Složenost algoritama
9. predavanje

Saša Singer

singer@math.hr

web.math.hr/~singer

PMF – Matematički odjel, Zagreb

Blokovsko množenje matrica

Množenje matrica

Problem: Zadan je prirodni broj $n \in \mathbb{N}$ i 3 matrice A, B i C, reda n. Treba izračunati izraz

$$C := C + A * B.$$

Znamo da je realizacija po elementima trivijalna

$$c_{ij} := c_{ij} + \sum_{k=1}^{n} a_{ik} \cdot b_{kj},$$

za sve indekse

$$i = 1, \ldots, n, \quad j = 1, \ldots, n.$$

Dakle, "programski" — treba "zavrtiti" tri petlje.

Množenje matrica — realizacija po elementima

Programska realizacija na "skalarnoj" razini (po elementima) ima ovaj opći oblik:

- 3 petlje po i, j, k, svaka od 1 do n,
  - operacija unutar tih petlji je
    $$c_{ij} := c_{ij} + a_{ik} \cdot b_{kj},$$
    tj. množenje i zbrajanje skalara.

Ove tri petlje smijemo permutirati pa dobivamo 6 različitih varijanti osnovnog algoritma:

- `ijk`, `ikj`, `jik`, `jki`, `kij`, `kji`.

Množenje matrica — podjela na blokove

Matrice A i B možemo podijeliti na blokove

$$A = \left[\begin{array}{c|c|c|c} A_{11} & A_{12} & \cdots & A_{1r} \\ \hline A_{21} & A_{22} & \cdots & A_{2r} \\ \hline \vdots & \vdots & \cdots & \vdots \\ \hline A_{p1} & A_{p2} & \cdots & A_{pr} \end{array}\right], \qquad B = \left[\begin{array}{c|c|c|c} B_{11} & B_{12} & \cdots & B_{1q} \\ \hline B_{21} & B_{22} & \cdots & B_{2q} \\ \hline \vdots & \vdots & \cdots & \vdots \\ \hline B_{r1} & B_{r2} & \cdots & B_{rq} \end{array}\right].$$

Ako su blokovi A_{ik} i B_{kj} takvi da se mogu množiti za sve indekse i, j, k, onda operaciju $C = C + A * B$ možemo izračunati "po blokovima", gdje je

$$C_{ij} = C_{ij} + \sum_{k=1}^{r} A_{ik} * B_{kj}, \quad i = 1, \ldots, p, \quad j = 1, \ldots, q.$$

Množenje matrica — blokovi (nastavak)

Podjela matrica A i B na blokove koji se mogu množiti inducira podijelu matrice C na blokove

$$C = \left[\begin{array}{c|c|c|c} C_{11} & C_{12} & \cdots & C_{1q} \\ \hline C_{21} & C_{22} & \cdots & C_{2q} \\ \hline \vdots & \vdots & \ldots & \vdots \\ \hline C_{p1} & C_{p2} & \cdots & C_{pq} \end{array}\right].$$

Pojednostavljenje: sve tri ulazne matrice su kvadratne reda n

- pa ih dijelimo na isti način u blokove.

Dakle, $p = q = r =$ (oznaka) $= N$, gdje je N tzv. "blok–red" matrice.

Množenje matrica — blokovi (nastavak)

Podjela sve tri matrice A, B i C ima isti oblik (napisan za C)

$$C = \left[\begin{array}{c|c|c|c} C_{11} & C_{12} & \cdots & C_{1N} \\ \hline C_{21} & C_{22} & \cdots & C_{2N} \\ \hline \vdots & \vdots & \cdots & \vdots \\ \hline C_{N1} & C_{N2} & \cdots & C_{NN} \end{array}\right].$$

Pojedini blokovi — podmatrice A_{ij}, B_{ij} i C_{ij} su

- matrice istog tipa, označimo ga s $n_i \times n_j$.

Uočite da blokovi ne moraju više biti kvadratne matrice — općenito su pravokutne.

Množenje matrica — blokovi (nastavak)

Za veličine blokova mora vrijediti

$$\sum_{i=1}^{N} n_i = n.$$

Kako se određuju veličine blokova n_i, za $i = 1, \ldots, N$ — malo kasnije.

Matrična operacija $C = C + A * B$ sad ima "blokovski" oblik

$$C_{ij} = C_{ij} + \sum_{k=1}^{N} A_{ik} * B_{kj}, \quad i = 1, \ldots, N, \quad j = 1, \ldots, N.$$

Množenje matrica — realizacija po blokovima

Programska realizacija na "blokovskoj" razini (po blokovima) ima ovaj opći oblik:

- 3 petlje po i, j, k, svaka od 1 do N,
  - operacija unutar tih petlji je

$$C_{ij} := C_{ij} + A_{ik} \cdot B_{kj},$$

tj. množenje i zbrajanje matrica.

Ova operacija ima isti oblik `xGEMM` kao i cijeli polazni problem ("rekurzija"), samo što matrice ne moraju biti kvadratne

$$(n_i \times n_j) = (n_i \times n_j) + (n_i \times n_k) * (n_k \times n_j).$$

Blokovsko množenje matrica — petlje

Tri petlje za blokove smijemo permutirati — pa dobivamo 6 različitih varijanti blokovskog algoritma:

- `ijk`, `ikj`, `jik`, `jki`, `kij`, `kji`.

Za "unutarnje" množenje pojedinih blokova, također, imamo odgovarajućih 6 varijanti osnovnog algoritma.

- Dakle, sve skupa, imamo 36 varijanti!

Tko hoće, neka proba sve. Ja neću.

U nastavku koristim

- istu varijantu (permutaciju petlji) i za blokovski i za osnovni (skalarni) algoritam.

Blokovsko množenje matrica — veličine blokova

Ideja: veličine blokova izabrati tako da se unutarnje množenje blokova

$$C_{ij} := C_{ij} + A_{ik} \cdot B_{kj}$$

(operacija `xGEMM`) obavlja u cacheu.

Postupak. Iz tablice brzina za odabrani osnovni algoritam

- nađemo približni maksimalni red n za koji još dobivamo punu “cache” brzinu.

Nazovimo taj red s n_{cache}.

Veličine blokova (nastavak)

Cilj podjele na blokove je

- unutarnje množenje blokova mora raditi s matricama veličine manje (ili jednake) n_{cache}.

Dakle, mora vrijediti

$$n_i \leq n_{\mathrm{cache}}, \quad i = 1, \ldots, N.$$

Tome dodajemo raniji uvjet

$$\sum_{i=1}^{N} n_i = n.$$

Veličine blokova (nastavak)

Za nalaženje n_i standardno se koriste dva pristupa.

- "equal–sized" — svi n_i imaju podjednaku veličinu, tj. razlika među njima je najviše 1.
- "greedy" — svi n_i imaju maksimalnu veličinu n_{cache}, osim, eventualno, jednog od njih (prvi ili zadnji).

U primjerima se koristi "equal–sized" podjela.

Napomena. Pravu (najbolju) vrijednost za n_{cache} određujemo

- testiranjem blokovskog algoritma!

(Taj treba biti što brži.)

Blokovsko množenje matrica — indeksi

I još, da nam se indeksi i oznake ne “pomiješaju”

- što indeksira blokove, a što elemente,

dodajemo podindeks “b” za sve što se odnosi na blokove.

Blokovsko množenje matrica — primjer

IVF s normal opcijom za `jik` petlju daje brzine:

- 1050 MFlops za $n \leq 50$,
- 205 MFlops za velike n.

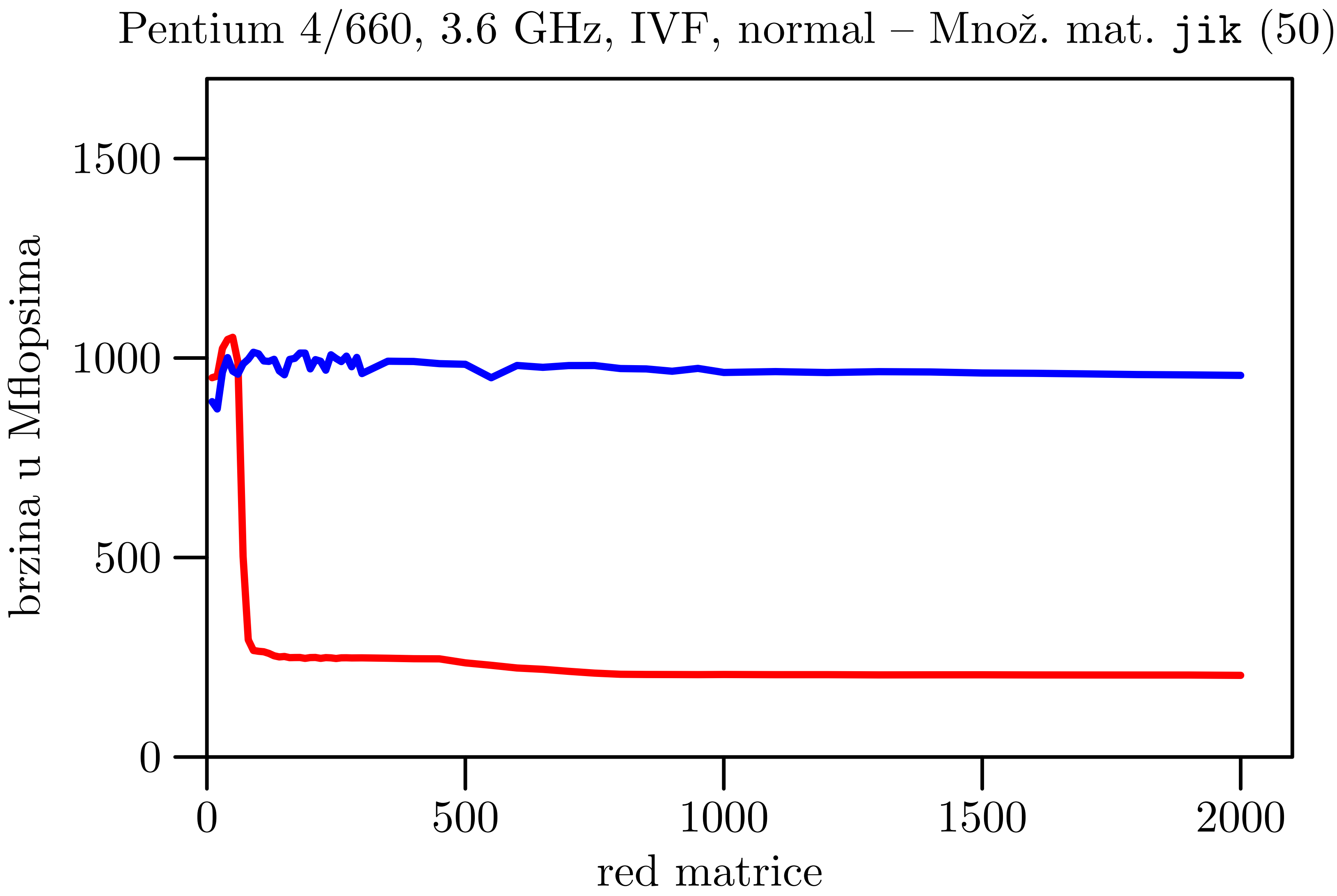
IVF s normal opcijom za `jki` petlju daje brzine:

- 1100 MFlops za $n \leq 300$,
- 840 MFlops za velike n.

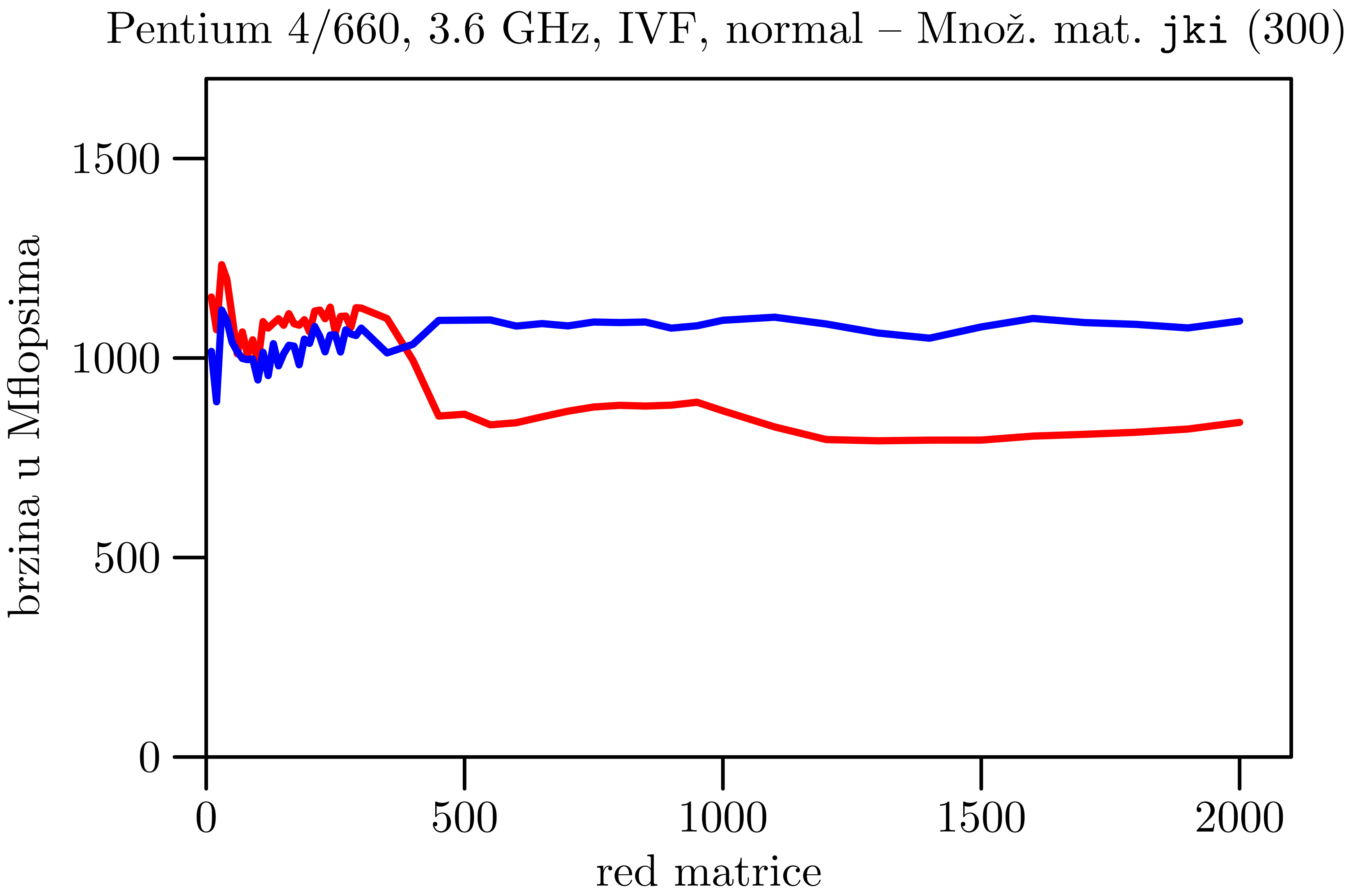
IVF s fast opcijom za `jki` petlju daje brzine:

- 2000 MFlops za $n \leq 300$,
- 1250 MFlops za velike n.

BabyBlue, IVF, normal — `jik` obični i blok (50)



BabyBlue, IVF, normal — `jki` *obični* i *blok (300)*



BabyBlue, IVF, fast — `jki` obični i blok (300)