Cache Pirating: Measuring the Performance Impact of Cache Sharing

Nikos Nikoleris, David Eklov, David Black-Schaffer and Erik Hagersten

Overview
Cache Pirating is a low overhead method for measuring application performance and off-chip bandwidth as a function of the available shared cache capacity on real hardware.

The Target application is co-run with a Pirate application that steals shared cache. The Pirate steals the desired amount of cache to control how much the Target gets.

Cache Pirating uses performance counters to measure the Target’s behavior. Any performance metric available on the hardware can be measured.

Application Insights from Cache Pirate Data

1. Miss Ratio \approx Fetch Ratio \rightarrow prefetching is not effective
2. Bandwidth and Miss Ratio \rightarrow 10x increase
3. But CPI constant \rightarrow insensitive to latency

2. LBM

The Pirate steals the desired amount of cache

The Target runs with the remaining cache space

Performance counters are used to measure the Target’s behavior

Bandwidth consumption?

Discussion

We run up to four instances of the same application on a quad core Nehalem with 8MB of shared cache
- Each instance takes equal portion of the cache.
- Naively expect: throughput to scale perfectly.
- Measured data: show that shared caches and bandwidth have a great impact on performance (not surprising).
- With Cache Pirating we can accurately predict the performance impact due to cache sharing on real hardware with 5.5% overhead.

Department of Information Technology, Uppsala University
http://it.uu.se/