Optimal Binary Switch Codes with Small Query Size

Zhiying Wang, Han Mao Kiah, Yuval Cassuto
Presenter: Junming Ke
junming.ke@ut.ee

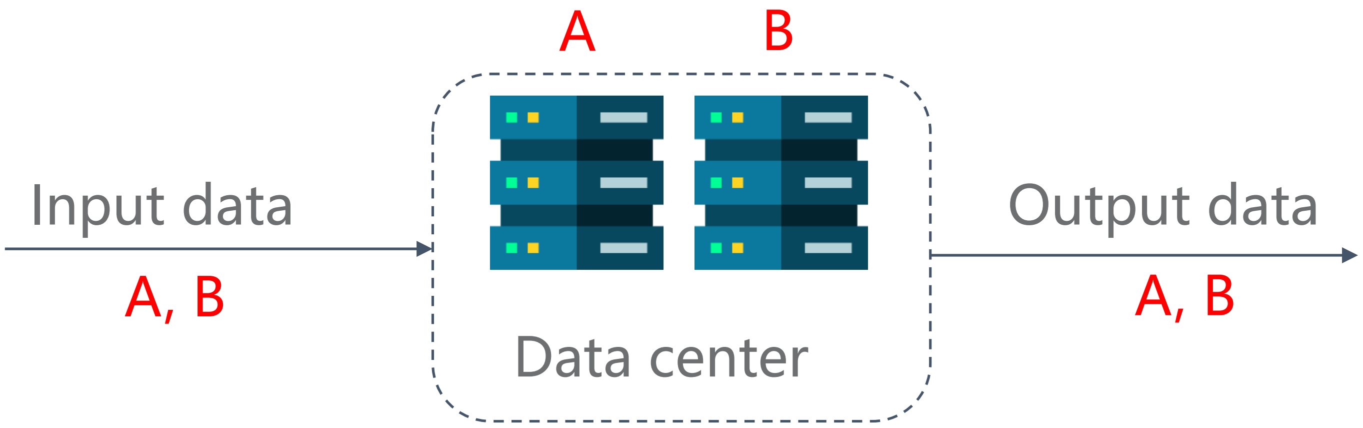
CONTENTS

1

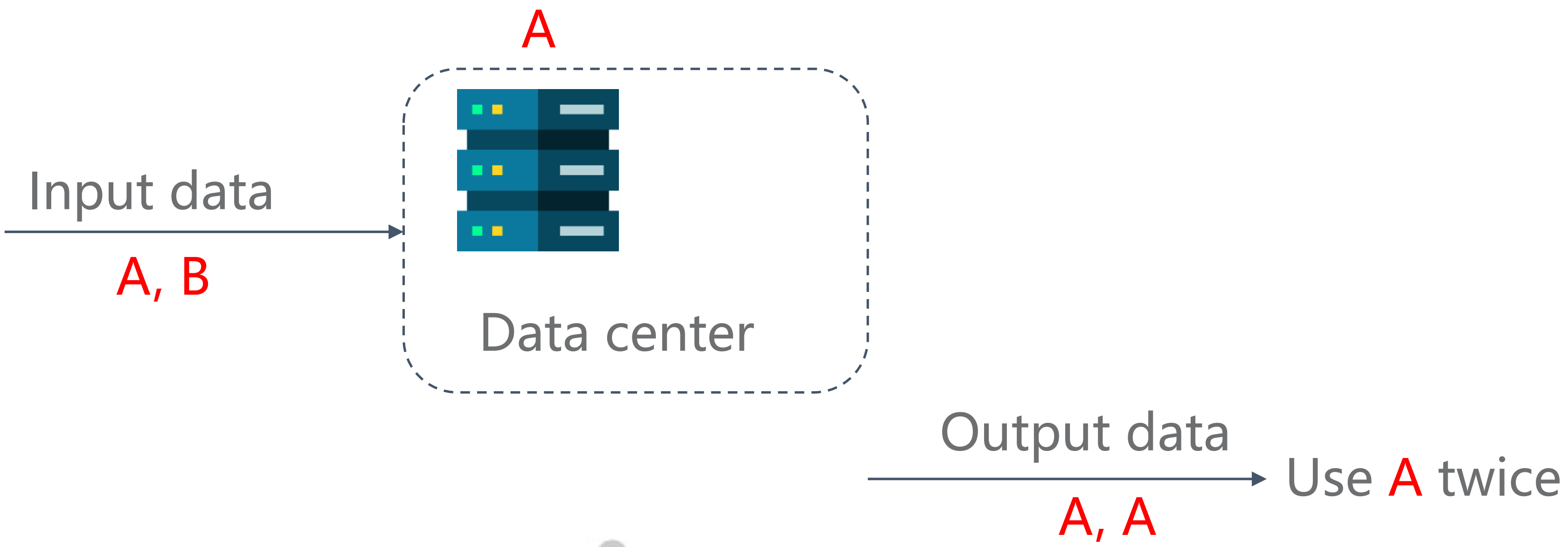
Background

Background

UNIVERSITY OF TARTU

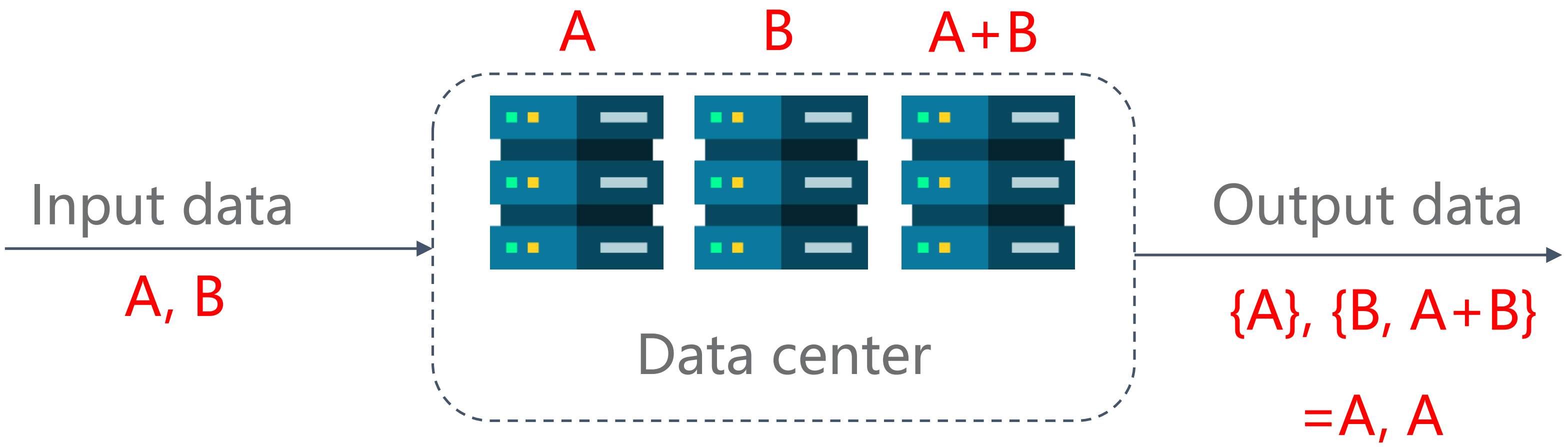


Background



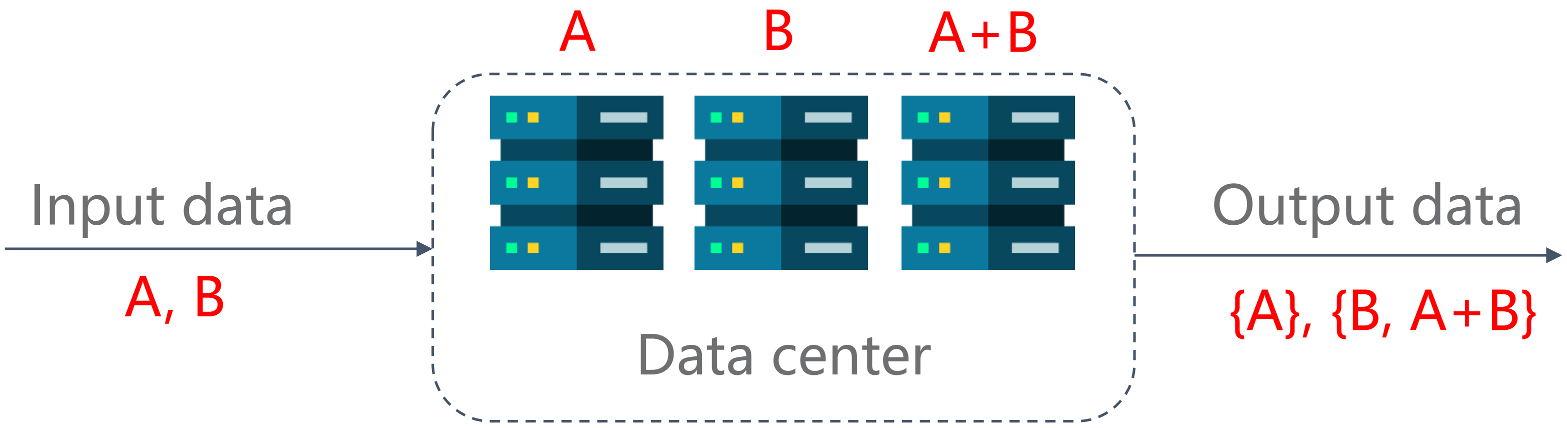
Background

UNIVERSITY OF TARTU



Background

UNIVERSITY OF TARTU



A: 10

B: 01

A+B: 11

Background

Data center

Problem:

Constructing switch codes solving arbitrary requests and having max query size 2.

2

Constructions

Definition

UNIVERSITY OF TARTU

(n, k, R) switch codes

- Code length n
- Dimension k
- Request size R
- Query size r

Definition

(n, k, R) switch codes

- Code length n
- Dimension k
- Request size R
- Query size r

$\downarrow$ A, B

A:	**10**
B:	**01**
A+B:	**11**

- Code length $n = 2$
- Dimension $k = 2$
- Request size $R = 2$
- Max query size $r = 2$

Construction

UNIVERSITY OF TARTU

1. Fix k, R such that $logR$ and $\frac{k}{1+logR}$ are integers.
2. Let $N = 2R - 1, K = 1 + logR$. (N, K) simplex code.
3. Codeword Length $\frac{k(2R-1)}{1+logR}$.

$$
\begin{array}{cc}
 & \begin{array}{cc} K & N-K \end{array} \\
K & \left[\begin{array}{cccc|cccc}
I_k & 0 & \dots & 0 & G & 0 & \dots & 0 \\
0 & I_k & \dots & 0 & 0 & G & \dots & 0 \\
\dots & \dots & \dots & 0 & \dots & \dots & \dots & 0 \\
0 & 0 & \dots & I_k & 0 & 0 & \dots & G
\end{array}\right]
\end{array}
$$

$\frac{k}{1+logR}$ groups, each group size $1 + logR$

The above construction solves arbitrary requests and has query size 2.

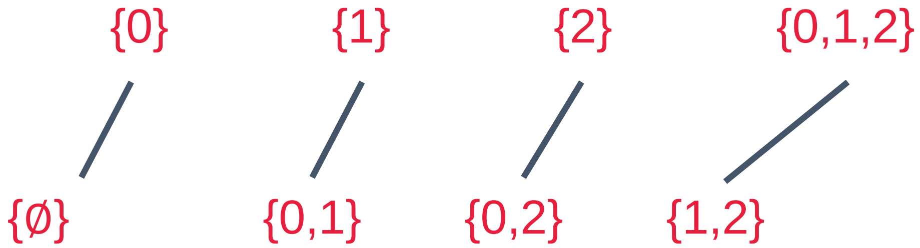
Construction example

$K = 3$

{0}
{1}
{2}
{0,1}
{0,2}
{1,2}
{0,1,2}

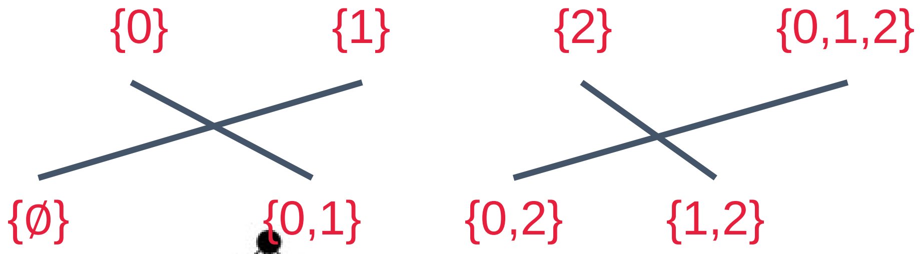
$L = (4,0,0)$ means request

- 4 times for {0}
- 0 times for {1}
- 0 times for {2}



$L = (0,4,0)$ means request

- 0 times for {0}
- 4 times for {1}
- 0 times for {2}



Definition

UNIVERSITY OF TARTU

A request vector $\boldsymbol{L}$ on K input bits is said to be short if its length satisfies:

$$|\boldsymbol{L}| \le f(K) \triangleq \frac{K}{K+1} 2^{K-1}$$

A solution to a request vector is said to be type I if singletons are not used in the solution, and the query size is 2.

$K = 3$

$$\frac{K}{K+1} 2^{K-1} = \frac{3}{3+1} 2^{3-1} = 3$$

$\boldsymbol{L} = (4,0,0)$ is not short, it uses singletons.
$\boldsymbol{L} = (0,4,0)$is not short, it uses singletons.
$\boldsymbol{L} = (3,0,0)$ is short, it has type *I* solution.

3

Proofs

Lemma 6

There is a type I solution to any request of length $2^{K-1} - K$, for $K \leq 8$.

Proof : By computer search.
It is easy to verify for $K = 3, 4$.

There is a type I solution to any short request for $K = 8$.

Proof : $2^{K-1} - K \geq \frac{K}{K+1} 2^{K-1}$

$K = 3$

$2^{K-1} - K = 1$

$\boldsymbol{L} = (1,0,0)$ has type *I* solution.
$\boldsymbol{L} = (0,1,0)$ has type *I* solution.
$\boldsymbol{L} = (0,0,1)$ has type *I* solution.

$K = 4$

$2^{K-1} - K = 4$

$\boldsymbol{L} = (1,1,1,1)$ has type *I* solution.
Example:
{0,1,2}, {1,2}
{1,2,3}, {2,3}
{0,2,3}, {0,3}
{0,1,3}, {0,1}

Lemma 6

There is a solution of query size 2 to any request of length 2^{K-1}, for $K \leq 8$.

***Proof (sketch)*:**
type I solution to any request of length $2^{K-1} - K$, and singletons solution to any request of length K.

Lemma 6

UNIVERSITY OF TARTU

There is a solution of query size 2 to any request of length 2^{K-1}, for $K \leq 8$.

Proof (sketch):
type I solution to any request of length $2^{K-1} - K$, and singletons solution to any request of length K.

There is a type I solution to any short request for $K \geq 8$.

Proof :
By induction.
$K = 8$: there is a type I solution to any short request.
$K > 8$:
let $\boldsymbol{L} = (l_0, l_1, \dots, l_{K-1})$, assume $l_0 \geq l_1 \geq \cdots \geq l_{K-1}$. (next slide)

Lemma 7

There is a type I solution to any short request for $K \geq 8$.

Proof : $\boldsymbol{L} = (l_0, l_1, \dots, l_{K-1})$.

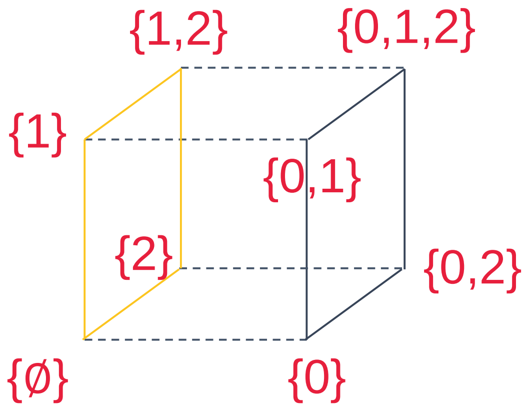
Let $\boldsymbol{L}' = \left(0, \left\lceil\frac{l_1}{2}\right\rceil, \left\lceil\frac{l_2}{2}\right\rceil \dots, \left\lceil\frac{l_{K-1}}{2}\right\rceil\right)$, $\boldsymbol{L}'' = (l_0, 0,0 \dots, 0)$, we need to show $2\boldsymbol{L}' + \boldsymbol{L}''$ has type I solution.

$$|\boldsymbol{L}'| \leq \frac{1}{2}\left(\sum_{i=1}^{K-1} l_i + \mathrm{K} - 1\right) \leq \frac{1}{2}\left(\frac{K-1}{K} f(K) + \mathrm{K} - 1\right)$$
$$\leq f(K-1)$$

Since $\boldsymbol{L}'$ has type I solution for $K-1$ inputs, each pair $(S, R) \in B$ has $(S \cup \{0\}, R \cup \{0\}) \in C$ to reconstruct $\boldsymbol{L}'$ for K inputs.

Since $\boldsymbol{L}'$ can be solved in two groups, $B \in A_0$ and $C \in \overline{A_0}$.
If we pick one element $S \in A_0/B$, then $S/\{0\} \in \overline{A_0}/C$. They can reconstruct $\{0\}$.

Example:



Except singletons, $|A_0/B| - K = 2^{K-1} - 2|\boldsymbol{L}'| - K$
$\geq 2^{K-1} - (|\boldsymbol{L}| - l_0 + K - 1) - K$
$\geq 2^{K-1} - f(K) + l_0 - K + 1 - K$
$\geq l_0$

We have type I solution for $2\boldsymbol{L}' + \boldsymbol{L}''$, for $\boldsymbol{L}$.

Theorem 8

UNIVERSITY OF TARTU

There is a solution of query size 2 to any request of length 2^{K-1}.

Proof :
$\boldsymbol{L} = (l_0, l_1, \dots, l_{K-1})$.
Let
$\boldsymbol{L'} = \left(0, \left\lfloor \frac{l_1}{2} \right\rfloor, \left\lfloor \frac{l_2}{2} \right\rfloor \dots, \left\lfloor \frac{l_{K-1}}{2} \right\rfloor\right),$
$\boldsymbol{L''} = (l_0, 0,0 \dots, 0)$, and
$\boldsymbol{L'''} = (0, l_1 \bmod 2, \dots, l_{K-1} \bmod 2)$.

$\boldsymbol{L} = \boldsymbol{L''} + \boldsymbol{2L'} + \boldsymbol{L'''}$
$\boldsymbol{L'}$ is a short request on $K-1$ inputs, thus $2\boldsymbol{L'}$ can be solved on K inputs.
$\boldsymbol{L'''}$ can be solved with singletons.
Similar to the Lemma 7,
$\boldsymbol{L''}$ can be solved by l_0 pairs.

The above proof is the recursive algorithm to find the solution.

Example

$$L = (62, 59, 58, 55, 51, 50, 49, 45, 42, 41)$$

$$L = L_1 + 2L_2 + L_3$$
$$L_1 = (62, 0, 0, 0, 0, 0, 0, 0, 0, 0)$$
$$L_2 = (0, 29, 29, 27, 25, 25, 24, 22, 21, 20)$$
$$L_3 = (0, 1, 0, 1, 1, 0, 1, 1, 0, 1)$$

$$L'_2 = (29, 30, 28, 26, 26, 24, 22, 22, 20)$$
$$|L'_2| > |L_2|$$
$$L'_2 = L_4 + 2L_5$$
$$L_4 = (29, 0, 0, 0, 0, 0, 0, 0, 0)$$
$$\boxed{L_5 = (0, 15, 14, 13, 13, 12, 11, 11, 10)}$$

Thanks for your attention.