf{b}_n^*\|^2 \end{array}$$

after LLL / BKZ - reduction:

$$\|\mathbf{b}_1^*\| \geq \|\mathbf{b}_2^*\| \geq \ldots \geq \|\mathbf{b}_n^*\|$$

# Random Sampling

TECHNISCHE UNIVERSITÄT DARMSTADT

$\mathbf{v} \in \mathcal{L} \quad , \quad \nu_i \in \mathbb{R}$

$$\begin{array}{rcccccccc} \mathbf{v} & = & \nu_1 \mathbf{b}_1^* & + & \nu_2 \mathbf{b}_2^* & + \ldots + & \nu_{n-1} \mathbf{b}_{n-1}^* & + & \nu_n \mathbf{b}_n^* \\ \|\mathbf{v}\|^2 & = & \nu_1^2 \left\|\mathbf{b}_1^*\right\|^2 & + & \nu_2^2 \left\|\mathbf{b}_2^*\right\|^2 & + \ldots + & \nu_{n-1}^2 \left\|\mathbf{b}_{n-1}^*\right\|^2 & + & \nu_n^2 \left\|\mathbf{b}_n^*\right\|^2 \end{array}$$

after LLL / BKZ - reduction:

$$\|\mathbf{b}_1^*\| \geq \|\mathbf{b}_2^*\| \geq \ldots \geq \|\mathbf{b}_n^*\|$$

$\mathcal{S}_{u,\mathbf{B}}$ is the set of all vectors $\mathbf{v} = \sum_{i=1}^{n} \nu_i \mathbf{b}_i^*$ satisfying

$$|\nu_i| \leq \begin{cases} 0.5 & \text{for } 1 \leq i < n-u \\ 1 & \text{for } n-u \leq i \leq n \end{cases}$$

# Random Sampling

TECHNISCHE UNIVERSITÄT DARMSTADT

$\mathbf{v} \in \mathcal{L} \quad , \quad \nu_i \in \mathbb{R}$

$$\begin{array}{rcccccccc} \mathbf{v} & = & \nu_1 \mathbf{b}_1^* & + & \nu_2 \mathbf{b}_2^* & + \ldots + & \nu_{n-1} \mathbf{b}_{n-1}^* & + & \nu_n \mathbf{b}_n^* \\ \|\mathbf{v}\|^2 & = & \nu_1^2 \left\|\mathbf{b}_1^*\right\|^2 & + & \nu_2^2 \left\|\mathbf{b}_2^*\right\|^2 & + \ldots + & \nu_{n-1}^2 \left\|\mathbf{b}_{n-1}^*\right\|^2 & + & \nu_n^2 \left\|\mathbf{b}_n^*\right\|^2 \end{array}$$

after LLL / BKZ - reduction:

$$\|\mathbf{b}_1^*\| \geq \|\mathbf{b}_2^*\| \geq \ldots \geq \|\mathbf{b}_n^*\|$$

$\mathcal{S}_{u,\mathbf{B}}$ is the set of all vectors $\mathbf{v} = \sum_{i=1}^{n} \nu_i \mathbf{b}_i^*$ satisfying

$$|\nu_i| \leq \begin{cases} 0.5 & \text{for } 1 \leq i < n-u \\ 1 & \text{for } n-u \leq i \leq n \end{cases}$$

# Function SAMPLE

TECHNISCHE UNIVERSITÄT DARMSTADT

## Sampling Algorithm

- SAMPLE [Lud05]
  - uses bits of a random seed
  - allows for more control
  - seed $\{1, 2, \ldots, i\}$ , $i = 2^{u_{max}}$
- generate (random) vectors in $\mathcal{S}_{u,\mathbf{B}}$

# Function SAMPLE

TECHNISCHE UNIVERSITÄT DARMSTADT

## Sampling Algorithm

- SAMPLE [Lud05]
  - uses bits of a random seed
  - allows for more control
  - seed $\{1, 2, \ldots, i\}$ , $i = 2^{u_{max}}$
- generate (random) vectors in $\mathcal{S}_{u,\mathbf{B}}$

$\rightarrow$ SIMD

# Function SAMPLE

TECHNISCHE UNIVERSITÄT DARMSTADT

**Algorithm 2:** SAMPLE

**Input:** Lattice basis $\mathbf{B} \in \mathbb{Z}^{n \times n}$, GS-coefficients $\mathbf{R} \in \mathbb{Q}^{n \times n}$, $x \in \mathbb{Z}$
**Output:** vector $\mathbf{v} \in \mathcal{S}_{u,\mathbf{B}}$

$\mathbf{v} \leftarrow \mathbf{b}_n$, $r \leftarrow r_n$
**for** $j = n - 1$ *to* $1$ **do**
  $y \leftarrow \lfloor x_j - 0.5 \rceil$
  **if** $x = 1 \mod 2$ **then**
    **if** $x_j - y \leq 0$ **then** $y \leftarrow y - 1$
    **else** $y \leftarrow y + 1$
  **end**
  $x \leftarrow \lfloor x/2 \rfloor$, $\mathbf{v} \leftarrow \mathbf{v} - y\mathbf{b}_j$, $r \leftarrow r - y\mathbf{r}_j$
**end**
**return** $\mathbf{v}$

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Introduction

Simple Sampling Reduction (SSR)

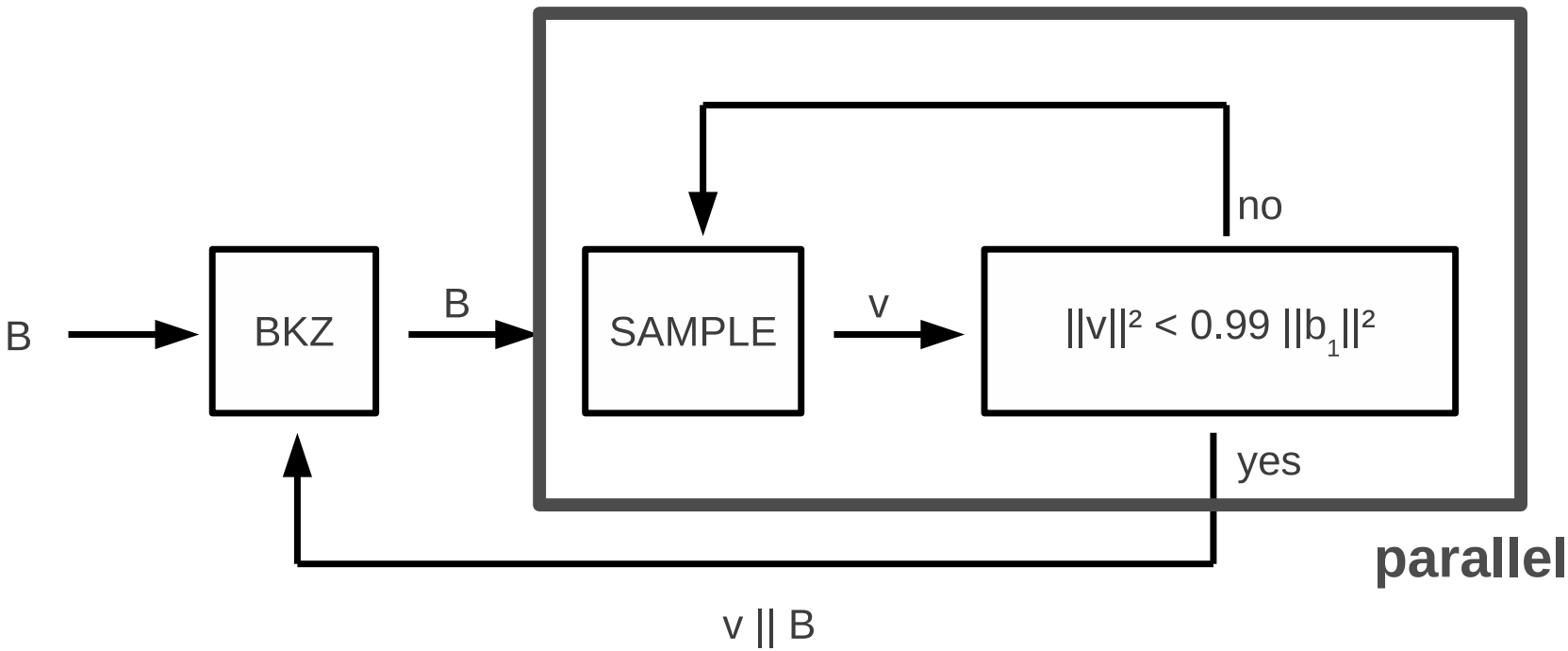
GPU-SSR

Results

Conclusion

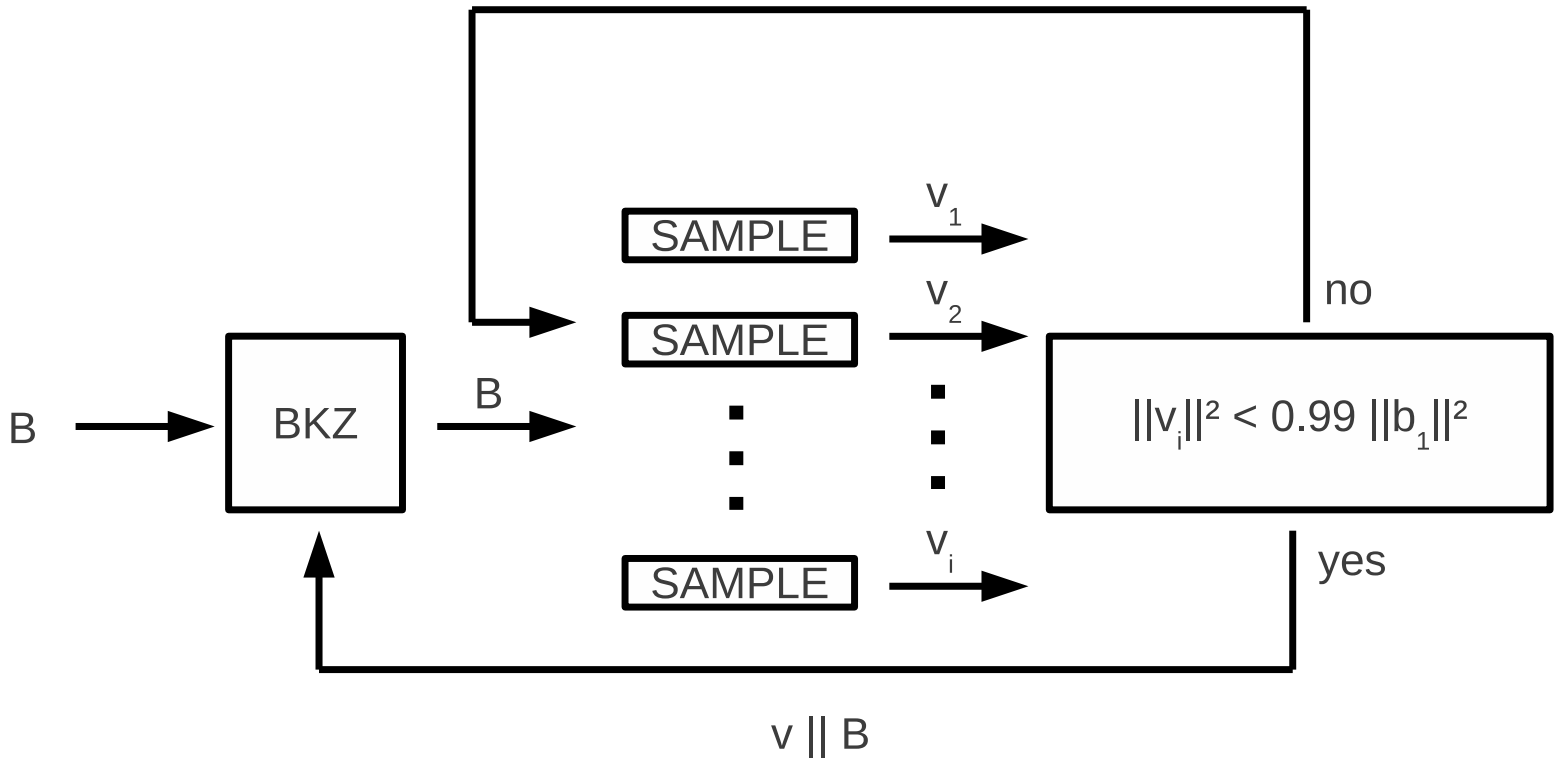
# Parallel SSR

TECHNISCHE UNIVERSITÄT DARMSTADT



# GPU-SSR

TECHNISCHE UNIVERSITÄT DARMSTADT




→ Slide 19

# Sampling Rate $n = 180$

TECHNISCHE UNIVERSITÄT DARMSTADT

| | |
|---|---|
| sampling rate CPU: | 160 samples/s |
| sampling rate GPU: | 120,000 samples/s |

# Experiments

TECHNISCHE UNIVERSITÄT DARMSTADT

## Setting:

- goalnorm: stop if $\|\mathbf{b}_1\| \leq 1.0219^n \cdot \det(\mathcal{L})^{1/n}$ $\quad (\rightarrow$ BKZ-20)
- 10 random lattices each dimension
- LLL pre-reduction

# Experiments

TECHNISCHE UNIVERSITÄT DARMSTADT

## Setting:

- goalnorm: stop if $\|\mathbf{b}_1\| \leq 1.0219^n \cdot \det(\mathcal{L})^{1/n}$ $(\rightarrow$ BKZ-20)
- 10 random lattices each dimension
- LLL pre-reduction

## Improvements:

- prepend multiple vectors $\mathbf{v}_i$
- parallel norm computations

# Experiments

TECHNISCHE UNIVERSITÄT DARMSTADT

## Setting:

- goalnorm: stop if $\|\mathbf{b}_1\| \leq 1.0219^n \cdot \det(\mathcal{L})^{1/n}$ ($\rightarrow$ BKZ-20)
- 10 random lattices each dimension
- LLL pre-reduction

## Improvements:

- prepend multiple vectors $\mathbf{v}_i$
- parallel norm computations

## CPU

- Intel Core Duo E8400 (3GHz)

## GPU

- NVIDIA GTX 295

$\rightarrow$ Slide 16

# Results - Norm

TECHNISCHE UNIVERSITÄT DARMSTADT

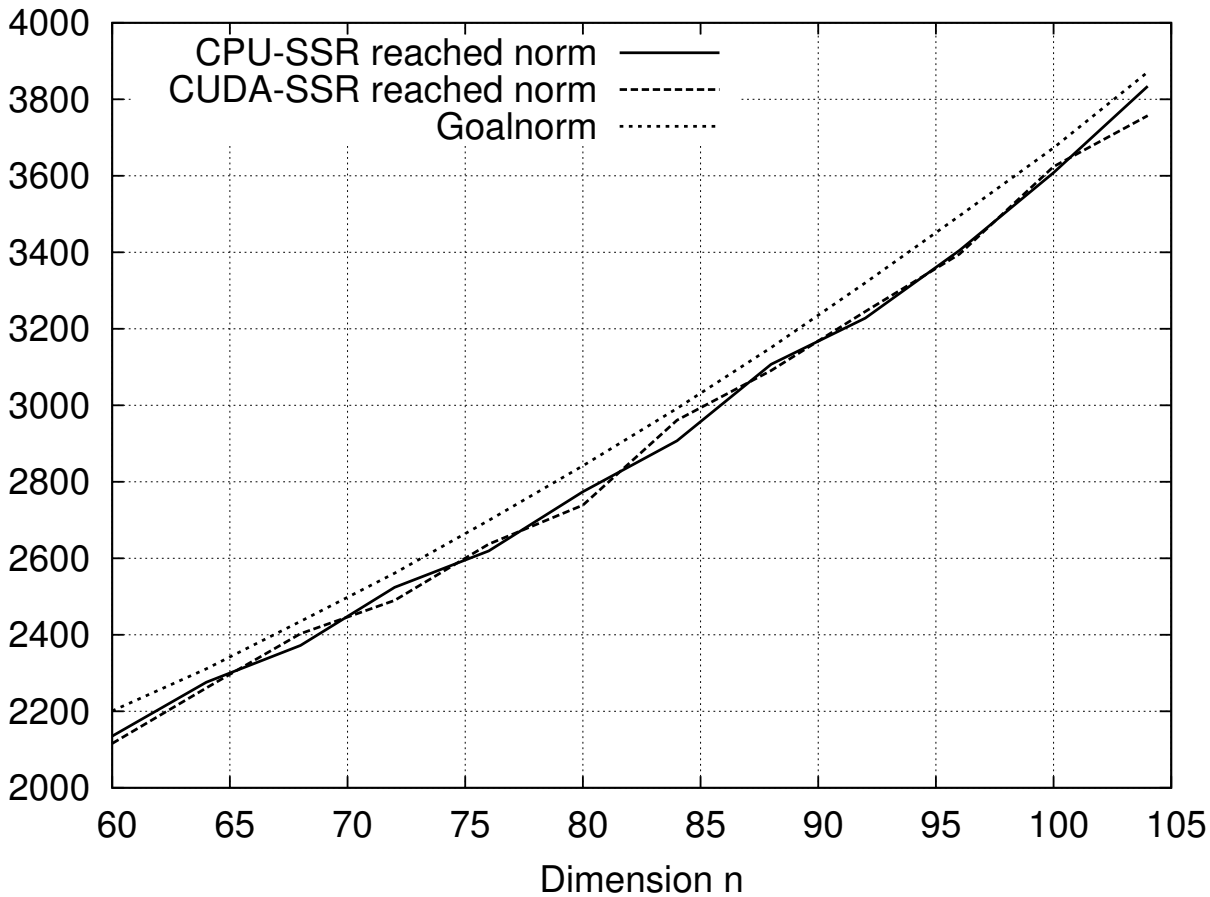



Figure: Reached norm

# Results - Time

TECHNISCHE UNIVERSITÄT DARMSTADT

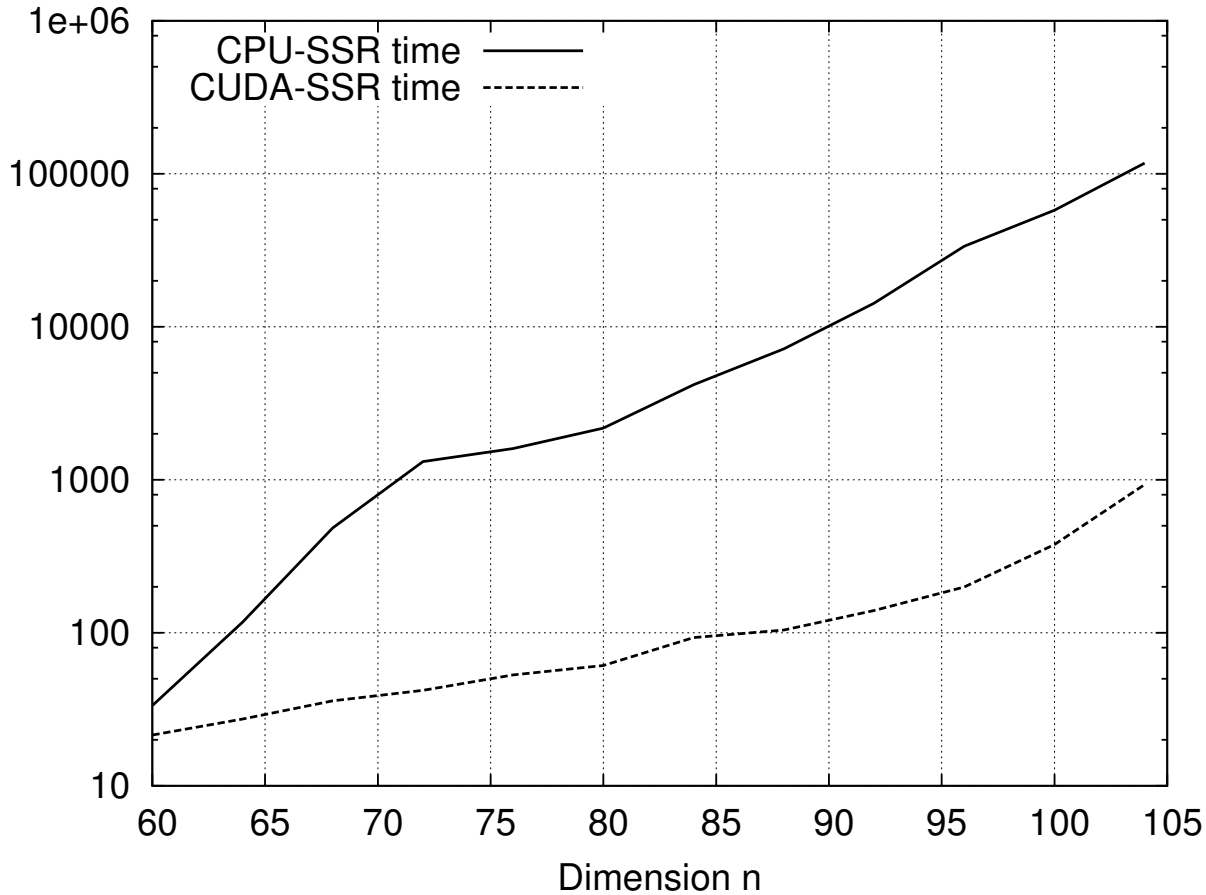



Figure: Runtime [s]

# Results - Speedup

TECHNISCHE UNIVERSITÄT DARMSTADT

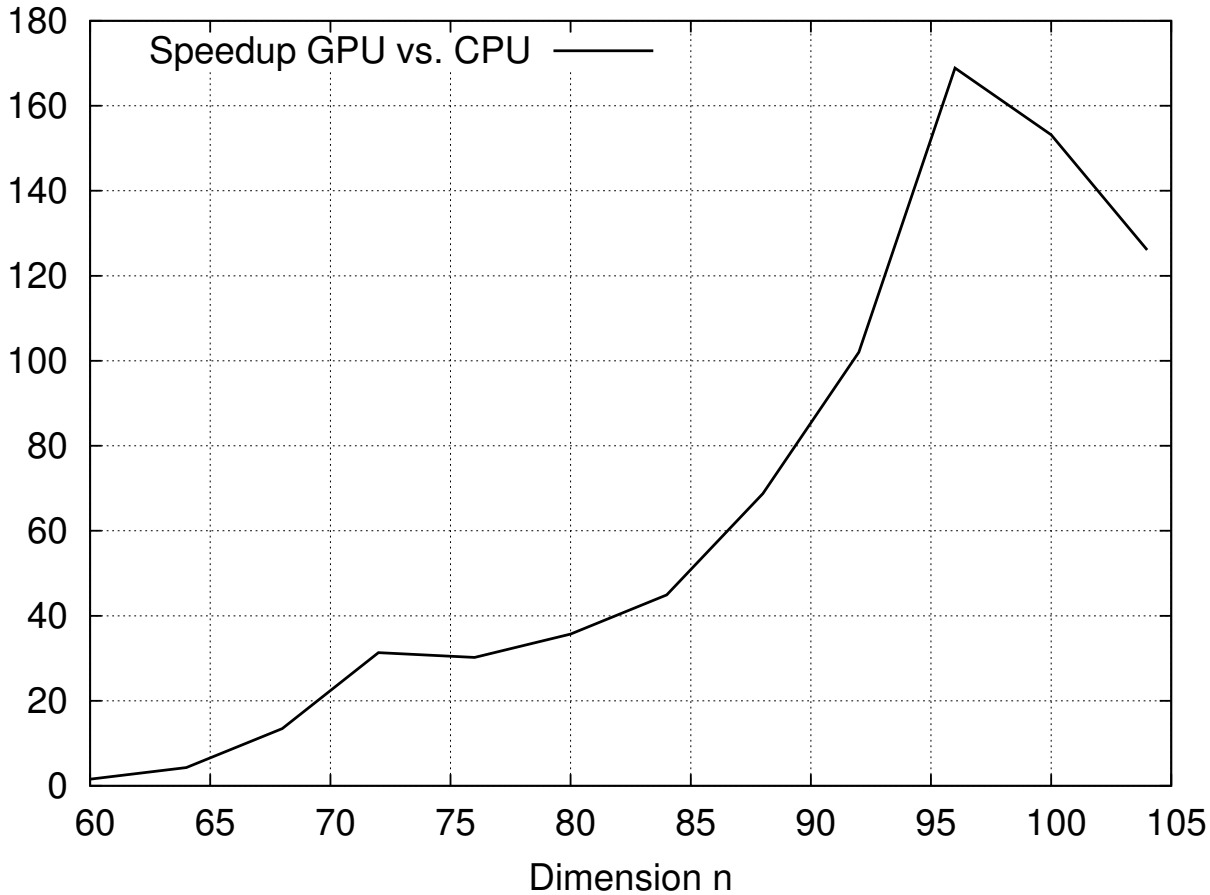



Figure: Runtime Speedup

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Introduction

Simple Sampling Reduction (SSR)

GPU-SSR

Results

Conclusion

# Conclusion

TECHNISCHE UNIVERSITÄT DARMSTADT

## Speedup

- GPU-SSR up to 160 times faster than CPU-SSR

# Conclusion

TECHNISCHE UNIVERSITÄT DARMSTADT

## Speedup

- GPU-SSR up to 160 times faster than CPU-SSR

## Implementation

- CPU and GPU version of SSR online:
  `www.cdc.informatik.tu-darmstadt.de/mitarbeiter/mischnei.html`

# Conclusion

TECHNISCHE UNIVERSITÄT DARMSTADT

## Speedup

- GPU-SSR up to 160 times faster than CPU-SSR

## Implementation

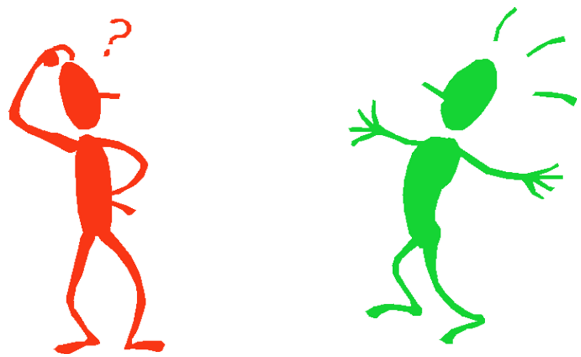
- CPU and GPU version of SSR online:
  `www.cdc.informatik.tu-darmstadt.de/mitarbeiter/mischnei.html`

## Future Work

- Better search spaces
- CPU / GPU clusters
- decrease LLL / BKZ time (LLL: 67% in dimension 100)
  - to 50%?!?

# Finally...

TECHNISCHE UNIVERSITÄT DARMSTADT



Thank you!

# Bibliography I

TECHNISCHE UNIVERSITÄT DARMSTADT


Christoph Ludwig.

*Practical Lattice Basis Sampling Reduction*.
PhD thesis, Technische Universität Darmstadt, 2005.
`http://elib.tu-darmstadt.de/diss/000640/`.

Claus-Peter Schnorr and M. Euchner.

Lattice basis reduction: Improved practical algorithms and solving subset sum problems.
*Mathematical Programming*, 66:181–199, 1994.