PRESEK
List za mlade matematike, fizike, astronome in računalnikarje

ISSN 0351-6652
Letnik **21** (1993/1994)
Številka 5
Strani 290–295

Sandi Klavžar:

HIPERKOCKE IN RAČUNALNIK CM

Ključne besede: računalništvo, hiperkocke, računalnik CM.

Elektronska verzija: http://www.presek.si/21/1186-Klavzar.pdf

HIPERKOCKE IN RAČUNALNIK CM

Uvod

Recimo, da moram napisati vsa števila med 1 in 10000. Na pomoč mi priskoči devet prijateljev. Domenimo se, katerih tisoč števil bo vsak napisal, in delo je (skoraj) desetkrat hitreje opravljeno. Po delu pride jelo. Lačen sem za deset piškotov. No, tu mi prijatelji ne morejo "pomagati", če se želim najesti. Najprej pojem prvi piškot, nato drugega in nazadnje desetega. Kaj želim pravzaprav povedati? Nekatera opravila lahko razdelimo na manjše naloge in vsako naredimo neodvisno (istočasno), zopet druga opravila pa ne dopuščajo takega pristopa.

Podobno je v računalništvu. Nekateri problemi, ki jih rešujemo, so po naravi taki, da jih ne moremo razdeliti na manjše podprobleme in le-teh reševati istočasno ali pa tak pristop ne bi bil bistveno hitrejši. Zopet drugi problemi kar kličejo po "vzporednem" reševanju. Na primer prikazovanje grafične slike, simulacija vremena, simulacija zračnega toka preko letalskega krila, simulacija delovanja možganov, določeni fizikalni procesi in še bi lahko naštevali.

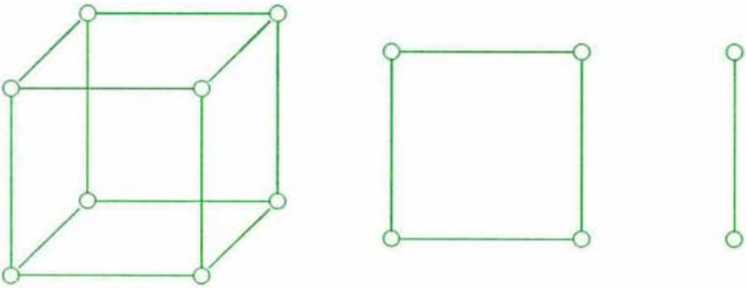
Osnovna lastnost *vzporednih računalnikov* je, da imajo več procesorjev, ki istočasno rešujejo skupen problem in po potrebi med seboj komunicirajo (si izmenjujejo sporočila). Eden prvih uspešnih poskusov izdelave vzporednega računalnika je računalnik CM (CM je kratica za *Connection Machine*), ki je bil izdelan v letih 1986 in 1987, najprej različica CM1 in nato še CM2. Kot bomo spoznali, so njegovi procesorji razporejeni v obliki hiperkocke. Naj bo to zadostna motivacija, da si najprej ogledamo, kaj je to hiperkocka.

Hiperkocke

Kaj je kocka, vemo vsi. Če v vsakem oglišču kocke narišemo krogec in krogca, ki sta povezana z robom, povežemo s črto, dobimo risbo, ki je prikazana na sliki 1(a).

Temu, kar imamo na sliki, rečemo *graf* (za več znanja o grafih velja pobrskati po starih Presekih), krogci so *točke* grafa, črte pa so njegove *povezave.* Graf s slike 1(a) imenujemo 3-kocka in ga označimo s Q_3. To je torej kocka v prostoru, v ravnini pa imamo 2-kocko Q_2 in na premici 1-kocko Q_1. Tudi grafa Q_2 in Q_1 si lahko ogledamo na sliki 1.

Kaj pa je graf Q_4 in, še naprej, kaj je graf Q_n, tj. n-kocka? Definirajmo n-kocko na dva načina; prva definicija nam bo pomagala dobiti predstavo o tem, kakšen je ta graf, druga definicija pa bo primernejša za "računanje".

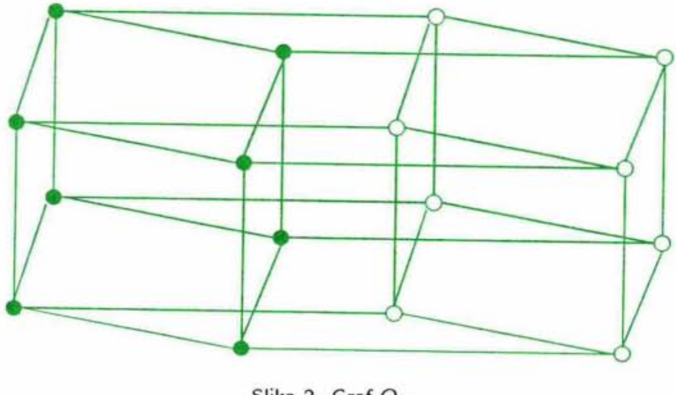


(a) (b) (c)

Slika 1. Grafi Q_3, Q_2 in Q_1.

Definicija 1. Graf Q_1 je graf z dvema točkama in eno povezavo. Graf Q_n, $n \geq 2$, je graf, ki ga dobimo tako, da vzamemo dve kopiji grafa Q_{n-1} in s povezavami povežemo istoležne točke v obeh kopijah.

Na sliki 2 je prikazano, kako iz dveh grafov Q_3 sestavimo graf Q_4. Za lažjo predstavo so točke ene izmed obeh 3-kock pobarvane.



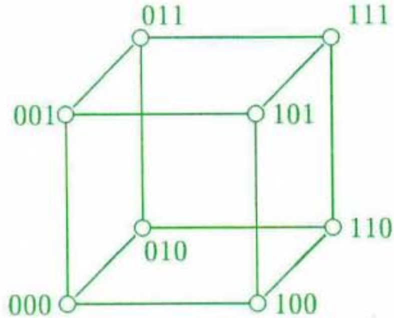
Slika 2. Graf Q_4.

Definicija 2. Graf Q_n, $n \geq 1$, je graf, katerega točke so vsi nizi oblike

$$b_1 b_2 \ldots b_n ,$$

kjer je $b_i \in \{0, 1\}$, $i = 1, 2, \ldots, n$. Točki grafa Q_n sta povezani natanko tedaj, ko se razlikujeta na natanko enem mestu (bitu).

Z drugimi besedami, točke grafa Q_n so vsi nizi dolžine n, sestavljeni iz ničel in enic. Na sliki 3 je prikazano, kako po tej definiciji ponazorimo graf Q_3.



Slika 3. Graf Q_3 še enkrat.

Ker nismo definirali, kaj pomeni, da sta dva grafa enaka, tudi ne moremo formalno dokazati, da smo v gornjih definicijah dobili enake grafe – *hiperkocke*. Kakorkoli, utemeljitev je naslednja: V definiciji 2 najprej pogledamo tiste točke, ki imajo prvi bit enak 0. Opazimo, da določajo za eno dimenzijo manjšo hiperkocko. Podobno velja, da táko hiperkocko določajo tudi vse točke, ki imajo prvi bit enak 1. Ker so nadalje istoležne točke v obeh kopijah povezane s povezavo, res dobimo hiperkocko iz definicije 1.

Grafi Q_n imajo nekatere lepe lastnosti, zaradi katerih so načrtovalci računalnika CM izbrali prav enega od teh grafov za osnovno shemo, v katero so razporedili procesorje. Najprej ugotovimo, da ima n-kocka 2^n točk. Res, trditev za $n = 1$, 2 in 3 lahko preverimo kar na sliki 1. Predpostavimo sedaj, da graf Q_{n-1} vsebuje 2^{n-1} točk. No, po obeh definicijah ima tedaj graf Q_n dvakrat toliko točk, torej $2 \cdot 2^{n-1} = 2^n$.

Da ima graf Q_n natanko 2^n točk, je gotovo zelo lepa lastnost, saj vemo, da imajo potence števila 2 zaradi dvojiškega številskega sestava posebno vlogo v računalništvu. Spomnimo se le na procesorje s 16 ($=2^4$) in 32 ($=2^5$) biti, da 1K pomeni 2^{10} zlogov, ...

Zaporedje različnih točk grafa imenujemo *pot*, če sta poljubni dve zaporedni točki povezani s povezavo. *Dolžina poti* je za ena zmanjšano število točk na poti (torej število povezav na poti). Na primer, zaporedje točk grafa Q_3: 001, 011, 111, 110, 010 določa pot dolžine 4.

Naslednja pomembna lastnost hiperkock je:

Trditev 1. V grafu Q_n je vsak par točk povezan s potjo, ki ni daljša od n.

Da se prepričamo v veljavnost gornje trditve, uporabimo definicijo 2. Naj bosta $b_1 b_2 \ldots b_n$ in $c_1 c_2 \ldots c_n$ poljubni točki n-kocke in recimo, da se razlikujeta v k bitih. No, tedaj po vrsti spremenimo vsakega izmed teh k bitov točke $b_1 b_2 \ldots b_n$ in tako po k povezavah pridemo do točke $c_1 c_2 \ldots c_n$. Ker se točki lahko razlikujeta v največ n bitih, smo svojo trditev preverili.

Če smo pozorno sledili gornjemu dokazu, smo lahko opazili vsaj še dvoje:

- Če se točki razlikujeta v k bitih, med njima obstaja pot dolžine k.
- Načinov, kako pridemo iz ene točke v drugo, je veliko.

Ponazorimo drugo ugotovitev s primerom: V grafu Q_4 lahko pridemo po povezavah iz točke 1001 do točke 0010 na naslednje načine (vedno je podčrtan bit, ki ga spremenimo):

$$
\begin{array}{ccccccc}
\underline{1}001 & \longrightarrow & 00\underline{0}1 & \longrightarrow & 001\underline{1} & \longrightarrow & 0010 \\
\underline{1}001 & \longrightarrow & 000\underline{1} & \longrightarrow & 00\underline{0}0 & \longrightarrow & 0010 \\
10\underline{0}1 & \longrightarrow & \underline{1}011 & \longrightarrow & 001\underline{1} & \longrightarrow & 0010 \\
10\underline{0}1 & \longrightarrow & 101\underline{1} & \longrightarrow & \underline{1}010 & \longrightarrow & 0010 \\
100\underline{1} & \longrightarrow & \underline{1}000 & \longrightarrow & 00\underline{0}0 & \longrightarrow & 0010 \\
100\underline{1} & \longrightarrow & 10\underline{0}0 & \longrightarrow & \underline{1}010 & \longrightarrow & 0010
\end{array}
$$

Različnih načinov je torej toliko, na kolikor različnih načinov lahko izberemo vrstni red tistih k bitov, ki jih spremenimo. To lahko naredimo na $k! = k \cdot (k-1) \cdot \ldots \cdot 2 \cdot 1$ načinov. Povzemimo:

Trditev 2. Če se točki n-kocke razlikujeta v k bitih, med njima obstaja $k!$ različnih poti dolžine k.

Računalnik CM

Pri načrtovanju vzporednega računalnika naletimo na več problemov, med katerimi je zelo pomemben naslednji: Kako zagotoviti, da bo računalnik čimbolj univerzalen, tak torej, da bomo z njim lahko učinkovito reševali raznovrstne probleme? Računalnik CM je zamišljen tako, da njegovemu uporabniku ni potrebno poznati njegove zgradbe. Računalnik CM je namreč priključen na glavni računalnik, na katerem delamo v običajnem operacijskem sistemu in z običajnimi programskimi jeziki. Procesorji računalnika CM pa so povezani z glavnim računalnikom podobno kot običajni pomnilnik. Zato lahko računalnik CM razumemo tudi kot pomnilnik glavnega računalnika. Dejansko nam torej za uporabo ni potrebno poznati njegove zgradbe.

Vseeno pa si oglejmo, kako je sestavljen. Po 16 procesorjev je združenih na enem čipu, na vsaki plošči (tiskanega vezja) je 32 čipov s procesorji, 16 plošč je združenih v eno kocko, osem kock pa tvori celotni računalnik (glej sliko na zadnji strani ovitka). Povzemimo:

8 kock po 16 tiskanin po 32 čipov po 16 procesorjev

znese skupaj

$$2^3 \cdot 2^4 \cdot 2^5 \cdot 2^4 = 2^{16} = 65536$$

procesorjev. Računalnik CM torej sestavlja zares veliko število procesorjev, kar po drugi strani pomeni, da le-ti ne morejo biti preveč zmogljivi (cena!). Zato ni preveliko presenečenje, ko izvemo, da so v resnici zelo primitivni – enobitni. Računalnik CM je tako zelo primeren za naloge, pri katerih mora posamezni procesor skrbeti za neki preprosti problem, recimo za nadzor pike na zaslonu. Slednje (natančneje nadzor grafične slike) je bila ena prvih uporab tega računalnika.

Praviloma vsak problem od procesorjev zahteva, da si izmenjujejo informacije. Lepo bi bilo, če bi vsak procesor povezali z vsakim, saj bi bila tako komunikacija najhitrejša. No, če bi to naredili na računalniku CM, bi potrebovali

$$\binom{65536}{2} = 2147450880$$

komunikacijskih linij, kar pa je seveda neizvedljivo. Zato je za komunikacije poskrbljeno takole: Procesorji na posameznem čipu so povezani neposredno. V vsak čip je vgrajen *usmerjevalnik*, ki skrbi za povezavo tega čipa z ostalimi. Kot vemo, je vseh čipov 2^{12}. (Sedaj pride tisto, zaradi česar smo si pravzaprav ogledali hiperkocke.) Čipi predstavljajo točke grafa Q_{12}, torej je vsak usmerjevalnik neposredno povezan z 12 drugimi usmerjevalniki. Zaradi trditve 1 vemo, da nobeno sporočilo ne bo potovalo dlje kot 12 korakov.

Oglejmo si še, kako neko sporočilo pride do pravega čipa. Vsak čip ima svoj naslov, to je zaporedje 12 ničel in enic v skladu z definicijo 2. Sporočilo potuje od začetnega do končnega čipa preko vmesnih usmerjevalnikov. Vmesni usmerjevalnik pozna končni naslov, kamor mora priti sporočilo. To mu tudi zadostuje, saj ve, da mora poslati sporočilo usmerjevalniku, ki je za en bit bližje končnemu naslovu. Pošiljanje se nadaljuje, dokler sporočilo ne pride do končnega naslova.

Med izvajanjem nekega procesa je lahko precej komunikacijskih linij zasedenih, tako da sporočil nekaj časa ne moremo poslati. Zaradi trditve 2 vemo, da imamo na razpolago veliko različnih poti do končnega naslova, torej nam zgradba 12-kocke omogoča, da do takih zastojev ne pride prepogosto. Najpreprosteje lahko to dejstvo uporabimo tako, da damo usmerjevalniku svobodo: sporočilo oddaj naprej po katerikoli trenutno prosti povezavi, ki vodi proti ciljnemu naslovu.

Za konec omenimo, da v zadnjih letih znanstveniki intenzivno proučujejo različne modele povezovanja procesorjev, večina teh modelov pa je tako ali drugače povezana z n-kockami. In še to: leta 1991 je bil izdelan računalnik CM5, ki je zasnovan drugače kot osnovni računalnik CM. O tem pa morda kdaj drugič.

Sandi Klavžar

IGRA "NIM*"

V 3. številki Preseka je Matjaž Željko predstavil igro "NIM". Rad bi omenil še različico te igre, ki jo bom imenoval "NIM*":

Na mizi se nahaja nekaj kupov s po nekaj kamni. Igralca izmenoma pobirata kamne z mize. V vsaki potezi mora igralec vzeti vsaj en kamen (lahko tudi vse) iz poljubnega kupa. **ZMAGA** *tisti, ki ne more več narediti poteze.*

(Pri "NIM" tisti, ki ne more več narediti poteze, IZGUBI).

Vabim bralce, da poiščejo zmagovito strategijo za to igro. Nasvet: Poizkusite z enako strategijo kot velja za igro "NIM", vendar jo spremenite tik pred zdajci (kdaj in kako ?).

Naj omenim, da smo to različico igrali že v davnem letu 1963 na prvem računalniku v Sloveniji (ZUSE), ki je deloval še na elektronke. V reklamnem paketu za prikaz "izrednih zmogljivosti računalnika" je bil tudi program za to igro (ne spominjam se, kako se je takrat imenovala).

Mitja Rosina