Map-reduce.

Mnożenie macierzy

Dane $A, B \in M_{n\times n}(\mathbb{R})$

CEL: $A \circ B$ $(=C)$

$A = (a_{ij})_{i,j=1..n}$

$B = (b_{jk})_{j,k=1...n}$

$C = (c_{ik})$

$c_{ik} = \sum_{j=1}^{n} a_{ij} \cdot b_{jk}$

(1) Iloczniki $a_{ij} \cdot b_{jk}$

dane: A : "A", i, j, a_{ij}

B : "B", j, k b_{jk}

map: $(\text{"A"}, i, j, a_{ij}) \rightsquigarrow (j, (\text{"A"}, i, a_{ij}))$ $\rightsquigarrow$ $h(j)$

$(\text{"B"}, j, k, b_{jk}) \rightsquigarrow (j, (\text{"B"}, k, b_{jk}))$

reducer (j, L) {

1) Posortuj L po pierw. współ.

2) Podziel L $\rightsquigarrow L_1; L_2$

3) forall $("A", i, x) \in L_1$

forall $("B", k, y) \in L_2$

emit $(i, K, x \cdot y)$

}

$$L = [\overbrace{("A", 1, a_{1j}), \ldots, ("A", n, a_{nj})}^{L_1},$$

$$\underbrace{("B", 1, b_{j1}), \ldots, ("B", n, b_{jn})}_{L_2}]$$

$x = a_{ij}$

$y = b_{jk}$

1	1	$a_{11} \cdot b_{11}$
1	1	$a_{12} \cdot b_{21}$
1	1	$a_{13} \cdot b_{31}$

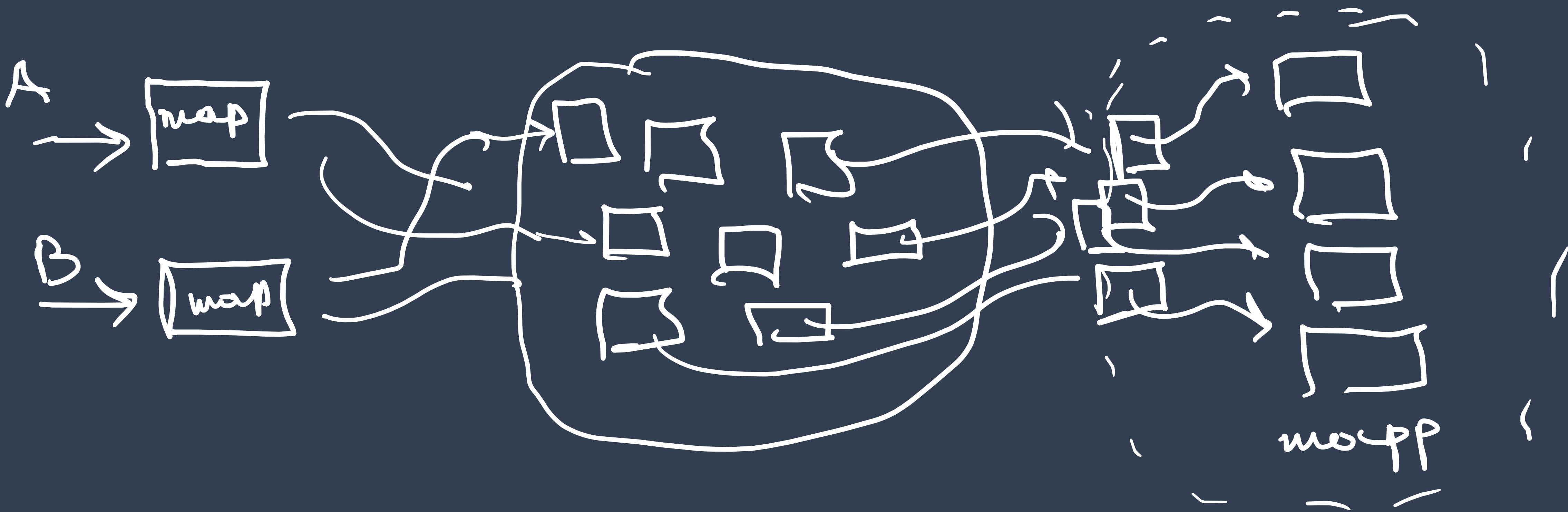
URUCHAMIAMY II RAZ MapRed.

map $(i, j, \cancel{x})$ {emit$((i,j), z)$}

reduce $((i,j), L)$ {emit $(i, j, \sum L)$;}

HADOOP : 2 razy odpalić MR

SPARK : to można zadawać wewnątrz jednego odpalenia



Inny pomysł na $A \cdot B$

$a_{11} \cdot b_{11}$

$a_{21} \cdot b_{12}$

a_{11}

b_{jk}

tu potrz.

a_{11}

tu b_{jk} jest potrzebne

map: $(\text{"A"}, i, j, a_{ij}) \longrightarrow ((i,1),(\text{"A"}, j, a_{ij})), ((i,2),(\text{"A"}, j, a_{ij})) \ldots ((i,n),(\text{"A"}, j, a_{ij}))$

$(\text{"B"}, j, k, b_{jk}) \longrightarrow ((1,k),(\text{"B"}, j, b_{jk})), \ldots ((n,k),(\text{"B"}, j, b_{jk}))$

reducer ZADANIE

$h((i,j)) \in \{0, \ldots, R-1\}$

Zadanie: mamy procesor 10 GHz.

Jaką odległość pokona światło w jednym cyklu procesora?

Metode 3.

$$\left[\begin{array}{c:c} A_{11} & A_{12} \\ \hdashline A_{21} & A_{22} \end{array}\right] \circ \begin{bmatrix} B_{11} & B_{12} \\ B_{22} & B_{22} \end{bmatrix} = \begin{bmatrix} A_{11}B_{11} + A_{12}B_{21} & \cdots \\ A_{21}B_{11} + A_{22}B_{22} & \cdots \end{bmatrix}$$

$$\mathcal{M}_{n \times n} = (M_{n,n}(R), +, \circ)$$

$$\mathcal{M}_{k \times k}(\mathcal{M}_{n \times n}) \underset{\mathrm{ISO}}{\cong} \mathcal{M}_{k \cdot n \times k \cdot n}$$

TW.

$$(A, i, j, a_{ij}) \rightsquigarrow \{ (i \operatorname{div} k, 0), (A, i \bmod k, j \bmod k, a_{ij}) \}$$

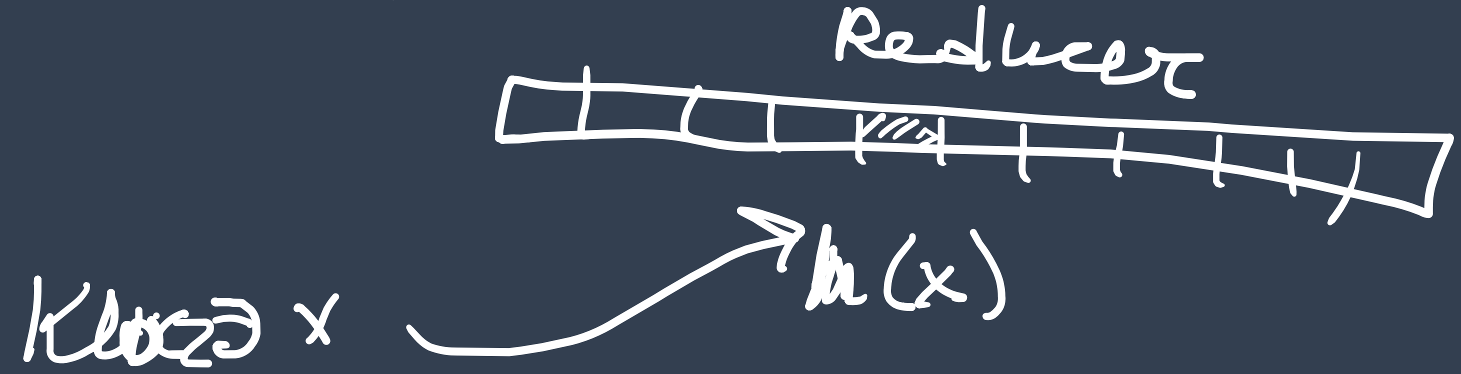
$$A = (a_{ij})_{i,j=0,\dots,kn-1}$$

$$C: \begin{bmatrix} \underbrace{A_{11} \cdot B_{11} + A_{12}B_{21} + \dots}_{n}, & \\ & \end{bmatrix}$$

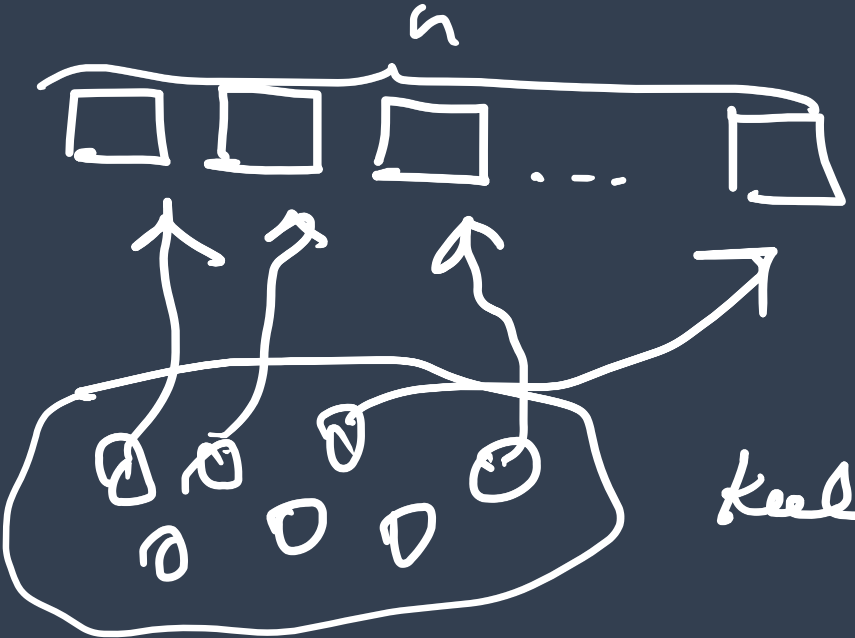
$$\frac{(kn)^2}{k} + \frac{(kn)^2}{k} = 2kn^2$$

$$(kn)^3 = k^2 n^2$$

HASHOWANIE



MODEL KULE-URNY



urny / bucket.

Rzucamy kulkę w urny losowo, rozkład jedn. niezależnie

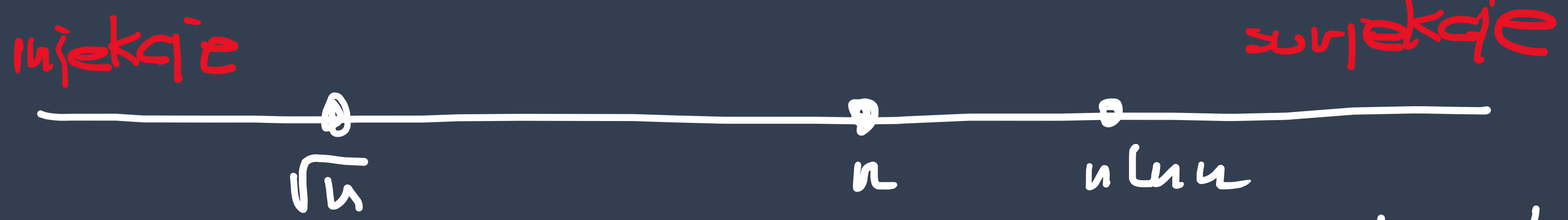
kulki / klucze

1. paradoks URODZINOWY:

$\sim$ po $\sqrt{n}$ urn są kolizje

2. coupon coll. problem (CCC)

$\sim n \ln n$: znikają puste urny



k - l. kul

$k = n$: Ile jest pustych urn?

$$E[L] = \sum_{i=1}^{n} P[U_i \text{ jest pusta}] = n \cdot \left(\frac{n-1}{n}\right)^n \simeq \frac{n}{e} \approx \frac{1}{3} n.$$

WNIOSEK: liczba reducerów R nie może być zbyt duża

$R \cdot \ln R \ll$ liczby różnych kluczy

dane $\rightsquigarrow$ (klucz, x)

Q: { universal hashing
k-indep. hashing

Zał. że mamy funkcje haszujące

$$h: \Sigma \longrightarrow \{0, 1, .., n-1\}$$

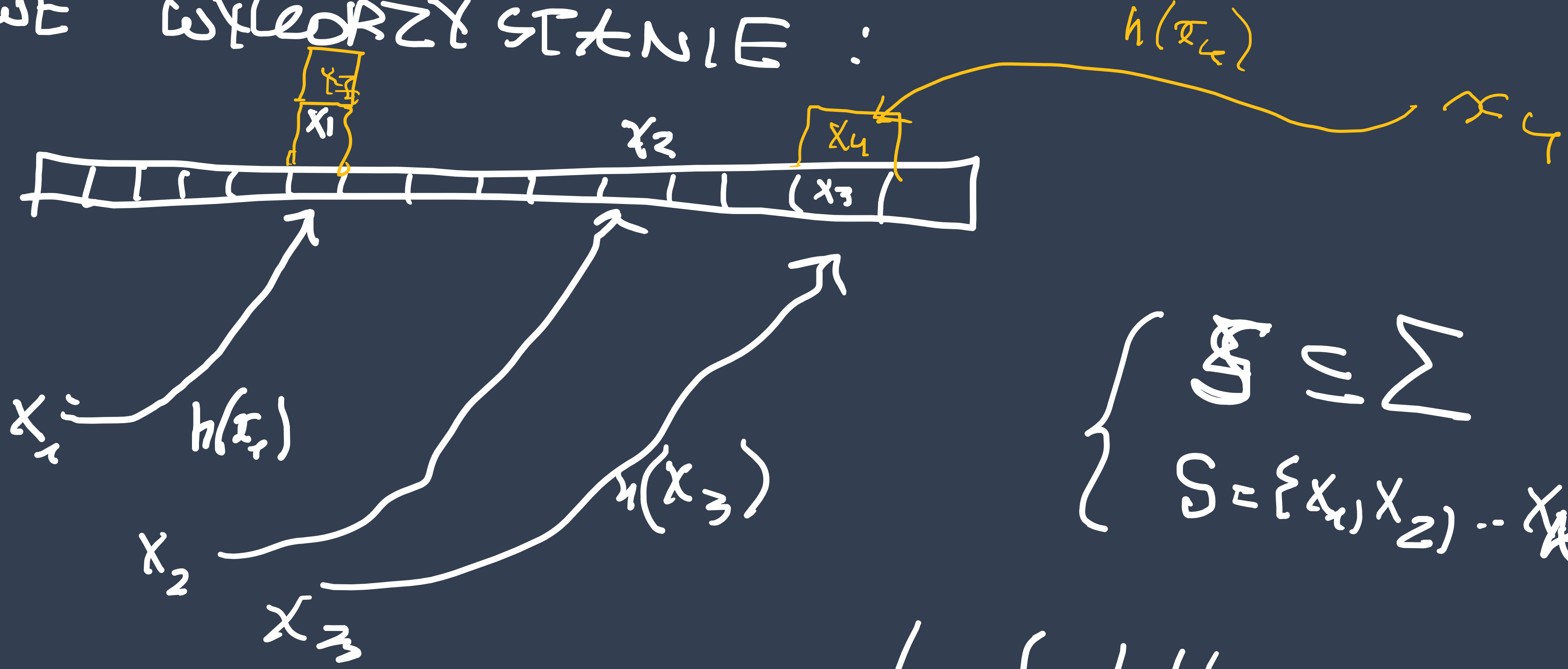
- bardzo porządn. "losowych probab."
  - $h(x)$ – rozkł. się jedn. w $\{0, .., n-1\} = [n]$
  - $x \neq y$: $h(x), h(y)$ są niezależne od siebie (z dużym prawdop.)

$$Pr[h(x) = h(y)] \approx \frac{1}{n}$$

$|\Sigma| > n$ h nie jest 1-1

BIG : MURMUR hash function

TYPOWE WYKORZYSTANIE :



$$\left\{ \begin{array}{l} S \subseteq \Sigma \\ S = \{x_1, x_2, \ldots x_n\} \end{array} \right.$$

$U_{h(x)} \ni x$

hash table

Q : $x \in S$?