HW6

Zhijiia Chen

November 26, 2018

4.10 For the vector computer, the memory acess of the vector portion of codes requires $\frac{200MB+100MB}{20GB/s} = 10ms$, and the execution of scalar portion of codes requires 400 ms. Suppose that vector computations are overlapped with memory acess, then the total time for the vector computer = 410 ms.

For the hybrid computer, the memory acess of the vector portion of codes requires $\frac{200MB+100MB}{150GB/s} = 2ms$, and the execution of scalar portion of codes requires 400 ms, the I/O transering data from host memory to GPU local memory requires $\frac{200MB+100MB}{10GB/s} = 30ms$. Suppose that GPU vector memory access is overlapped with memory transering, then the total time for the vector computer = 430 ms.

4.13

a. $1.5GHz \times 0.85 \times 0.8 \times 0.7 \times \frac{1}{4} \times 32 \times 10 = 57.12$ GFLOP/s.

b. **(1)** By increasing the number of lanes to 16, each SIMD instruction can produce 32 results in 2 cycles. So the improved throughput $= 1.5GHz \times 0.85 \times 0.8 \times 0.7 \times \frac{1}{2} \times 32 \times 10 = 114.24$ GFLOP/s, and the speedup is 2.

(2) The improved throughput $= 1.5GHz \times 0.85 \times 0.8 \times 0.7 \times \frac{1}{4} \times 32 \times 15 = 85.68$ GFLOP/s, and the speedup is 1.5.

(3) The improved throughput $= 1.5GHz \times 0.95 \times 0.8 \times 0.7 \times \frac{1}{4} \times 32 \times 10 = 63.84$ GFLOP/s, and the speedup is 1.118.

4.14

b. True dependences:
S2 depends on S1 through A[i].
S4 depends on S3 through A[i].
Output dependences:
WAW hazard between S1 and S3.
Antidependences:
WAR hazard between S3 and S4.
Rewritten codes:

```
for (i = 0; i < 100; i++)
{
    A[i] = A[i] * B[i]; /*S1*/
    B[i] = A[i] + c; /*S2*/
    E[i] = C[i] * c; /*S3*/
    F[i] = D[i] * E[i]; /*S4*/
}
```

c. There are dependences between S1 and S2 in iteration i and i + 1 on B, so this loop is not parallel. To make it parallel, we can rewrite the code as following:

```
for (i = 0; i < 100; i++)
{
    B[i+1] = C[i] * D[i]; /*S2*/
}

for (i = 0; i < 100; i++)
{
    A[i] = A[i] * B[i]; /*S1*/
}
```

The second loop must run after the first loop, and all iterations in each loop can run in parallel.

4.16 Peak throughput $= 1.5 \times 16 \times 16 = 384$ GFLOPS/s. Suppose each single precision operation unit operates on 2 four-byte operands, and outputs one four-byte result, then the throughput requires 12 bytes/FLOP × 384 GFLOPS/s = 4.608 TB/s of memory bandwidth, which is not suitable for the given bandwidth.