ASK: Lista 14

$x_{14} = 6$

Większość materiału przerabianego na tej liście jest omówione w książce "*Computer Organization and Design: The Hardware / Software Interface*" (wydanie piąte) w rozdziale piątym i szóstym.

Zadanie 1

Wyjaśnij prawo Amdahla. Rozpatrzmy algorytm wyszukiwania binarnego (klasyczne dziel i zwyciężaj), 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 w/w architektur podaj rzeczywistą implementację.

Zadanie 3

Zapoznaj się z rozdziałem 3 i 4 dokumentu "The *MIPS32® SIMD Architecture Module*"[1]. Jaka jest ilość, szerokość i organizacja rejestrów koprocesora wektorowego? Wymień kategorie instrukcji i wytłumacz ich przeznaczenie. Wyjaśnij szczegółowo działanie instrukcji `ADDS_U`, `PCNT` i `VSHF`.

Zadanie 4

Używając jedynie instrukcji *MIPS32® SIMD Architecture* napisz w asemblerze makroinstrukcję "`SORTPAIR.B wd, ws`" o następującym znaczeniu:

```
for i in 0 .. WRLEN/16-1
  WR[wd]_{16i+7..16i}   ← min(WR[ws]_{16i+7..16i}, WR[ws]_{16i+15..16i+8})
  WR[wd]_{16i+15..16i+7} ← max(WR[ws]_{16i+7..16i}, WR[ws]_{16i+15..16i+8})
```

Innymi słowy chcemy posortować sąsiadujące pary bajtów w słowie. Notacja semantyki instrukcji jest zaczerpnięta bezpośrednio z dokumentu z poprzedniego zadania. **Uwaga!** Nie można używać instrukcji sterowania i skalarnych.

---

[1] Do ściągnięcia tu: http://cahirwpz.cs.uni.wroc.pl/teaching-pl/architektury-systemow-komputerowych-lato-2014.

Zadanie 5

Zdecydowana większość współczesnych komputerów – np. PC, smartfony, konsole – bazuje na architekturze `SMP` (ang. shared memory multi-processor). Przedstaw na diagramie poszczególne elementy takiej architektury tj. procesory, pamięci podręczne, szyny, pamięć główną, urządzenia. Porównaj architekturę SMP z UP (ang. uni-processor) – w jaki sposób procesory komunikują się ze sobą i pozostałymi komponentami? Jakie problemy tu zauważasz? Co jest wąskim gardłem w tej architekturze (wyjaśnij to pojęcie)?

Zadanie 6

Załóżmy, że posiadamy proces składający się z wielu wątków **wspólnie modyfikujących** te same dane. Rozmiar danych przekracza pojemność pamięci podręcznych co najmniej dwa rzędy wielkości. Porównajmy maszynę `SMP` z `N`-procesorami i maszynę `SMT` (ang. *simultaneous multithreading*) z $\frac{n}{4}$-procesorami 4-wątkowymi. Która z nich ma szansę być w tym przypadku wydajniejsza? Dlaczego? Wyjaśnij zjawisko fałszywego współdzielenia (ang. *false sharing*).

Zadanie 7

W maszynach wieloprocesorowych znaczącym problemem jest zbudowanie wydajnych pamięci podręcznych – w szczególności utrzymanie spójności (ang. *cache coherence*). Dlaczego? Znajdź informacje nt. dwóch mechanizmów utrzymywania spójności – podglądanie szyny, katalog informacji o wierszach – i porównaj je. Który z nich się nie skaluje (wyjaśnij to pojęcie) ? Narysuj diagram stanów protokołu `MESI` i wyjaśnij jego działanie.

Zadanie 8

Rozważmy architekturę `CC-NUMA` (ang. *cache coherent non-uniform memory architecture*). Wyjaśnij ją 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 9

Popularną metodą zrównoleglania pracy w architekturach rozproszonych jest `MapReduce`. Wyjaśnij na czym ona polega? Załóżmy, że posiadamy klaster setek komputerów przechowujący miliony dokumentów tekstowych. Jak z użyciem tej metody można zliczyć częstość występowania danego słowa? Jakie warunki musi spełniać nasze zadanie, by można je było efektywnie przetwarzać za pomocą metody `MapReduce`?

Zadanie 10

Rozważmy podstawy architektury GPGPU[2]. 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?

---

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