

Przetwarzanie równoległe

Zadanie domowe III

Jarosław Marek Gliwiński
#indeksu 74839

16 stycznia 2010

1 Wstęp

1.1 Wykaz skrótów i oznaczeń

W pierwszej kolejności przedstawione zostaną używane w pracy oznaczenia, następnie podstawowe definicje miar koniecznych do uzyskania rozwiązania

d liczba wymiarów kostki

p liczba węzłów (procesorów), z definicji hiperkostki równa 2^d

n liczba sumowanych liczb

n_i początkowa liczba liczb w i -tym węźle

T_{sum}, T_{comm} odpowiednio czas sumowania dwóch liczb oraz przesłania liczby. W warunkach zadania $T_{comm} = 10T_{sum}$

$T_{seq}(n), T_{par}(n)$ odpowiednio czas działania algorytmu sekwencyjnego oraz równoległego dla n liczb

S *speed-up* przyspieszenie obliczeń

E *efficiency* efektywność wykorzystania procesorów

C *cost* koszt zrównoleglenia

$$(1) \quad S = \frac{T_{seq}}{T_{par}}$$

$$(2) \quad E = \frac{S}{p}$$

$$(3) \quad C = pT_{par} - T_{seq}$$

1.2 Założenia

Przede wszystkim przyjęto równomierny¹ rozkład liczb w procesorach. Motywacja takiego warunku wynika z faktu, iż w przypadku rozkładu innego niż równomierny niemożliwe byłoby uzyskanie deterministycznych wyników, ponieważ system równoległy pracowałby z parametrami zależnymi od konkretnej wartości wektora losowego $X(\omega) = (X_1(\omega), X_2(\omega), \dots, X_p(\omega))$. Nie jest to jedyny mankament tego podejścia – dodatkowo rozkłady zmiennych X_i *omega* są nieznanne, a same zmienne z pewnością nie są niezależne². Ewentualnie możliwe jest wyznaczenie przypadku teoretycznie najgorszego i najlepszego.

2 Analiza

Dokonamy teraz określenia zależności czasu przetwarzania systemu „hiperkostkowego” od ilości danych wejściowych mierzonej liczbą sumowanych wartości oraz rzędu struktury systemu, tj. liczby wymiarów. Poczynając od systemu jednowymiarowego pokażemy ogólne zależności obowiązujące dla dowolnego $D \in \mathbb{N}$, wykorzystując na każdym etapie wyniki i wnioski pośrednie otrzymane na etapie rozważania kolejnych przypadków.

2.1 Kostka 0D

Przypadek zerowymiarowy jest trywialny – hiperkostka zerowymiarowa jest zdegenerowana do punktu, a zatem nieodróżnialna od systemu jednoprocessorowego. Jedyne przeprowadzane operacje nie różnią się niczym od tych przeprowadzanych w ramach algorytmu sekwencyjnego. Zatem

$$(4) \quad T_{par}^{0D} = T_{seq}$$

2.2 Kostka 1D

Kostka jednowymiarowa przybiera postać odcinka – przedstawiono to na rysunku 1b. Zatem przebieg algorytmu podanego w treści zadania będzie następujący:

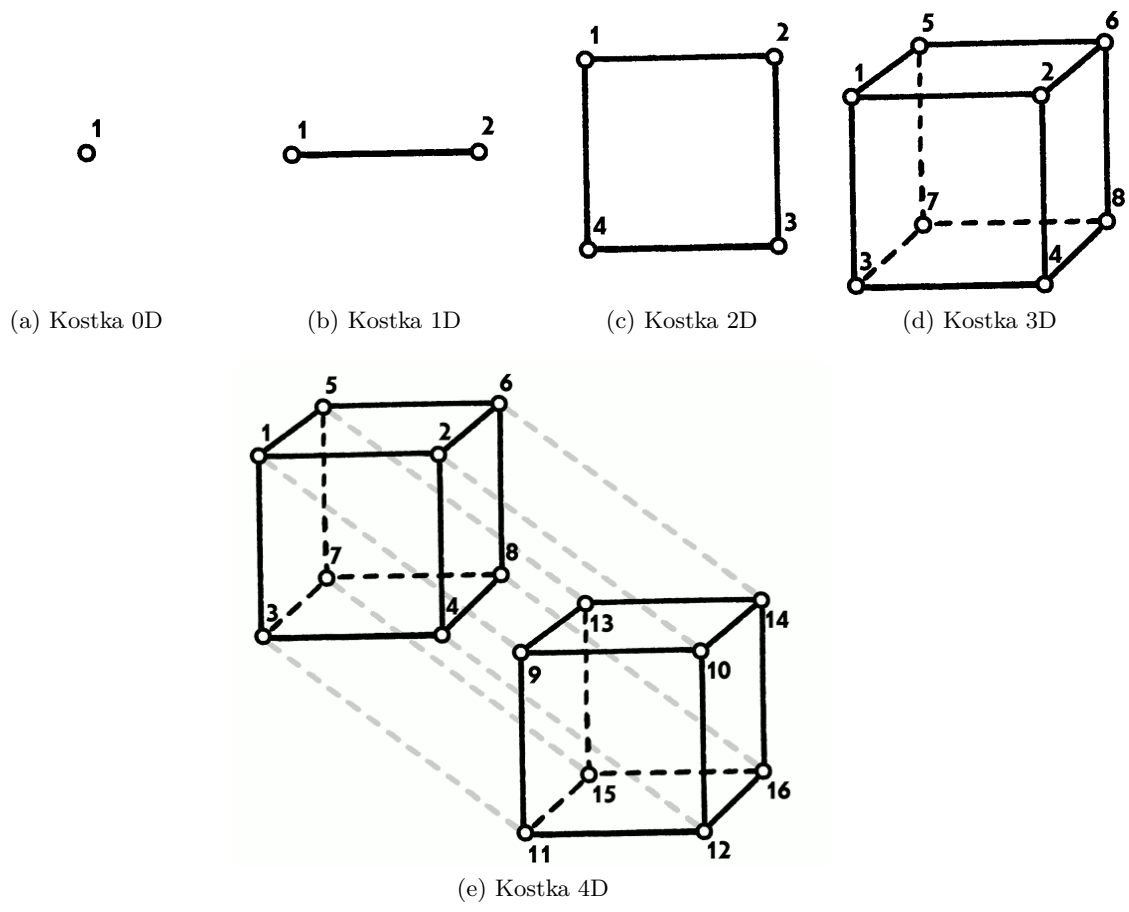
1. Sumowanie liczb w każdym z węzłów, przy czym liczba sumowań w zależności od liczby liczb w węźle n_i wynosi:

$$(5) \quad s(n_i) = \begin{cases} 0 & n_i < 2 \\ n_i - 1 & n_i \geq 2 \end{cases}$$

2. Wymiana sum częściowych między jedynymi dostępnymi sąsiednimi węzłami 1 i 2 w jedynym dostępnym wymiarze – *poziomym*
3. Dodawanie w każdym z węzłów otrzymanej i własnej sumy częściowej

¹z dokładnością ± 1 , w przypadkach, gdy $n \bmod p \neq 0$

² $X_1(\omega) + X_2(\omega) + \dots + X_p(\omega) = n$



Rys. 1: Przykładowe realizacje d -wymiarowej hiperkostki – dla rozważanych wartości d

Kroki 2 i 3 składają się z pojedynczych operacji elementarnych trwających odpowiednio T_{sum} i T_{comm} . Nieco precyzyjniejszego omówienia wymaga krok pierwszy i wynikająca z niego synchronizacja systemu. Mianowicie punkt drugi – równoległe dodawanie sum częściowych – jest możliwy jedynie po zakończeniu dodawania początkowych n_i liczb w każdym z węzłów. W praktyce oznacza to, że krok drugi może rozpocząć się dopiero po czasie $\max_i s(n_i)$

Łatwo zatem wyciągnąć wniosek, że przypadek optymistyczny występuje, gdy liczby są rozdzielone równomiernie pomiędzy węzły a wartość wynikająca ze wzoru $\max_i s(n_i)$ jest minimalna. Przypadkowi pesymistycznemu odpowiadałaby oczywiście sytuacja odwrotna – maksymalne $s(n_i)$ występujące w przypadku agregacji wszystkich liczb w jednym z węzłów. W tym wypadku pierwszy krok byłby powieleniem wykonania algorytmu sekwencyjnego, a w kolejnych krokach następowałaby jedynia propagacja wyniku do wszystkich węzłów³. Dodatkowo możliwe jest kombinatoryczne wyznaczenie przypadku średniego. Jednakże, jako że jak wspomniano wcześniej przyjęto model z rozkładem równomiernym, nie będziemy zajmować się przypadkami innymi niż optymistyczny.

W przypadku równomiernego rozkładu jego równomierność jest rzecz jasna umowna – jak wspomniano wcześniej. Zależność na czas kroku pierwszego zamiast równania przybierze zatem w praktyce prostszą postać $\lceil \frac{n}{p} \rceil$ – co oczywiście sprowadza się do tego samego wyniku dla $n > 2^{d+1}$. Podsumowując, dla przypadku jednowymiarowego uzyskujemy zatem następującą zależność:

$$(6) \quad T_{par}^{1D}(n) = \left(\left\lceil \frac{n}{2} \right\rceil - 1 \right) T_{sum} + T_{comm} + T_{sum}$$

2.3 Kostka 2D

Kostka 2D to struktura o topologii pierścienia składająca się z czterech węzłów – przedstawiono ją na rysunku 1c. Ponownie, pierwszym wykonanym krokiem będzie sumowanie początkowych wartości w każdym z węzłów i synchronizacja w momencie uzyskania sumy częściowej o największej liczbie składników. Następnie nastąpi równoczesna wymiana sum częściowych w wymiarze *poziomym* kostki, a więc sumy częściowe będą równe odpowiednio w węzłach (1, 2) oraz (3, 4). Na tym etapie przetwarzania w grafie systemu możemy wyróżnić dwa obszary według wartości przechowywanych w węzłach. Jeśli narysujemy graf połączeń tych obszarów uzyskamy kostkę jednowymiarową w wymiarze *pionowym*, z jedną liczbą w obydwu węzłach. Odpowiada to sytuacji w systemie, ponieważ operacje, jakie teraz pozostały do wykonania, to wymiana w wymiarze *pionowym* oraz dodanie sum częściowych, które kończy algorytm pozostawiając w każdym z węzłów tę samą wartość wyniku. Jak łatwo zauważyć, skonstruowana przed chwilą *pionowa* kostka 1D pozwala przez wykonanie algorytmu równoległego dokładnie zamodelować te kroki i uzyskać ten sam wynik w tym samym czasie.

Czas działania w tym przypadku sprowadzi się zatem do sumy kroku pierwszego o czasie wykonania $\lceil \frac{n}{4} \rceil T_{sum}$ i czasu wykonania algorytmu równoległego w systemie 1D dla dwóch liczb:

$$(7) \quad T_{par}^{2D}(n) = \left(\left\lceil \frac{n}{4} \right\rceil - 1 \right) T_{sum} + T_{par}^{1D}(2)$$

³plus ewentualne obliczenia polegające na zsumowaniu zera z propagowaną sumą – to już pozostaje kwestią implementacji algorytmu

2.4 Kostka 3D

Ogólnie znaną pod nazwą kostki strukturę kostki trójwymiarowej przedstawia w rzucie na płaszczyznę rysunek 1d. Kostkę tę można po wykonaniu sumowania początkowego i wymiany w wymiarze *głębokości*, nalogicznie do kostki 2D, k *zredukować* do dwóch kostek niższego rzędu – a więc w tym przypadku dwuwymiarowych z czterema liczbami każda. Znamy zatem wszystkie informacje konieczne do sformuowania wzoru:

$$(8) \quad T_{par}^{3D}(n) = \left(\left\lceil \frac{n}{8} \right\rceil - 1 \right) T_{sum} + T_{par}^{2D}(4)$$

2.5 Kostka 4D i przypadek ogólny

Hiperkostkę 4D, o strukturze *tesseractu*, przedstawiono w rzucie na płaszczyznę na rysunku 1e. Niemożność interpretacji przez ludzki wzrok właściwej struktury we właściwej ilości wymiarów przedstawionego grafu czyni jeszcze bardziej oczywistym niż w poprzednich przypadkach spostrzeżenie, iż tesseract *składa się* z dwóch sześciątów. Postępujemy zatem analogicznie jak poprzednio:

$$(9) \quad T_{par}^{4D}(n) = \left\lceil \frac{n}{16} \right\rceil T_{sum} + T_{par}^{3D}(8)$$

Ze względu na symetrię struktur i przekształceń *składania* oczywistym jest już, że ogólna zależność będzie postaci rekurencyjnej:

$$T_{par}^{dD}(n) = \left(\left\lceil \frac{n}{2^d} \right\rceil - 1 \right) T_{sum} + T_{par}^{(d-1)D}(2^{d-1})$$

Rekurencję tę można usunąć zauważając, że w każdym przypadku trzeba wykonać *składanie* hiperkostki do o wymiar mniejszej, a jest to operacja odpowiadająca wymianie sum częściowych i sumowaniu w każdym węźle sumy wymienionej z własną – a więc o stałym czasie wykonania $T_{comm} + T_{sum}$ ⁴. Pozwala to zamienić każde odwołanie rekurencyjne tą właśnie sumą. Rekurencja zatrzymuje się na $d = 0$, zatem ilość tych wywołań będzie równa liczbie wymiarów. Wzór w uproszczonej postaci będzie miał zatem postać

$$(10) \quad T_{par}^{dD}(n) = \left(\left\lceil \frac{n}{2^d} \right\rceil - 1 \right) T_{sum} + d(T_{comm} + T_{sum})$$

co kończy rozważania w tej sekcji i pozwala przejść bezpośrednio do zadanych obliczeń.

⁴ponieważ w ułamku $\left\lceil \frac{n}{2^d} \right\rceil$ zachodzi $n = 2^d$

3 Rozwiązanie

Na podstawie otrzymanych zależności uzyskamy teraz wyniki liczbowe wymagane przez zadanie.

3.1 Kostka 3D, 1000 liczb

Pierwsze polecenie dotyczy obliczeń w systemie o topologii kostki trójwymiarowej dla $n = 1000$ liczb. Sprowadza się to do uzyskania wyników z równań 1–3. Niezbędne do tego wartości, poza wyznaczonym wyżej T_{par} , należą do danych. Dla uproszczenia przyjmujemy $T_{sum} = 1$. Zatem najpierw uzyskujemy czas przetwarzania systemu równoległego:

$$T_{par}^{3D}(1000) = \left(\left\lceil \frac{1000}{8} \right\rceil - 1 \right) + 3 \cdot 11 = 125 - 1 + 33 = 157$$

A następnie na tej podstawie obliczamy wymagane miary efektywności:

$$(11) \quad S = \frac{T_{seq}(n)}{T_{par}^{3D}(n)} = \frac{999}{157} \approx 6.363$$

$$(12) \quad E = \frac{S}{p} = \frac{6.363}{8} = 0.795$$

$$(13) \quad C = pT_{par} - T_{seq} = 8 \cdot 157 - 999 = 257$$

3.2 Porównanie z kostką 4D

Kolejne polecenie dotyczy wymaganego n , które pozwoliłoby uzyskać systemowi 4D uzyskanie efektywności co najmniej takiej samej, jak w przypadku 3D. Sprowadza się to do porównania i sprawdzenia, czy istnieje n' dla którego zachodzi:

$$(14) \quad \frac{\frac{T_{seq}(n)}{T_{par}^{3D}(n)}}{8} \leq \frac{\frac{T_{seq}(n')}{T_{par}^{4D}(n')}}{16}$$

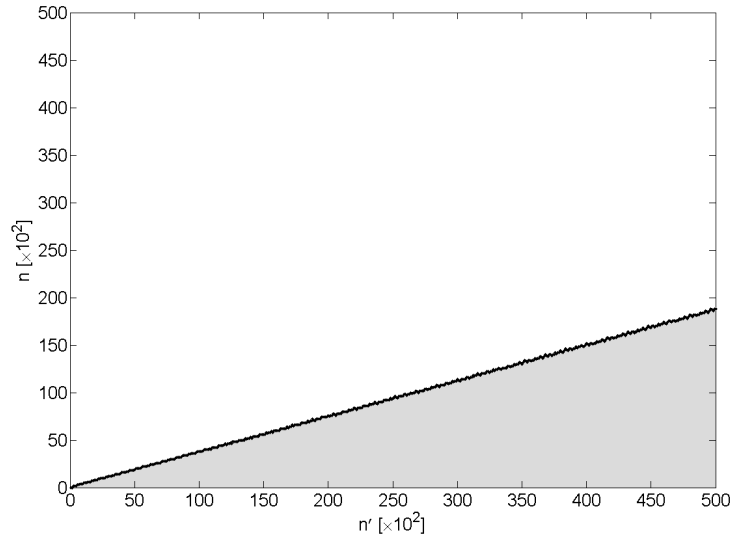
Można rozważyć dwie alternatywne interpretacje nierówności 14:

1. $n = n'$
2. $n \neq n'$

3.2.1 Interpretacja 1

Odpowiada na pytanie, który z systemów efektywniej wykorzystuje potencjalny czas przetwarzania węzłów w rozwiązywaniu tego samego problemu (tych samych danych wejściowych). W tym wypadku nierówność upraszcza się do postaci

$$\frac{2}{T_{par}^{3D}(n)} \leq \frac{1}{T_{par}^{4D}(n)}$$



Rys. 2: Wizualizacja wyników interpretacji 2. na płaszczyźnie $n \times n'$. Kolorem szarym oznaczono obszar, w którym nierówność 14 jest spełniona

A następnie kolejno:

$$2T_{par}^{4D}(n) \leq T_{par}^{3D}(n)$$

$$2 \left\lceil \frac{n}{2^4} \right\rceil - 2 + 88 \leq \left\lceil \frac{n}{2^3} \right\rceil - 1 + 33$$

$$2 \left\lceil \frac{n}{16} \right\rceil + 54 \leq \left\lceil \frac{n}{8} \right\rceil$$

Użycie funkcji sufitu uniemożliwia proste uproszczenie do $2 \frac{n}{16} + 54 \leq \frac{n}{8}$ prowadzące do sprzeczności $54 \leq 0$. Jednakże $\lceil x \rceil - x \in (0, 1)$, a zatem nawet skumulowane różnice w skali wszystkich składników nierówności nie są w stanie wprowadzić zmiany rzędu wielkości odpowiedniego do zmiany nierówności w niesprzeczną.

3.2.2 Interpretacja 2

Odpowiada na pytanie, jak dużo większy zbiór danych należy zapewnić w systemie 4D, aby jego efektywność co najmniej dorównała systemowi 3D. W tym wypadku konieczne jest rozważenie niezmienionej nierówności 14. Wobec występowania weń dwóch niewiadomych najprościej jest wykreślić na płaszczyźnie $n \times n'$ punkty, w których lewa i prawa strona są równe. Wykres wykonano numerycznie w środowisku Matlab, jest widoczny na rysunku 2. Wyznaczono także współczynniki regresji liniowej – współczynnik kierunkowy krzywej wyniósł około 0.375. Oznacza to w praktyce, że system czterowymiarowy osiąga efektywność porównywalną z trójwymiarowym dla zbiorów danych około 2.667 razy większych. W szczególności oznacza to, że dla $n = 1000$ wymagane jest $n' \approx 2667$. Potwierdza się to w sprawdzeniu za pomocą obliczeń numerycznych – choć nie sposób tu uzyskać precyzyjną wynikającą z funkcji regresji odpowiedź ze względu na występowanie funkcji sufitu we wzorze 10.