① Simulace velkého PRAM na malém téhož typu **36PAR**

- p procesorů, t kroků

p' procesorů: $t' = O(tp/p')$ kroků; $p' < p$

každý procesor simuluje $\frac{p}{p'}$ procesorů: 1, všechny READ a LOC; 2, všechny WRITE

- $m' < m$, $p' = \max(p, m')$, $O(tm/m')$ kroků

každý procesor má na starosti m/m' buněk, přes $M'[i]$ se přistupuje ke všem simulovaným buňkám

Simulace silnějšího PRAM na slabším

- 1 krok v $O(\log p)$ krocích s $m.p$ buňkami paměti
  - pomocné buňky jsou vnitřní uzly úplného bin. stromu

- 1 krok v $O(\log p)$ krocích s $m+p$ buňkami paměti
  - pomocné pole A, P_k chce přistoupit k $M[i] \Rightarrow A[k] := (i, k)$
  - setřízení A ($O(\log p)$ kroků), nastavení příznaku zda je první procesor žádající o buňku v poli

write: P'_k přečte (i, j, s) z $A[k]$ a zapíše do $A[j]$, P'_k přečte (i, k, s) a podle příznaku s provede write nebo ...

read: P'_k přečte (i, j, s) z $A[k]$, pokud $s=1$, tak $A[k] := (i, j, M[i])$; rozkopírování $(i, *, j) \to (i, *, M[i])$

Simulace EREW PRAM na APRAM

- základní třífázový cyklus lze simulovat v $O(b(p))$, simulace bude trvat $\Theta(b(p)t)$,
- $C = p.t \Rightarrow C' = \Theta(p\, b(p)\, t)$
- $p_A = p/b(p)$ čas je $O(b(p)t)$

HYPERKRYCHLE Q_n

$|V| = 2^n$; $|E| = n.2^{n-1} = \frac{n}{2} \cdot 2^n$; $\Phi = n$, $deg = n$, $bw_e = 2^{n-1}$, $n!$ permutací dimenzí, 2^n posunutí ($x := x \oplus u \oplus v$),

$Q_n = Q_{n-1} \times Q_{n-1}$, uzly u a v, $\varrho(u,v) = k$, k cest délky k, $n-k$ cest délky $k+2$, pointa $\Rightarrow$ bipartita

vnoření cest a kružnic: grayovy kódy ($b = b_{n-1} \dots b_0$, $G_n(b) = g_{n-1} \dots g_0$, $g_{n-1} = b_{n-1}$, $g_i = b_{i+1} \oplus b_i$)

vnoření stromů: CBT_n není podgrafem Q_{n+1}, protože CBT_n není a Q_{n+1} je vyvážený bipartitní graf

$\Rightarrow$ trik: zdvojení kořene: load = ecng = 1, dil = 2

vnoření mřížky stromů: $MT_n \subset CBT_n \times CBT_n$, $CBT_n \to Q_{n+1}$ (dil = 2) $\Rightarrow Q_{2n+2}$

vnoření mřížek a toroidů: $M(z)$ je podgrafem $Q_{\lceil \log z \rceil}$ (použijem grayův kód), $M(2^{k_1}, \dots 2^{k_n})$ je faktorem $Q_{k_1 + \dots + k_n}$, alg. vyjádření: $M \to Q$: převod jednotlivých souřadnic do Gray. kódu + zřetězení

vnoření CCC_n: $CCC_n \subset Q_n \times T(n) \subset Q_n \times Q_{\lceil \log n \rceil} = Q_{n + \lceil \log n \rceil}$

② **MŘÍŽKA $M(z_1, z_2, \dots z_n)$**

$|V| = \prod_{i=1}^{n} z_i$, $|E| = \sum_{i=1}^{n} (z_i - 1) \prod_{j \neq i} z_j$, $\phi = \sum_{i=1}^{n} (z_i - 1)$, $bw_e = \frac{\prod_{i=1}^{n} z_i}{\max_i z_i}$ (z_i sudé), $\max_i z_i$ liché $\Rightarrow + \min_i z_i$ $(+1)$

$M(z_1, z_2, \dots, z_n) = M(z_1) \times M(z_2) \times \dots \times M(z_n)$, počet disjunktních cest mezi 2 uzly je roven minimu ze stupňů koncových uzlů, ~~alespoň~~ bipartitní

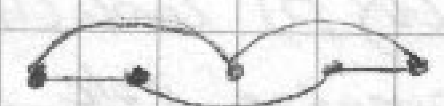
TOROID $T(z_1, z_2, \dots, z_n)$

$|V| = |M|$, $|E| = n \cdot \prod_{i=1}^{n} z_i$, $\phi = \sum_{i=1}^{n} \lfloor \frac{z_i}{2} \rfloor$, $deg = 2n$, $bw_e = 2 \cdot bw_e(M)$, prohození $x_1 \oplus_{z_1} u_1 \oplus_{z_2} v_1$,

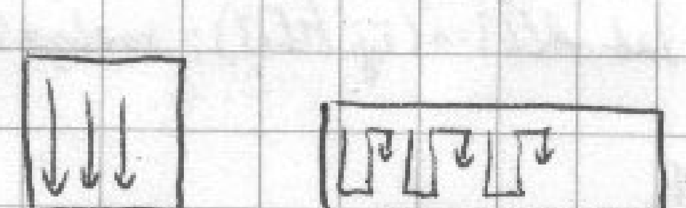
$T(z_1, z_2, \dots, z_n) = T(z_1) \times T(z_2) \times \dots \times T(z_n)$, bipartitní $\Leftrightarrow$ délky všech kružnic sudé

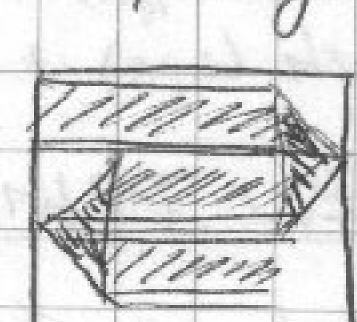
vnoření lin. polí a kružnic do M a T: Peanova křivka ($dil = \sqrt{N}$)  relaxování,
T má Ham. kružnici, M má H. cestu

vnoření mřížky do toroidu: M je podgrafem T

vnoření toroidu do mřížky: $load = 1$, $dil = ecng = 2$ 

vnoření □ M do ▭ M: $M(\sqrt{z_1 z_2}; \sqrt{z_1 z_2})$, $dil = \lceil z_1 / z_2 \rceil$, $load = 2$, $ecng = 1 + \lceil z_1 / z_2 \rceil$





vnoření ▭ M do □ M: $dil = 1$, $load = ecng = 2$

KRUŽNICE PROPOJENÉ KRYCHLÍ DIMENZE n CCC_n

$|V| = n \cdot 2^n$, $|E| = \underbrace{n \cdot 2^{n-1}}_{\text{krychle}} + \underbrace{n \cdot 2^n}_{\text{kružnice}}$, $\phi = (2n - 2) + \lfloor \frac{n}{2} \rfloor$, $bw_e = 2^{n-1}$, n sudé $\Rightarrow$ vyvážený bipartitní

wBF_n ZABALENÝ MOTÝLEK

$|V| = n \cdot 2^n$, $|E| = n \cdot 2^{n+1}$, $\phi = n + \lfloor \frac{n}{2} \rfloor$, $bw_e = 2^n$, $deg = \{4\}$

oBF_n OBYČEJNÝ MOTÝLEK

$|V| = (n+1) 2^n$, $|E| = n \cdot 2^{n+1}$, $\phi = 2n$, $bw_e = 2^n$, $deg = \{2, 4\}$

2D MŘÍŽKA STROMŮ VÝŠKY n MT_n

$MT_n \subset CBT_n \times CBT_n$

vnoření do hyperkubických sítí (oba typy motýlků jsou výpočetně ekvivalentní)

① CCC_n je faktorem wBF_n $\varphi((i, x)) = (i \oplus_n parity(x), x)$

② $wBF_n \rightarrow CCC_n$ $dil = ecng = 2$ 2×2 motýlek $\rightarrow$ 3 hranová cesta v CCC

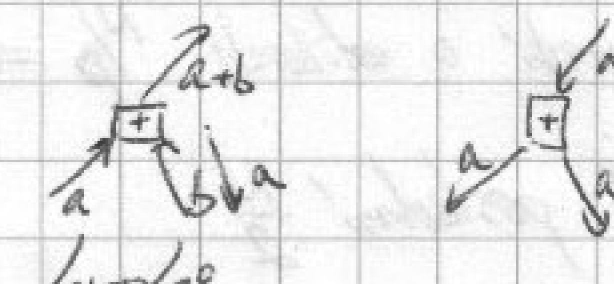
③ $oBF_n \rightarrow wBF_n$ $load = 2$, $dil = 1$, sloučím koncové body $\Rightarrow$ dostanu kružnici
$\Rightarrow$ konstantní zpomalení

③ vnoření lineárního pole a kružnice do obecné sítě

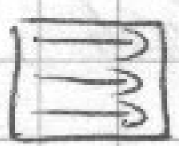
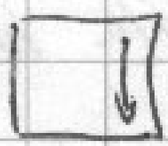
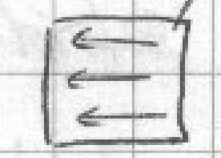
N-vrcholovou kružnici C lze vložit do jakékoliv N-uzlové sítě G s $load=1$; $dil \leq 3$; $ecng=2$

1, udělat kostru T_G; 2, uzly rozdělit na sudé (V_0) a liché (V_1) úrovně; 3, procházení zleva doprava a umisťování C do T_G pokud: a, x je ve V_1 a je to první návštěva x
b, $x \in V_0$ a je to poslední návštěva x

spodní mez dilatace: při vnoření G do H = $dil = \frac{\phi H}{\phi G}$

PPS na stromu: nepřímý  přímý – sčítání i umístění uzly
$O(\log n)$ kroků

PPS na Q_n: jako AAB na SF, žlutý jen co nás zajímá, zelený všechno, N kroků

PPS na MaT:  PPS v řádkách   doinformuju zbytek

PPS na EREW PRAM: $\lceil \log n \rceil$ kroků; $y_i := y_i + M[i-2^j]$, $M[i] := y_i$

zhušťovací problém: procesory s paketem $f_i=1$; ostatní $f_i=0$, PPS přes pole f

paralelní sčítačka s predikcí přenosu: $YX \rightarrow B \xrightarrow{PPS} B' \rightarrow C$ (B' shift bit)

⊙	s	p	g
s	s	s	s
p	s	p	g
g	g	g	g

segmentový PPS:

⊕	b	|b
a	a⊕b	|b
|a	|(a⊕b)	|b

Quicksort: dělení do segmentů podle pivota, $T(n,p) = O(\log n) \cdot T_{PPS}(n,p)$

výpočet předchůdců z následníků: Pred[i] := i; if Succ[i] ≠ i then Pred[Succ[i]] := i

výpočet pořadí pomocí přeskoku ukazatelů: $\lceil \log n \rceil$ kroků: Rank[i] += Rank[Succ[i]];
Succ[i] := Succ[Succ[i]]; počáteční inst. Rank: if Succ[i] = i then Rank[i] := 0 else Rank[i] := 1

Eulerovské cesty: souvislý graf je eulerovský pokud je sudý stupeň každého uzlu,
strom: hrana → 2 hrany (hrany-dvojčata), seznam uzlů AL, seznam hran EL,
seznam následníků euler. kružnice ET[e] := (EL[e].Sib↑).Next
pole čísel hran eul. cesty EA (výpočet z Ranku)

rodičové: Dir[xy] := Rank[xy] < Rank[yx] ? 'F' : 'R'; Rank = pořadí hran v Euler cestě

velikost podstromů: 1, počet uzlů v podstromu je roven polovině počtu hran podstromu
(Rank[ij] - Rank[ji] + 1)/2 kde j = ~~Par~~ Parent[i]
2, počet uzlů podstromu je roven počtu jeho dopředných hran
Weight[e] := Dir[e] = 'F'; PPS přes Weight; Weight[xy] - Weight[yx]

(4) Úroveň uzlů: 1, if $Dir[e] = F$ then $Weight[e] := 1$ else $Weight[e] := -1$, 2, PPS přes Weight

číslování preorder: kořen, levý, pravý, číslo = počet dopředných hran

NEPŘÍMÉ TŘÍDICÍ SÍTĚ: $N/2$ komparátorů ve sloupci, datově necitlivé

Přímé třídicí sítě: CE a MS operace mezi páry uzlů, spodní mez na par. čas je $\varnothing$ sítě

naivní PRAM třízení: 1, P_1 předá ostatním N, 2, každý proc. setřídí svých N/p čísel

3, $P/2$ proc. sloučí $p/2$ dvojic posloupností o velikosti $N/p \Rightarrow P/2$ setříděných posloupností

o délce $2N/p$, 1 proc. sloučí 2 poslední $\frac{N}{2}$

sudo-liché transpozice na lin. poli: paralelní bubble sort, N kroků, operace CE

ShearSort na 2D mřížce: cílem je setřídit čísla hadovitě po řádcích

$\lceil \log n \rceil + 1$ třídění řádků, $\lceil \log n \rceil$ třídění sloupců

$$T(N,N) = \underbrace{\Theta n}_{EOTSort}(\underbrace{\log n + 1}_{řádky}) + \underbrace{n}_{EOTSort} \cdot \underbrace{\log n}_{sloupce} = \Theta(\sqrt{N} \log N)$$

Třídění na 3D mřížce $M(n,n,n)$: 1, třízení xz rovin, 2) třízení yz rovin

3, třídění xy rovin střídavě; 4, licho-sudá a sudo-lichá transpozice ve

všech sloupcích paralelně; 5, třídění xy rovin

Sudo-lichý MergeSort $O(\log^2 N)$ CE kroků

$$EOMS(a_0, \ldots a_{2N-1}) = EOMerge(EOMS(a_0, \ldots, a_{N-1}), EOMS(a_N, \ldots, a_{2N-1}))$$ $\log n$ kroků

$$EOMerge(A,B) = PárovanéCE(Promíchání(EOMerge(sudý(A), lichý(B)), EOMerge(lichý(A), sudý(B))))$$

Bitonický MergeSort $N = 2^n$ čísel na $Q_n \Rightarrow O(\log^2 N)$ CE kroků

$$BSort^+(a_0, \ldots a_{2N-1}) = BMerge(BSort^+(a_0, \ldots, a_{N-1}) BSort^-(a_N, \ldots a_{2N-1}))$$

$$BMerge(A) = BMerge(A_L) BMerge(A_H), \text{ kde } (A_L A_H) = BR(A)$$

na 2D mřížce Peanovo indexování, $S = \sum_{i=1}^{\log N} \sum_{j=1}^{i} 2^{\lfloor \frac{j-1}{2} \rfloor} \approx 7\sqrt{N}$

celková délka komunikačních cest na jeden procesor

⑤ SMĚROVÁNÍ

$t_{SF} = t_s + d \mu t_m$ $\qquad$ $t_{WH} = t_s + d \cdot t_d + \mu t_m$

ZABLOKOVÁNÍ V NEPRAVIDELNÝCH TOPOLOGIÍCH – algoritmus NAHORU/DOLŮ

sestrojení 1, kostra grafu ⇒ strom ⇒ strom má hladiny

2, pro každou hranu určím orientaci (NAHORU/DOLŮ)

směrování: nejdřív používám hrany co jdou nahoru a pak ty co jdou dolů

všeport SF $M_k(n_1, \dots, n_k)$ – permutace v $k(N-1)$ krocích

– minimální směrování FF = Farthest First = nejvzdálenější nejdříve $(N-1)$ kroků

~~ve~~ – V mřížkách používám XY směrování + FF

MINIMALIZACE PAMĚŤOVÝCH POŽADAVKŮ PŘI PERMUTACÍCH V M ~ nechci aby se pakety potkávaly

Randomizace perm. směr. – přes náhodně vygenerovaný mezilehlý uzel nebo přes náhodně zvolený uzel ve stejné části sloupce

Perm. směrování založené na třídění: 1, třídění do hada po sloupcích podle adres cílových sloupců, 2, každý druhý sloupec převrácen, 3, ~~posun~~ permutace do cílových sloupců, 4, permutace do cílových řádků

OFFline permutační směrování: 1, předpočítání permutací, aby byl v řádku max. 1 paket určený pro jakýkoliv daný sloupec, 2, ve všech sloupcích se provede permutace, 3, sloupc., 4, řádky

OAB v SF: všeport ($\rho = 0$), 1-port ($\rho = \max(\phi, \log|V|)$), pro všeport je triv. alg. ZÁPLAVOVÝ

EREW PRAM: bin. zdvojování $\quad$ úplný graf a hyperkrychle: binomiální kostra

mřížky: dimenzionálně uspořádané kostry

OAB v WH: k-portová síť $\rho = \log_{k+1}|V|$

hyperkrychle: Algoritmus dvojitého stromu

mřížka 1port a toroid 1port: binární zdvojování

všeport M a T: $\rho = \log_{2n+1} N$, 3 fázový diagonální algoritmus

– 1, do všech řádků, 2, na hlavní diagonálu, 3, do 4 diagonálních pásů rekurzivně

MC: hyperkrychle: setřídění + bin zdvojování

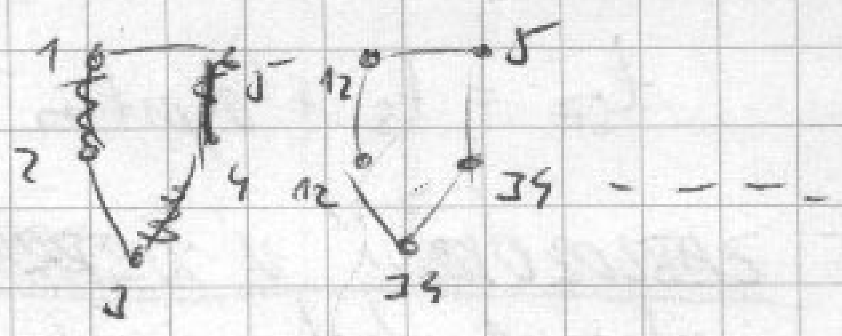
mřížky: setřídit, rozdělit na poloviny, je-li zdroj ve spodní polovině, tak pošle prvnímu uzlu v horní polovině, jinak poslednímu uzlu spodní poloviny } pro 1-port WH RD

OAS nekombinující: 1port: FF, všeport: $\rho = \frac{N-1}{\text{stupeň zdrojového uzlu}}$

OAS kombinující: jako OAB, ale jiné velikosti zprávy, SF: postupné přeposílání, WH: binární zdvojování, hyperkrychle: binom. kostra

⑥ AAB v komb. SF, FD:

mřížky: sudo-liché výměny $\left(r = \begin{cases} z-1 & z \text{ sudé} \\ z & z \text{ liché} \end{cases}\right)$



toroidy: 〃 $\left(r = \begin{cases} \frac{z}{2} & z \text{ sudé} \\ \frac{z+3}{2} & \text{liché} \end{cases}\right)$

hyperkrychle: n kroků, velikost zprávy se v každém kroku zdvojnásobí

AAB v nekomb. SF, FD: $\rho = \left\lceil \frac{N-1}{\text{minimální stupeň uzlu}} \right\rceil$

2D M: uzel vyšle svůj paket oběma směry po Ham. kružnici

T_n a Q_n: časově hranově disjunktní kostry pro všechny uzly (generic kostra + posunutí)

AAS: SF s komb.: standardní (SV) výměna na hyperkrychli $t_{AAS}(Q_n) = n.(t_s + \mu t_m 2^{n-1})$

1D T: cyklické obíhání

WH s komb.: stejné jako WH nekomb., nebo SF SV na Q_n

M a T: binární výměna = rekurzivní půlení, $\log N$ fází, ale každá má několik kroků kvůli zahlcení. WH kanály, každý krok = výměna $N/2$ paketů

WH bez kombinace: Q_n - přímá výměna 2^n-1 permutací $\pi_j: x \to x \text{ xor } j$
$t_{AAS} = (2^n-1)(t_s + n t_d/2 + \mu t_m)$

LIN. ALGEBRA

transpozice matic: proužkové mapování → AAS, SF 2D Mřížka: submatice v řádcích vysílány horizontálně k hl. diagonále,

$$T(n^2,p) = t_s + \underbrace{2(\sqrt{p}-1)}_{\text{vzdálenost}} \underbrace{\frac{n^2}{p}}_{\text{velikost 1 submatice}} + \underbrace{O\left(\frac{n^2}{p}\right)}_{\text{lokální}}$$

Q_n: $\to M(\sqrt{2^q}, \sqrt{2^q})$ virtuální mřížka $\quad T(n^2,p) = \left(t_s + 2\frac{n^2}{p} t_m\right)\frac{q}{2} + O\left(\frac{n^2}{p}\right)$

Násobení matice vektorem:

proužkové mapování po řádcích: 1, každý procesor pošle svou část $\vec{x}$ ostatním (AAB)
2, sekvenčně skalární součin

proužkové mapování po sloupcích: sekv. skal. součiny, redukce se sčítáním, příp. rozeslání výsledku

šachovnicové mapování: 1, právé procesory pošlou $\vec{x}$ na diagonálu, 2, ty je rozešlou ve sloupci, 3, lokální násobení, par. redukce v řádcích

Násobení matic: naivní algoritmus $M(m,l,n)$, $(m \times l) \times (l \times n)$, součin + paralelní redukce

méně naivní: blokově šachovnicové mapování, v řádcích se rozešlou všechny submatice A ve sloupci se rozešlou všechny submatice B, na každém procesoru se vypočítá výsledná submatice C

Cannonův algoritmus: toroid, předrotace A doleva, B nahoru, $\sqrt{p}$ kroků (P_{ik} vynásobí momentální submatici, přičte k $C_{\dagger}$, udělá rotaci A doleva, B nahoru)

⑦ <u>Foxův algoritmus:</u> $\sqrt{p}$ kroků (DAB submatice A z diagonály, přičtení součinu submatice, rotace B)

<u>Tridiagonální soustava rovnic:</u> $\log n$ redukcí, kdy se do sudých rovnic dosazují $x_i =$ tvary lichých rovnic

<u>LU dekompozice:</u> $A = L \times U$ $\quad L = \left[\begin{smallmatrix} 1 & 0 \\ x & 1 \end{smallmatrix}\right]$ $\quad U = \left[\begin{smallmatrix} u & u \\ 0 & u \end{smallmatrix}\right]$, střídavě se počítají řádky

U a sloupce L, blok. cyklické mapování (šachovnice)

$M(r,r)$, $r = \sqrt{p}$, $n = rsq$, submatice jsou $s \times s$, procesor jich má q^2

<u>Jacobiho iterační metoda:</u> $\vec{x}_i(t+1) = \dfrac{\vec{b} - A' \cdot \vec{x}(t)}{a_{i,i}}$; 1 krok $\approx O(A \cdot \vec{x}(t))$

– počet kroků není dopředu známý, while $(\|\vec{r}(t)\| > \varepsilon)$