THE UNIVERSITY OF RHODE ISLAND
COLLEGE OF ENGINEERING

Sandbox Based Optimal Offset Estimation [DPC2]

Nathan T. Brown and Resit Sendag

Department of Electrical, Computer, and Biomedical Engineering

THINK BIG WE DO℠

Outline

- Motivation
- Background/Related Work
  - Sequential Offset Prefetchers
  - Sandbox Prefetching
- Optimal Offset Estimation
  - Overview/Architecture
  - Memory Access Latency Estimation
  - Scoring
  - Experimental Results/Conclusion
- Acknowledgements

Motivation

- Balance between aggressive and conservative prefetching:
  - *Confirmation*: Feedback Directed Prefetching (FDP).
  - *Immediate*: Next line/Sequential Offset.
  - *Combination*: Sandbox Prefetching.
- Performance is highly architecture and application dependent:
  - Adaptive mechanisms perform well.
  - Limited bandwidth (memory prioritization).
    - Main memory open row policy/prioritization.

Related Work: Sequential Offset Prefetchers

- Sequential Offset Prefetchers
  - Always produces a single prefetch per access (i.e. next line prefetcher).
  - Upon a cache access to address *A*, the address *A* + *O* (offset) is immediately prefetched.
  - Majority of benchmarks exhibit favorable performance for a given fixed offset.
  - Aggressiveness proves damaging in unfavorable applications as they cannot adapt to changing conditions.

Related Work: Sequential Offset Prefetchers Benchmark Profile

	Sequential Offset																															
SPEC2006	**-16**	**-15**	**-14**	**-13**	**-12**	**-11**	**-10**	**-9**	**-8**	**-7**	**-6**	**-5**	**-4**	**-3**	**-2**	**-1**	**+1**	**+2**	**+3**	**+4**	**+5**	**+6**	**+7**	**+8**	**+9**	**+10**	**+11**	**+12**	**+13**	**+14**	**+15**	**+16**
art	0.87	0.97	0.86	0.86	0.97	0.85	0.85	0.97	0.85	0.84	0.97	0.84	0.83	0.97	0.83	0.82	0.81	0.80	1.07	0.80	0.81	1.09	0.82	0.83	1.11	0.82	0.83	1.11	0.83	0.83	1.11	0.84
ammp	0.86	0.86	0.86	0.85	0.85	0.85	0.85	0.84	0.84	0.84	0.84	0.83	0.86	0.88	0.88	0.89	0.97	0.92	0.92	0.92	0.86	0.86	0.86	0.86	0.86	0.87	0.87	0.87	0.87	0.87	0.88	0.88
parser	1.01	1.01	1.01	1.01	1.02	1.01	1.01	1.02	1.02	1.02	1.01	1.01	1.02	1.02	1.02	1.01	1.08	1.10	1.10	1.10	1.09	1.09	1.09	1.08	1.08	1.08	1.08	1.08	1.08	1.07	1.07	1.07
perlbench	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.21	1.21	1.20	1.27	1.23	1.18	1.18	1.20	1.24	1.16	1.16	1.17	1.20	1.16	1.14	1.14
bzip2	0.98	0.98	0.98	0.97	0.98	0.97	0.98	0.98	0.98	0.98	0.97	0.98	0.98	0.99	1.02	0.97	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.96
gcc	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.01	1.19	1.22	1.23	1.23	1.22	1.22	1.22	1.21	1.20	1.20	1.19	1.19	1.18	1.17	1.16	1.16
bwaves	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.23	1.25	1.35	1.36	1.36	1.36	1.35	1.34	1.33	1.32	1.31	1.31	1.30	1.28	1.27	1.26
gamess	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.99	0.98	0.98	0.98	0.99	0.97	0.97	0.98	0.97	0.98	0.99	0.98	0.98	0.99	0.99	0.99
mcf	0.97	0.98	0.97	1.00	0.98	1.00	1.00	0.99	1.03	1.02	1.03	1.04	1.03	1.05	1.04	1.02	1.02	1.04	1.04	1.03	1.05	1.04	1.04	1.05	1.03	1.04	1.02	1.01	1.02	0.99	1.00	0.97
milc	0.93	0.93	0.93	0.94	0.95	0.96	0.95	0.94	0.92	0.92	0.90	0.91	0.91	0.91	0.96	0.99	1.02	0.99	0.95	0.97	0.97	0.95	0.95	0.95	0.95	0.97	0.98	0.97	0.96	0.95	0.95	0.96
zeusmp	1.18	1.17	1.16	1.15	1.15	1.13	1.09	1.05	1.03	1.03	1.02	1.02	1.02	1.02	1.01	1.02	1.54	1.49	1.44	1.39	1.34	1.29	1.19	1.14	1.12	1.11	1.10	1.09	1.08	1.08	1.07	1.08
gromacs	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.20	1.22	1.23	1.23	1.22	1.22	1.21	1.21	1.20	1.20	1.19	1.19	1.19	1.18	1.18	1.17
cactusADM	0.98	1.01	1.01	1.01	1.01	1.01	1.01	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	0.99	1.75	1.72	1.69	1.66	1.65	1.61	1.57	1.51	1.52	1.51	1.49	1.47	1.46	1.44	1.43	1.42
leslie3d	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.27	1.34	1.35	1.36	1.36	1.35	1.34	1.33	1.33	1.32	1.31	1.31	1.31	1.30	1.30	1.28
namd	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.00	1.01	1.01	1.01	1.00	1.01	1.00	1.01
gobmk	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.99	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
dealII	0.99	1.00	1.00	1.00	1.00	1.01	1.01	0.99	1.01	0.99	1.01	1.01	1.01	1.01	1.01	1.02	1.11	1.12	1.10	1.10	1.10	1.09	1.08	1.09	1.08	1.08	1.08	1.08	1.07	1.06	1.07	1.06
soplex	1.13	1.13	1.13	1.14	1.14	1.14	1.14	1.14	1.14	1.15	1.15	1.15	1.15	1.16	1.16	1.15	1.03	1.02	1.02	1.02	1.02	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01
povray	0.98	0.97	0.96	0.97	0.98	0.98	0.98	0.98	0.97	0.98	0.98	0.99	0.98	0.99	1.00	1.00	1.00	1.00	1.00	0.98	0.98	0.98	0.98	0.99	0.98	0.98	0.98	0.99	0.99	0.99	1.00	0.99
hmmer	1.00	1.00	1.01	1.00	1.01	1.00	1.01	0.99	1.00	1.00	0.99	1.00	1.00	0.99	0.99	0.99	1.31	1.40	1.41	1.40	1.38	1.36	1.33	1.32	1.32	1.31	1.30	1.28	1.28	1.27	1.27	1.25
sjeng	0.97	0.97	0.97	0.98	0.97	0.97	0.98	0.97	0.97	0.97	0.97	0.96	0.97	0.97	0.96	0.97	0.98	0.98	0.98	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.98	0.97	0.98	0.98	0.97
GemsFDTD	0.99	0.99	0.99	0.99	0.98	0.99	0.99	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.98	0.99	1.19	1.22	1.22	1.22	1.21	1.21	1.21	1.21	1.20	1.20	1.20	1.19	1.19	1.20	1.19	1.19
libquantum	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.02	1.02	1.02	1.02	1.02	1.01	1.02	1.02	1.01	1.02	1.02	1.02	1.01	1.01

**Results are for configuration 1 across SPEC2006.*

Related Work: Sandbox Prefetching

- Prefetchers (32)
  - Sequential offsets between -16 and +16 evaluated in a real-time simulation environment.
- Sandboxes (16)
  - 2048 bit bloom filter with 1% false positive rate (256).
  - Stride detection: For *O of* +3, As *A* is checked so is *A*-3, *A*-6, and *A*-9. If detected score is extended.
  - After access period, the results are *scored* and highest performers are allowed to go *live* while lowest 4 are cycled out (priority given to lowest positive offsets).
    - 1 to 8 possible live prefetches.

Sandbox Based Optimal Offset Estimation: Overview

- Competition architecture favors fewer, more accurate, and timely prefetches:
  - *Accuracy*: Will the line be used in the near future?
  - *Timeliness*: Would the line have arrived in time for use?
  - *Usefulness:* Accuracy vs. Timeliness.
- Adjust scoring to reflect cache fill time:
  - Estimate fill time by tracking lower level memory latency for demand misses.
  - Identify if sandbox hit would have been filled in time for access (coarse).

Sandbox Based Optimal Offset Estimation: Architecture

- Prefetchers (9)
  - Sequential offsets between -1 and +8 (experimental).
  - Dedicated *Sandbox* with 1024 entries (experimental).
  - *Sandbox*, *Late*, and *Useful* score registers and logic.
- Shared *Cycles to Arrival* buffer.
- Lower Level Memory Access Latency Estimation
  - Utilizes user MSHR as well as additional cycle buffer.
- Prefetch Buffer (32 entries).
- User MSHR (16 entries).

Sandbox Based Optimal Offset Estimation: Architecture cont.

- Selection
  - Optimal offset prefetcher is identified as that with the highest *Useful Score*.
  - *Sandbox Score* for chosen prefetcher must be greater than a quarter the maximum possible score to go *live*.
- Prefetches
  - Provided the prefetch passes filtering (MSHR/PB), the request will go to the L2 if MSHR is less than half occupancy. Otherwise, it will go the LLC (L3).

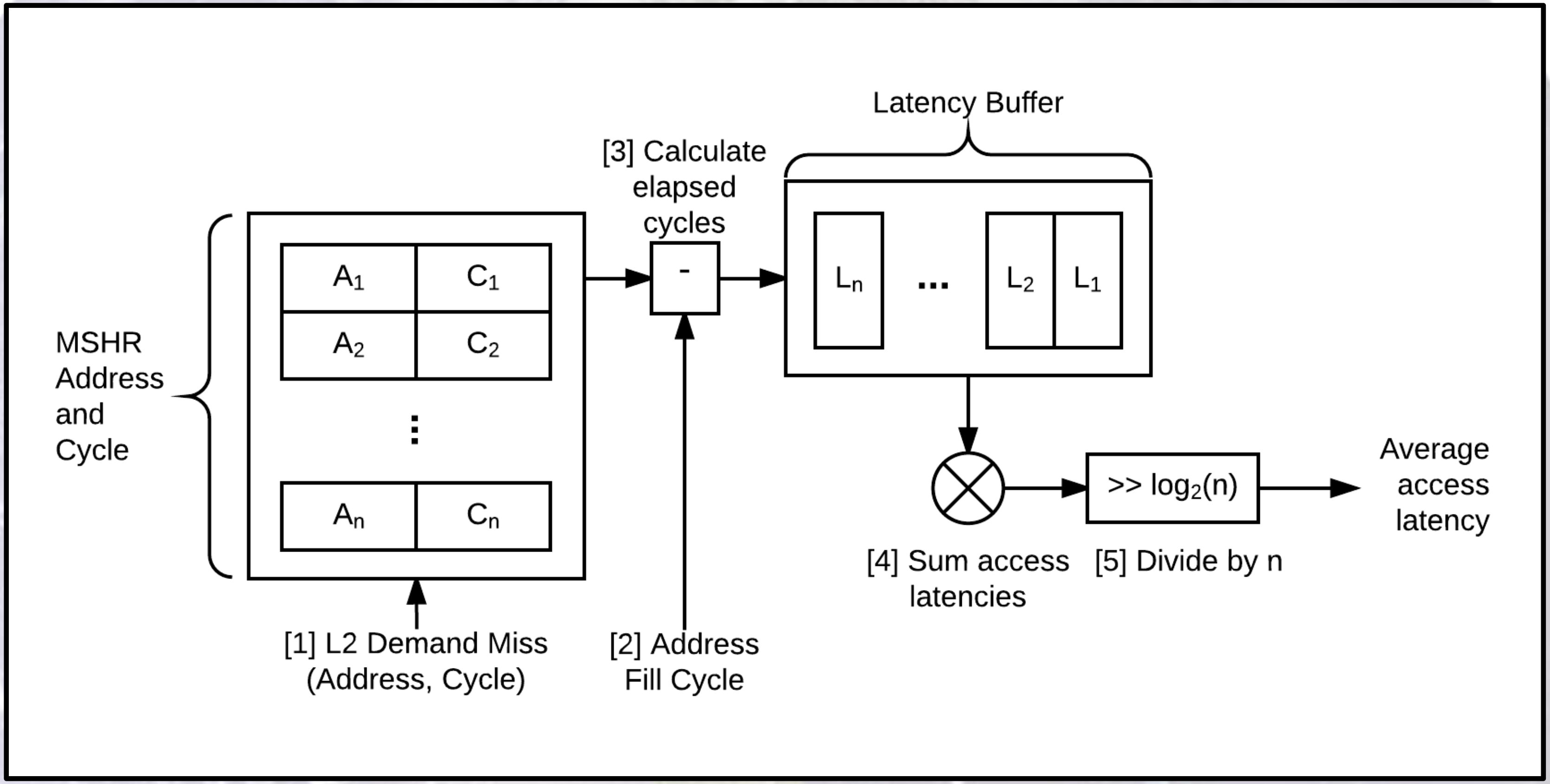
Sandbox Based Optimal Offset Estimation: Scoring

- *Sandbox Score* (*S_S*): L2 accesses during the evaluation period which were present in the sandbox (hits).
- Late Score (*S_L*): Hits in the sandbox which had non-zero *Cycles to Arrival* field when the hit occurred.
- *Useful Score* (*S_U*)*:* Sandbox score adjusted by the late score.

$$\boxed{\boldsymbol{S_S - S_L = S_U}}$$

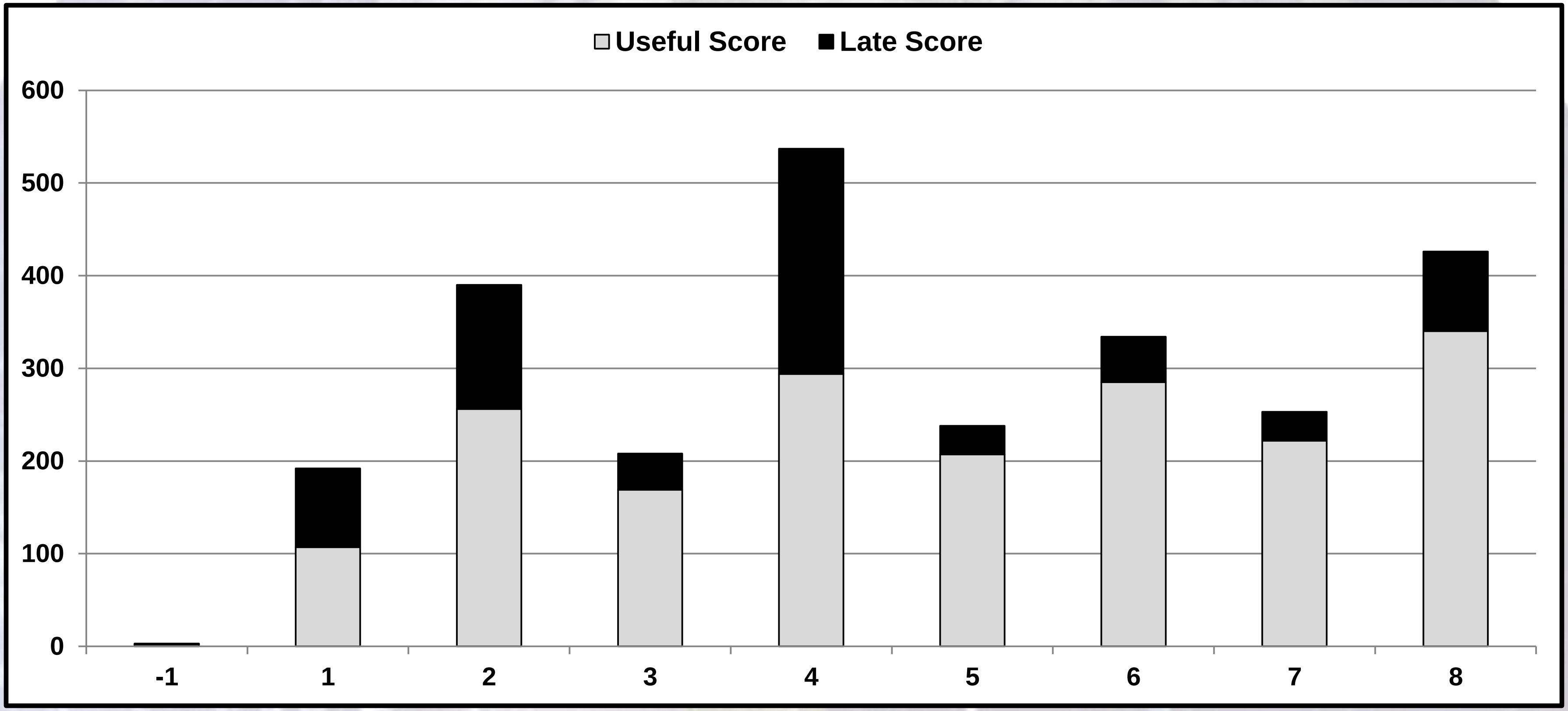
THINK BIG WE DO

Memory Access Latency Estimation



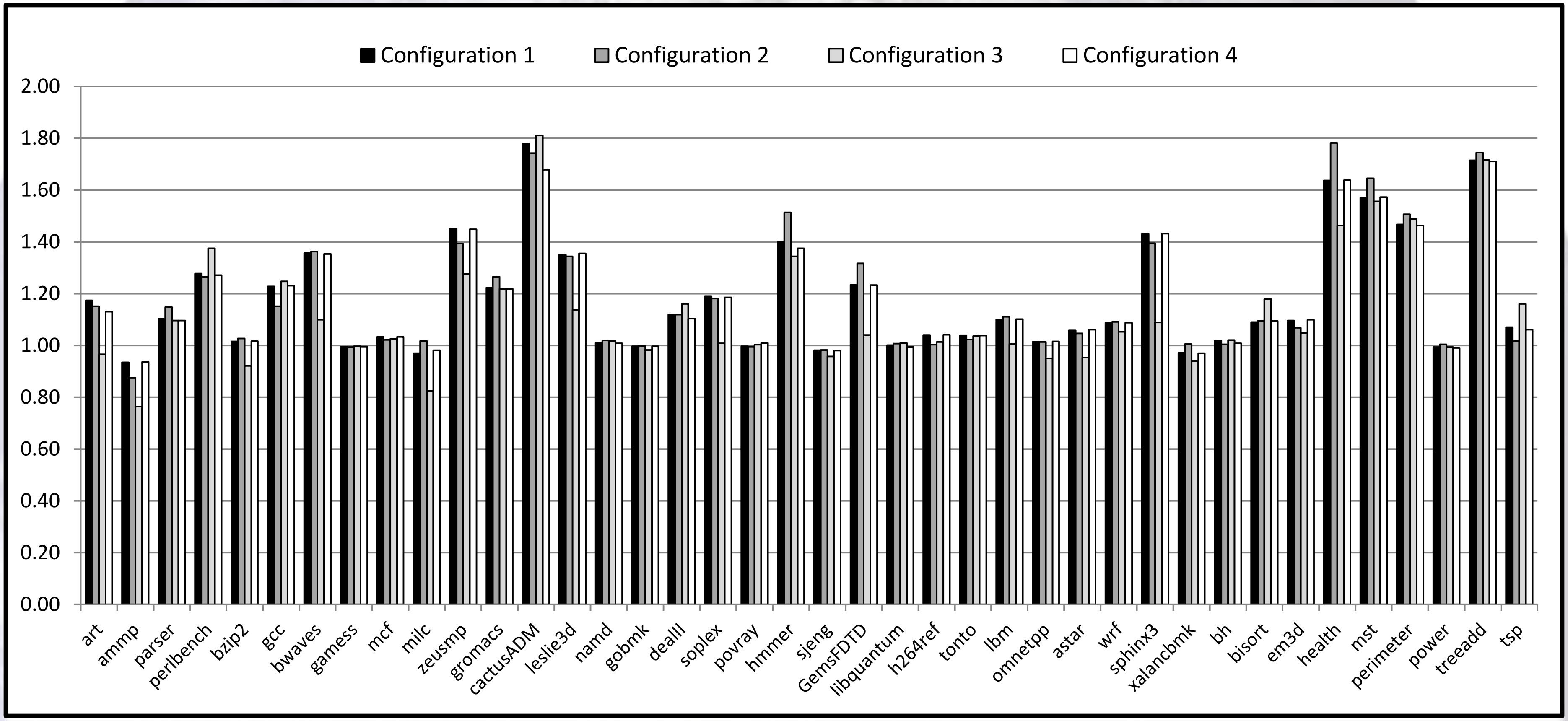
**Can be implemented without buffer.*

Scoring Example: GCC Single Evaluation Period: *Useful*, *Late*, and *Sandbox* Score



**Maximum sandbox score of 1024.*

Experimental Results



**Benchmarks from SPEC2006 and SPEC2000*

THINK BIG WE DO™

THE UNIVERSITY OF RHODE ISLAND

Hardware Budget

Component	Paper Cost	New Cost
Access Latency Estimation	898 bits	-
Prefetch Buffer	1,036 bits	-
Offset Prefetchers (9)	580 bits	-
Sandboxes (9)	240,138 bits	148,298 bits
Total	**242,652 bits**	**150,892 bits**
Percentage	**92.56%**	**57.53%**

- Following submission several changes were made to reduce hardware without any effect on performance:
  - Share *Cycles to Arrival* buffer between sandboxes.
  - Reduce the *Cycles to Arrival buffer from 1024 to 32 entries.*
    - *Fill time rarely if ever exceeds this threshold.*

Conclusion

- Overall Score:
  - *AMPM-Lite: **4.511***
  - *Sandbox Implementation: **4.578***
  - *Sandbox Based Optimal Offset Estimation: **4.589***
- Thoughts and Potential Improvements:
  - Costly (hardware) but effective on this architecture.
  - Potentially utilize wider array of offset prefetchers and share sandboxes (like original).
  - Addition of PC based prefetcher.

Acknowledgements

- We would like to thank the organizers of the 2nd Data Prefetching Championship (DPC2).
- We would like to thank the anonymous reviewers for their helpful suggestions and feedback.
- This work is partly supported by the National Science Foundation under grant CNS-1405862.

Questions?