

Metody Programowania Równoległego

Oscar Teeninga
Tomasz Markiewicz

Zbadać poprawność danych wejściowych dla założeń sortowania kubełkowego

Poprawność danych oznacza, że dane będą rozproszone równomiernie dla każdego przedziału odpowiadającemu każdemu kubełkowi. Założyłem wstępnie, że tablica zawiera wartości z przedziału 0-1000. Sprawdziłem błąd(maxdiff), który oznacza największą różnicę między spodziewaną liczbą elementów z danego przedziału do rozmiaru tablicy oraz realną liczbą elementów z danego przedziału do rozmiaru tablicy. Dla przykładu, mając 100 liczb oraz 3 kubełki, spodziewamy się, że jeden będzie zawierał 0.33 elementów, drugi i trzeci też 0.33 elementów. Realnie założmy, że jeden kubełek zawierał 0.4 elementów, a drugi i trzeci 0.3 elementów. Oznacza to, że największa różnica to $0.4 - 0.33 = 0.07$. Natomiast błąd względny (error) to jest stosunek maxdiff do spodziewanego stosunku elementów do rozmiaru tablicy. W poprzednim przykładzie byłoby to $0.07 / 0.33$. Zgodnie z wynikami testu zauważyć można, że im większy rozmiar problemu tym lepsze rozłożenie osiągamy. Ciekawy jest również fakt, że ze względu na to, że rozmiar tablicy dzieli się przez 2, to parzysta liczba kubełków daje lepsze efekty. Przy wzroście liczby kubełków rośnie również błąd, co jest spodziewane, ponieważ jest większe prawdopodobieństwo niedokładnego losowania oraz kubełki nie są równe między sobą (1000 nie dzieli się przez 16, 15, 14, 13, 12, 11, 9, 7).

Czy będzie możliwe dotrzymanie tych założeń? przy jakich warunkach?

Osiągnięte błędy względne nie przekraczają 2%, a nie planujemy zwiększać liczby kubełków powyżej 16, co oznacza, że bez problemu możemy założyć, że kubełki są równomiernie rozłożone. Warunkami jest jedynie odpowiedni stosunek przedziału do liczby kubełków. W całym laboratorium założyliśmy przedział 0-1000, który zapewnia, że dla 16 kubełków błędy wynikające z dyskretnych przedziałów liczby typu całkowitego są praktycznie pomijalne (1000 nie dzieli się przez 16, a więc kubełki nie są między sobą identyczne).

Rozmiar problemu 100000

size	buckets	maxdiff	error
1000000	1	0,000000000	0,000000000%
1000000	2	0,000325000	0,065000000%
1000000	3	0,000598333	0,179500000%
1000000	4	0,000899000	0,359600000%
1000000	5	0,000614000	0,307000000%
1000000	6	0,000930667	0,558400000%
1000000	7	0,001703143	1,192200000%
1000000	8	0,000593000	0,474400000%
1000000	9	0,000621111	0,559000000%
1000000	10	0,000482000	0,482000000%
1000000	11	0,001387091	1,525800000%
1000000	12	0,000981333	1,177600000%
1000000	13	0,001230077	1,599100000%
1000000	14	0,000890571	1,246800000%
1000000	15	0,001111667	1,667500000%
1000000	16	0,000812000	1,299200000%

Rozmiar problemu 1000000

size	buckets	maxdiff	error
10000000	1	0,000000000	0,000000000
10000000	2	0,000402300	0,000804600
10000000	3	0,000416333	0,001249000
10000000	4	0,000394200	0,001576800
10000000	5	0,000152000	0,000760000
10000000	6	0,000959867	0,005759200
10000000	7	0,001103543	0,007724800
10000000	8	0,000300800	0,002406400
10000000	9	0,000341411	0,003072700
10000000	10	0,000223900	0,002239000
10000000	11	0,001129391	0,012423300
10000000	12	0,000499433	0,005993200
10000000	13	0,001093777	0,014219100
10000000	14	0,000646271	0,009047800
10000000	15	0,000797367	0,011960500
10000000	16	0,000669400	0,010710400

