Paralelní a distribuované algoritmy – Mesh Multiplication

Radek Duchoň, xducho07

3. května 2021

1 Popis algoritmu

Mesh multiplication je paralelní algoritmus pro spočtení součinu matic. Pro vynásobení dvojice matic o rozměrech $m \times k$ a $k \times n$ prvků je tímto algoritmem vyžadováno $m \times n$ procesorů, které jsou uspořádány do konfigurace 2D mřížky. Procesory pracují dle následujícího algoritmu:

1. Procesory na levém okraji 2D mřížky přijímají „zleva“ data první matice po řádcích a procesory na horním okraji 2D mřížky přijímají „shora“ data druhé matice po sloupcích.
2. Procesory ve 2D mřížce přijímají k dvojic hodnot – jednu zleva a druhou shora a jejich součin přičítají ke svému výsledku, který je zpočátku 0.
3. Všechny procesory, které nejsou na pravém okraji 2D mřížky, zasílají hodnotu obdrženou zleva dále pravému sousednímu procesoru.
4. Všechny procesory, které nejsou na dolním okraji 2D mřížky zasílají hodnotu obdrženou shora dále dolnímu sousednímu procesoru.
5. Po vykonání kroků výše obsahují procesory jednotlivé hodnoty finální matice.

1.1 Analýza složitosti algoritmu

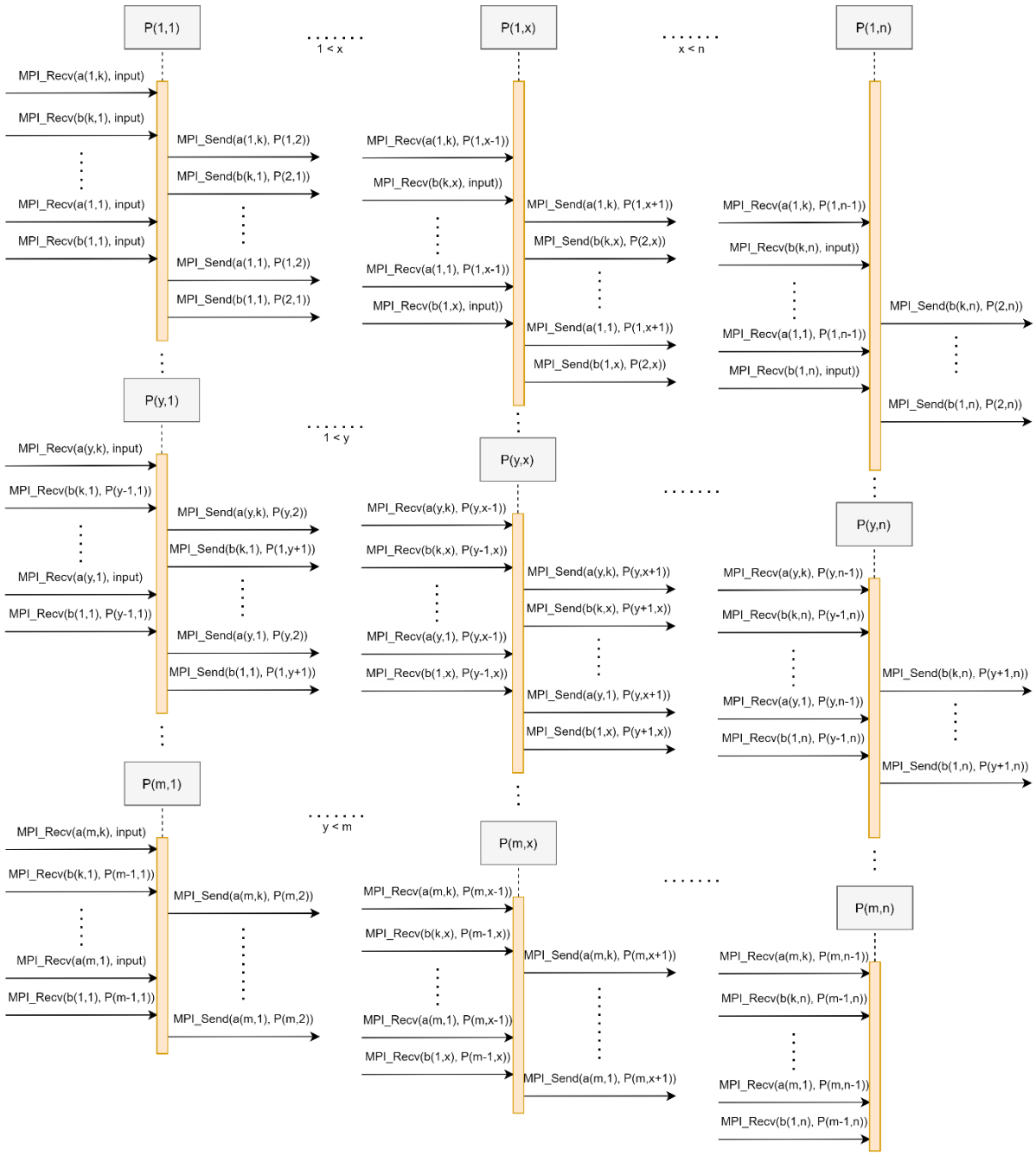
Uvažujme matici A o rozměrech $m \times k$ a matici B o rozměrech $k \times n$, pak poslední krok algoritmu bude proveden poté, co poslední hodnota $a_{m,1}$ posledního řádku m matice A a poslední hodnota $b_{1,n}$ sloupce n matice B dorazí do procesoru $P_{m,n}$ v pravém dolním rohu mřížky $m \times n$. Aby se tyto dvě hodnoty mohly začít distribuovat, musí se nejprve k procesoru $P_{m,1}$ dostat k prvků přes $m-1$ procesorů a k procesoru $P_{1,n}$ se musí dostat k prvků přes $n-1$ procesorů. Po $m+k-1$ krocích se tak začne distribuovat hodnota $a_{m,1}$, která musí projít dalšími $n-1$ procesory a po $n+k-1$ krocích se začne distribuovat hodnota $b_{1,n}$, která musí projít dalšími $m-1$ procesory.

Výpočet v procesoru $P_{m,n}$ bude dokonán po $k+m-1+n-1 = m+n+k-2$ krocích. Pro čtvercové matice stejných rozměrů lze vzorec zjednodušit do tvaru $n+n+n-2$, dostáváme tak lineární časovou složitost $3n-2 = O(n)$.

Prostorovou složitost lze odvodit z počtu procesorů v mřížce $p(m,n) = m * n$. Pro čtvercové matice kde $m = n$ lze funkci zjednodušit na tvar $p(n) = n * n = n^2$, získáváme tak kvadratickou prostorovou složitost $O(n^2)$.

Výsledná cena algoritmu tak je $O(n^2) * O(n) = O(n^3)$, což není optimální, jelikož víme že časová složitost ideálního sekvenčního algoritmu by byla $O(n^x)$, kde platí $2 < x < 3$.

1.2 Sekvenční diagram komunikace



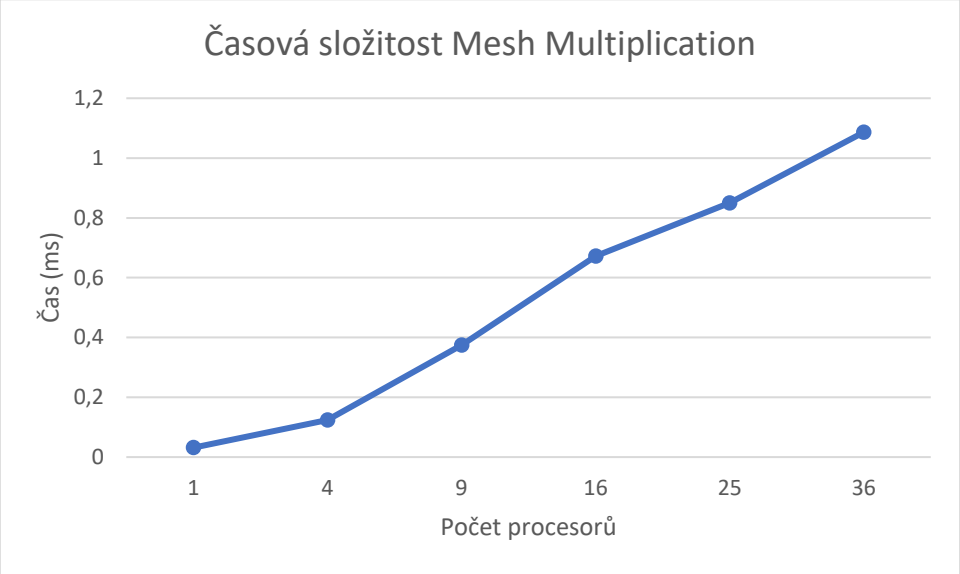
Sekvenční diagram výše znázorňuje komunikaci mezi jednotlivými procesory vedoucí ke stavu, pro přehlednost jsou procesory vyskládány do mřížky. Sloupce jsou vůči sobě výškově posunuty pro naznačení, že trvá nějaký čas, než se k nim dostanou data z předchozích procesorů.

2 Měření časové složitosti

Pro experimenty byl využit server merlin. Měření probíhala opakovaně na náhodně generovaných maticích a výsledky byly zprůměrovány.

Testovány byly čtvercové matice o rozměrech 1×1 až 6×6. Větší matice, pro které by bylo třeba alokovat více procesorů, nelze na serveru vzhledem k omezení prostředků na uživatele zprovoznit.

Pro každý zkoumaný rozměr bylo provedeno 20 měření za pomoci funkce *MPI_Wtime*. Měřen byl pouze čas provádění samotného algoritmu, nikoliv čas zpracování vstupu a výstupu. Průměrné naměřené hodnoty v milisekundách lze vidět na grafu níže. Z experimentů je patrné, že časová složitost algoritmu je dle teoretického předpokladu lineární.



3 Závěr

Algoritmus Mesh Multiplication byl implementován v jazyce *C++* za využití knihovny *OpenMPI*. Následně byl proveden jeho rozbor a analýza složitosti. Lineární časová složitost byla nakonec úspěšně ověřena pomocí experimentálního měření s náhodně generovanými maticemi na serveru merlin.