Hybrid Data Prefetchers

Inspired by Slim AMPM by Young et al. and Variable Length Delta Prefetcher (VLDP) by Shevgoor et al.

Slim AMPM is a hybrid prefetcher:

- Uses the Delta Correlating Prediction Tables (DCPT) and Access Map Pattern Matching (AMPM) prefetchers.
- DCPT used when AMPM prefetcher cannot be used or for workloads that do not benefit from it.
- Evaluation performed to choose the best one to use.

VLDP:

- Variable Length Delta Prefetcher (VLDP) outperforms AMPM.
- Can use more complex regular access patterns.

Aneesh Neelam <aneelam@ucsc.edu>

Background

Slim AMPM: Optimized AMPM + Evaluation framework

Evaluation framework to choose between DCPT and AMPM in Slim AMPM.

DCPT more suited for irregular access patterns.

AMPM for more regular access patterns.

VLDP is an improvement over AMPM, can use more complex regular patterns.

Introduction

Devised two new hybrid prefetchers:

1. VLDP - AMPM hybrid prefetcher
2. DCPT - VLDP hybrid prefetcher

Ported evaluation code from Young et al. to both hybrid prefetchers.

AMPM style Access Map used for evaluation.

Evaluation Framework

Exact Results reproducible by running traces again (Check README):

- Results compared with Slim AMPM and VLDP (Their best case results)
- Average Case results for hybrid prefetchers (Framework averaging) used.

Software stack:

- Ubuntu 16.04 VirtualBox machine

Prefetcher/Traces	VLDP-AMPM	DCPT-VLDP	Slim AMPM (DCPT-AMPM)	VLDP
GemsFDTD	3.450102	3.448920	0.860191	0.833527
gcc	0.331036	0.328462	2.902273	2.901642
lbm	1.984225	1.966279	1.705119	1.692381
leslie3d	1.299102	1.243556	0.975834	0.808902
libquantum	3.279885	3.273719	2.874937	2.861325
mcf	0.375427	0.371777	0.761822	0.737965
milc	1.203386	1.200224	1.16502	1.03397
omnetpp	2.251103	2.245062	0.880231	0.948249

Prefetcher/Traces	VLDP-AMPM	DCPT-VLDP	Slim AMPM (DCPT-AMPM)	VLDP
GemsFDTD	3.450102	3.448920	0.859272	0.842652
gcc	0.334651	0.330768	2.815107	2.819697
lbm	1.768515	1.768936	1.734494	1.729084
leslie3d	1.196737	1.149879	0.893055	0.763178
libquantum	3.279885	3.273719	2.916293	2.905574
mcf	0.236937	0.237254	0.793876	0.774975
milc	1.117191	1.114823	1.247686	1.110049
omnetpp	1.535260	1.528507	0.855185	0.933967

Results, small_llc

Prefetcher/Traces	VLDP-AMPM	DCPT-VLDP	Slim AMPM (DCPT-AMPM)	VLDP
GemsFDTD	3.411516	3.411539	0.430172	0.431387
gcc	0.182784	0.183068	2.616083	2.587423
lbm	0.987513	0.986207	0.851235	0.844733
leslie3d	0.689992	0.678101	0.399688	0.367698
libquantum	3.180119	3.204195	2.455232	2.447332
mcf	0.224346	0.223630	0.561072	0.566711
milc	0.615020	0.613283	0.580039	0.597089
omnetpp	1.749399	1.738227	0.591656	0.547526

Prefetcher/Traces	VLDP-AMPM	DCPT-VLDP	Slim AMPM (DCPT-AMPM)	VLDP
GemsFDTD	3.426760	3.423804	0.855348	0.831578
gcc	0.331000	0.328068	2.886941	2.882598
lbm	1.979308	1.982051	1.697693	1.699459
leslie3d	1.260684	1.207869	0.962062	0.799942
libquantum	3.278289	3.274490	2.881643	2.853581
mcf	0.378711	0.373476	0.764595	0.734879
milc	1.191201	1.187882	1.157804	1.03203
omnetpp	2.254144	2.245905	0.931988	0.960259

Analysis

Both hybrid prefetchers perform similarly.

Both prefetchers are not used simultaneously to prevent cache pollution.

But cache pollution can occur if prefetches are done improperly.

Evaluation framework does not choose correct framework on gcc trace.

Cache pollution causes gcc trace to perform extremely poorly.

Conclusion

Can create a hybrid prefetcher within the storage state limits.

Storage state used by both hybrid prefetchers is less than 8 KB.

Ran for all four configurations of DPC simulation framework

Increasing the size of data structures (state storage) did not change results.

Hybrid prefetchers do not perform better for all traces (gcc, mcf)

References

"Access Map Pattern Matching Prefetch: Optimization Friendly Method"
Yasuo Ishii, Mary Inaba, and Kei Hiraki

"Towards Bandwidth-Efficient Prefetching with Slim AMPM"
Vinson Young, Ajit Krisshna

"Efficiently Prefetching Complex Address Patterns"
Manjunath Shevgoor, Sahil Koladiya, Rajeev Balasubramonian, Zeshan Chishti

Documentation (Extra Slide)

Count to keep track of number of prefetchers done for every operation.

Evaluation period to determine if the first prefetcher is viable.

If first prefetcher runs, then the second is omitted.

This code has been ported from Young et al.'s Slim AMPM.

This is to prevent both prefetchers from polluting caches.

Documentation (Extra Slide)

VLDP modified to work with the Evaluation framework of Slim AMPM.

Omit prefetching completely if performance is bad.

This is coded into VLDP, DCPT, AMPM by their respective authors.

Evaluation framework uses AMPM style Access Map.

Map needed to keep track of the pages already prefetched by other prefetcher.