Homework 1 Solution

Parallel and Vector Architectures

1.3 Let *s* be the fraction of work that must be done sequentially and T_1 is the sequential execution time. Then, the sequential work (T_1 *s*) takes the same time on a multiprocessor as on a uniprocessor. If the remainder $T_1(1\text{-}s)$ is perfectly parallelized on *p* processors, it will be accelerated by a factor of *p*. If there is redundant work performed, communication, or synchronization, the parallel execution time will increase and the speedup will be reduced from the ideal case. This is why the formula for speedup is not exact, but is only an upper bound.

$$T_p \geq T_1\left(s + \frac{1-s}{p}\right) \quad S_p = \frac{T_1}{T_p} \leq \frac{1}{s + \frac{1-s}{p}}$$

1.4 Let f_i be the fraction of cycles on an ideal machine in which *i* instructions can issue in parallel. T_{ideal} is the ideal execution time, $T_{1\text{-}issue}$ is the execution time on a *1-issue* processor. This is the total amount of work done. $T_{k\text{-}issue}$ is the execution time on a *k-issue* processor. To compute $T_{k\text{-}issue}$, we view all issue slots with no more than *k* instructions as taking a single cycle and take all the work in slots with more than *k*-way parallelism and doing it *k* instructions at a time.

$$T_{1-issue} = T_{ideal}\left(f_0 + \sum_{i=1}^{\infty} f_i \times i\right)$$

$$T_{k-issue} \geq T_{ideal}\left(\sum_{i=0}^{k} f_i + \frac{1}{k}\sum_{i=k+1}^{\infty} f_i \times i\right)$$

$$S_k \leq \frac{f_0 + \sum_{i=1}^{\infty} f_i \times i}{\sum_{i=0}^{k} f_i + \frac{1}{k}\sum_{i=k+1}^{\infty} f_i \times i}$$

1.8 Ignoring the boundary element cases, with a per-element decomposition there are 4(neighbors)×1024×1024(elements)×8(bytes per element) = 32 MB of data communicated per step.

The most efficient decomposition is into blocks of 128×128 elements each. Then, ignoring the boundary cases, there are 4(neighbors)×128(elements communicated to each neighbor by each processor)×64(processors) ×8(bytes per element) = 256 KB of data communicated per step.

1.16 At 100 MIPS and 1% instruction cache miss, 1 Million instructions cause cache misses each second. Each instruction miss causes a block (32 bytes) to be transferred. Therefore, 32 Million bytes of code have to be transferred to the instruction cache each second, consuming a bus bandwidth of 32MB/s.

20% of instructions are loads and 10% are stores. This means that 20 Million loads and 10 Million stores are executed per second. The data cache miss rate is 5%, which implies that 1 Million loads and 0.5 Million stores cause cache misses. 1 block is fetched for a load miss and 2 blocks are fetched for a store miss. Therefore, 32 Million bytes are fetched for load misses and another 32 Million bytes for store misses. A total of 64 Million bytes are transferred from/to data cache per second.

Total bus traffic is 32 MB/s (instruction) + 64 MB/s (data) = 96 MB/s per processor.

With a 250 MB/s bus, we can have only one processor below the 125 MB/s, which is 50% of peak bus bandwidth.