Processing of Control Transfer Instructions

Tópico 4

Instruction-level Parallel Processing

Tópico 2 : Exploiting Instruction-level Parallelism (ILP) – An Overview.

Tópico 3 : Instruction-level Parallel Processors How architectures exploit ILP.

Tópico 4 : *Processing Control Transfer Instructions.*

Tópico 5 : Code Scheduling for Instruction-level Parallel Processors.

Processing of Control Transfer Instructions

Branch instructions seem to be the most straightforward instruction type for a processor, since there is nothing more to do than modify, conditionally or unconditionally, the value of the program counter (PC). However, this easy-to-perform operation has turned out to be one of the most serious obstacles to increasing the performance of ILP-processors.

Processing Branch Instructions

- Types and checking of results – The branch problem;
- Branch statistics and performance;
- Basic approaches to branch handling;
- Branch processing – prediction and speculation;
- Multiway branching;
- Guarded execution;

Hardware-based Speculation

Hardware-based speculation combines three key ideas:

1. dynamic branch prediction to choose which instructions to execute;

2. speculation to allow the execution of instructions before control dependencies are resolved; and

3. dynamic scheduling to deal with the scheduling of different combinations of basic blocks.

Advantages of Hardware-based Speculation

Some of the advantages of hardware-based speculation over software-based version include the following:

- To speculate extensively, one must be able to disambiguate memory references. This is difficult to do at compile-time, for example, for integer programs that contain pointers. In a hardware-based scheme dynamic runtime disambiguation of memory addresses is done using techniques such as *load/store reordering.*

Advantages of HW Speculation$_{(cont)}$

- Hardware-based speculation is better when hardware-based branch prediction is superior to software-based branch prediction done at compile time. This is true for many integer programs. Note that speculative instructions may slow down computation when the prediction is incorrect.
- Hardware-based speculation maintains a completely precise exeception model even for speculative instructions.
- Hardware-based speculation does not require compensation or bookkeeping code.

Advantages of HW Speculation$_{(cont)}$

- Hardware-based speculation with dynamic code scheduling does not require different code sequences to achieve good performance for different implementations of an architecture. Compiler-based speculation and scheduling often require code sequences tuned to the machine, and older or different code sequences can result in much slower performance. In contrast, while hardware speculation and scheduling can benefit from scheduling and tuning processors, using the hardware-based approaches is expected to work well even with older or different code sequences.

Against these advantages stands a major disadvantage: supporting speculation in hardware is complex and requires substantial hardware resources.