

Accurate Branch Prediction for Short Threads

Bumyong Choi, Leo Porter, Dean M. Tullsen

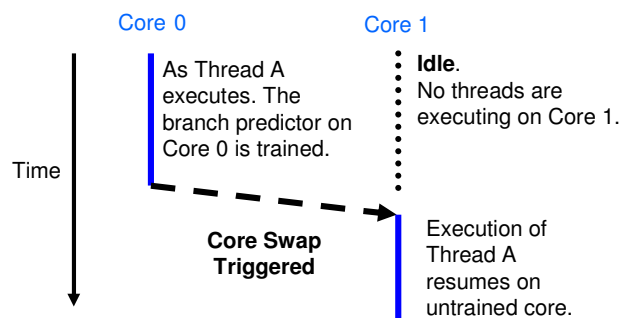
Two Conflicting Trends

The advent of Single Chip Multiprocessors (CMPs) has made it easier to transfer a thread between cores. Many recent proposals to improve performance, power, and/or reliability take advantage of this property causing the duration of a thread on a core to become *shorter*.

To increase branch prediction accuracy, modern branch predictors are using increasingly *longer* histories.

Hence, modern branch predictors are optimized for long threads and do not handle short threads well.

Short Threads Impact Prediction Accuracy



Recent Proposals that use Short Threads

Speculative Multithreading – Thread level parallelism is leveraged to improve performance by executing speculative threads on idle cores.

Dynamic Compilation – Idle cores are used to perform runtime compiler optimizations to improve performance.

Heterogeneous CMPs – Threads at different phases of execution migrate between cores with different resources to improve performance or power usage.

Thermal Scheduling – Threads are migrated between cores to reduce thermal stresses and power consumption.

Global History

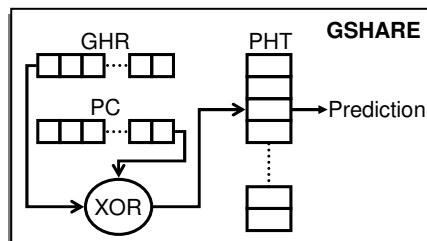
Global History – The outcome of every branch is shifted into a register called the **Global History Register (GHR)**. That register is then used to produce branch predictions.

Code Example where global history is essential

```
Do:
int A = random(0, 9) // A is between 0 and 9
If (A>4) // branch alternates with no predictability
...
If (A<4) // branch is highly correlated to branch above
... // with global history it can be predicted accurately
i++;
While (i<100000)
```

Global History is essential for accurate branch prediction. The following modern branch predictors use Global History:

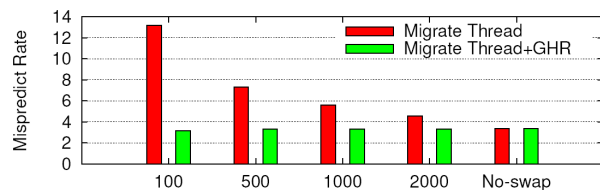
- GSHARE
- Perceptron
- 2Bc-gskew
- Alpha 21264
- YAGS
- O-GEHL
- L-TAGE
- PPM



The GHR is the Key!

Both predictor state and the GHR value matter. However, in time, the predictor state is learned and only the GHR matters.

To test this hypothesis we migrated a thread from one core to another after a fixed number of instructions



The results above show that just migrating the GHR, and not any additional predictor state, is all that is required to improve the branch prediction accuracy.

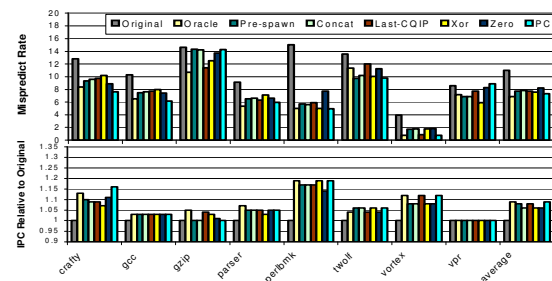
How to construct a meaningful GHR

In Speculative Multithreading, the correct GHR is **not known when the thread is created so it cannot be transferred**. Two categories of techniques for GHR generation were examined with novel policies proposed.

1. Attempt to construct the correct GHR. Policies include: Pre-spawn, Concat, Last CQIP
2. Attempt to provide a consistent starting point for new threads. Policies include: Zero, PC, XOR

These policies were compared against the oracle (real) GHR value from the future (and hence unknowable by hardware).

Results



These results show that the pc policy performs as well as an unknowable oracle policy

Contributions

1. In the presence of frequent core-swaps, providing a useful GHR is the only requirement for accurate branch predictions.
2. Using the program counter of the spawn-triggering instruction for the GHR provides highly accurate branch predictions. **This simple and free technique reduces branch mispredicts by 29% and improves IPC by as much as 13% for select SPEC2000 benchmarks.**