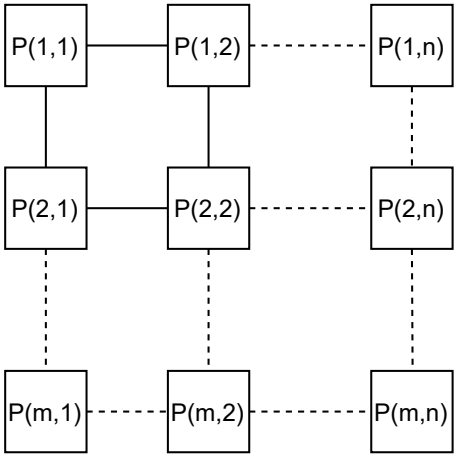
# Paralelní a distribuované algoritmy
# **Projekt 2**

Branislav Mateáš (xmatea00)
xmatea00@stud.fit.vutbr.cz

## Matrix mesh multiplication

Je algoritmus pre násobenie matíc. Procesory sú zapojené v topológii 2D mriežky, viď obr. 1. Pre dve matice o veľkostiach $m*k$ a $k*n$ je potrebných $m*n$ procesorov. Algoritmus funguje tak, že každý procesor čaká na hodnotu jeho ľavého a horného suseda. Po prijatí oboch hodnôt tieto hodnoty procesor vynásobí a pripočíta k premennej. Po vykonaní operácie násobenia rozposiela ďalej v smere mriežky prijaté hodnoty. Tzn., hodnotu prijatú od ľavého suseda posiela ďalej pravému susedovi a hodnotu prijatú od horného suseda posiela ďalej spodnému susedovi. Ak je procesor na kraji mriežky, hodnoty nikomu neposiela.




Obr. 1: 2D mriežka $m * n$ procesorov

### Časová a priestorová zložitosť

Pre dve vstupné matice $A_{(m*k)}$ a $B_{(k*n)}$ bude posledný krok algoritmu vykonaný až, keď prídu prvky $a_{(m,1)}$ a $b_{1,n}$ do posledného procesoru P(m, n). To bude v kroku:

$$k + max(m, n) - 1$$

a teda jeho časovú zložitosť možno vyjadriť:

$$t(m.n.k) = k + max(m, n) - 1$$

Priestorovú zložitosť je možné odvodiť z veľkosti matíc:

$$p(m, n) = m * n$$

Celková cena algoritmu odpovedá:

$$c(m, n, k) = p(m, n) * t(m.n.k) = m * n(k + max(m, n) - 1)$$

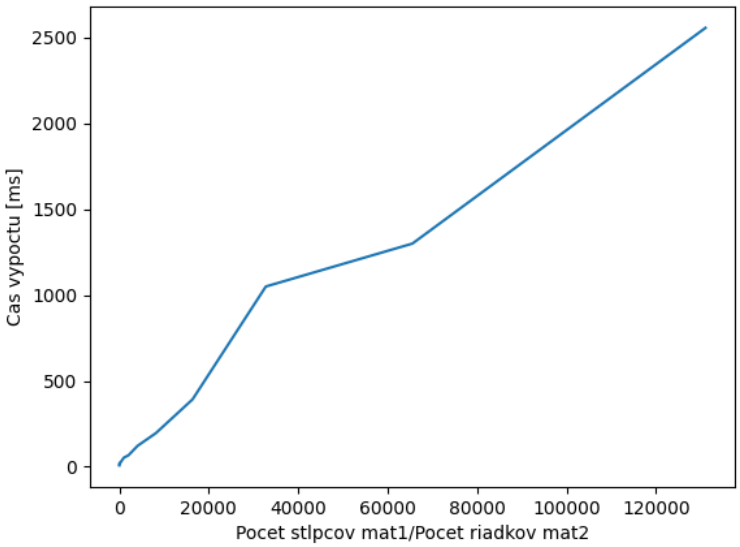
# Implementácia

Algoritmus je implementovaný v programovacom jazyku C++ a komunikácia medzi procesormi zaistená použitím knižnice OpenMPI. Každý procesor obsahuje cyklus for, v ktorom čaká na prijatie oboch hodnôt a výslednú hodnotu pripočíta k premennej prod. Ďalej tieto hodnoty posiela svojim susedom v smere mriežky. K orientácii v mriežke sú pripravené funkcie, ktoré pracujú so šírkou/výškou matice a aktuálnym ID Procesoru. Po skončení výpočtu posiela hodnotu premennej prod hlavnému procesoru $P_0$.

Časť implementácie $P_0$ sa navyše od ostatných procesov líši v tom, že v cykle for navyše posiela krajným horným a ľavým procesorom mriežky zvyšné hodnoty matíc v respektívnom smere. A hodnotu prod neposiela, ale naopak po skončení odosielania príjme od všetkých procesorov hodnotu prod a vypíše ju na štandardný výstup.

# Experimenty

Meranie bolo vykonané C++ knižnicou chrono. Meranie začína, keď hlavný proces začne spracovávať prvé dve čísla a meranie končí, keď hlavný procesor dostane výsledok od posledného procesoru. Meranie prebiehalo na pevnom počte procesorov 4x4. Každý test bol spustený 10−krát, hodnoty boli spriemerované a spracované do grafu 2.




Obr. 2: Výsledky merania

# Záver

Pre tento algoritmus bola odvodená teoretická časová zložitosť lineárna. Meranie však istú lineárnu závislosť preukázalo. Treba brať do úvahy, že réžia veľkého počtu procesov na menšom počte jadier môže mať vplyv na jemné odchýlky a anomálie, a teda bez potrebného HW nie je možné určiť presnú časovú zložitosť.