Paralelní a distribuované algoritmy
Mesh multiplication

Jan Wrona
xwrona00@stud.fit.vutbr.cz

1 Rozbor a analýza algoritmu

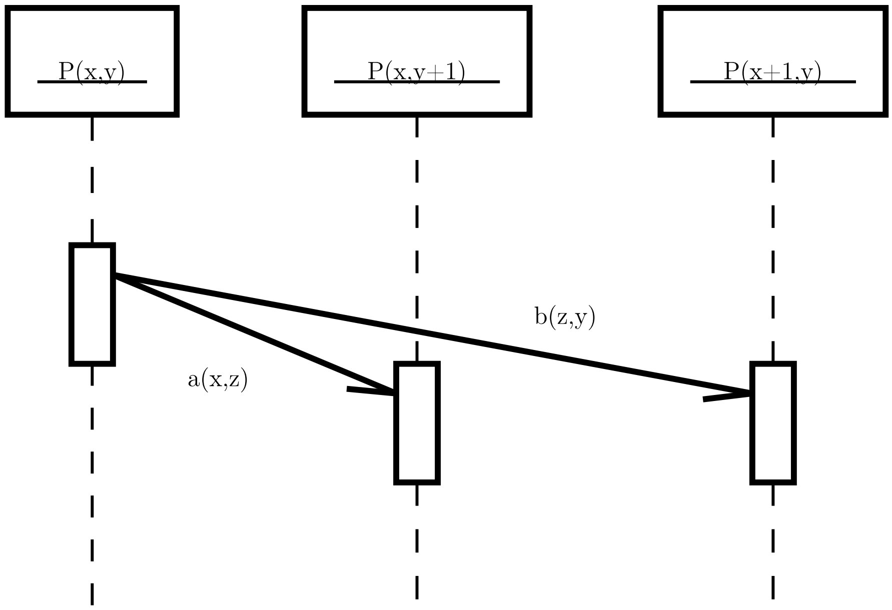
Algoritmus *mesh multiplication* je algoritmus pro násobení matic. Formálně je násobení matic definováno jako binární operace nad množinou matic. Pokud jsou operandy matice A a B, kde A (násobenec) má rozměr $m \times n$ a B (násobitel) má rozměr $n \times k$, výsledek je matice C o rozměru $m \times k$. Prvky matice C jsou dány vztahem

$$c_{ij} = \sum_{s=1}^{n} a_{is} \times b_{sj}, \qquad 1 \leq i \leq m, 1 \leq j \leq k. \tag{1}$$

V sekvenčním prostředí pro násobení matic existuje řada algoritmů, jejichž teoretická časová složitost je $O(n^x), 2 < x \leq 3$. Není známo, zda nejrychlejší z těchto algoritmů je optimální, žádný algoritmus však nemůže dosahovat složitosti lepší než $O(n^2)$, protože n^2 je počet prvků výstupní matice. Například naivní sekvenčním algoritmus se třemi vnořenými cykly dosahuje časové složitosti $O(n^3)$.

Pro následující rozbor algoritmu je jeden výpočetní krok složen z přijetí operandů procesorem, provedení příslušné operace a následné distribuce operandů. Mesh multiplication algoritmus využívá $m \times k$ procesorů, které jsou logicky uspořádány do matice. Rozměry této matice procesorů odpovídají rozměrům výsledné matice po násobení. V počátečním stavu jsou matice A a B rozmístěny napříč krajními procesory. Každý procesor z prvního sloupce zná jeden řádek matice A a každý procesor z prvního řádku zná jeden sloupec matice B. Procesor v levém horním rohu tedy zná první řádek matice A a první sloupec matice B atp. Procesor musí v jednom kroku vykonat několik primitivních úkonů, které se odvíjejí od jeho logického umístění. Prvním z těchto kroků je zisk dvou operandů. Procesory, jenž v počátečním stavu znají některý řádek a/nebo sloupec vstupní matice, jako operand použijí poslední prvek z řádku/sloupce. Ostatní procesory čekají na zprávu od svého souseda, která operand obsahuje. Prvky matice A jsou přijímány od levého souseda, prvky matice B od horního souseda. Po zisku obou operandů je možné provést jejich násobení. Násobky jsou v rámci procesoru akumulovány. Posledním úkonem je distribuce operandů. Procesory, které nejsou logicky umístěny v posledním sloupci odesílají zprávou prvek matice A svému pravému sousedovi, obdobně procesory mimo poslední logický řádek odesílají prvek matice B svému spodnímu sousedovi. Tento proces každý procesor opakuje $n$krát.

Čekání ná operandy je určitá forma synchronizace. V prvním kroku jsou oba operandy dostupné pouze procesoru $P(1, 1)$, ostatní čekají. V druhém kroku začínají pracovat také $P(1, 2)$ a $P(2, 1)$. Řádek i matice A se tedy začne používat až v kroku i, obdobně sloupec j matice B se



Obrázek 1: Obecný sekvenční diagram znázorňující zasílání zpráv jedním procesorem.

začne používat až v kroku j. Tímto je zajištěno, že prvky a_{is} a b_{sj} budou během jednoho kroku operandy v procesoru $P(i, j)$. Práce procesoru končí vyčerpáním všech prvků příslušného řádku a sloupce. Na konci algoritmu hodnota akumulovaná procesorem $P(i, j)$ odpovídá rovnici 1.

Procesor $P(1, 1)$ začíná pracovat v čase 1, provede n výpočetních kroků a končí tak čase n. $P(m, 1)$ začíná v čase m a končí v $m + n - 1$, procesor v protějším rohu logické matice $P(1, k)$ začíná v čase k a končí v $k + n - 1$. Z předchozího lze odvodit, že $P(m, k)$ svůj výpočet začne v čase $m + k - 1$ a po provedení n kroků ukončí výpočet v $m + k - 1 + n - 1 = m + k + n - 2$. Protože je poslední krok procesoru $P(m, k)$ posledním krokem celého algoritmu, je teoretická časová složitost lineární v závislosti na rozměrech vstupních matic. Za předpokladu, že $m \leq n$ a $k \leq n$ platí

$$t(n) = O(m + k + n - 2) = O(n).$$

Počet procesorů $p(n)$ roste kvadraticky s rozměry vstupních matic m a k, cena algoritmu je tak

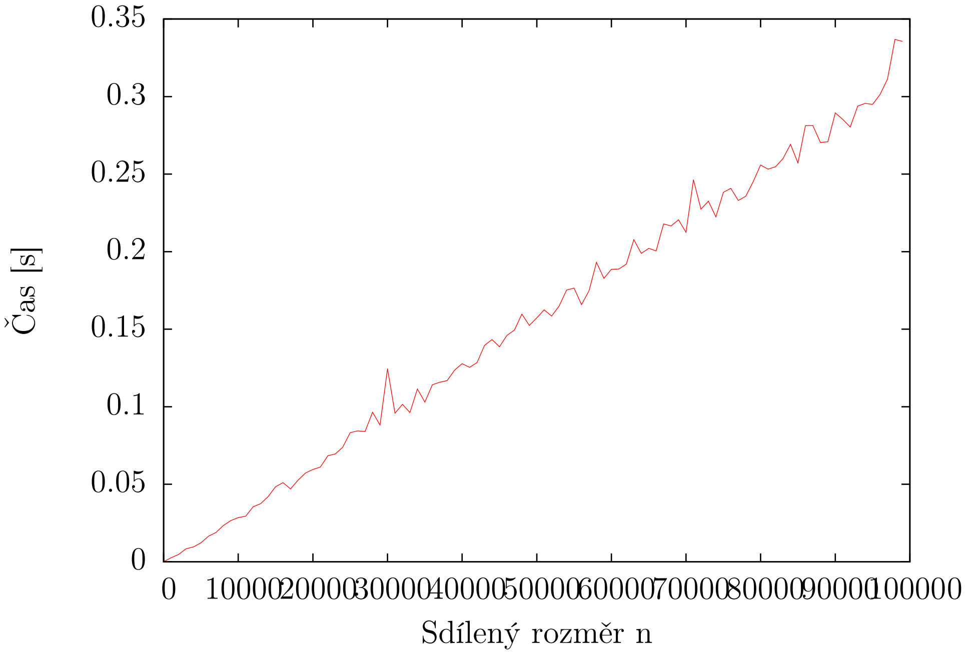
$$c(n) = O(n) * n^2 = O(n^3),$$

což odpovídá teoretické časové složitosti naivního sekvenčního algoritmu pro násobení matic a mesh multiplication tedy není optimální. Prostorová složitost je kvadratická: krajní procesory mají mezi sebe rozdělené matice A a B. Dále si každý procesor musí pamatovat částečný výsledek, který se po ukončení výpočtu stává prvkem matice C.

Zasílání zpráv mezi procesory je znázorněno sekvenčním diagramem na obrázku 1. Jde o obecný diagram pro jeden procesor a platí $1 \leq x \leq m - 1, 1 \leq y \leq k - 1, 1 \leq z \leq n$, tedy procesory v posledním sloupci již prvky matice A dále nezasílají, obdobně pro poslední řádek a matici B.

2 Testování a experimenty

První experiment měl za úkol prakticky ověřit časovou složitost $O(n)$, která byla odvozena dříve. Dodrženy byly předpoklady $m \leq n$ a $k \leq n$. Rozměry m a k byly zvoleny shodné (konkrétně



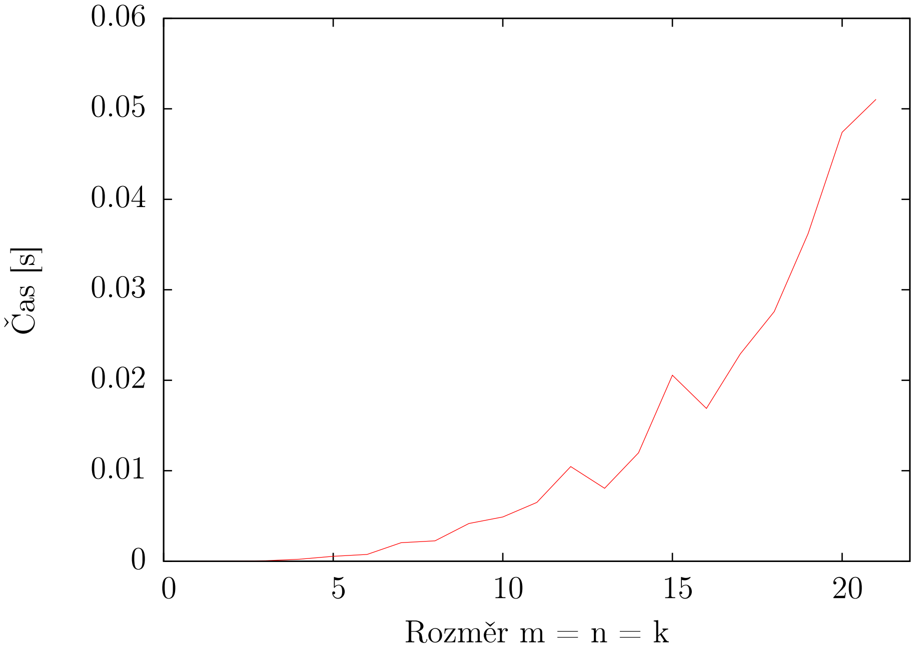
Obrázek 2: Experimentální ověření časové složitosti pro $m = k = 4$.

4), tak aby počet prvků výsledné matice odpovídal počtu fyzických jader na testovacím stroji (16). Program byl spouštěn s mapováním procesu na jádro, čímž byl podstatně omezen vliv přepínání procesů operačním systémem. Velikost sdíleného rozměru n byl graduálně zvyšován až do velikosti 100000. Jak lze vidět na obrázku 2, experiment byl úspěšný a prakticky potvrdil lineární teoretickou časovou složitost v závislosti na velikosti sdíleného rozměru matic n.

Cílem druhého experimentu bylo zjistit, jak se algoritmus chová při změnách rozměrů m a k. Vstupem byly čtvercové matice se shodnými rozměry $m = n = k$, pro každé měření se tedy měnil také počet potřebných procesorů. Při rozměru větším něž 4 již nebylo možné provést mapování procesu na jádro, tak jako v prvním experimentu. Pravděpodobně je to příčinou odchylek v měření zobrazeného grafem 3, nicméně i přesto lze pozorovat exponenciální průběh. Exponenciální růst času byl očekáván, počet prvků vstupních čtvercových matic totiž roste s druhou mocninou jejich rozměru.

3 Závěr

Rozbor v sekci 1 ukazuje, že teoretická časová složitost algoritmu mesh multiplication je lineární v závislosti na počtu prvků vstupních matic. Kvůli kvadraticky rostoucímu počtu procesorů ale algoritmus není optimální. Testy a experimenty uvedené v sekci 2 si kladly za cíl především ověřit teoretickou časovou složitost. Naměřené časy pro různé velikosti vstupních matic odpovídají lineárnímu průběhu a teoretickou složitos tak potvrzují také prakticky.



Obrázek 3: Experimentální ověření časové složitosti.