Flynn Taxonomy

The Flynn Taxonomy divides methods of computation into four categories. For each category provide an example of an application or declare it uncommon. You should only declare categories uncommon if they are almost never used.

1. SISD (Single Instruction Single Data): The model you are most familiar with - each instruction is executed in order, and acts on a single data stream.

   Just about any program they've written that wasn't written for hadoop

2. SIMD (Single Instruction Multiple Data): Each instruction is executed in order, but may act on multiple data at once (e.g. compute four multiplications simultaneously).

   SSE Intrinsics

3. MISD (Multiple Instruction Single Data): Multiple instructions are executed simultaneously, but all act on the same data stream.

   uncommon

4. MIMD (Multiple Instruction Multiple Data): Multiple Instructions are executed simultaneously, and act on multiple data at once.

   map reduce, mutliple thread programs

Amdahl's Law

Amdahl's law states that

$$\text{speedup} = \frac{1}{(1-F) + \frac{F}{S}}$$

where F is the fraction of the program sped up, and S is the speedup on that fraction. This may seem arbitrary, but is in fact fairly easy to derive. Do so, starting from the definition of speedup.

Let T be the time needed to execute the program initially.

$$\text{speedup} = \frac{\text{time before}}{\text{time after}} = \frac{T}{(1-F)T + F \cdot \frac{T}{S}} = \frac{1}{(1-F) + \frac{F}{S}}$$

AMAT

AMAT stands for Average Memory Access Time. It refers to the time necessary to perform a memory access on average. It does *NOT* refer to the time necessary to execute an instruction which accesses memory. Those of you in CS70 may find it helpful to think of AMAT as $\mathrm{E}(\tau)$, where τ is a random variable whose value is the time necessary to perform a given memory access. The AMAT of a simple system with only a single level of cache may be calculated as

$$\text{AMAT} = \text{hit time} + \text{miss rate} \times \text{miss penalty}$$

This formula can be extended to more complicated memory hierarchies by replacing miss penalty with the AMAT for the next level in the memory hierarchy. That is

$$\text{AMAT}_i = \text{hit time}_i + \text{miss rate}_i \times \text{AMAT}_{i+1}$$

where i refers to the level of the memory hierarchy.

1. Suppose you have a system with the following properties:

   - L1$ hits in 1 cycle (local hit rate 75%)
   - L2$ hits in 20 cycles (local hit rate 80%)
   - L3$ hits in 100 cycles (local hit rate 90%)
   - Main memory hits in 1000 cycles (always hits)

   Calculate the AMAT.

   $$\text{AMAT} = 1 + 0.25(20 + 0.20(100 + 0.10(1000 + 0))) = 16 \text{ cycles}$$

2. Suppose you have a system with the following properties:

   - L1$ hits in 1 cycle (global miss rate 25%)
   - L2$ hits in 20 cycles (global miss rate 20%)
   - L3$ hits in 100 cycles (global miss rate 10%)
   - Main memory hits in 1000 cycles (always hits)

   Calculate the AMAT.

   $$\text{AMAT} = 1 + 0.25(20) + 0.20(100) + 0.10(1000) = 126 \text{ cycles}$$