Branch Prediction

CS441 System Architecture
Jeffrey Waldron
Caleb Hellickson
What is Branch Prediction?

- Given a tree, try to predict how many branches it will grow as well as the leave fall pattern
What is a branch?

- An instruction that may cause the compiler to start the execution of a different instruction sequence.
What is a branch?

● Types of branch instructions:
  ○ Unconditional: Always results in a branch
  ○ Conditional: Doesn’t always result in a branch

● Compiler fetches instructions in sequence.

if (unconditional branch instruction or conditional satisfied)
{
    Fetch from different instruction sequence;
}
What is a branch?

if(conditional branch instruction not satisfied)
{
    Keep fetching in same sequence;
}

Common branch instructions

- Unconditional: GOTO!

- Conditional: Any of the “Jumps ifs” (CS301)
  - JZ, JNZ
  - JO, JNO
  - JL, JLE
  - etc....
What is a branch predictor?

- A circuit designed to predict which way a branch will go before it gets there.
- Without this, the processor would have to wait for a conditional jump instruction to finish executing before it could fetch the next instruction.
What is a branch predictor?

- Branch most likely taken is speculatively executed
  - If the predictor is wrong, associated instructions are thrown out
  - Pipeline starts again with correct branch (delay).
  - Delay between 10 and 20 clock cycles.

- Let’s see some approaches...
Static Prediction

- Simplest
  - Looks at just the instruction, not the history

- SPARC and MIPS always said conditional was not taken

- Other forms:
  - Backward branches will always be taken
    - Target address is always lower
Next Line Prediction

- Fetching instructions with pointer to next line

- When groups are pointing to:
  - 2, 4, or 8 instructions, branch is not likely the first instruction so initial instructions fetched are wasted

- Since branch will not likely be last instruction in group, instructions after branch are discarded.
1 Bit Prediction

- Does what was done before
- Saves in a 1 Bit field
- So easy a caveman could do it
2 Bit / Saturating

- State Machine
- When branch evaluated, state machine updated
- Must deviate twice before new prediction
2 Level/adaptive

- Remembers history of last $n$ occurrences and uses a saturated counter for $2^n$ history patterns

- Branch history register responsible for choosing which of possibilities to take.

- Great for patterns
  - Ex: Every 3rd condition is true
Local Branch Prediction

- Separate history buffer for each conditional jump instruction
- Uses a 2 Level/Adaptive table
- Combines the two to prediction
Global Branch Prediction

- Shared history for all conditional jump instructions

- Good:
  - Any correlation is used for prediction

- Bad:
  - Saturated with irrelevant data
  - History table must be extensive for good prediction
Alloyed Branch Prediction

- LBP+GBP = ABP
- Concatines LBG and GBP histories
Agree Predictor

- 2 Level/Adaptive Predictor + Globally Shared History Buffer = AP

- Outputs of local and global predictors are XOR’d to produce prediction

- Reduces contentions when opposite productions share same entry in history table
Hybrid Predictor

● Implements multiple predictors and takes a vote!

● Benchmarks say about as good as local predictor
Loop Prediction

● For Loops
  ○ Condition at bottom of loop repeated $N$ times will be taken $N-1$ times
  ○ If at top of loop taken $N-1$ times and then once

● Most microprocessors today have these
Natural Branch Prediction

- Ability to exploit long histories with linear growth instead of exponential

- Advantages
  - 5.7% improvement over global predictor

- Disadvantages:
  - High Latency
Jeff’s Super Awesome Idea of Greatness

● Branch? Copy Machine w/ 2 processors
  ○ 100% accuracy w/ no slow down!

● Pipelining, means possible 4, 8, or 16 processors

● Don’t take anything beyond 4 perhaps?
  ○ Not 100% anymore :(
References

Branch Photo: http://www.bhmpics.com/view-autumn_maple_tree_branches-normal.html

Branches and Branch Prediction:
http://en.wikipedia.org/wiki/Branch_%28computer_science%29
http://en.wikipedia.org/wiki/Branch_predictor