Architektury systemów komputerowych

Ćwiczenia 7: "Architektury równoległe i rozproszone"

Należy przygotować się do zajęć czytając następujące rozdziały książek:

- *Computer Design and Organization: The Hardware/Software Interface* (5 edycja); 6.2 – 6.6
- *Computer Architecture: A Quantitative Approach* (4 edycja)

Należy być przygotowanym do wytłumaczenia **wytłuszczonych** haseł.

Zadanie 1

Wyjaśnij prawo *Amdahla*. Rozpatrzmy algorytm wyszukiwania binarnego, który znajduje element o wartości X w posortowanej N-elementowej tablicy A i zwraca jego indeks:

```
BinarySearch(A[0..N-1], X) {
   low = 0
   high = N -1
   while (low <= high) {
        mid = (low + high) / 2
        if (A[mid] >X)
           high = mid -1
        else if (A[mid] <X)
           low = mid + 1
        else
           return mid // found
   }
   return -1 // not found
}
```

Przyjmij, że posiadasz maszynę z K procesorami.

- Zakładając, że K jest dużo mniejsze niż N, jakiego przyśpieszenia możemy się spodziewać w stosunku do wersji jednoprocesorowej? Narysuj wykres zależności między K i N.
- Jak zmieni się Twoja odpowiedź jeśli K = N ?

Zadanie 2

Zapoznaj się z taksonomią Flynna. Wytłumacz sposób przetwarzania danych w architekturze: **`SISD`**, **`SIMD`**, **`MISD`**, **`MIMD`**. Jakie potrzeby obliczeniowe uzasadniają ich stosowanie? Dla każdej z wyżej wymienionych architektur podaj rzeczywistą implementację.

Zadanie 3 [§4.5]

Wytłumacz w jaki sposób instrukcje `LL` i `SC` procesora mogą posłużyć do synchronizowania wątków wykonywanych w systemach **wieloprocesorowych z pamięcią współdzieloną**. W jaki sposób można zaimplementować te dwie instrukcje w systemie z protokołem synchronizacji pamięci podręcznej? Upewnij się, że Twoje rozwiązanie będzie odporne na **problem ABA**.

Zadanie 4 [§4.6]

W systemach wieloprocesorowych z pamięcią współdzieloną problemem staje się określenie semantyki operacji na pamięci. Wyjaśnij różnice między **spójnością sekwencyjną** (ang. *sequential consistency*), **porządkiem liniowym na zapisach** (ang. total store order), **porządkiem częściowym na zapisach** (ang. *partial store order*) i **spójność zwalniania** (ang. *release consistency*). W jakim celu udostępnia się programistom specjalne instrukcje zwane **barierami pamięciowym** (ang. *memory barrier*) ?

Zadanie 5 [§3.5]

Czym różni się **wielowątkowość drobnoziarnista** od **gruboziarnistej**? Rozpatrzmy architekturę out-of-order (Tomasulo) realizującą sprzętową wielowątkowość (ang. simultaneous multithreading). Które za zasobów procesora są prywatne. a które współdzielone między sprzętowe? Jakie są wady i zalety architektury SMT ?

Zadanie 6

Raport techniczny The Landscape of Parallel Computing Research: A View from Berkeley definiuje pewne klasy problemów obliczeniowych stanowiących wyzwanie dla projektantów systemów komputerowych i języków programowania. W jaki fundamentalny sposób różnią się od siebie, z punktu widzenia ASK, następujące problemy: mnożenie macierzy, znajdowanie najkrótszej ścieżki w grafie, kompresja tekstu.

Zadanie 7

Wyjasnij architekturę ***CC-NUMA*** (ang. *cache coherent non-uniform memory architecture*) z użyciem diagramu. W takim systemie koszt dostępu do **nielokalnej pamięci** może poważnie redukować zysk z posiadania wielu procesorów. Załóżmy, że czas trwania zapisu do pamięci lokalnej to 20 cykli; do pamięci nielokalnej to 100 cykli; mamy proces w którym częstości zapisów do pamięci lokalnej i nielokalnej wynoszą 50%. Pomijamy rozważanie odczytów. Instrukcje zapisu nie posiadają zależności danych, nie musimy czekać na ich zakończenie i pojawiają się równomiernie w trakcie wykonania naszego programu. W danej chwili pamięć może odpowiadać tylko na jedno żądanie dostępu. Jaki wpływ ma na czas działania naszego programu fakt, że odwołujemy się do pamięci średnio co (a) 50 cykli (b) 100 cykli.

Zadanie 8

Rozważmy architekturę GPGPU[1]. Czym różni się model przetwarzania danych przez procesor z **instrukcjami wektorowymi** i przez pojedynczy procesor SIMT (ang. *Single Instruction Multiple Threads*) karty graficznej? W jaki sposób GPGPU radzi sobie z opóźnieniami związanymi z oczekiwaniem na sprowadzenie danych z pamięci? Co dzieje się w przypadku wyrażeń warunkowych `if-then-else`, jeśli dla części wątków warunek jest niespełniony?

Krystian Bacławski

[1] http://courses.cs.washington.edu/courses/cse471/13sp/lectures/GPUsStudents.pdf