

MESH MULTIPLICATION

3. května 2015

Jan Bednařík (xbedna45)

1 Úvod

V tomto dokumentu je popsána problematika maticového násobení pomocí algoritmu `Mesh multiplication` na multiprocessorovém systému. Práce vznikla v rámci projektu v kurzu Paralelní a distribuované algoritmy (PRL), kdy bylo úkolem algoritmus implementovat v jazyce `C++` s použitím knihovny `OpenMPI` a experimentálně ověřit jeho předpokládanou časovou složitost.

2 Rozbor algoritmu

Algoritmus staví na topologii 2rozměrná mřížka procesorů, jež svými rozměry odpovídá rozměrům matice výsledku maticového násobení. Necht' \mathbf{A} , \mathbf{B} , \mathbf{C} jsou matice, platí $\mathbf{C} = \mathbf{AB}$, rozměry matice \mathbf{A} jsou $m \times n$ a rozměry matice \mathbf{B} jsou $n \times k$. Následující odstavce podávají odvození časové a paměťové složitosti a potřebný počet procesorů.

Časová složitost Řádky matice \mathbf{A} a sloupce matice \mathbf{B} jsou postupně čteny procesory v prvním sloupci a v prvním řádku 2rozměrné mřížky, přičemž každý procesor následně prvky zasílá procesoru napravo a procesoru dole. Každý řádek i_a matice \mathbf{A} je při horizontálním průchodu mřížkou o jeden krok opožděn oproti řádku $i_a - 1$ a stejně tak každý sloupec j_b matice \mathbf{B} je při vertikálním průchodu mřížkou procesorů o jeden krok opožděn oproti sloupci $j_b - 1$, díky čemuž se v každém kroku algoritmu do každého procesoru $P_{i_c j_c}$ dostanou odpovídající si hodnoty matic \mathbf{A} a \mathbf{B} . Časová složitost potom odpovídá počtu kroků s potřebných k tomu, aby byl poslední řádek matice \mathbf{A} a poslední sloupec matice \mathbf{B} zpracován procesorem P_{mk} :

$$s = (m - 1) + (k - 1) + n = m + n + k - 2 \quad (1)$$

Časová složitost algoritmu je tedy *lineární*, $O(N)$.

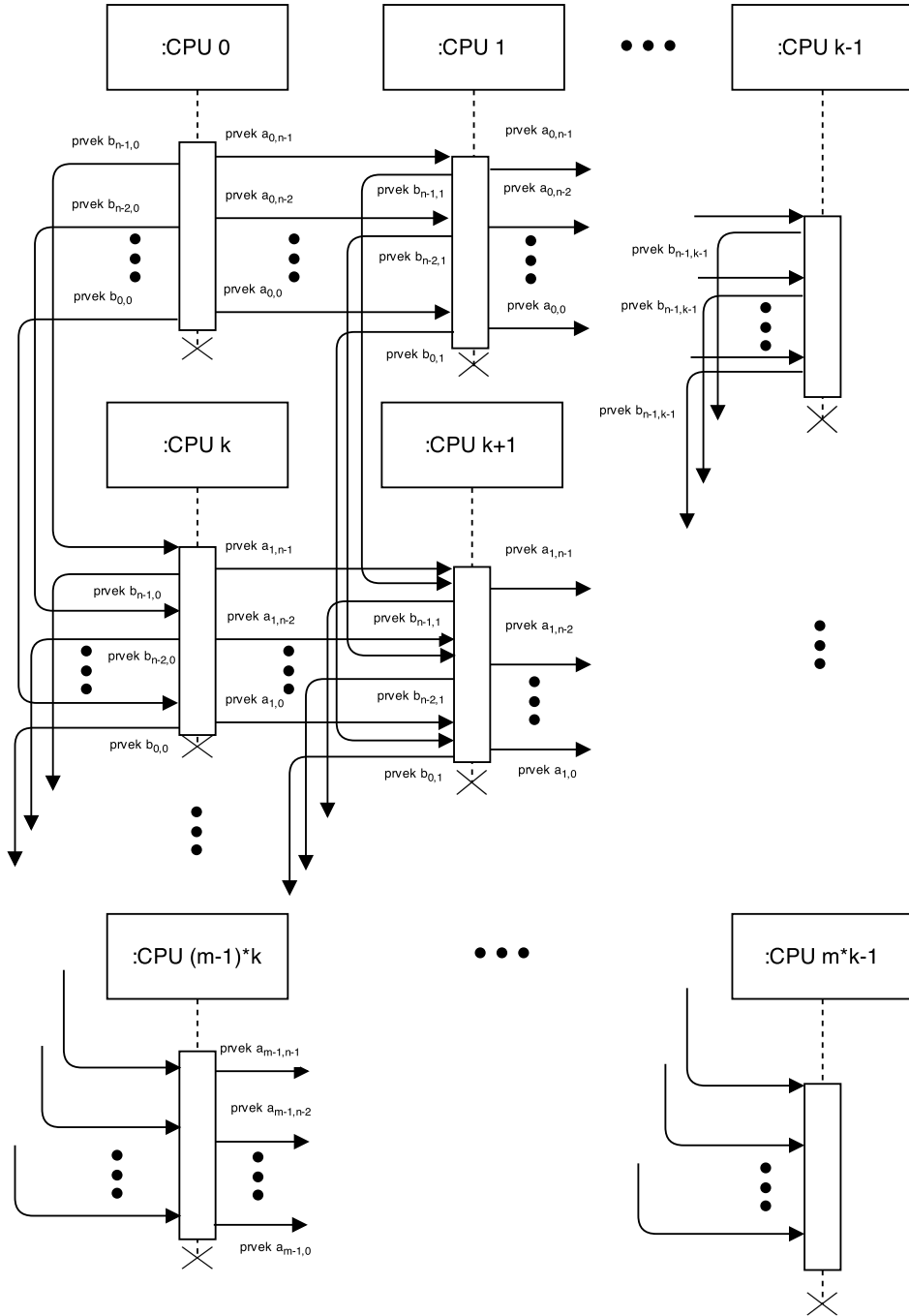
Počet procesorů Procesory tvoří 2rozměrnou mřížku odpovídající rozměrům výsledné matice. Počet procesorů lze tedy vyjádřit jako $P = m * k$, jedná se o *kvadratickou* složitost $O(N^2)$.

Paměťová složitost Každý procesor disponuje pouze pamětí pro dvě vstupní hodnoty a akumulátorem, paměťová složitost je tak *lineární*, $O(N)$

Cena Cena algoritmu je dána jako násobek počtu procesorů a časové složitosti, tedy $C(N) = N^2 * N = N^3$. Algoritmus `Mesh multiplication` není optimální.

3 Implementace

Program je implementován v C++ s použitím knihovny OpenMPI, jež staví na zasílání zpráv mezi procesory. Komunikační protokol použitý pro implementaci algoritmu *Mesh multiplication* je znázorněn ve schématu 1.



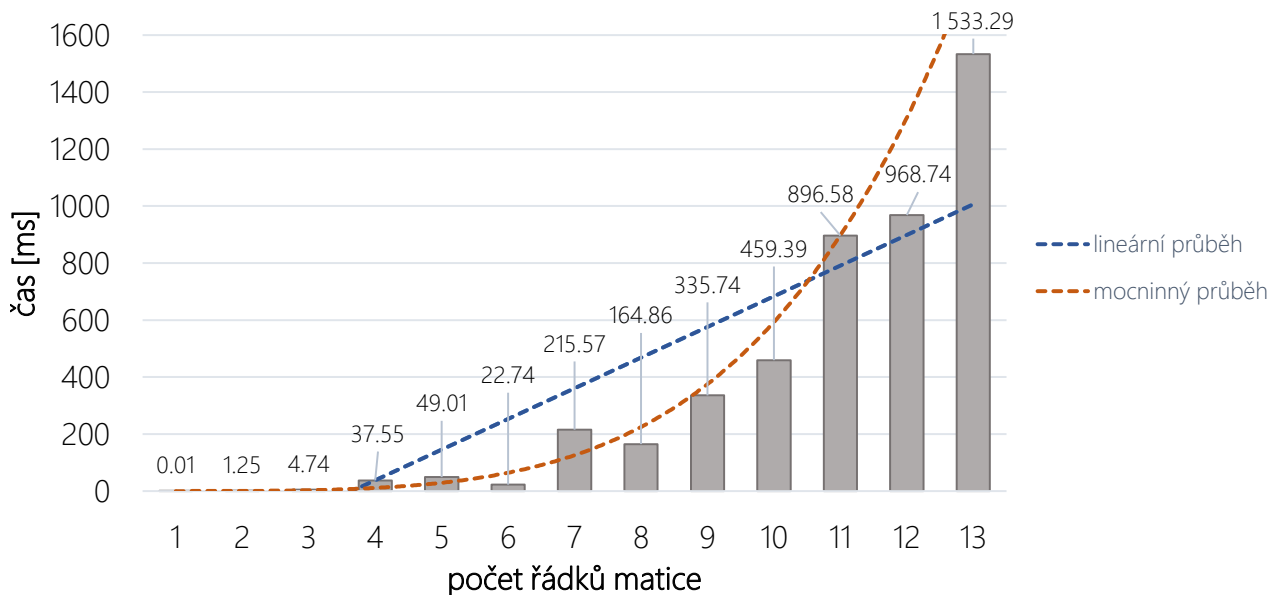
Obrázek 1: Komunikační protokol procesorů vyjádřený sekvenčním diagramem pro vstupní matice \mathbf{A} a \mathbf{B} obecných rozměrů $m \times n$ a $n \times k$.

Oproti teoretickému popisu průběhu algoritmu se implementace liší tím, že veškerá vstupní data načítá pouze procesor P_{00} a stará se o zasílání hodnot všem procesorům v prvním řádku a prvním sloupci 2-rozměrné mřížky procesorů.

4 Experimenty

Běh algoritmu byl otestován na jednoprocessorovém systému (2 jádra, 4 vlákna) ve virtualizovaném prostředí operačního systému **Lubuntu 14.04**. Testovány byly vstupní čtvercové matice o stejných rozměrech $n \times n$, kde $n \in \mathbb{N}, 1 \leq n \leq 13$. Čas běhu staví na knihovní funkci `MPI::Wtime()` pro přesné měření času. Měření provádí pouze první procesor, kdy spouští časovač po načtení všech vstupních dat a před započítáním zasílání prvků ostatním procesorům. Jakmile přijme výslednou hodnotu od posledního procesoru (P_{mk}), zastaví časovač. Doba běhu tak není zatížena vstupně výstupními operacemi.

Výsledky měření jsou uvedeny v grafu 2. Měřené časy pro jednotlivé velikosti vstupních matic jsou proloženy přímkou znázorňující očekávaný lineární průběh. Jak je z grafu patrné, lineární průběh je respektován pouze při nižších velikostech matic, zatímco se zvyšující se velikostí matic časovou složitost lépe modeluje kubický průběh. Tento je způsoben faktem, že byl systém testován na jednoprocessorovém systému, kde jsou fyzické procesory emulovány jako procesy, tedy vlivem nutnosti přepínat kontext paralelní zpracování postupně degraduje na sekvenční.



Obrázek 2: Závislost doby běhu programu na počtu vstupních prvků.

Pro dosažení co nejvěrnějších výsledků byl pro každou délku posloupnosti N test proveden desetkrát, z výsledků byly odstraněny dvě nejmenší a tři největší hodnoty a zbylých pět hodnot bylo průměrováno.

5 Závěr

V rámci projektu byl implementován algoritmus *Mesh multiplication*, byla odvozena jeho teoretická časová a paměťová složitost i cena a provedeny experimenty nad reálnými daty. Testy potvrdily lineární časovou složitost algoritmu, přesto byly zaznamenány nezanedbatelné odchylky. Ty jsou způsobené především proto, že je testovací stroj jednoprocessorový, tedy jednotlivé procesory požadované algoritmem jsou nahrazeny procesy sdílejícími výpočetní prostředky a neběží tak zcela paralelně. Při vyšších rozměrech vstupních matic tak složitost postupně degraduje na mocninnou podobně jako u sekvenčního přístupu.