PAR-cv-1/1 5.10.2006

PAR - CV

Paralelní redukce

- $\sum_{i=1}^{n} a_i$

$\boxed{\bullet \text{ ZADÁNÍ}}$

- operace: čtení z paměti do reg.
  zápis —"—
  součet —"—
  komunikace
  } vše trvá jednotkový čas

sekvenčně: $SL = \Omega(n)$

spodní mez

$SU(n) = \underbrace{n}_{\text{čtení}} + \underbrace{n-1}_{\text{sčítání}} = 2n-1 = \Theta(n)$

nejlepší horní odhad

paralelně:

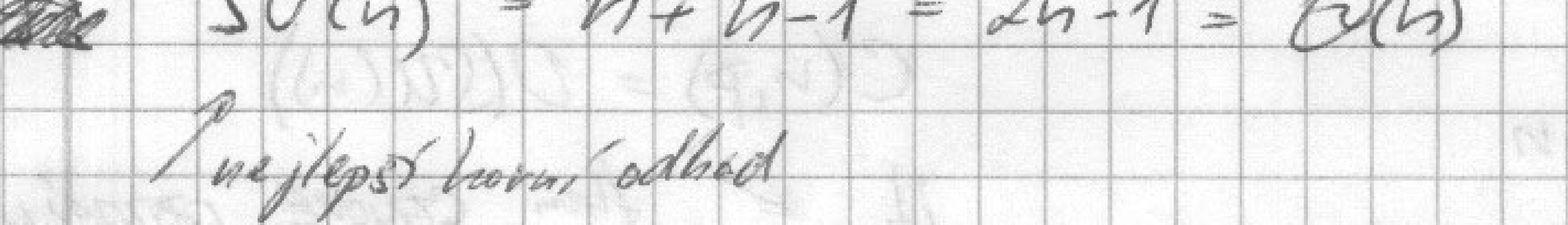
- $p = n$ počet čísel $\quad n = 2^k$

počet procesorů

- − propojovací síť: úplný graf (každý spojený s každým)

PŘ řešení: na začátku data v registrech

$\log_2 n$



binární strom, jeho výška ~~počet~~ $\log_2 n$

$T(n,p) = T(n,n) = \underbrace{2}_{\text{poslání a součet}} \log n = \Theta(\log n)$

$\Rightarrow n \neq 2^k \quad T(n,n) = 2 \cdot \lceil \log n \rceil = \Theta(\log n)$

PRÁCE:

$W(n,n)$ = kolik práce provedou všechny procesory v globálu

$$= 2\left(\frac{n}{2}+\frac{n}{4}+\dots+1\right) = 2\sum_{1}^{\log n}\frac{n}{2^i}$$ geometrická řada

2 časové jednotky

$$\cancel{\dots} = \sum_{i=0}^{\log n}\frac{n}{2^i} = n\,\frac{2(n-1)}{n} = 2n-2$$

$n \neq 2^k$ $\boxed{W(n,n) = 2(n-1)}$

vždy

počet šipek

~~počet op~~

kolik práce stojí jedna šipka

CENA ALGORITMU: používá se k porovnání efektivity paralelního řešení k sekvenčnímu

$$C(n,p) = p\cdot T(n,p)$$

$$C(n,n) = 2n\log n$$

$C(n,p) = O(SU(n))$

if ← then CENOVĚ OPTIMÁLNÍ

$2n\log n \neq O(n) \Rightarrow$ NENÍ CENOVĚ OPTIMÁLNÍ

⇒ algoritmus je rychlejší, ale není efektivní, nevytíží p na 100%.

⇒ zrychlení není lineární

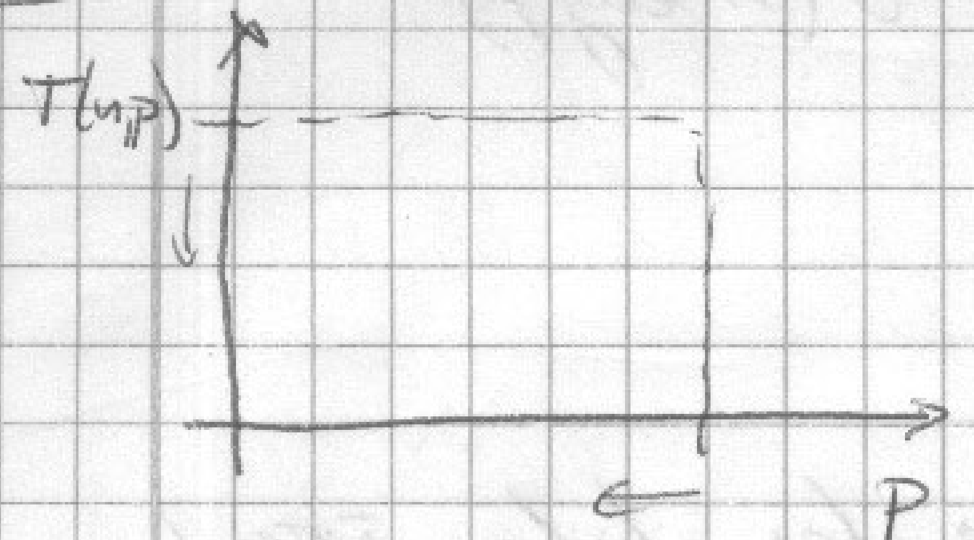
ZRYCHLENÍ:

$$S(n,n) = \frac{SU(n)}{T(n,n)} = \frac{2n-1}{2\log n} = O\left(\frac{n}{\log n}\right) \qquad S \in \langle 1;p\rangle$$

EFEKTIVITA:

$$E(n,n) = \frac{SU(n)}{C(n,n)} = \frac{2n-1}{2n\log n} = O\left(\frac{1}{\log n}\right) \qquad E \in (0;1\rangle$$

CENA



Př: $\sum_{i=1}^{n} a$; $p < n$

2 postupy
- zanovem od začátku navrhovat nový algoritmus
- simulujeme n virtuální procesorů na p

SIMULACE PO ŘÁDCÍCH = způsob mapování

12	13	14	15
8	9	10	11
4	5	6	7
0	1	2	3
0	1	2	3

$p = 4$, $n = 16$

(1)

virtuální	12 ← 13		14 ← 15	
	8 ← 9		10 ← 11	
	4 ← 5		6 ← 7	
	0 ← 1		2 ← 3	
fyzické	0	1	2	3

je nutné ukládat do paměti

(2)

25	13	29	15
17	9	21	11
9	5	13	7
1	1	5	3
0	1	2	3

(3)

54	13	29	15
38	9	21	11
22	5	13	7
6	1	5	3
0	1	2	3

120

trvá jeden mikrokrok (načíst z MEM, poč., sečti, uložím)

$$T(n,p) = \underbrace{4 \cdot \left(\frac{n}{p}\right) \cdot \log p}_{\text{paralelní redukce}} + \underbrace{4\left(\frac{n}{p} - 1\right)}_{\text{sekvenčně sečtu (krok 3)}}$$

počet mikrokroků v (1)

$$\doteq 4\frac{n}{p}\log p + 4\frac{n}{p} = 4\frac{n}{p}(\log p + 1) = \Theta\left(\frac{n}{p}\log p\right) \qquad 1 < p < n$$

jednoduché pravidlo: co můžeme vytknout, konst. zanedbat, zpět pronásobit (n a p musí vždy zůstat)

$$C(n,p) = p \cdot T(n,p) = p \cdot \frac{n}{p} \log p = O(n \log p)$$

$$n \log p = O(n)$$

předpoklad $p = O(1) \Rightarrow k \cdot n = O(n)$ = matematické řešení

správné, ale degeneroval jsem ~~to~~ čas na SEK. $\Rightarrow$

není řešení

NENÍ CENOVĚ OPTIMÁLNÍ

<u>ŘEŠIT JINAK:</u>

SIMULACE PO SLOUPCÍCH

(1)

3	7	8	12
2	6	7	11
1	5	6	10
0	4	5	9
0	1	2	3

(2)

3	7	11	15
5	13	21	29
1	5	9	13
6	22	38	54
0 ←	1	2 ←	3

(3)

3	7	11	15
5	13	21	29
1	5	9	13
28	22	92	54
0	1	2	3

$T(n,p)$ (1) 2× čtení, součet, zápis $4\left(\frac{n}{p} - 1\right)$

(2) a (3) poslat, součet $2 \log p$

$$T(n,p) = 4\left(\frac{n}{p} - 1\right) + 2 \log p = \Theta\left(\frac{n}{p} + \log p\right)$$

$$C(n,p) = \Theta(n + p \cdot \log p)$$

$n + p \cdot \log p = \Theta(n)$ odečtu n

$p \cdot \log p = O(n)$

$n = \Omega(p \cdot \log p)$ algoritmus ~~má~~ za tohoto předpokladu je cenově optimální, musíme najít přesný vztah

PAR-CV-1/5

$$\sum_{i=1}^{n} a_i \,, \quad p < n$$

$$0 < E_0 < 1$$

~~...~~ HLEDÁNÍ PŘESNÉHO ŘEŠENÍ

$$\forall n_p = \Omega(\psi_1(p)) : E(n_p, p) \geq E_0$$ známe p, kolik musí být dat n, aby bylo cenově efektivní

$$\forall p_n = O(\psi_2(n)) : E(n, p_n) \geq E_0$$ kolik musí být p aby bylo lepší než E_0

Výpočet:

$$E(n,p) = \frac{SU(n)}{C(n,p)} = \frac{2n-1}{4(n-1)+2p\log p} \geq E_0$$

$$\doteq \frac{2n}{4n+2p\log p} \geq E_0$$

$$2n \geq E_0(4n + 2p\log p)$$

$$\cancel{n \geq \frac{2-4E_0}{2E_0} p \log p}$$

$$n \geq \frac{2E_0}{2-4E_0} p \log p$$

$$\cancel{n \geq \frac{1-2E_0}{E_0} p \log p}$$

$$n \geq \frac{E_0}{1-2E_0} p \log p$$

$$k = \frac{E_0}{1-2E_0}$$

$$n \geq k \cdot p \log p$$

$$\psi_1(p): n = \Omega(p \log p)$$ ⇐ ŠKÁLOVATELNOST ALGORITMU JE DOBRÁ, FCE JE POMALU ROSTOUCÍ

PAR-CV-1/6

Ψ_2: $k \cdot p \log p \leq n$ $\qquad \beta = \frac{1}{k}$

$p \cdot \log p \leq \beta \cdot n$

Jak to vyřešit?

PAR-CV-2/1 19.10.2006

$\Psi_2 = ?$ $\qquad p \cdot \log p \leq \beta n$

$p = f(n)$

$p = \log n$ splňuje

$p = \log^2 n$ splňuje

⋮

takových odhadů je relativně hodně

v $p \log p \cdot \beta n$ nám vadí $\log p$ ⇒ najdeme pro něj aproximaci

platí $1 < p < n$

1, zmenšení LS

$\log p \Rightarrow 1$ substituce

$p = \beta \cdot n$ je <u>horní mez</u>

$\beta < \beta \cdot n$

zpětné tvrzení $\beta n \log \beta n \geq \beta n$ platí

2, zvětšení LS

$\log p \Rightarrow p$ substituce

$p^2 = \beta n$

$p = \sqrt{\beta n}$ <u>dolní mez</u>

ověření pomocí zpětného dosazení

$\sqrt{\beta n} \log \sqrt{\beta n} \leq \beta n$

$\frac{1}{2} \sqrt{\beta n} \log \beta n \leq \beta n$ <u>platí</u>

$\boxed{\sqrt{\beta n} < p < \beta n}$ na obou stranách není řádově stejná funkce ⇒ neznáme přesné řešení

$\log \sqrt{\beta n} \le p < \log \beta n$

$\boxed{\frac{1}{2} \log \beta n < \log p < \log \beta n}$ LS i PS je řádově stejná fce

$p \cdot \log \beta n \le \beta n$

$$p \le \frac{\beta n}{\log \beta n}$$

$$\psi_2 : p = O\left(\frac{\beta n}{\log \beta n}\right)$$

$\sum_{i=1}^{n} a_i , p < n$ design nového algoritmu

$T(n,p) = \underbrace{2\frac{n}{p}}_{\text{sek. část}} + \underbrace{2 \log p}_{\text{par. č.}}$ dvojky jsou počty operací (součet a uložení; poslání a součet)

řádově stejné jako simulace

pokud bude komunikace trvat m: $T(n,p) = 2\frac{n}{p} + (m+1) \cdot \log p$

řádově pořád stejná

⇒ fce izoefektivnosti budou stejné, pouze jiné konstanty

$$k = \frac{(m+1)E_0}{1-2E_0} \qquad \beta = \frac{1}{k} = \frac{1-2E_0}{(m+1)E_0}$$

$p_1 = 8 \quad n_1 = 10^{12} \quad E_0 = 0{,}75$

$p_2 = 12 \quad n_2 = ?$

$\psi_1(p) : n = \Omega(p \log p) \qquad \psi_2(n) : p = O\left(\frac{n}{\log n}\right)$

$n_1 = k p_1 \log p_1$ ⇒ $k = \frac{n_1}{p_1 \cdot \log p_1}$

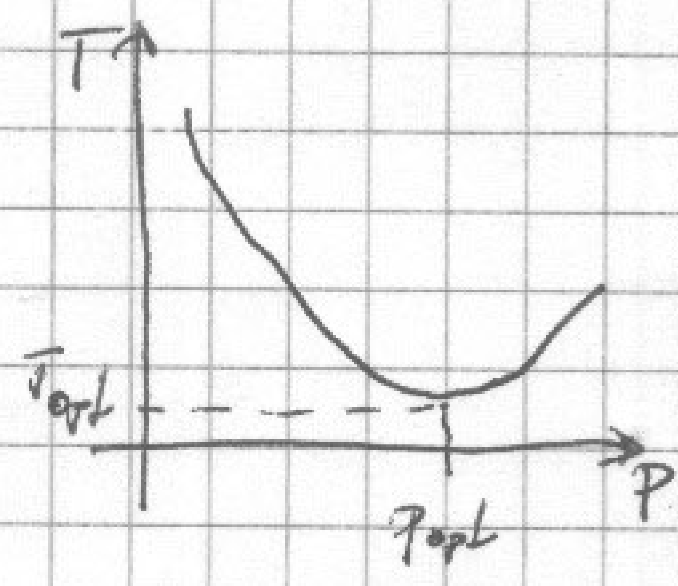
$n_2 = k p_2 \log p_2$

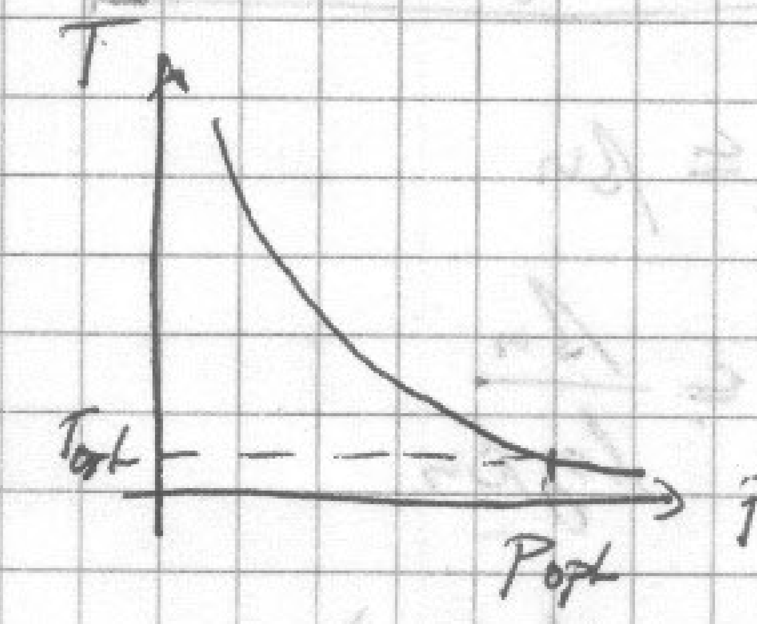
$$n_2 = \frac{n_1}{p_1 \log p_1} \cdot p_2 \cdot \log p_2$$

$$n_2 = \frac{10^{12}}{8 \cdot \log 8} \cdot 12 \cdot \log 12$$

$$\Psi_3: \forall p = \Omega(\Psi_3(n)): T(n,p) < T(n, p_{opt}) = T_{opt}(n,p)$$

optimální čas ⇒ minimální čas





$p = n$ pro většinu algoritmů

$p^2 = n$ pro maticové operace

$$T(n,p) = 2\frac{n}{p} + 2\log p$$

$$\frac{dT(n,p)}{dp} = -\frac{2n}{p^2} + \frac{2}{p \cdot \ln 2} = 0$$

$$p_m = n \cdot \log 2 = 0{,}6\, n$$

$n = 10$

$p = 6$ ⇒ $4\times$ 2 čísla

$2\times$ 1 číslo ← to je zbytečné, stačil by 1 procesor ⇒ $p_m = \frac{1}{2} n$

$$T_{min} = T\left(n, \frac{n}{2}\right) = 2\frac{n}{\frac{n}{2}} + 2\log\frac{n}{2} = 4 + 2\log n - 2\log 2 = \underline{2\log n + 2 = T_{min}}$$

$2\frac{n}{p} + 2\log p = O(\log n)$ řešíme řádově!

rozpadá se to na 2 rovnice

1) $2\frac{n}{p} = O(\log n)$

$$\frac{n}{\log n} = O(p)$$

$$p = \Omega\left(\frac{n}{\log n}\right)$$

2) $2\log p = O(\log n)$

$$p = O(n)$$

..

řešení: $\left\langle \frac{n}{\log n} ; n \right\rangle$

hledáme minimum, tj. levou část intervalu $\Rightarrow$ 4_2: $p = \Omega\left(\frac{n}{\log n}\right)$
časově optimální

$p = \Theta\left(\frac{n}{\log n}\right) \Rightarrow$ cenově i časově optimální

PRAM

- výpočetní model paralelního počítače se sdílenou pamětí, synchronní, model neřeší konflikty přístupu (musí řešit algoritmus)

3 operace: čtení, lokální výpočet, zápis

obecný algoritmus: $\langle RLW \rangle^{t_p}$

$T_p(n, p_p) = 3 . t_p$

$C_p(n, p_p) = 3 . p_p . t_p = O(p_p t_p) \Rightarrow O(SU(n))$
je cenově optimální

APRAM

- asynchronní (operace nemusí začínat ve stejném okamžiku a trvají různě dlouho)

4 operace:

- čtení, zápis ... m
- lok. výpočet ... 1
- bariéra (synchronizace) ... $b(p)$
- pipeline čtení, zápis ... $m+k-1$

číslo $b = m . \log p$ (paralelní redukce)

(čtení nebo zápis za sebou)

obecný algoritmus: $\langle RBLWB \rangle^{t_p}$ bariéry vždy stačí 2, můžou být také počet R a W

$T_a = m + b(p_p) + 1 + m + b(p_p) = 2m + 2b(p_p) + 1 = O(b(p_p))$ $p_a = p_p$
$\doteq 2b(p_p)$

$T_a(n_a, p_a) = T_a(n, p_p) = 2 . b(p_p) . t_p$

$C_a = 2 . b(p_p) . p_p . t_p \neq O(C_p)$ simulace není cenově optimální

řešení: $p_a' = \frac{p_a}{b(p_a)} = \frac{p_p}{b(p_p)}$

PRAM na APRAMu

PRAM: $\dfrac{L, R, W}{1}$

(*)

L	1
R/W	m
bariéra… B	$b(p) = m.\log p$
$\langle R\rangle^d$, $\langle W\rangle^d$	$\}\ m+d-1$

$m << p$

PRAM: p-procesorů $\langle RLW\rangle^t$ $3t$ $C(n,p) = 3tp$ VĚTA 3.17

APRAM: $\boxed{p_A = \dfrac{p}{b(p)}}$ $\langle RBLWB\rangle^t$

lépe: $\langle R^j B L^j W^j B\rangle^t$

$j = b(p)$ – doba vykonávání RLW bude stejná jako čekání na bariérách

$\langle R^{b(p)} B L^{b(p)} W^{b(p)} B\rangle^t$

dosadím (*) bariéry jen na APRAMu

$$T(n,p) = t.(\underbrace{2(m+b(p)-1)}_{R,W} + \underbrace{b(p)}_{L} + \underbrace{2b(p_A)}_{\text{bariéra}})$$

$$b(p_A) = m.\log\left(\frac{p}{b(p)}\right) = m.\log\left(\frac{p}{m.\log p}\right) = m.\log p - m.\log\log p - m.\log m =$$

$$= \leq m.\log p = b(p) \quad \text{omezení shora}$$

dosadím:

$$T(n,p) = t(2m + 5b(p))$$

zanedbám, řádově menší

$$C_A(n, p_A) = p_A.t.(2m + 5b(p)) = t\left(\frac{2m.p}{b(p)} + 5p\right) = 5pt + O\left(\frac{p}{\log p}\right).t$$

$$\doteq 5pt$$

PRAM na APRAM

Př: PAR. REDUKCE 1) $T(n,p) = 4\frac{n}{p} + 2\log p$

2) $p = \frac{n}{\log n}$

dosadíme 2 → 1

$T(n,p) = 4.\log n + 2\log n - 2\log\log n \doteq 6\log n$

$C(n,p) \doteq 6n$

Simulace PAR. REDUKCE na APRAM

(využití věty 3.17)

APRAM: $p_A = \frac{p}{b(p)} = \frac{\frac{n}{\log n}}{m.\log(\frac{n}{\log n})} = \frac{n}{\log n(m.\log n - m.\log\log n)} =$

$\doteq \frac{n}{n.\log^2 n}$

$C(n,p_A) = 5t_p = 5.6.\log n . \frac{n}{\log n} = \underline{\underline{30n}}$ ← dosadíme časovou složitost alg.

<u>NÁVRH cenově optimálního</u> algoritmu pro APRAM, PAR. REDUKCE

$p_A = \frac{p}{b(p)} = \frac{n}{m.\log^2 n}$

provádí se zřetězeně $p = \frac{n}{\log n}$

$R\left\langle \begin{matrix} R \\ L \end{matrix} \right\rangle^{n/p_A} BW \langle B\tilde{R}\tilde{R}LBW \rangle^{\log p = \log n}$

pipelinové provádění

$T(n,p) = m + \frac{n}{p_A} + m + b(p_A) + \log n(2b(p_A) + 2m + 1) =$

$= 2m + \frac{n}{p_A} + b(p_A) + \underline{2\log n(b(p_A) + m)}$ ← při DYN. redukci se změní 2 → 1

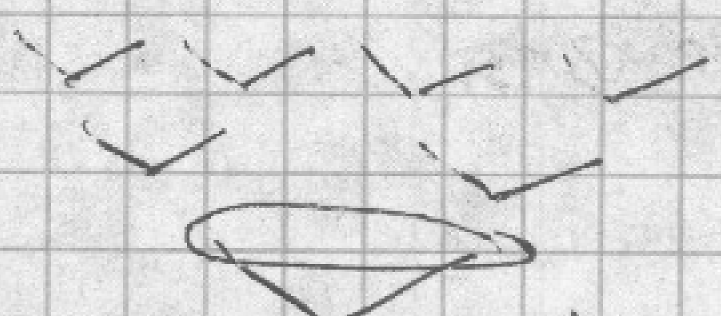
– při počítání ceny se chová omezit na řádově nejvyšší ceny

$$C(n, p_A) = \cancel{p_A} \cdot \frac{n}{\cancel{p_A}} + 2 p_A \log n \cdot m \log n =$$

$$= n + \frac{2 n m \log^2 n}{m \log^2 n} \doteq \underline{3n + O(n)}$$

DYNAMICKÁ SYNCHRONIZACE

- synchronizují se jen aktivní procesory ve stromu pro paralelní redukci



$$2b(p_A) + 2b\left(\frac{p_A}{2}\right) + 2b\left(\frac{p_A}{4}\right) + \dots + 2b(2) =$$

(poslední synchronizuje 2 proc.)

$$= 2\left(m \cdot \log p_A + m \cdot \log \frac{p_A}{2} + m \cdot \log \frac{p_A}{4} + \dots + m \log 2\right) =$$

$$= 2m\left(\log p_A + \log p_A - 1 + \log p_A - 2 + \dots + 2 + 1\right) =$$

aritmetická řada s diferencí 1

$$= 2m \sum_{i=1}^{\log p_A} i = 2m \frac{\log p_A (\log p_A + 1)}{2} \doteq \underline{\underline{m (\log p_A)^2}} \leq m \log^2 n$$

$$C_D(n, p_A) \doteq 2n + O(n)$$

VNOŘOVÁNÍ

vexp – expanze vrcholu

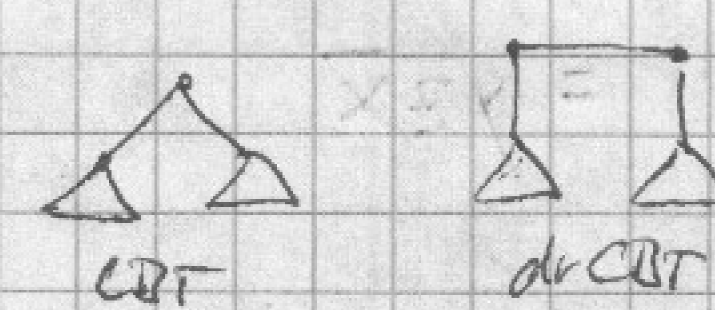
load – zátěž

dil – prodloužení

ecng

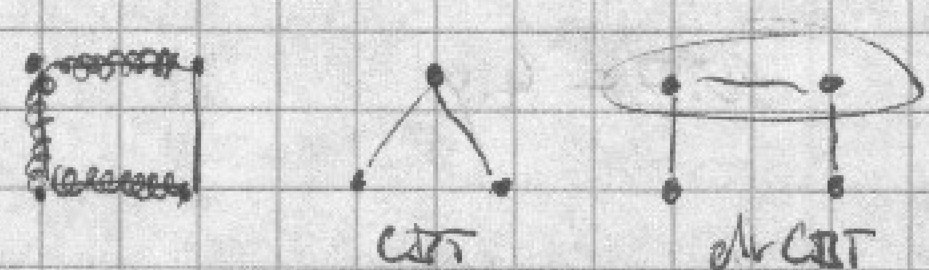
CBT → Q (úplný bin. strom do hyperkrychle)

dvCBT

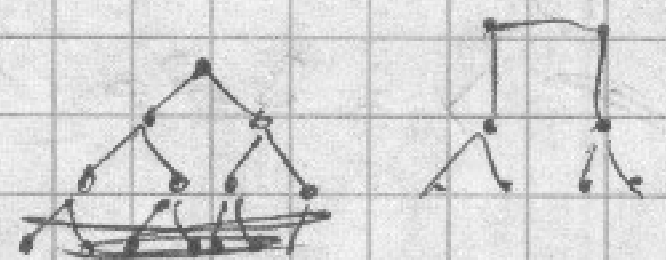


Q_1

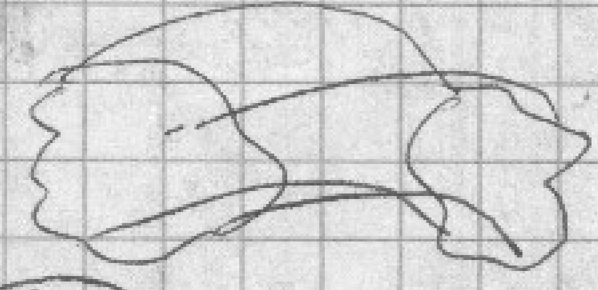
Q_2



dil = 1
load = 1

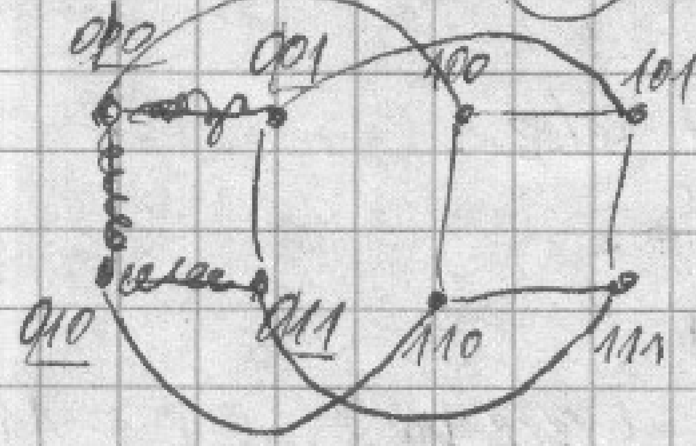


MAT. INDUKCÍ NALÉZT OBECNÝ POSTUP

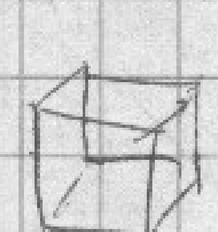


Z hyperkrychle a vnořením je propojím

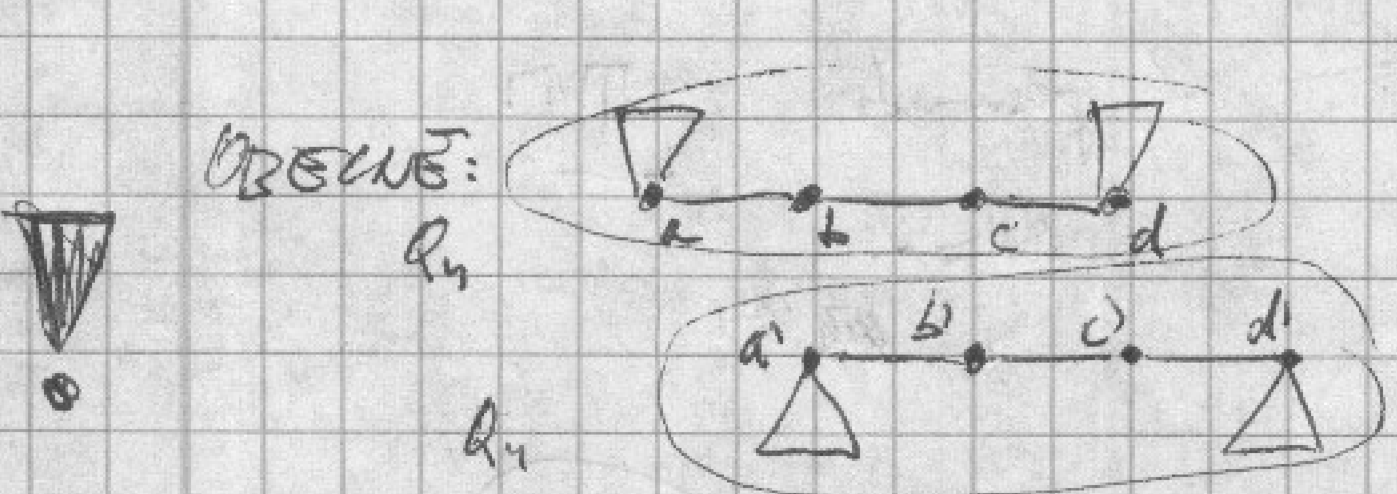
Q_3



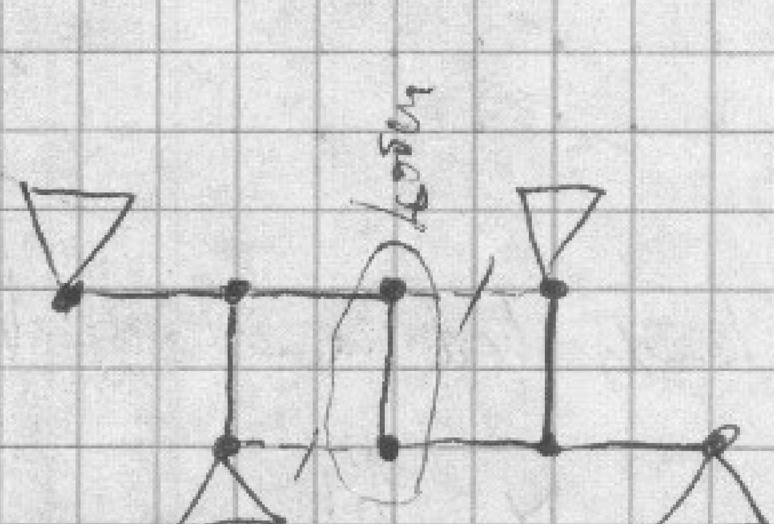
přidám 3 bit tak vrcholy se stejnými bity ⟨1:0⟩ propojím



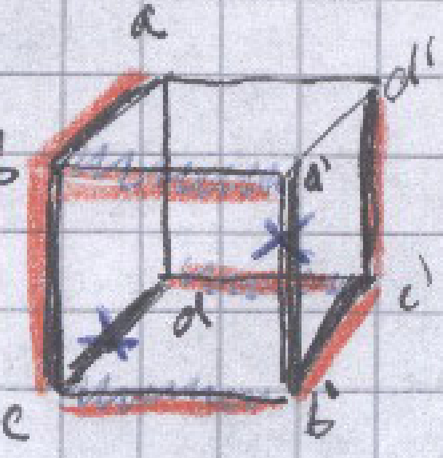
OBECNĚ:

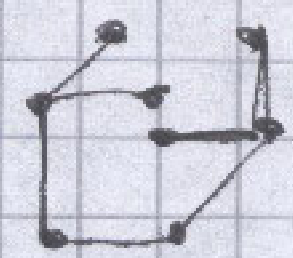


⟹



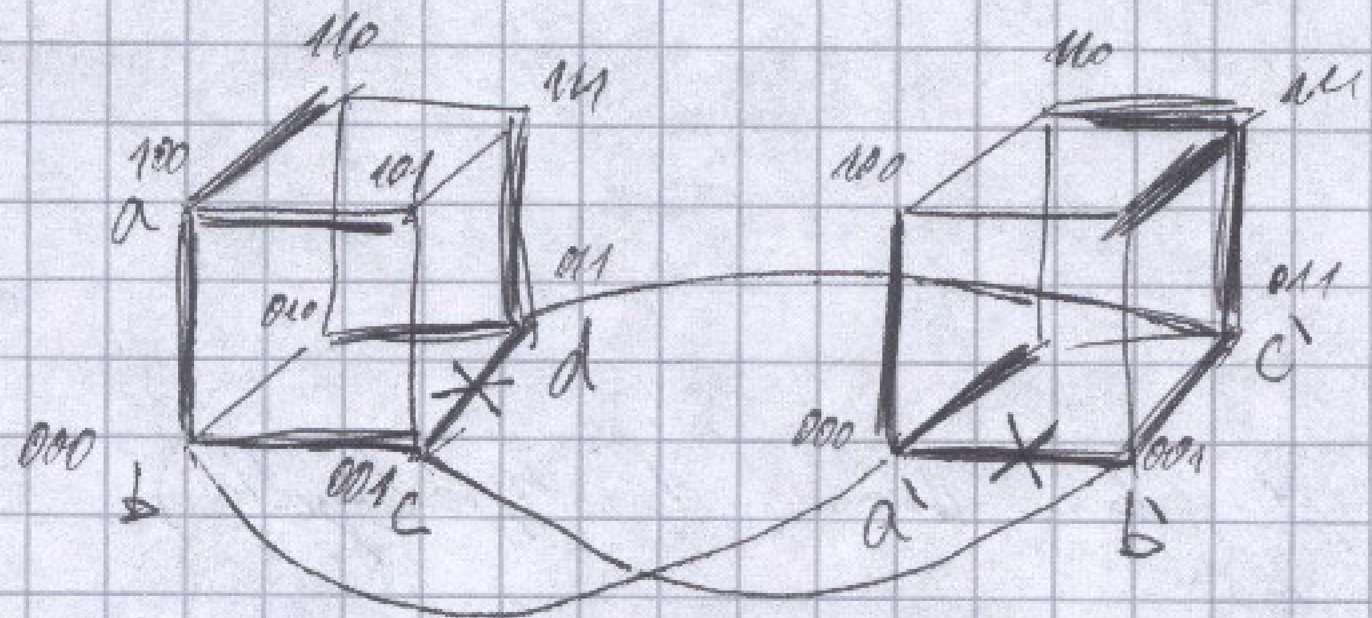
PAR-CV-4/12





TRANSFORMACE

Q_3	Q_3'
a b c	a' b' c'
a = 100	a' = 000
b = 000	b' = 001
c = 001	c' = 011



TRANSFORMACE ZOBRAZENÍ: $\varphi(xyz) = yz\overline{x}$ (1. 2. 3. bity)

2D M → 2D M (mřížka do mřížky)

$M(10,7) \rightarrow M(4,?)$

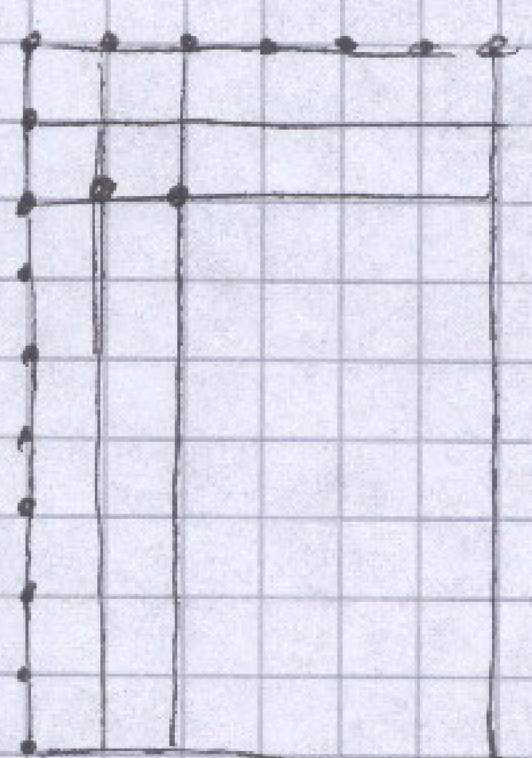
load = 1

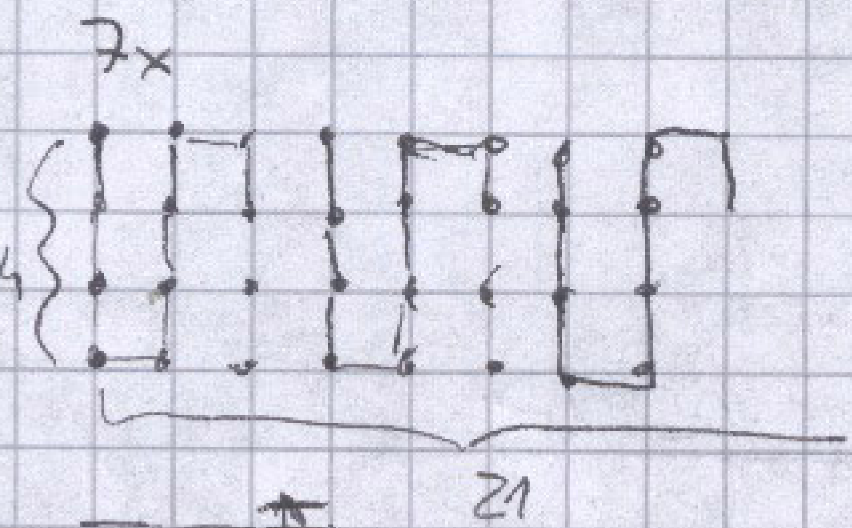
dil = ?

eeng = ?

$y \geq 18$

$10 \cdot 7 \leq 4 \cdot y$



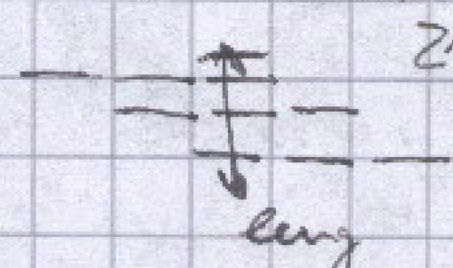


$y = 3 \cdot 7 = 21$

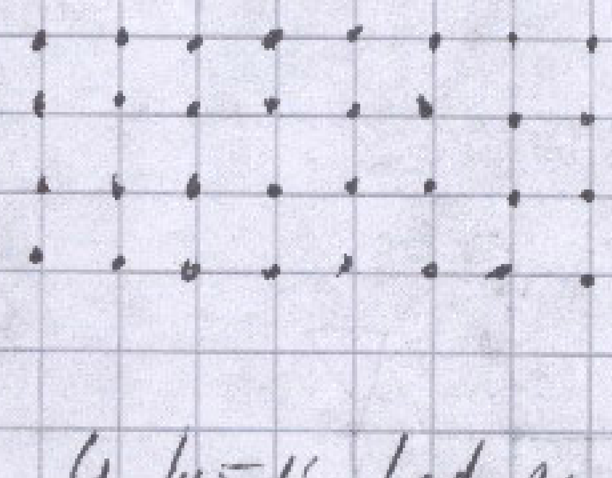
#val = 4 · 21 = 84

dil = 3 prodloužení

eeng = 4 zahlcení hran (max.)



load = 2 několik možností



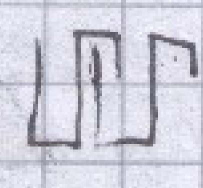
1, hady na sebe

dil = 3

eeng = 8

$y = 3 \cdot 3 + 2 = 11$

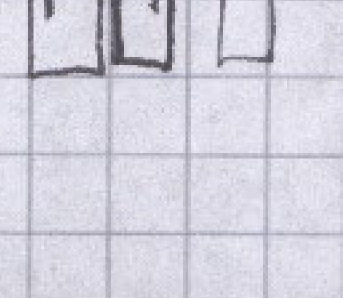
2, posunutí



dil = 2

y = 14

3, zrcadlově



y = 11

dil = 5

4, každý had na sebe

y = 14

dil = 2

PAR-CV-5/1

TŘÍDĚNÍ

SUDO-LICHÉ TŘÍDĚNÍ NA LINEÁRNÍM POLI (BOUBLE SORT)

- kolečko procesor
- číslo = data

4	8	10	11	1	6	7	8
4	8	10	11	1	6	7	8
4	8	10	1	11	6	7	8
4	8	1	10	6	11	7	8

- 2 CPU si pošlou vzájemně čísla
- levý procesor si nechá menší číslo
- pravý větší

OBECNĚ:

lokální setřídění

(n−1) krát dášm data od 2CPU dohromady

$$T(n,p) = \frac{n}{p}\log\frac{n}{p} + (p-1)\frac{n}{p} = O\left(\frac{n}{p}\log\frac{n}{p} + n\right)$$

- lokálně setřídím
- vymění 2CPU data, nechá si menší polovinu

$$C(n,p) = p \cdot T(n,p) = O\left(n\log\frac{n}{p} + np\right)$$

efektivita:

$$\frac{n\log n}{n\log\frac{n}{p} + np} \geq E_0$$

vyřešíme pro n a p, dostaneme Ψ_1, Ψ_2

$$\log n \geq E_0(\log n - \log p) + E_0 p$$

$$\log n \geq \frac{E_0}{1-E_0}(p - \log p)$$

subst.: $k = \frac{E_0}{1-E_0}$

PAR-cv-5/2

$$n \ge 2^{k(p-\log p)} = \cancel{\left(\frac{2^p}{p}\right)^k} = \left(\frac{2^p}{p}\right)^k$$

funkce škálovatelnosti:

$$\boxed{\psi_1 : n = \Omega\left(\left(\frac{2^p}{p}\right)^{O(1)}\right)}$$

je špatná, dobrá by byla když malá změna vyvolá malou odezvu. Tady je exponenciální

Řešení pro p:

$$\underbrace{p - \log p}_{<p \;\ldots\; \text{úvaha}} \le k' \log n$$

$$p - \log p \le \boxed{p \le k' \log n}$$

$$\boxed{\psi_2 : p = O(\log n)}$$

jinak

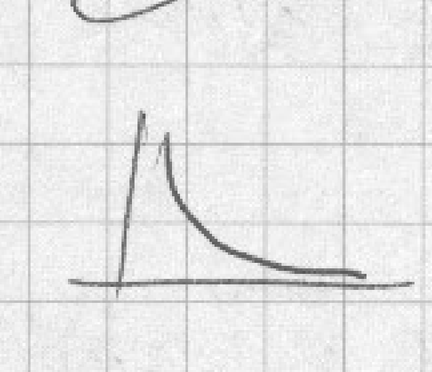
$$p < \log n + \log p \le k'' \log n$$

$$\log p \le \log n$$

$$p < n$$

$T_{min} = n \quad \leftarrow$ pro $p = n \quad T(n, p) = \frac{n}{p} \log \frac{n}{p} + n$

permanentně klesající



$\psi_3 : \; p = \Omega(f(n))$

$T = O(T_{min})$

$$\frac{n}{p} \log \frac{n}{p} + n = O(n)$$

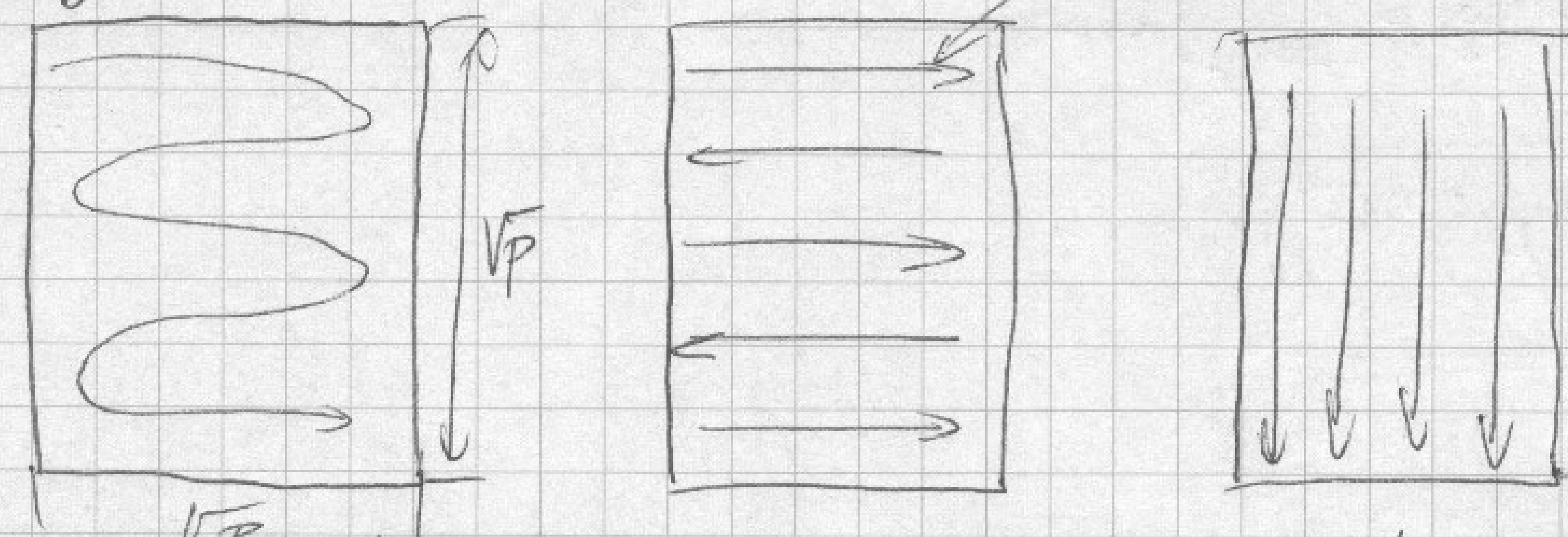
$$\psi_3 : p = \Omega(\log n)$$

PAR-W-5/3

SHEART SORT 2-D MESH

fáze algoritmu:

setřídím tímto směrem



výsledná posloupnost setříděných dat

1. fáze

2. fáze $\log \sqrt{p}$

$\log \sqrt{p} + 1$

cykly se střídají

$p < N$

počet (velikost) řádku

$$T(N,p) = \underbrace{\frac{N}{p} \log \frac{N}{p}}_{} + \underbrace{(\log p + 1)}_{} \underbrace{\sqrt{p}}_{} \frac{N}{p}$$

lokální rozdělení a setřídění

složitost BUBBLESORTU ve řádku

počet fází

$$T(N,p) = \frac{N}{p} \log \frac{N}{p} + \frac{N}{\sqrt{p}} \log p$$

$$C(N,p) = N \log \frac{N}{p} + N \sqrt{p} \log p$$

$$E(N,p) = \frac{\cancel{N} \log N}{\cancel{N} \log \frac{N}{p} + \cancel{N} \sqrt{p} \log p}$$

$$\frac{\log N}{\log N - \log p + \sqrt{p} \log p} \geq E_0$$

$$\log N \geq \frac{E}{1 - E_0} (\sqrt{p} \log p - \cancel{\log p})$$

$$\log p (\sqrt{p} - \cancel{1})$$

$$\log N \geq k \cdot \sqrt{p} \log p \quad ; \quad k = \frac{E_0}{1-E_0}$$

$$N \geq 2^{k \cdot \sqrt{p} \cdot \log p} = p^{k\sqrt{p}}$$

$$\boxed{\Psi_1 : N = \Omega\left(p^{O(1)\sqrt{p}}\right)}$$

výpočet Ψ_2: $\sqrt{p} \log p \leq k' \cdot \log N$ $\qquad k' = \frac{1-E_0}{E_0}$

řešíme pro p

1. možnost : úprava levé strany

$$\left.\begin{array}{l} a,\ \log p \Rightarrow 1 \\ b,\ \log p \Rightarrow \sqrt{p} \end{array}\right\} \text{substituce}$$

CPC

2. možnost. (zlogaritmujeme obě strany):

$$\frac{1}{2} \log p + \cancel{\log \log p} \leq \log (k' \log N)$$

$$\underline{\log p} \doteq 2 \log (k' \log N) \text{ --- odhad pro } \log p$$

↳ nemůžeme z toho vypočítat p, protože jsme zanedbali člen

$$\sqrt{p} \leq \frac{k' \cdot \log N}{2 \log (k' \log N)}$$

$$p \leq \frac{k'^2 \cdot \log^2 N}{4 \log^2 (k' \log N)} \text{ --- } = \Psi_2$$

$$E(n,p) = \frac{\log N}{\log N + \sqrt{p} \log p}$$

dosadím Ψ_2

$$k' = \frac{1-E_0}{E_0}$$

——— musí vyjít E_0

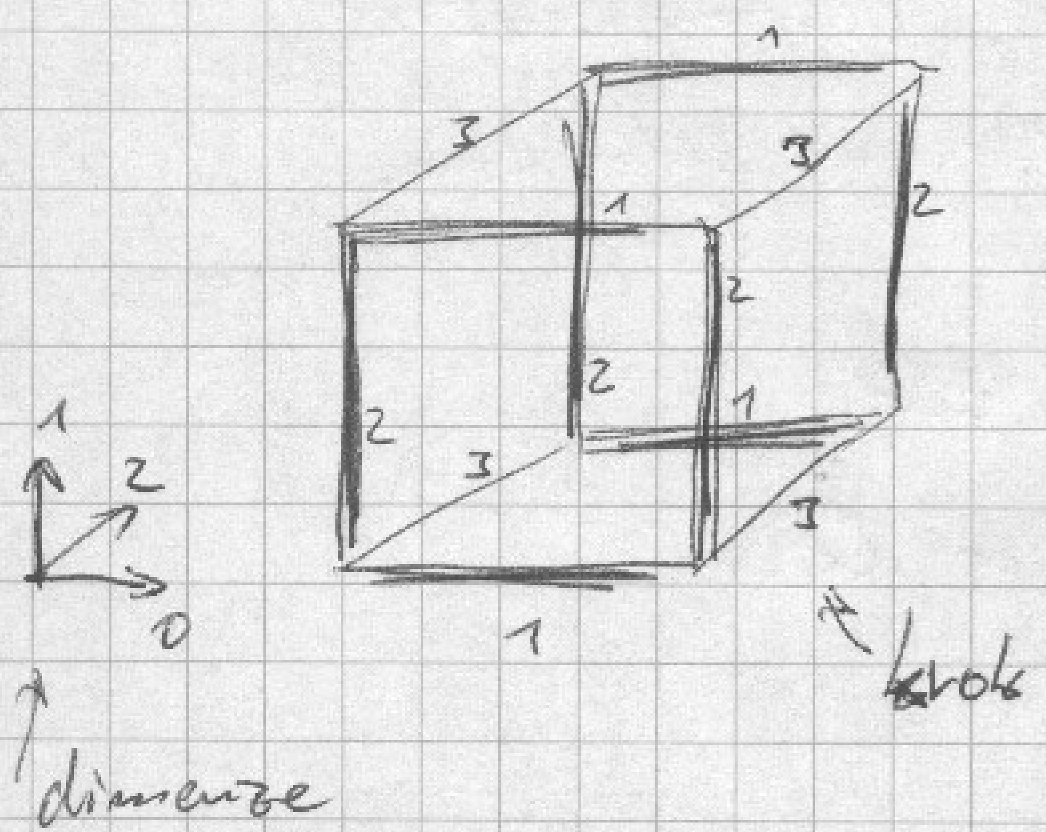
par-cv-5/5

$$T_{min} = T(N,N) = \sqrt{N} \cdot \log N$$

$$\psi_3 = (\text{rozdělím na 2 nerovnice}) = \ldots = \Omega(\sqrt{N})$$

BITONICKÝ MERGE SORT

- přesně funguje na hyperkrychlích



	KROK	DIMENZE
3D krychle	1	0
	2	1, 0
	3	2, 1, 0
	⋮	
n-krychle	n + 1	n, n−1, ..., 0

- použiju novou dimenzi + všechny předtím

$P < N$

$$T(n,p) = \underbrace{\frac{n}{p} \log \frac{n}{p}}_{\text{sekvenční setřídění}} + \frac{n}{p} (1 + \log p) \frac{\log p}{2}$$

$$\sum_{i=1}^{\log p} \sum_{j=1}^{i} 1 = \sum_{i=1}^{\log p} i = \overbrace{(1 + \log p) \frac{\log p}{2}}$$

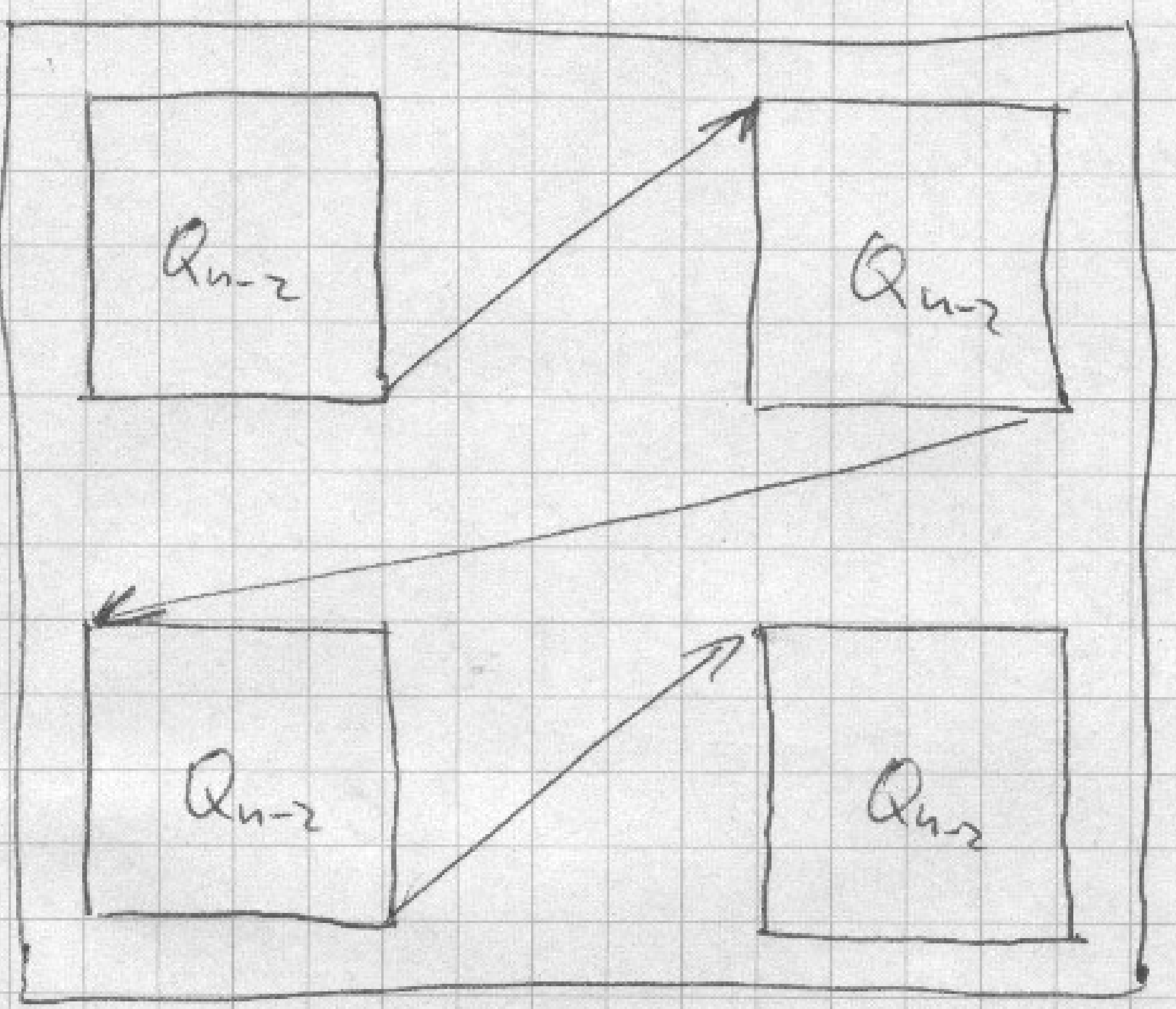
$$T(n,p) = \frac{n}{p} \log \frac{n}{p} + \frac{n}{p} \log^2 p$$

PAR-11-5/6

$$\Psi_1(p) = p^{k \log p}$$

$$\Psi_2(n) = 2^{\sqrt{k! \log n}}$$

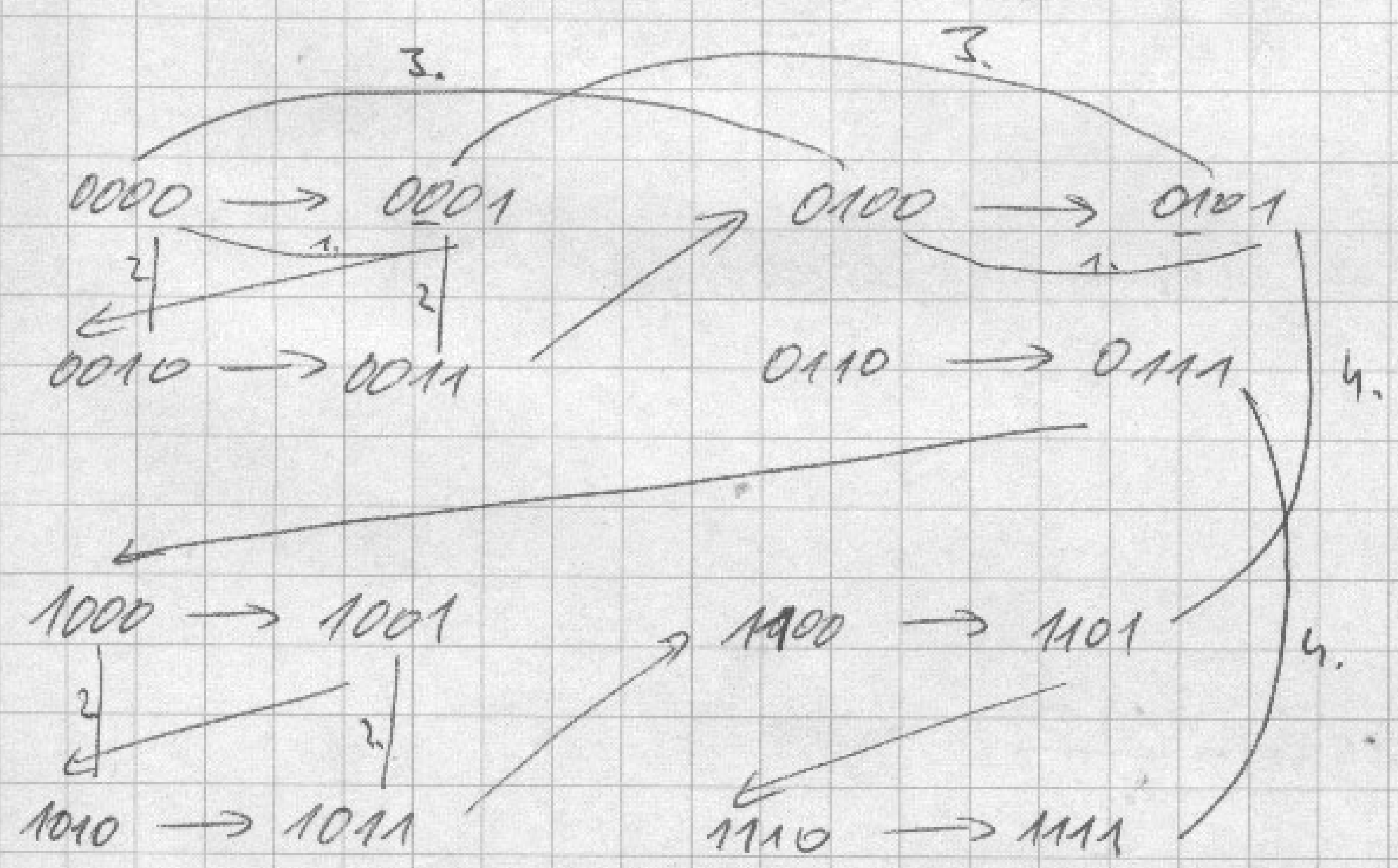
BITONICKÝ MERGE SORT 2-D MESH $M(2^k, 2^k)$; $n=p$



obecné dekompoziční rekurzivní schéma

2D MESH

+ zobrazení Q_n



Vzdálenost i-tého bitu

číslováno od 1 : $2^{\lfloor \frac{i-1}{2} \rfloor}$

– spočítat počet kroků

$$\sum_{i=1}^{\log n} \sum_{j=1}^{i} 2^{\lfloor \frac{i-1}{2} \rfloor}$$

tímto vzorcem dosáhneme řešení, protože $n=p$,

rozdělit na 2 řady

PAR-cv-5/7

pomocný vzorec: $2^0+2^1+\dots+2^k = 2^{k+1}-1$ $\quad 2^{\lceil \frac{\log n}{2} \rceil +1} \approx \sqrt{n}$

výpočet vnitřní Σ

i je liché; $i = 2k+1$

$$\sum_{j=1}^{2k+1} 2^{\lfloor \frac{j-1}{2} \rfloor} = 2^0+2^0+2^1+2^1+\dots+2^{k-1}+2^{k-1}+2^k = 2^{\lceil \frac{i}{2} \rceil} + 2^{\lfloor \frac{i}{2} \rfloor} - 2$$

i je sudé; $i = 2k$ $\qquad k = \frac{i}{2}$

$$\sum_{j=1}^{2k} 2^{\lfloor \frac{j-1}{2} \rfloor} = 2^0+2^0+\dots+2^{k-1}+2^{k-1} = 2(2^0+\dots+2^{k-1}) = 2(2^k-1) =$$

$$= 2(2^{\frac{i}{2}}-1) = 2^{\frac{i}{2}+1}-2$$

dám to dohromady:

$$\sum_{i=1}^{\log n} 2^{\lceil \frac{i}{2} \rceil} + 2^{\lfloor \frac{i}{2} \rfloor} - 2 + 2^{\frac{i}{2}+1} - 2 \Longrightarrow$$ rozepíšu řadu a zkusím to vykoukat

$$2^{\lceil \frac{1}{2} \rceil} + 2^{\lfloor \frac{1}{2} \rfloor} - 2 + 2^{\frac{2}{2}+1} - 2 +$$

$$2^{\lceil \frac{3}{2} \rceil} + 2^{\lfloor \frac{3}{2} \rfloor} - 2 + 2^{\frac{4}{2}-1} - 2 + \dots$$

$$2^{\lceil \frac{\log n}{2} \rceil} + 2^{\lfloor \frac{\log n}{2} \rfloor} - 2 + 2^{\frac{\log n}{2}-1} - 2$$

1. 2. 3. řady ... pomocný vzorec

$\Downarrow$

$$2^{\lceil \frac{\log n}{2} \rceil +1} + 2^{\lfloor \frac{\log n}{2} \rfloor +1} + 2^{\frac{\log N}{2}+2} + O(\log N)$$

– upravím pomocí pomocného vzorce

$$2\sqrt{n} + \sqrt{n} + 4\sqrt{n} \approx \underline{7\sqrt{n}}$$

PAR-CV-6/1

14.12.2006

KOMUNIKACE

- závisí na typu sítě

komunikační topologie – portovost ~~[illegible]~~

– # sousedů

- kombinující / nekombinující

↑ např. 3 různé zprávy v 1 taktu

- duplexní, poloduplexní, jednoduplexní
- technologie směrování — STORE and FORWARD (SF)

WORM HOLE (WH)

$t_{SF} = t_s + d\mu t_m$ ← přenosová rychlost

vzdálenost

↑ start up time (řádově největší)

$t_{WH} = t_s + d t_d + \mu t_m$ ← poslání vlastní zprávy

↑ čas na vytvoření spojení (čas na vytvoření virtuálního kanálu)

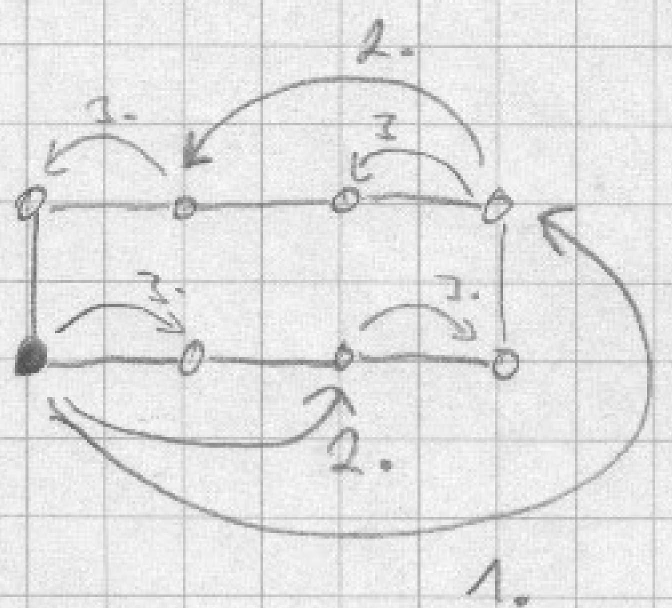
$\mu t_m \gg d t_d$

$\Rightarrow$ WH je NECITLIVÝ NA VZDÁLENOST

Př: toroid – jednoportový, full duplex, WH

$K(Z)$, $Z = 2^k$, algoritmus OAB $\Rightarrow$ SYMETRICKÁ SÍŤ

⇓ je jedno kde začínám



řešení $\log Z$ kroků

$$
\left.\begin{aligned}
T_{OAB} = {} & t_s + \frac{z}{2} t_d + \mu t_m + \\
& t_s + \frac{z}{4} t_d + \mu t_m + \\
& \vdots \\
& t_s + t_d + \mu t_m =
\end{aligned}\right\} \log z
$$

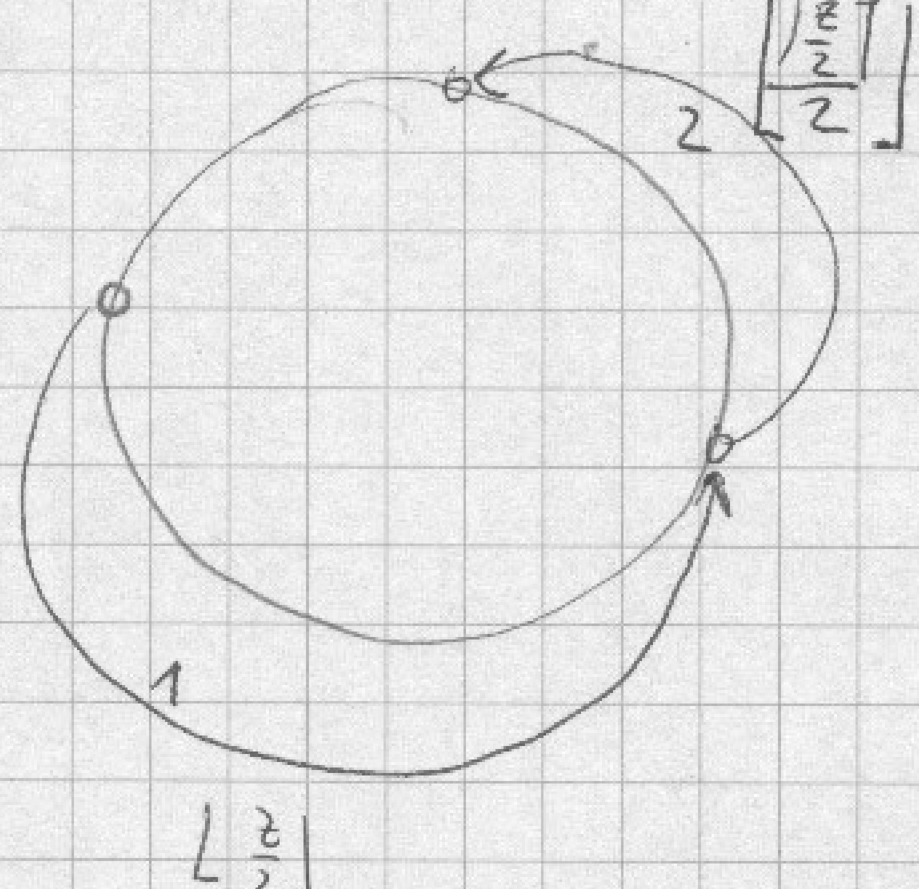
$$= \log z (t_s + \mu t_m) + \sum_{i=1}^{\log z} \frac{z}{2^i} t_d$$

spočítat tuhle řadu → ~~geometrickou~~ geometrická

$$\sum_{i=1}^{\log z} \frac{1}{2^i} = \frac{1}{2} \cdot \frac{1 - \left(\frac{1}{2}\right)^{\log z}}{1 - \frac{1}{2}} = \frac{1}{2} \cdot \frac{\frac{z-1}{z}}{\frac{1}{2}} = \frac{z-1}{z}$$

$$= \log z (t_s + \mu t_m) + z \frac{z-1}{z} t_d = \underline{\log z (t_s + \mu t_m) + (z-1) t_d}$$

Př: toroid $K(z)$, $\underline{\underline{z \neq 2^k}}$, FD, WH, OAB

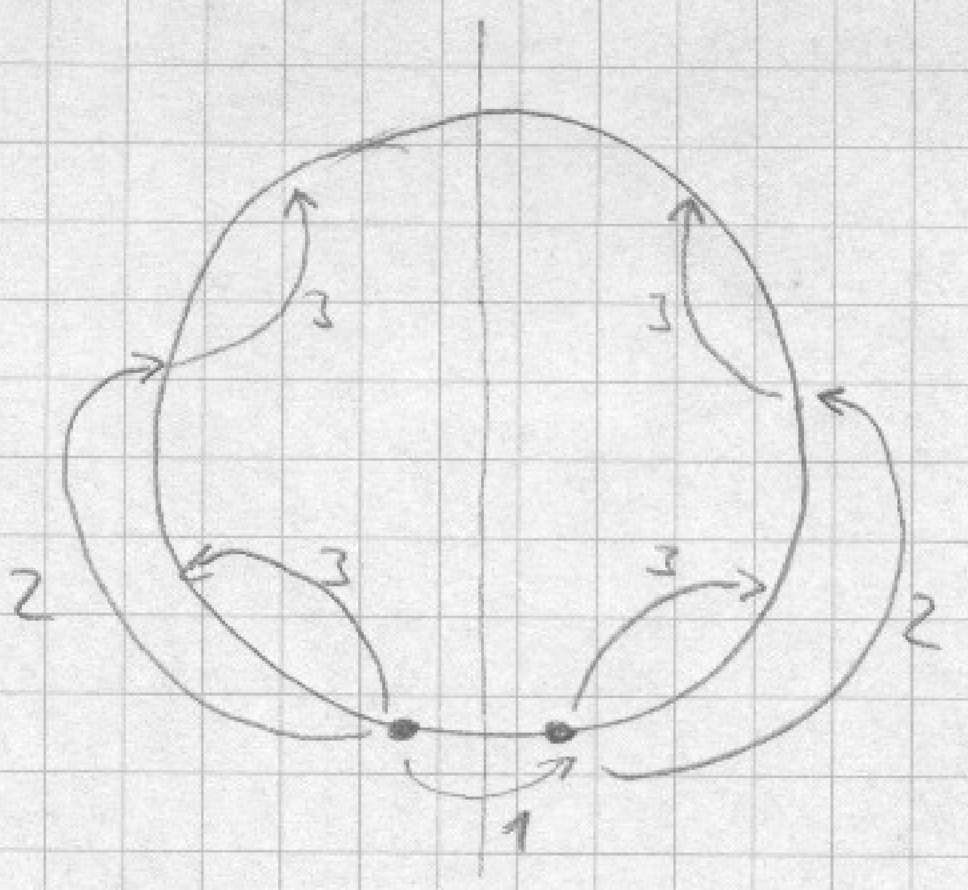


$$\left.\begin{aligned} &\left\lceil \frac{z}{2} \right\rceil \\ &\left\lfloor \frac{z}{2} \right\rfloor \end{aligned}\right\}$$ ~~Když~~ u OAB je jedno kam to pošlu dřív, ale pro asymptotickou složitost OAS nejdřív $\lfloor \frac{z}{2} \rfloor$

$$
\begin{aligned}
T_{OAB} = {} & t_s + \left\lfloor \frac{z}{2} \right\rfloor t_d + \mu t_m + \\
& t_s + \left\lfloor \frac{\lceil \frac{z}{2} \rceil}{2} \right\rfloor t_d + \mu t_m + \\
& t_s + \left\lfloor \frac{\left\lceil \frac{\lceil \frac{z}{2} \rceil}{2} \right\rceil}{2} \right\rfloor t_d + \mu t_m + \dots \\
& \vdots
\end{aligned}
$$

$$= \lceil \log z \rceil (t_s + \mu t_m) + (z-1) t_d$$

Př: Vylepšení prvního příkladu



$$T_{OAB} = \left.\begin{array}{l} t_s + t_d + \mu t_m + \\ t_s + \frac{z}{4} t_d + \mu t_m + \\ \vdots \\ t_s + t_d + \mu t_m = \end{array}\right\} \lceil \log z \rceil$$

$$= \lceil \log z \rceil \cdot (t_s + \mu t_m) + t_d + \sum_{i=2}^{\log z} \frac{z}{2^i} t_d =$$

1. člen

vykrátí se

$$\frac{1}{4} \cdot \frac{1 - \left(\frac{1}{2}\right)^{\log z - 1}}{1 - \frac{1}{2}} = \frac{1}{4} \cdot \frac{1 - \frac{2}{z}}{\frac{1}{2}} = \frac{1}{2} = \frac{z-2}{z} =$$

z se pokrátí $\quad \frac{z-2}{2} = \frac{z}{2} - 1 = \lceil \frac{z}{2} \rceil$

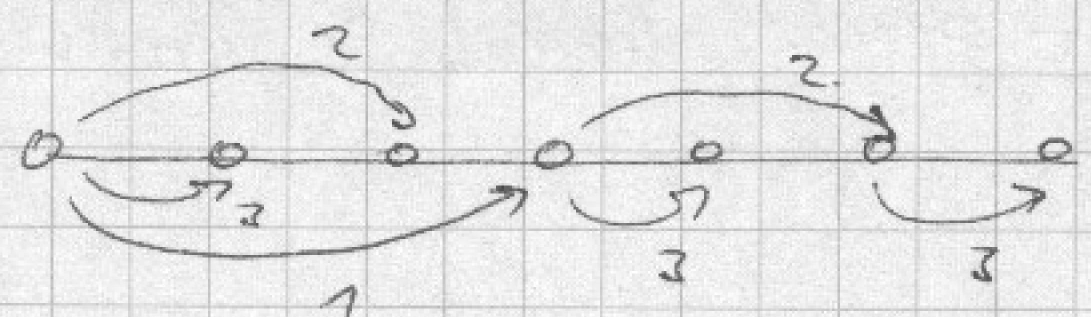
$$= \lceil \log z \rceil (t_s + \mu t_m) + t_d \left(\lceil \frac{z}{2} \rceil - 1 + 1 \right) =$$

$$= \lceil \log z \rceil (t_s + \mu t_m) + \lceil \frac{z}{2} \rceil t_d$$

PAR – W – 6/5

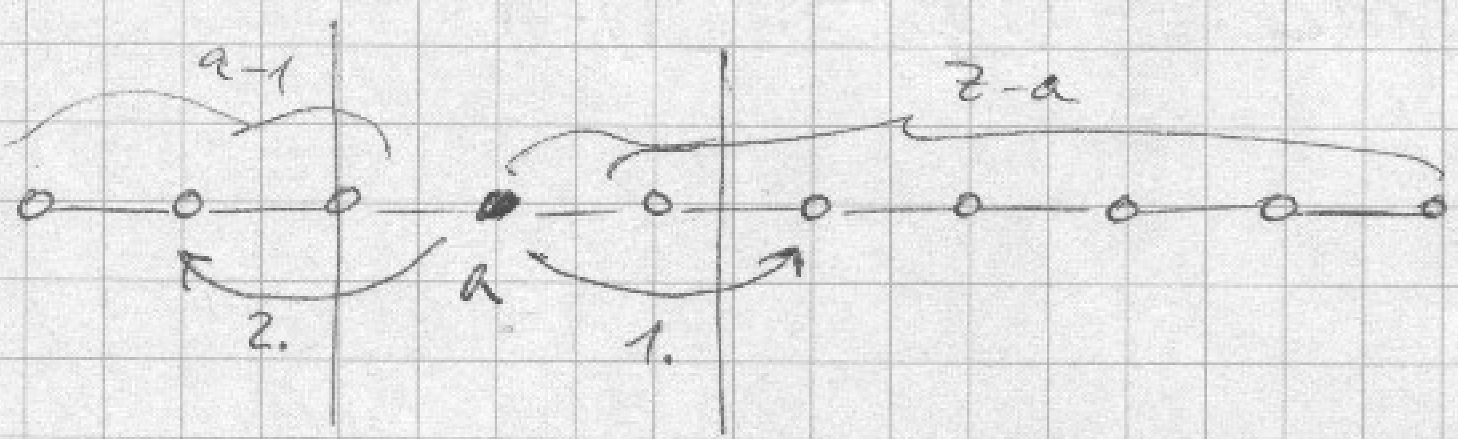
Mřížka (z), $z \neq 2^k$, 1-portový, FD, WH, OAB

$\longrightarrow$ není SYMETRICKÁ



1. případ – stejné jako toroid; vysílám z kraje

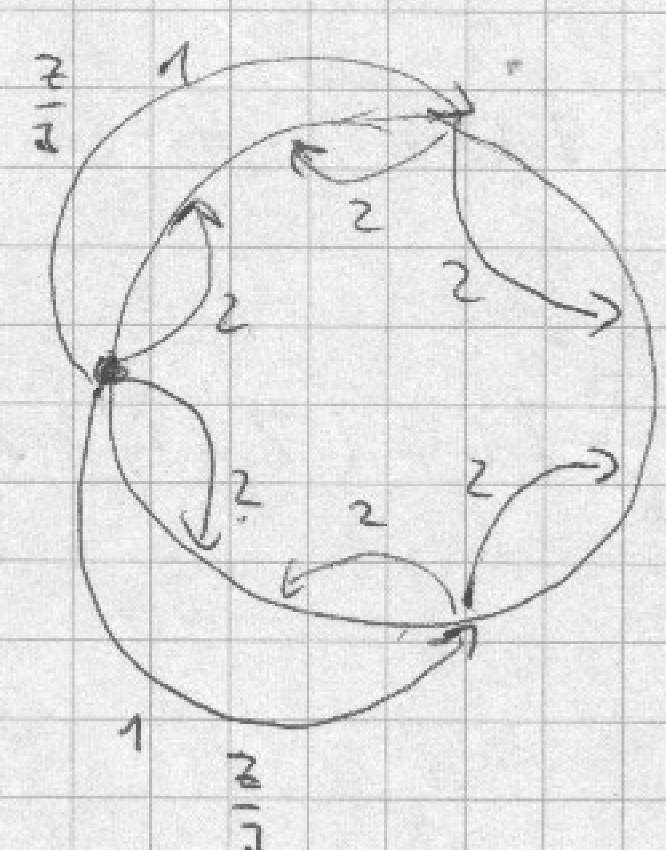
2. případ – převedu na předchozí případ



$$T_{OAB} = \underline{\lceil \log z \rceil (t_s + \mu t_m) + \max(a-1, z-a)\, t_d} \; \ldots$$

výsledek, pouze jsme aplikovali předchozí výsledek

$K(z)$, $z \neq 3^k$, 2-portový, FD, WH



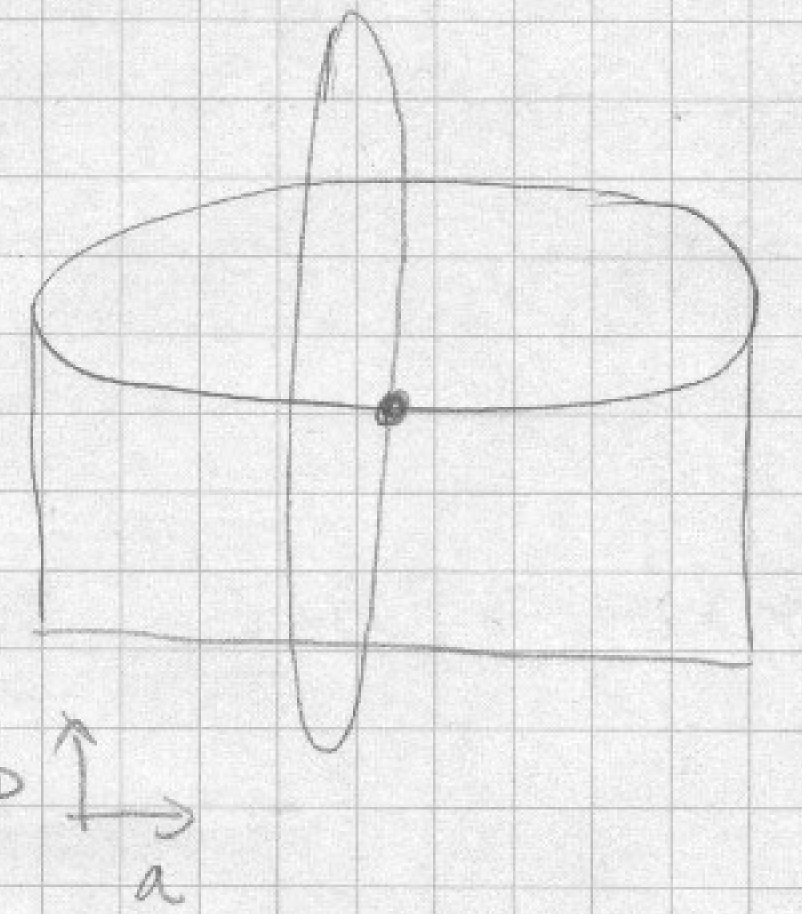
– dělím na $\frac{1}{3}$

$$T_{OAB} = \left.\begin{array}{l} t_s + \frac{z}{3} t_d + \mu t_m + \\ t_s + \frac{z}{3^2} t_d + \mu t_m + \\ \vdots \\ + t_s + t_d + \mu t_m \end{array}\right\} \log_3 z$$

$$= \underline{(\log_3 z)(t_s + \mu t_m) + \frac{z}{2} t_d}$$

PAR-W-6/5

$K(a,b)$, 1-portový, FD, WH, OAB

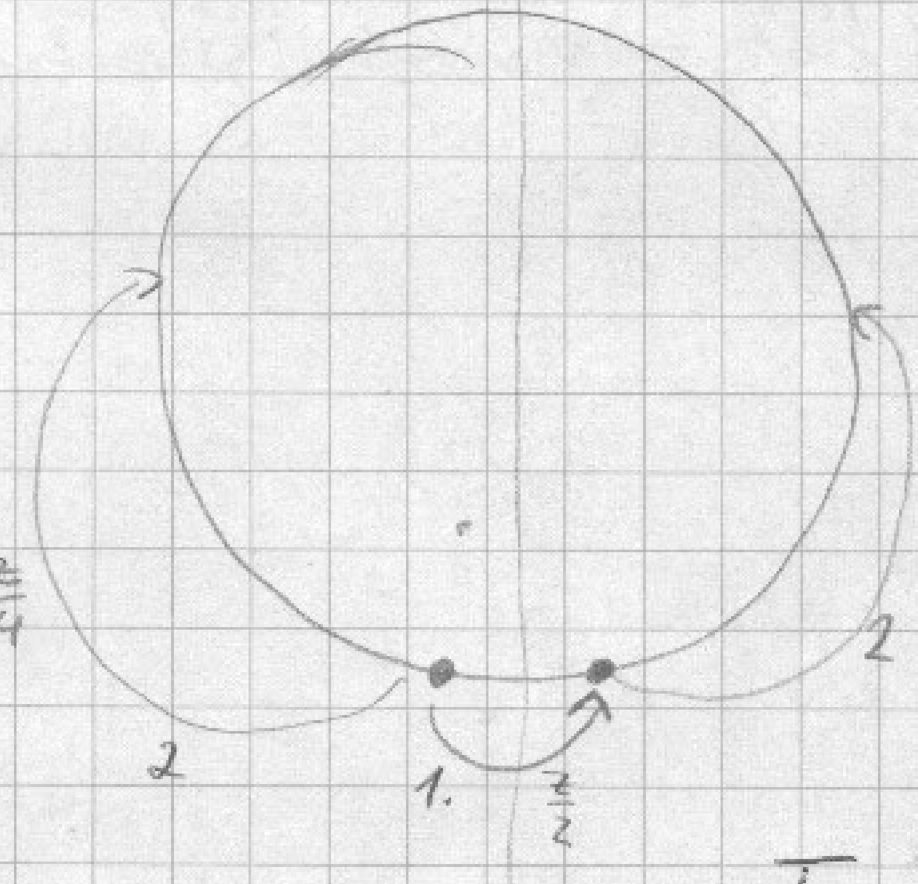


– pouze součet 2 toroidů 1-D

$$T_{OAB} = (\lceil \log a \rceil + \lceil \log b \rceil)(t_s + \mu t_m) + \left(\left\lceil \frac{a}{2} \right\rceil + \left\lceil \frac{b}{2} \right\rceil\right) t_d$$

dimenze a   b

$K(z)$, 1-portový, FD, WH, OAS = každému něco posílám, ale jiného



jen sousedovi

$$T_{OAS} = t_s + 1 t_d + \frac{z}{2} \mu t_m +$$

vzdálenost   počet paketů

$$t_s + \frac{z}{4} t_d + \frac{z}{4} \mu t_m +$$

$$\vdots$$

$$t_s + t_d + 1 \cdot \mu t_m =$$

log z

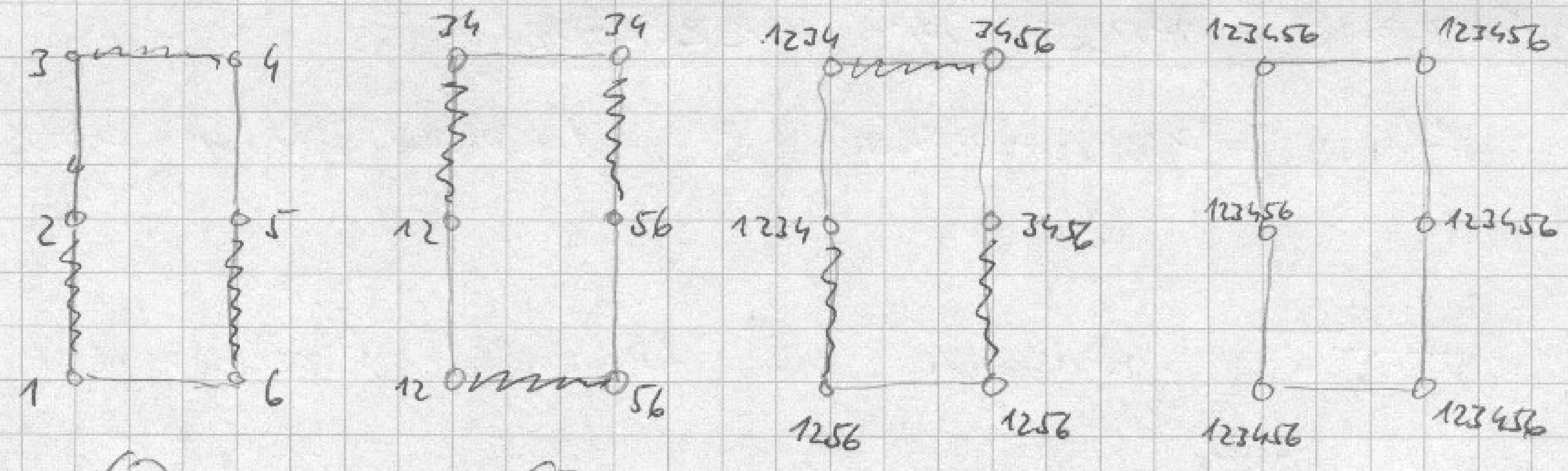
$$T_{OAS} = \lceil \log z \rceil t_s + \frac{z}{2} t_d + (z-1) \mu t_m$$

– pro více-D vnořením

PAR – cv – 6/6

K(z), 1-portový, FD, SF, kombinující, AAB

$z = 6$



(1) velikost zprávy max 1 paket

(2) max 2.

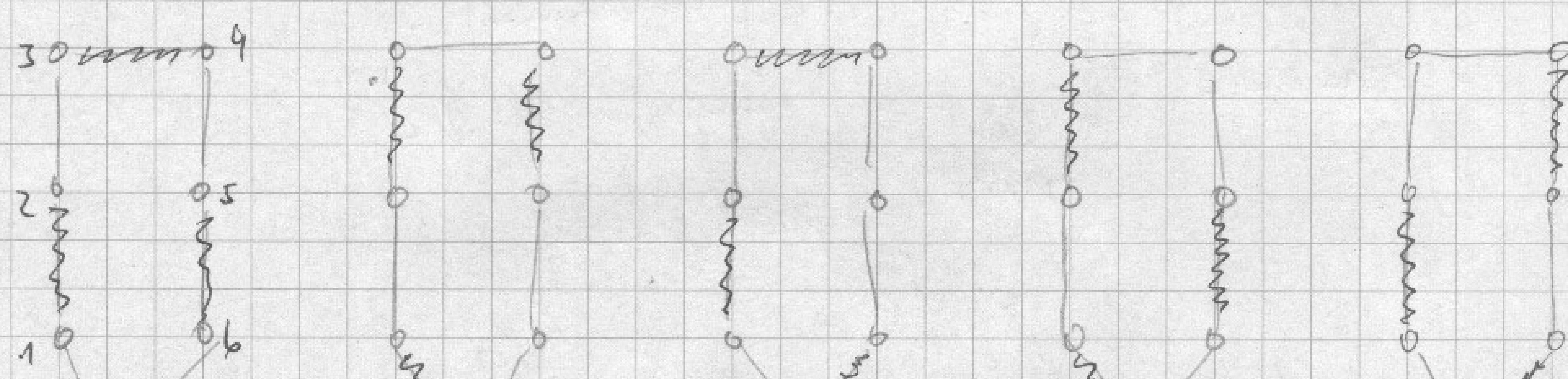
(2) ← jinak by docházelo k duplicitě

$$T_{AAB} = \left.\begin{array}{l} t_s + \mu t_m + \\ t_s + 2\mu t_m \\ \vdots \\ t_s + 2\mu t_m \end{array}\right\} \frac{z}{2}$$

$$T_{AAB} = \frac{z}{2} t_s + \mu t_m + \left(\frac{z}{2} - 1\right) 2 \mu t_m = \underline{\underline{\frac{z}{2} t_s + (z-1)\mu t_m}}$$

platí pouze pro sudé

$z = 7$



max. počet vynásobovaných paketů: (1) (2) (3) (3) (1)

počet kroků: $\lceil \frac{z}{2} \rceil + 1 = \frac{z+3}{2}$

$$T_{AAB} = \frac{z+3}{2} t_s + \mu t_m \left(1 + 2 + 3 \cdot \left(\frac{z+3}{2} - 3\right) + 1\right) =$$

3 atypické kroky (2 na začátku + 1 na konci)

$$= \underline{\frac{z+3}{2} t_s + \mu t_m \frac{3z-1}{2}}$$

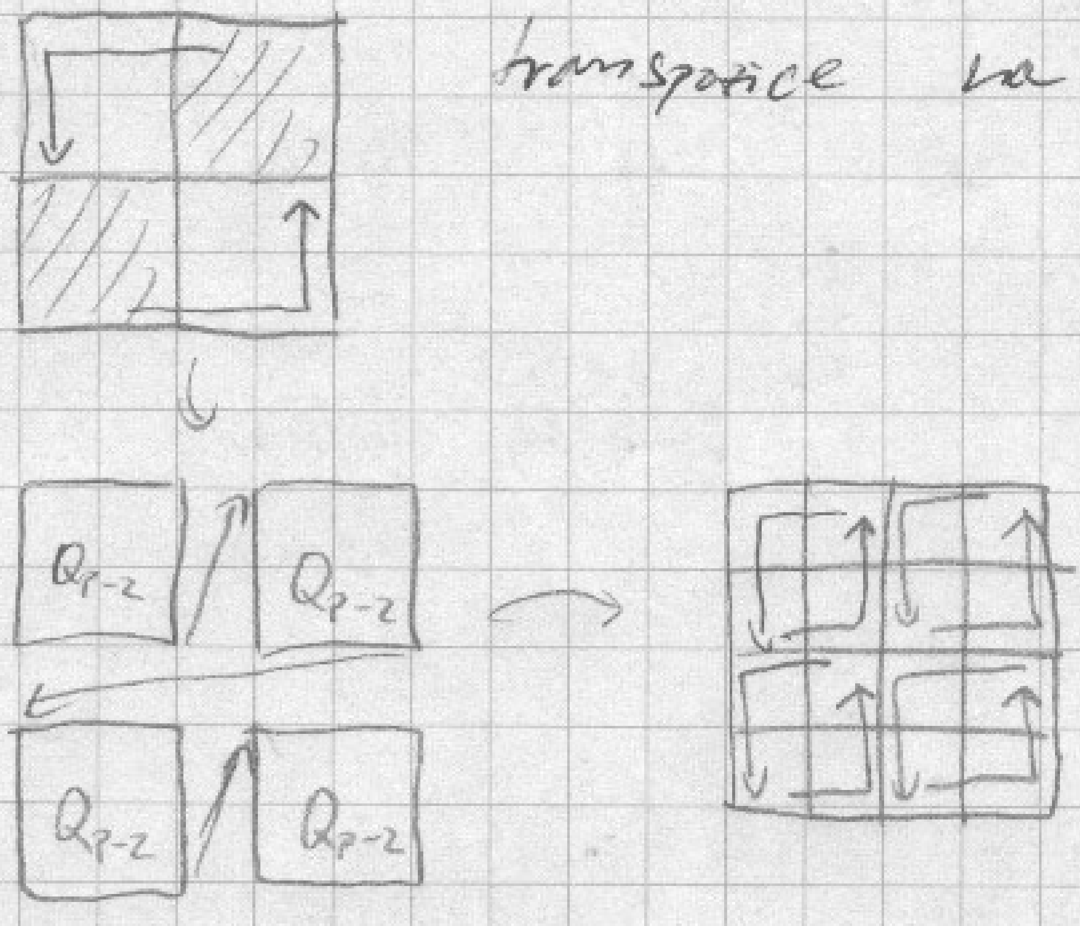
PAR_CV – 7/1

11.1.2006

LINEÁRNÍ ALGEBRA

- TRANSPOZICE MATICE NA HYPERKRYCHLI (SLIDE 6)

$A(n,n)$, POLODUPLEXNÍ, SF, 1-PORTOVÁ, $N = n^2$, $p = 2^q$



transpozice na 2D mřížce (rekurzivně)

- komunikace vždy na vzdálenost (2)

- rekurzivně dělím stejným způsobem až do 1 procesoru

$$T_{TR}(N,p) = \left(t_s + 2\frac{N}{p}t_m\right)\frac{q}{2} + O\left(\frac{N}{p}\right)$$

– lokální transpozice

konst. zanedbáme

komunikace

počet dimenzí

vzdálenost

velikost zprávy

zmenšuju po 2^2

počet kroků

$SU(N) = O(N)$

zanedbáme t_m, t_s vůči $\frac{N}{p}$ (matice jsou velké)

$C(N,p) = p \cdot T(N,p) = 2N\frac{q}{2} + N$

$$E(N,p) = \frac{SU(N)}{C(N,p)} = \frac{N}{2N\frac{q}{2} + N} \geq E_0$$

$$\frac{1}{q+1} \geq E_0$$

$\log p$

$$\frac{1}{\log p + 1} \geq E_0 \Rightarrow \frac{1 - E_0}{E_0} \geq \log p$$

$p = O(1)$

PAR-CV-7/2

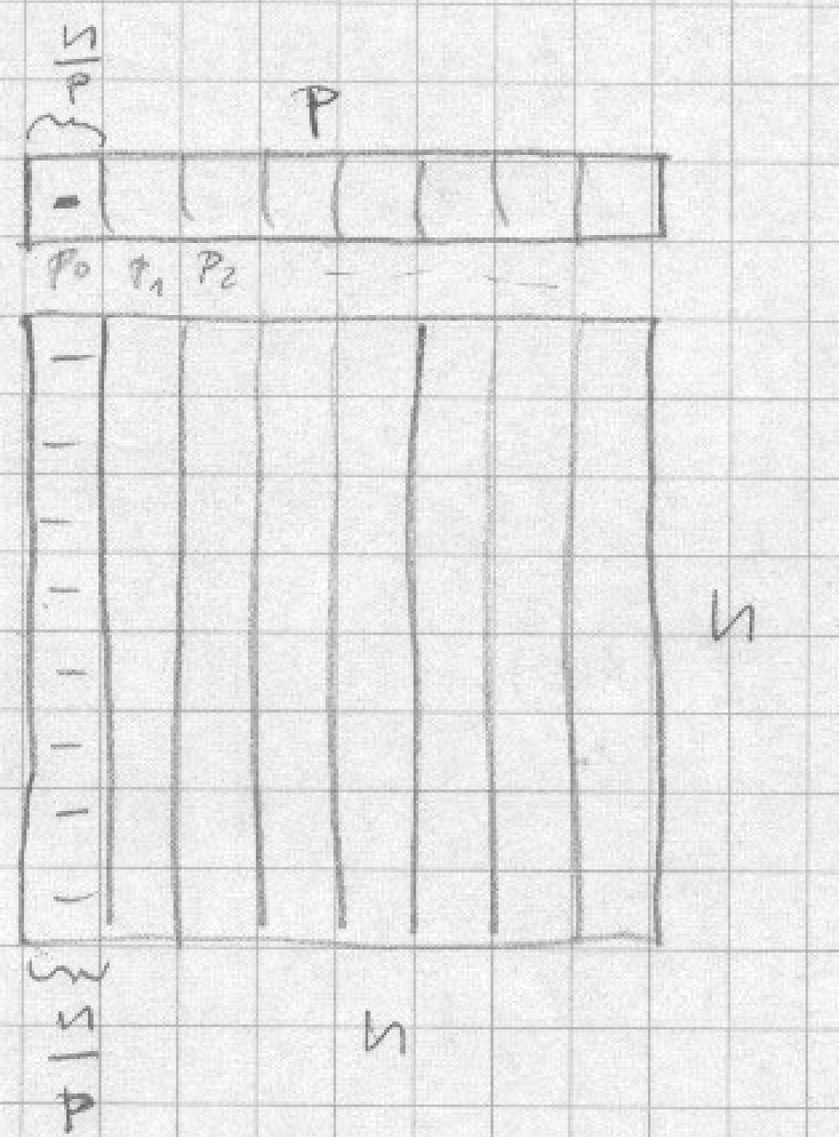
⇒ algoritmus je efektivní již od konstantního počtu řádku

NÁSOBENÍ MATICE VEKTOREM (SLIDE 9)

lineární pole procesorů ⇒ 1D mřížka

SF, FD, 2-port, $M(p)$, mapování po sloupcích

$A(n,m)$, $N = n^2$, p, $p \neq n$, $X(n,1)$



← procesory $n \neq p$

← matice

2 fáze algoritmu:

F1: sekvenční skalární součin

$\frac{n}{p} \wedge \frac{n}{p}$

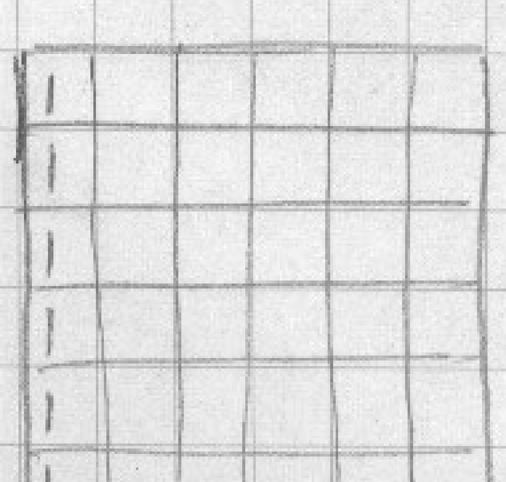
vektor matice

$$T_1 = n \underbrace{\left(\frac{n}{p} t_{nás} + \frac{n}{p} t_{souč}\right)}$$

počet řádků matice

1 skalární součin

F2: p paralelních redukcí

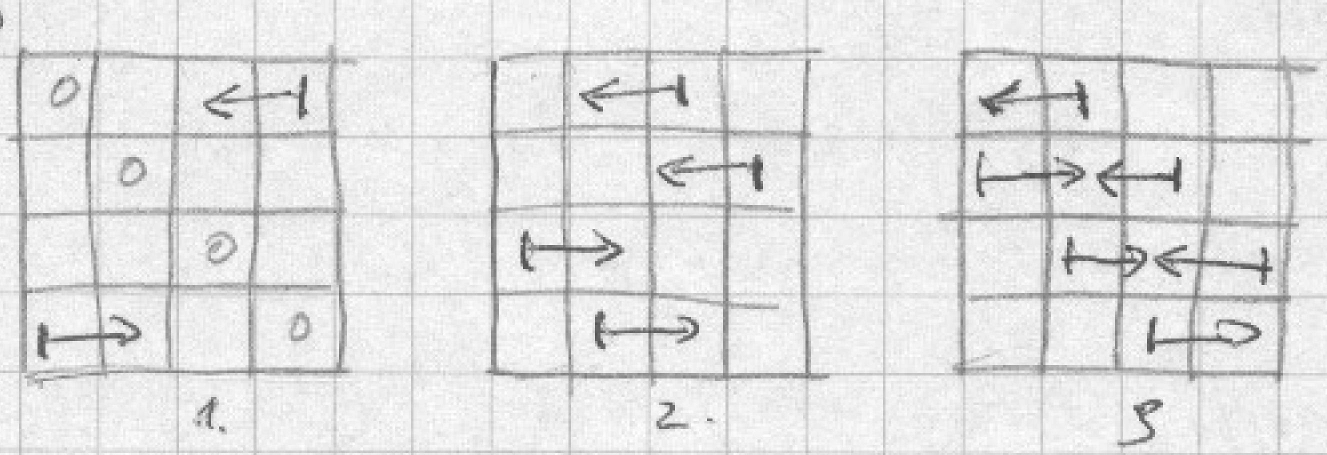


1 redukce p vektorů o velikosti $\frac{n}{p}$, celkově se posílá n dat.

PAR – CV – 7/3)

– protože komunikace je SF, použiju PIPELINE paralelní redukci (pro WH bych použil binární || redukci)

tady chci mít výsledek



– musí být nekonfliktní, tj. po promítnutí řádků matice do p procesorů se nesmí šipky překrývat

n dat na vzdálenost $p-1 = O(p)$ $\quad T_2 = O(np)$

výsledek: $T = T_1 + T_2 = \frac{n^2}{p} + np = \frac{N}{p} + \sqrt{N}\cdot p$

skalární součet sekvenčně

paralelní redukce

$C(N,p) = N + \sqrt{N}\cdot p^2$ $\qquad SU(n) = n^2 = N$

$E(N,p) = \frac{N}{N + p^2\sqrt{N}} \geq E_0$

$N \geq E_0(N + p^2\sqrt{N})$

$\sqrt{N} \geq \frac{E_0}{1-E_0} p^2$

$N \geq \left(\frac{E_0}{1-E_0}\right)^2 p^4$

PAR - ω - 7/4

SLIDE 70÷28: ŘEŠENÍ SOUSTAV LINEÁRNÍCH ROVNIC

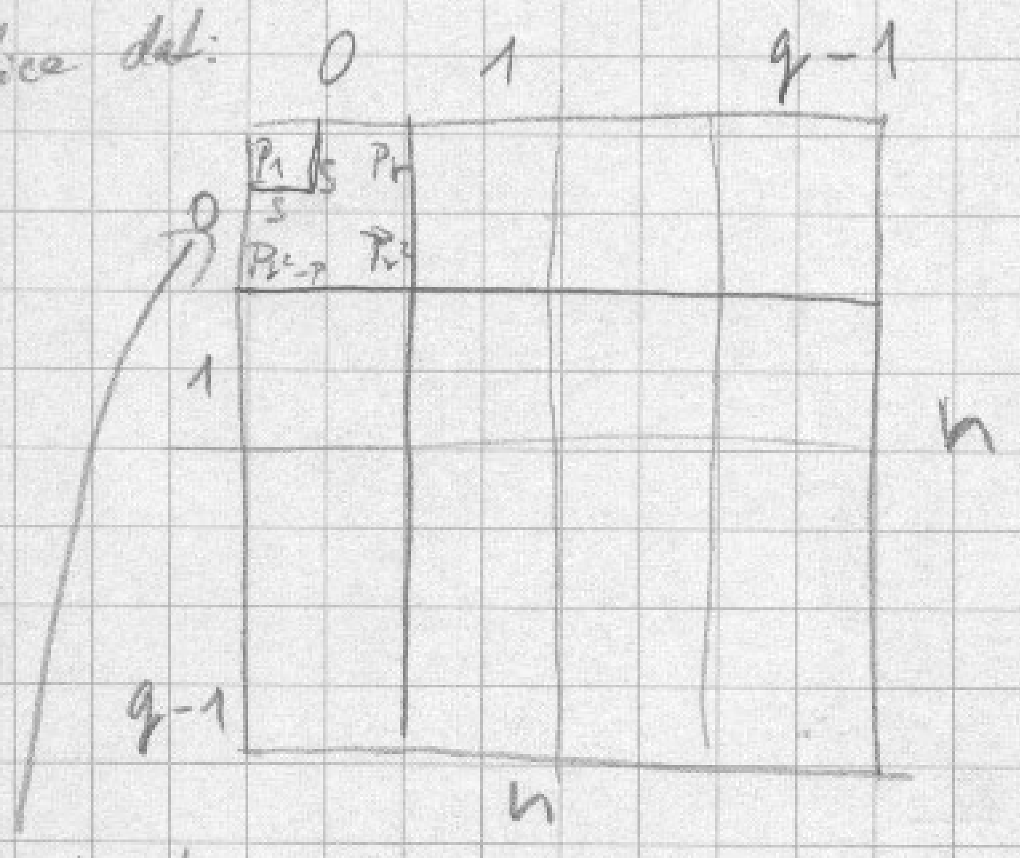
VNH, 2D - MŘÍŽKA, $(\sqrt{p}, \sqrt{p})$,

def: $p = r^2$

$n = rsq$

NÁS. MATIC $SU(n) = \alpha n^3$

matice dat:



v podmatici r^2 procesorů

- každý procesor p dostane q^2 submatic, každá má s^2 prvků

4 fáze: 1 - inverze
2 - broadcast po sloupcích
3 - —"— po řádcích
4 - součin matic

→ zpracovává data po diagonále → data ubývají

F1: počítání inverzní matice $\quad T_{inv}(n,n) = \beta n^3$

MATICE $s \times s \Rightarrow \beta s^3$

počet matic : rq

$\underline{T_1 = \beta r q s^3}$

F2, F3: BROADCAST

1. KROK - posílám vše ⇒ velikost zprávy $q^2 s^2 = \frac{N}{p}$

$T_{OAB}(N(z), m) = \log z (t_s + m t_m) + \cancel{\frac{z}{z} t_d}$ --- zadáno

r procesorů Broadcast

1 OAB $\quad T_{OAB} = \log r (t_s + \frac{N}{p} t_m)$ --- broadcast pro 1 procesor

r OAB $\quad T = r \log r (\frac{N}{p} t_m)$ ---- pro r procesorů

PAR - cv - 7/5

2. KROK $(q-1)^2 s^2$

r OAB $T = r \log r \, (t_s + (q-1)^2 s^2 t_m)$

q-tý krok $1 s^2$

r OAB $T = r \cdot \log r \, (t_s + s^2 t_m)$

posčítat všechny kroky:

$$T_{OAB} = r \cdot (\log r) \, t_m \cdot s^2 \cdot \underbrace{(q^2 + (q-1)^2 + \dots + 1)}_{q} = \frac{q^3}{3}$$

$$T_{OAB} = \frac{r \cdot (\log r) \cdot t_m \cdot s^2 \cdot q^3}{3}$$

F4: ~~ÚPRAVA~~ NÁSOBENÍ MATIC

$$T_4 = \underbrace{\alpha q^2 s^3 + \alpha q^2 s^3 + \dots}_{r} + \underbrace{\alpha (q-1)^2 s^3 + \alpha (q-1)^2 s^3 + \dots}_{r} + \underbrace{\alpha s^3 + \dots \alpha s^3}_{r} =$$

$$= \alpha r s^3 \underbrace{(q^2 + (q-1)^2 + \dots + 1)}_{\frac{q^3}{3}} = \alpha \frac{r s^3 q^3}{3} = \frac{\alpha}{3} \frac{n^3}{r^2} = \frac{\alpha}{3} \frac{\sqrt{N^3}}{p} = T_{NAS}$$

SEČTEM JEDNOTLIVÉ FÁZE:

$\sqrt{p} = r$

$$T = T_1 + T_2 + T_3 + T_4 = 2 T_{INV} + 2 \cdot T_{OAB} + T_{NAS} = \cancel{2 \beta \sqrt{p} q s^3} +$$

vádlová ucení? hod?

$$+ \frac{2}{3} \sqrt{p} \log \sqrt{p} \, s^2 q^3 t_m + \frac{\alpha}{3} r s^3 q^3 \doteq$$

$$\doteq \underbrace{\sqrt{p} \log \sqrt{p} \, s^2 q^3 t_m + r s^3 q^3}_{\text{výsledek}} = \underbrace{\sqrt{p} \, s^2 q^3 (\log \sqrt{p} + s)}_{\text{ustálení}}$$

$C(N, p) = O(r^3 s^2 q^3 (\log r + s))$ $\quad SU(n) = \underbrace{r^3 s^3 q^3}$

$n^3 \Leftrightarrow n = r s q$

PMR-CV-2/6

$$E(M,P) = \frac{r^2 s^3 q^3}{r^3 s^2 q^3 (\log r + s)} = \frac{s}{s + \log r} \doteq \frac{s}{s + \log P}$$