Hardware assisted Cache Prefetching Techniques

Aniket Pandey
IIT Kanpur, 160113
aniketp@iitk.ac.in

Aditya Rohan
IIT Kanpur, 160053
raditya@iitk.ac.in

ABSTRACT

Hardware prefetchers play an important role in hiding long data access latency in various multiprocessor systems. Significant number of prefetchers have been proposed using state-of-the-art techniques which independently augment the existing instruction pointer based and cache-address based prefetching. In this paper, we attempt to categorically evaluate the submissions of the 3rd Data Prefetching Championship (DPC3) [1]. Additionally, we also propose IPCP++, an enhanced version of Instruction Pointer Classifier based prefetcher (IPCP) [11] reinforced with IP-Delta based sequence predictor and IP-based stride prefetcher (Sangam++) [6]. The confluence of these IP based prefetching techniques achieves a speedup of 9.74% over no prefetching when averaged over 20 single-thread 6XX SPEC CPU 2017 traces. This is the second best performance after Berti (9.89%).

As a stretch goal, we also present a dynamic degree & distance stream prefetcher, written from scratch. The dynamic stream prefetcher (DSP) takes into account factors like prefetch accuracy, cache pollution and strength of the stream in a given window of 50 instructions to determine the prefetch degree and distance dynamically.

1. INTRODUCTION

Modern processors employ various techniques to overcome the memory wall problem. Data prefetching is an important solution for the same where prefetchers seek to understand the memory access pattern and initiate cache-line prefetching based on the learned pattern. Instruction pointer (IP) based prefetchers such as the IP-Stride prefetcher offer huge performance benefits that help in reducing the number of costly DRAM accesses [11].

In a multi-level cache hierarchy, data blocks can be brought into different levels of cache depending on confidence of a particular prediction. It may so happen that hierarchical prefetchers employ entirely difference schemes for retrieving data blocks. The submissions in DPC3 were observed to have followed similar prefetching schemes for both L1 Data Cache and L2 Cache. Since having a functional prefetcher is not significantly profitable at the last-level cache, we observe that [3, 13, 14] employ generic Next-Line prefetching while [6, 9, 11, 12], have no LLC prefetcher whatsoever.

We leverage the Signature Path Prefetching [4] framework at LLC due to the noticeable performance improvement over other prefetchers, as shown in Figure 1. For L1 Cache, we implement a crossover of Instruction Pointer Classifier with IP-Delta-based sequence predictor, IP-based stride prefetcher and adaptive next-line prefetcher [6] in the decreasing order of preference. The later schemes are invoked only when the confidence offered by earlier is below a predefined threshold. We exclude speculative next-line method from IPCP as it is ineffective in improving the IPC. The L2 prefetcher is set as Sangam++ due to the best observed results.

2. RELATED WORK

Sizeable number of contributions have been proposed in the field of data prefetching in recent years. Prefetchers to reduce instruction cache misses for warehouse-scale, database-centric systems have been proposed [2, 15]. Certain prefetchers for irregular memory accesses have also been studied [8]. Prefetchers leveraging cache-miss-address based techniques have been proposed earlier [7, 10].

3. UNDERLYING DESIGN OF IPCP++

The IPCP++ prefetcher consists of a mixture of 6 individual components. The parts borrowed from IPCP include the prefetching decision based on 3 different classes of IPs. IP Constant Stride (CS), IP Complex Stride (CPLX) and IP Global Stream (GS). Certain irregular accesses, which do not follow any pattern, simple or complex, but recur, have been classified into a separate class: Temporal and Irregular IP (TIP). The ISB prefetcher [8] suits these access patterns.

The 3 components of Sangam++ are IP-delta-based delta sequence predictor, IP-based stride predictor and an adaptive next-line prefetcher. We will now discuss the details of the individual components.

Prefetcher	L1D	L2C	LLC
IPCP++	IPCP++	Sangam++	SPP
Sangam	Sangam++	Sangam++	–
Bouquet	IPCP	IPCP	–
Enhancing	Next-line	SPP+PPF	SPP

Table 1: Components of the involved prefetchers

3.1 Classes for the IP Classifier

The IPCP module divides Instruction Pointers into 3 classes:

3.1.1 IP Constant Stride (CS)

Th access patterns covered by this class have a constant stride pattern, which enables 100% prefetch accuracy assuming no anomaly is observed. An example mentioned in [11] for an IP named IP_A: 0,2,4,6,8 elucidates the constant stride pattern. The prefetching confidence for CS is maintained by a simple 2-bit counter, as described in [16].

3.1.2 IP Complex Stride (CPLX)

For access pattern like IP_B: 0,1,2,3,5,6,7,8,10 with the stride pattern of $\{1,1,1,2\}x2$. Here, a CS classifier would assume a 75% accuracy and would be unable to ascertain the very definite yet complex stride pattern observed in this case. With CPLX, a signature for this delta would be created in the delta prediction table (DPT) and would enable maximal correctness whenever this pattern is observed next.

3.1.3 IP Global Stream (GS)

Certain stride patterns can be too generic to be classified as one of CS or CPLX. Such accesses are categorized within Global Stream (GS). These are the set of cache aligned accesses that although can be indexed to a particular IP and followed by the trigger stream. However, on careful analysis, we see that these are essentially independent of any IPs and it would be worthwhile to not consider such strides in either of CS or CPLX. Therefore, [11] prioritizes GS over other classes as it maps to multiple instruction pointers.

3.2 Components of Sangam++

Sangam++ is a confluence of 3 prefetching components:

3.2.1 IP-Delta-based Sequence Predictor

The IP-Delta sequence predictor uses source IP to distinguish all demand accesses. The general scheme looks at the recent deltas to predict next d deltas, where d is a baseline prefetch degree. The observation here is that within the accesses coming from a given IP, the sequence of deltas appearing after a given Δ essentially repeats itself [6].i.e Δ, δ_1, δ_2, δ_3,...,Δ, δ_1, δ_2, δ_3,... The predictor also uses two tables to store history of previous accesses. IP Table is used to store a FIFO list of last $d+1$ accesses corresponding to a given IP. Next, the IP-delta table is looked up using a concatenation of the IP of the current demand access and the current delta within the access stream sourced by that IP. Prefetching decision is usually made by looking up the IP-delta table using the current IP and delta. The sequence of deltas stored for the IP, delta combination is read out in case of a hit. In the case of a miss in the IP-delta table or IP table, no prediction can be generated.

3.2.2 IP-based Stride Prefetcher

Since the first step of IP-Delta sequence prediction looks up IP table anyways, it can be used without any further overhead to come up with a IP-stride prediction in case the confidence reflected by the last step is below a certain threshold.

3.2.3 Adaptive degree Next-Line Prefetcher

In case the IP-stride prefetcher and IP-delta sequence predictor cannot offer enough confidence, the third component, an adaptive feedback enabled next-line prefetching is employed. The feedback mechanism helps to avoid polluting the L1 cache with inaccurate next-line prefetches. Priority is given to lower-degree prefetches to avoid large lookahead in NL prefetcher, which can be quite inaccurate.

4. MODIFICATIONS IN IPCP++

IPCP uses speculative next-line prefetching as and when the confidence offered by IP classifier prediction is not significant enough. However, the performance benefit offered by speculative next-line prefetching is diminutive. The original IPCP framework has an average geomean speedup of 9.627% over no prefetching, while on eliminating speculative next-line, the speedup observed is 9.617%.

Our modification introduces IP-delta sequence predictor and IP-stride prefetcher in addition to the original Instruction pointer classifier based prefetcher at L1 cache. This change observes a speedup of 9.357% individually over no prefetching. The exclusive modification at L1 actually reduces the speedup as compared to IPCP. We replace L2 prefetcher with Sangam++ and LLC prefetcher with SPP [4] to make up for the lost speedup. Additionally, we turn off the $\Delta \neq 0$ assertion since the modified IPCP++ L1 prefetcher does not incorporate full IP-delta sequence based predictor and there is no communication between L1 and L2 prefetcher and hence, the encoded metadata does not contain the most up-to-date delta. Collective combination of these prefetching techniques results in a speedup of 9.74%, an increase of 1.1% from the original.

5. EVALUATION OF DPC3 SUBMISSIONS

The submissions for DPC3 were required to implement a data prefetching algorithm for L1, L2 & LLC cache, with the restriction being a maximum storage budget of 32KB for each core. The prefetchers were evaluated based on their performance on a set of benchmarks using the framework provided by the organizing committee [1]. To evaluate the performance of submissions for our own research, we considered the subset of benchmarks that were recommended in the Champsim repository [5]. Henceforth, the resulting performance difference is not consistent with the official results. The comprehensive evaluation result is tabulated in Tables 2 to 21. We only consider single-core CPU for the evaluation purpose.

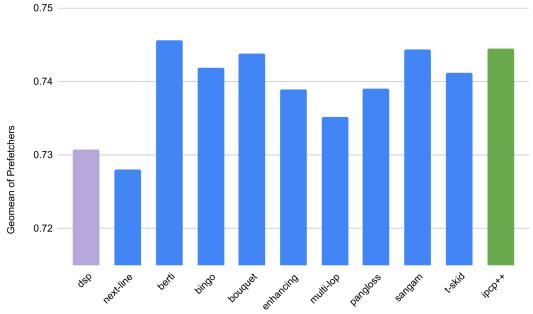

Figure 1: Geomean Comparison of DPC3 Submissions

The geomean of all prefetchers is compared in Figure 1. Difference in best and the worst performing scheme is 1.4%. Best performing submission in our case is Berti, with the observed speedup of 9.896% over no prefetching. This is closely followed by Sangam (9.716%) and IPCP (9.626%). The difference is pretty minimal for the top performers, with majority of the submissions reaching 9.0% mark, i.e Bingo (9.349%), T-Skid (9.250%), Pangloss (8.936%) & SPP (8.914%). Multi-lop however, which won the third place in DPC3 due to a good 4-core speedup, is quite behind the second last performer with a speedup of (8.362%). The reason for this can

be attributed to poor performance in case of 605-mcf where multi-lop is the sole prefetcher to exhibit a negative speedup (-6.52%) (See Table 5).

For most of the traces, we observe a significant positive speedup by almost every prefetcher. The speedups are nearly consistent for a prefetcher on a given trace suggesting predictable memory accesses which are properly discovered. Traces like 602-gcc, 619-lbm, 628-pop2, 649-fotonik offer multi fold increase in IPC, and are notable for the overall performance improvement. However, we also see that in case of 623-xalancbmk every prefetcher except for Berti and Bingo have negative speedups. This signifies that cache pollution is quite prevalent for unpredictable memory accesses and that prefetchers hamper the overall performance in such cases.

6. DYNAMIC STREAM PREFETCHER

In this section, we present a dynamic stream prefetcher modeled after the Intel stream prefetcher at the L2 cache level. We implement the prefetcher for both the L1D and L2C cache. In the context of the stream prefetcher, the "prefetch degree" is the number of lines the streamer prefetches, and the "prefetch distance" is how far ahead the prefetched cache-line is from the trigger line. The degree and distance can depend on various factors such as prefetch accuracy, cache pollution, and strength of the stream that triggers the prefetcher.

For L1D cache, we maintain a prefetch table of a capacity of eight block addresses, to keep track of the lines that are prefetched every time the prefetcher is triggered. We update the prefetching policy (distance and degree) every 1000 cycles. This window can be increased or reduced based on the resulting IPC. For any given window, we use the prefetch table elements to calculate the accuracy of the prefetcher. For accuracy above a certain threshold, we increase the prefetch degree.

We also maintain a history table that stores unique cache blocks accessed until now. We update the history table with a Least Recently Used policy. We use this table to identify the direction of the stream and the strength of the stream. For a very strong stream, we increase the prefetch degree by up to two. There are separate windows for identifying very strong, fairly strong, steady, and weak streams. We increase the degree by one for fairly strong streams; We leave it unchanged for steady streams and reduce it by one for weak streams.

To observe the effect of cache pollution, we monitor the number of cache hits in a window 1000 cycles. We change the prefetch degree depending on the amount of cache pollution. Higher the pollution, lower will be the cache hit rate in any given window. Taking into account all of these factors, we vary the degree dynamically for L1D cache. For the L2C cache, we also vary the distance depending on the aforementioned factors.

7. CONCLUSION AND FUTURE WORK

Acknowledgements

This work is heavily inspired from Sangam & IPCP, and is a course project of *CS622: Advanced Computer Architecture*, Fall-19, IIT Kanpur. We are extremely grateful to Dr. Mainak Chaudhuri, course instructor and co-author of Sangam, for guiding us at various stages of the project.

8. REFERENCES

[1] (2019) The 3rd data prefetching championship (dpc3). [Online]. Available: https://dpc3.compas.cs.stonybrook.edu/

[2] G. Ayers, N. P. Nagendra, D. I. August, H. K. Cho, S. Kanev, C. Kozyrakis, T. Krishnamurthy, H. Litz, T. Moseley, and P. Ranganathan, "Asmdb: understanding and mitigating front-end stalls in warehouse-scale computers," in *Proceedings of the 46th International Symposium on Computer Architecture*. ACM, 2019, pp. 462–473.

[3] M. Bakhshalipour, M. Shakerinava, P. Lotfi-Kamran, and H. Sarbazi-Azad, "Accurately and maximally prefetching spatial data access patterns with bingo," *The Third Data Prefetching Championship*, 2019.

[4] E. Bhatia, G. Chacon, E. Teran, P. V. Gratz, and D. A. Jiménez, "Enhancing signature path prefetching with perceptron prefetch filtering," *Delta*, vol. 1, p. 3.

[5] ChampSim. (2017) Champsim. [Online]. Available: https://github.com/ChampSim/ChampSim

[6] M. Chaudhuri and N. Deshmukh, "Sangam: A multi-component core cache prefetcher," *The Third Data Prefetching Championship*, 2019.

[7] T.-F. Chen and J.-L. Baer, "Effective hardware-based data prefetching for high-performance processors," vol. 44, no. 5. IEEE, 1995, pp. 609–623.

[8] A. Jain and C. Lin, "Linearizing irregular memory accesses for improved correlated prefetching," pp. 247–259, 2013.

[9] T. Nakamura, T. Koizumi, Y. Degawa, H. Irie, S. Sakai, and R. Shioya, "T-skid: Timing skid prefetcher," *The Third Data Prefetching Championship*, 2019.

[10] A. R. Omondi *et al.*, "Dstride: data-cache miss-address-based stride prefetching scheme for multimedia processors," in *Australian Computer Science Communications*, vol. 23, no. 4. IEEE Computer Society, 2001, pp. 62–70.

[11] S. Pakalapati and B. Panda, "Bouquet of instruction pointers: Instruction pointer classifier based hardware prefetching," *The Third Data Prefetching Championship*, 2019.

[12] P. Papaphilippou, P. H. Kelly, and W. Luk, "Pangloss: a novel markov chain prefetcher," *arXiv preprint arXiv:1906.00877*, 2019.

[13] A. Ros, "Berti: A per-page best-request-time delta prefetcher," *The Third Data Prefetching Championship*, 2019.

[14] M. Shakerinava, M. Bakhshalipour, P. Lotfi-Kamran, and H. Sarbazi-Azad, "Multi-lookahead offset prefetching," *The Third Data Prefetching Championship*, 2019.

[15] A. Sriraman, A. Dhanotia, and T. F. Wenisch, "Softsku: Optimizing server architectures for microservice diversity@ scale," in *Proceedings of the 46th International Symposium on Computer Architecture*. ACM, 2019, pp. 513–526.

[16] S. P. Vanderwiel and D. J. Lilja, "Data prefetch mechanisms," *ACM Computing Surveys (CSUR)*, vol. 32, no. 2, pp. 174–199, 2000.

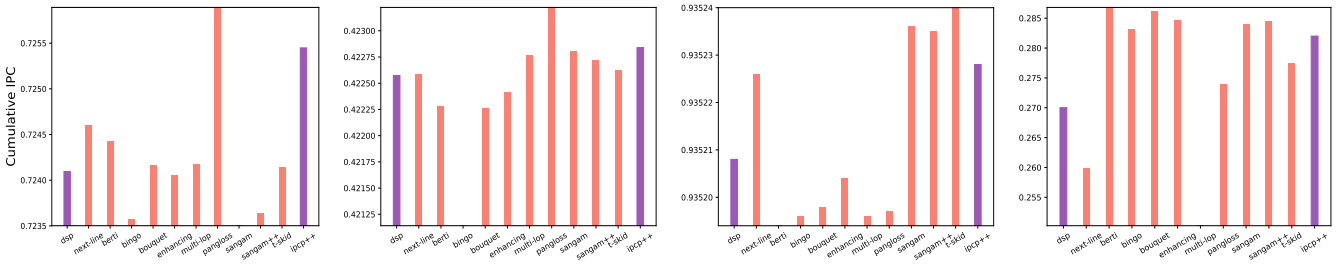

Figure 2: The plots for (a) 600.perlbench_s-210B, (b) 602.gcc_s-734B, (c) 603.bwaves_s-3699B, (d) 605.mcf_s-665B.

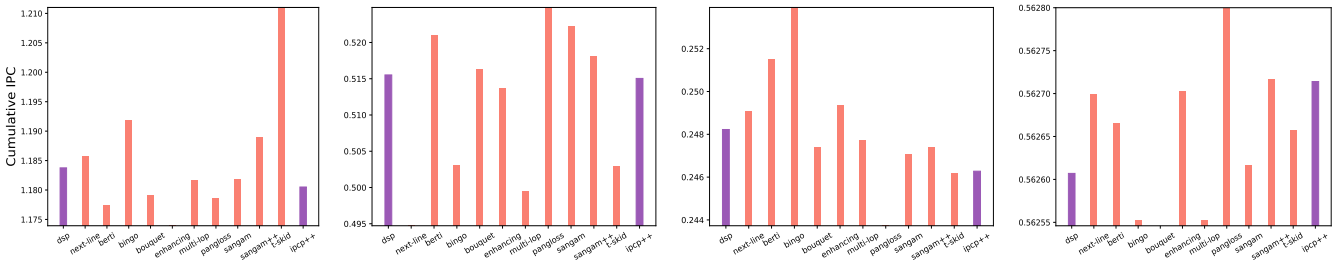

Figure 3: The plots for (a) 607.cactuBSSN_s-2421B, (b) 619.lbm_s-4268B, (c) 620.omnetpp_s-874B, (d) 621.wrf_s-575B.

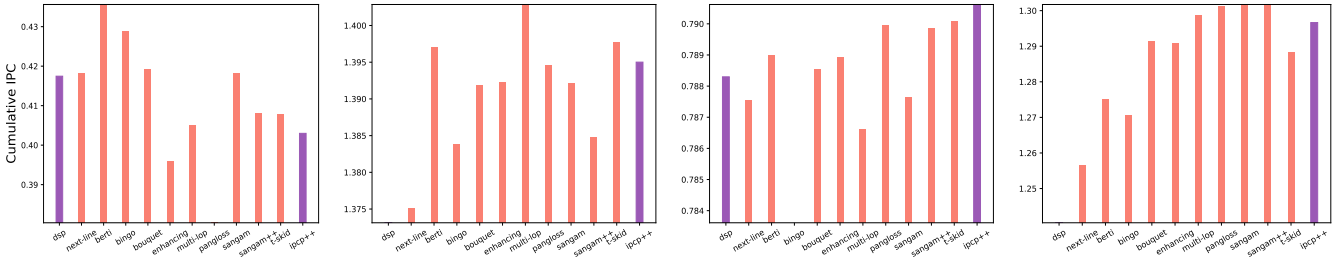

Figure 4: The plots for (a) 623.xalancbmk_s-700B, (b) 625.x264_s-18B, (c) 627.cam4_s-573B, (d) 628.pop2_s-17B.

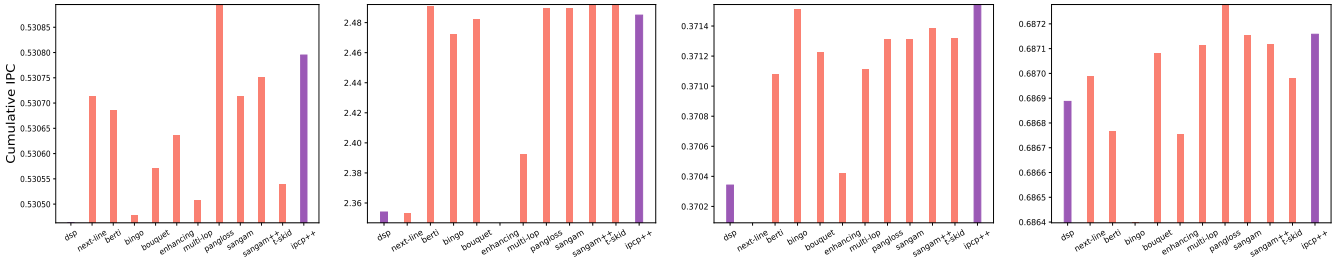

Figure 5: The plots for (a) 631.deepsjeng_s-928B, (b) 638.imagick_s-10316B, (c) 641.leela_s-800B, (d) 644.nab_s-5853B.

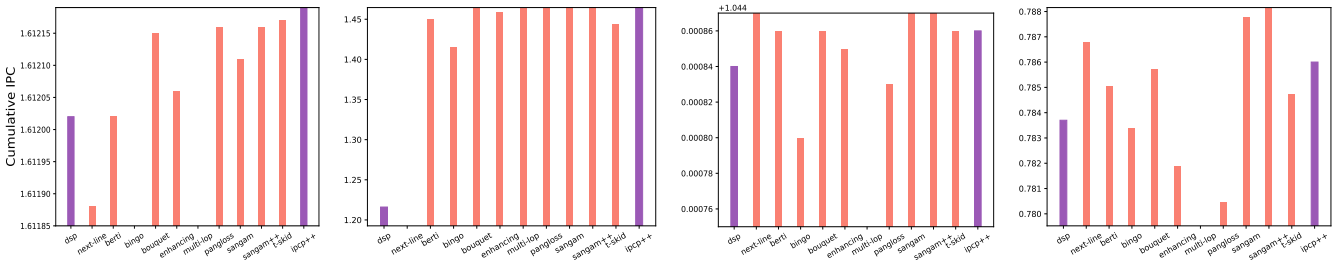

Figure 6: The plots for (a) 648.exchange2_s-1699B, (b) 649.fotonik3d_s-1176B, (c) 654.roms_s-842B, (d) 657.xz_s-3167B.

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	0.721404	–	3463564	1295	775	1127	0	1127
berti	0.724420	0.41%	3467440	1740	1514	1677	773	2094
bingo	0.723568	0.29%	3464970	2041	2539	1780	191	1780
bouquet	0.724167	0.38%	3763072	8495	15640	3044	4	3043
enhancing	0.724053	0.36%	3944527	3068	3590	1937	365	2105
multi-lop	0.724172	0.38%	3463453	6138	7066	2290	144	2290
pangloss	0.725891	0.62%	6345629	10952	62260	5619	461	5218
sangam++	0.723634	0.30%	4201868	6061	9917	2070	1	2070
t-skid	0.724139	0.37%	3462766	10278	13250	2224	512	2625
IPCP++	0.725448	0.56%	3953216	13512	25364	4478	138	4423

Table 2: Simulations for the trace 600.perlbench_s-210B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	0.355704	–	2996628	160748	81106	80698	348	80686
berti	0.422283	18.72%	3115451	161893	181160	82795	74870	84279
bingo	0.421141	18.40%	3112616	165382	91686	86947	685	86738
bouquet	0.422258	18.71%	6003627	172604	622669	86873	597	86615
enhancing	0.422416	18.75%	4147074	163941	248417	88226	1943	88537
multi-lop	0.422770	18.85%	3115956	178728	93118	87134	639	86996
pangloss	0.423223	18.98%	9605604	182059	926530	98763	1000	98106
sangam++	0.422716	18.83%	3767675	172425	240000	89570	468	89442
t-skid	0.422620	18.81%	3116929	169880	222740	84168	606250	88883
IPCP++	0.422840	18.87%	6232846	175567	329326	94494	1106	93888

Table 3: Simulations for the trace 602.gcc_s-734B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	0.935050	–	1619891	105	0	105	0	105
berti	0.935194	0.015%	1619906	108	11	111	20	112
bingo	0.935196	0.016%	1619890	105	17	105	0	105
bouquet	0.935198	0.016%	3025054	178	65	184	0	184
enhancing	0.935204	0.016%	2494996	114	21	126	2	124
multi-lop	0.935196	0.016%	1619890	105	17	105	0	105
pangloss	0.935197	0.016%	3007714	99	150	122	0	122
sangam++	0.935235	0.019%	1740826	116	58	125	0	125
t-skid	0.935240	0.020%	1619945	134	1	134	21	135
IPCP++	0.935228	0.019%	3052308	174	122	203	0	203

Table 4: Simulations for the trace 603.bwaves_s-3699B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	0.267712	–	3421585	390201	174925	265993	162896	129592
berti	0.286801	7.13%	3576554	449798	301672	321994	371265	213974
bingo	0.283072	5.74%	3611876	427718	338111	550201	375566	227483
bouquet	0.286180	6.89%	4436481	647587	525831	481662	320816	196235
enhancing	0.284624	6.32%	4253911	432309	304028	337909	319347	199971
multi-lop	0.250247	- 6.52%	3512406	920650	422970	881435	398343	534455
pangloss	0.273842	2.29%	6476745	452745	944078	684658	386246	337199
sangam++	0.284519	6.28%	4718624	704218	378462	482529	319614	199092
t-skid	0.277389	3.61%	3489169	1054759	565866	570901	373804	303798
IPCP++	0.282003	6.28%	4402873	778048	478122	518811	357185	223282

Table 5: Simulations for the trace 605.mcf_s-665B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	1.057070	–	3134933	733458	1009844	48612	27890	26476
berti	1.177460	11.39%	3353534	857270	1576833	53780	79009	26660
bingo	1.191910	12.76%	3358693	855668	1893787	49835	29340	26537
bouquet	1.179050	11.54%	4511672	1205028	2421986	47541	26714	26645
enhancing	1.173930	11.06%	3498508	840396	2185462	96065	80639	26415
multi-lop	1.181610	11.78%	3132349	785964	1973532	49309	28757	26638
pangloss	1.178660	11.50%	4412782	1234810	4174729	66016	45118	26884
sangam++	1.189040	12.48%	3912732	1265340	2180934	57317	36434	26681
t-skid	1.211010	14.56%	3492346	1168168	2056436	44601	154783	26716
IPCP++	1.180510	11.68%	4518531	1212419	2788048	56345	36545	26690

Table 6: Simulations for the trace 607.cactuBSSN_s-2421B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	0.438254	–	1993459	774713	961711	470311	353067	469523
berti	0.520927	18.86%	1674338	774698	1018971	470298	394481	469514
bingo	0.502995	14.77%	1634494	774704	975394	470313	353990	469515
bouquet	0.516328	17.81%	2113440	775294	1048624	470335	353084	469511
enhancing	0.513609	17.19%	2252507	777984	1014799	470636	354393	469597
multi-lop	0.499519	13.97%	1711843	803431	1007991	470439	357389	469545
pangloss	0.524830	19.75%	2455587	777143	1450705	470907	353608	469687
sangam++	0.518093	18.21%	1588855	783654	1035028	471097	353809	469725
t-skid	0.502912	14.75%	1690596	783909	973361	470323	357201	469517
IPCP++	0.515044	17.52%	2078256	775146	1030033	470423	355703	469535

Table 7: Simulations for the trace 619.lbm_s-4268B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	0.242485	–	4172615	180018	156388	117362	84361	91150
berti	0.251495	3.72%	4327688	227331	193644	204199	191764	215308
bingo	0.253927	4.72%	4284691	250884	273172	272765	159495	193242
bouquet	0.247397	2.03%	5446654	364952	297310	316955	155018	233276
enhancing	0.249334	2.82%	5015560	238219	249145	235453	163999	202453
multi-lop	0.247704	2.15%	4221970	309512	250849	189916	129821	139147
pangloss	0.243728	0.51%	9451386	448942	1007370	883232	271503	691358
sangam++	0.247377	2.02%	5335927	327819	257063	248198	129188	187171
t-skid	0.246158	1.51%	4217158	641534	382075	381409	174631	328258
IPCP++	0.246277	1.57%	5319056	395377	262248	308788	162163	234365

Table 8: Simulations for the trace 620.omnetpp_s-874B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	0.561930	–	1778689	518	21	773	0	773
berti	0.562665	0.13%	1778988	524	215	807	245	862
bingo	0.562553	0.11%	1778771	518	303	827	21	827
bouquet	0.562546	0.11%	1853275	614	377	950	0	950
enhancing	0.562703	0.14%	2282538	599	377	983	96	1006
multi-lop	0.562553	0.11%	1778771	518	303	827	21	827
pangloss	0.562800	0.15%	2570612	616	3899	1387	0	1387
sangam++	0.562717	0.14%	2239409	726	759	1058	0	1058
t-skid	0.562657	0.13%	1779014	604	140	852	771	1006
IPCP++	0.562714	0.14%	1854709	722	823	1101	15	1101

Table 9: Simulations for the trace 621.wrf_s-575B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	0.427485	–	2026532	395789	420292	23356	21180	14786
berti	0.435614	1.90%	2213948	424961	453312	32515	46516	18917
bingo	0.428859	0.32%	2213400	450003	687700	128946	152691	17719
bouquet	0.419142	- 1.95%	3809840	583190	672428	128227	107635	35835
enhancing	0.396017	- 7.36%	3674723	645820	711995	659381	791433	29146
multi-lop	0.404957	- 5.27%	2068098	1087320	977404	390961	448675	25951
pangloss	0.380342	- 11.03%	5122842	841083	2203876	1717845	1678691	58756
sangam++	0.408025	- 4.55%	2671411	862068	927646	664528	660136	23216
t-skid	0.407938	- 4.57%	2127583	1027852	639400	523762	603201	57026
IPCP++	0.402984	- 5.73%	3899166	959880	675053	800837	849848	52401

Table 10: Simulations for the trace 623.xalancbmk_s-700B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	1.36457	–	1430283	4372	535	3925	0	3925
berti	1.39697	2.37%	1436172	4943	3020	4304	2961	4574
bingo	1.38387	1.41%	1433800	4994	6328	4720	1317	4720
bouquet	1.39189	2.00%	2024099	7800	7440	4586	1	4586
enhancing	1.39224	2.03%	1705964	5648	5210	4934	5210	4934
multi-lop	1.40286	2.81%	1433621	7987	4185	4847	11	4847
pangloss	1.39454	2.19%	2380516	9400	28146	5608	9	5608
sangam++	1.38478	1.48%	1824785	6603	3845	4440	0	4440
t-skid	1.39769	2.43%	1433269	8328	4720	4454	503	4492
IPCP++	1.39501	2.24%	2086172	8252	7305	4592	551	4605

Table 11: Simulations for the trace 625.x264_s-18B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	0.773241	–	1612735	101341	108812	79371	83130	73597
berti	0.788991	2.04%	1634974	103068	115994	80606	88068	75319
bingo	0.783611	1.34%	1625861	102268	122178	80763	84947	74527
bouquet	0.788550	1.98%	1862433	107539	149807	81684	85131	75176
enhancing	0.788917	2.03%	1869135	102963	127686	81594	85832	75393
multi-lop	0.786623	1.73%	1631295	115825	128443	83179	87384	75413
pangloss	0.789953	2.16%	3432313	111407	273304	90323	92737	78436
sangam++	0.789852	2.15%	2161526	108228	146644	82937	86245	75805
t-skid	0.790094	2.18%	1633363	109410	140537	80913	90579	75365
IPCP++	0.790621	2.25%	1931485	109789	151376	83009	86718	75908

Table 12: Simulations for the trace 627.cam4_s-573B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	1.02064	–	1809231	202698	207082	86880	88261	30693
berti	1.27505	24.93%	2045125	212771	279920	101270	149539	33652
bingo	1.27064	24.49%	2026872	212179	270419	102891	111936	31746
bouquet	1.29122	26.51%	2867424	230962	536077	110219	113160	32298
enhancing	1.29066	26.46%	2682170	217698	376957	118681	128001	33023
multi-lop	1.29865	27.24%	2037988	303689	288436	123263	133508	33022
pangloss	1.30125	27.49%	4318091	249016	1089876	157160	161617	35090
sangam++	1.30148	27.51%	2953628	254919	473970	124098	127978	33164
t-skid	1.28837	26.23%	2017544	231862	428674	98321	217736	32648
IPCP++	1.29665	27.14%	3015958	247086	470910	135395	144170	34055

Table 13: Simulations for the trace 628.pop2_s-17B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	0.529114	–	3008244	6390	15415	3180	36	3179
berti	0.530686	0.29%	3033998	9013	20461	3732	783	5210
bingo	0.530478	0.26%	3025837	9015	27819	4708	253	4700
bouquet	0.530571	0.28%	3900964	41525	63846	8694	843	8597
enhancing	0.530636	0.29%	3935834	14397	29660	8605	1230	9344
multi-lop	0.530507	0.26%	3008975	13294	34447	7381	675	7337
pangloss	0.530895	0.34%	5319383	30966	176206	27390	3710	25491
sangam++	0.530751	0.31%	3979104	31837	51683	10281	1023	10141
t-skid	0.530539	0.27%	3006854	48002	54607	12420	1849	13236
IPCP++	0.530795	0.32%	4148569	65094	98987	17053	2185	16498

Table 14: Simulations for the trace 631.deepsjeng_s-928B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	2.26129	–	422187	89012	88764	391	0	391
berti	2.49089	10.15%	534258	90386	155933	309	272	310
bingo	2.47264	9.35%	526210	91341	107566	446	3	443
bouquet	2.48263	9.79%	1203541	122549	408491	3338	2948	395
enhancing	2.34682	3.78%	605646	100898	228194	1799	973	860
multi-lop	2.39248	5.80%	488087	105934	139022	464	0	464
pangloss	2.49000	10.11%	2180427	104249	1044432	7739	6758	981
sangam++	2.49188	10.19%	611616	105330	310406	577	1	576
t-skid	2.49203	10.20%	535261	102574	438901	1214	36255	1611
IPCP++	2.48499	9.89%	1211546	123171	362087	23852	23272	780

Table 15: Simulations for the trace 638.imagick_s-10316B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	0.369288	–	3263154	19618	22009	1291	6	1287
berti	0.371077	0.48%	3282503	25813	32798	2306	1310	2755
bingo	0.371513	0.60%	3286261	23254	40362	2799	402	2619
bouquet	0.371227	0.53%	4302682	42882	84333	2854	136	2737
enhancing	0.370421	0.31%	4151737	21942	38614	2700	838	2657
multi-lop	0.371116	0.49%	3268073	55744	87973	3316	318	3097
pangloss	0.371313	0.55%	7534037	41562	229544	10054	5640	4657
sangam++	0.371387	0.57%	4022092	48347	84411	3437	310	3152
t-skid	0.371318	0.55%	3271939	57787	66885	3878	616	3776
IPCP++	0.371543	0.61%	4510296	69016	133274	5655	1427	4395

Table 16: Simulations for the trace 641.leela_s-800B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	0.676815	–	3165561	43265	56197	2268	0	2268
berti	0.686767	1.47%	3224462	44370	66095	2378	1972	2424
bingo	0.686397	1.42%	3232734	45042	65183	2533	31	2533
bouquet	0.687080	1.52%	3538250	48350	201553	2584	0	2584
enhancing	0.686753	1.47%	3837642	46900	108024	2710	85	2712
multi-lop	0.687113	1.52%	3207553	69158	83415	2478	25	2478
pangloss	0.687278	1.55%	6426537	51661	456582	2902	9	2893
sangam++	0.687119	1.52%	4277947	49786	156189	2605	0	2605
t-skid	0.686980	1.50%	3207350	52553	173572	2334	18759	2587
IPCP++	0.687158	1.53%	3626239	50161	175320	2883	20	2883

Table 17: Simulations for the trace 644.nab_s-5853B.champsimtrace

30191

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	1.61182	–	2626773	60	0	60	0	60
berti	1.61202	0.012%	2626804	65	10	64	10	65
bingo	1.61185	0.002%	2626774	60	8	61	6	61
bouquet	1.61215	0.021%	3745616	73	21	72	0	72
enhancing	1.61206	0.015%	3668490	66	14	62	2	62
multi-lop	1.61185	0.002%	2626774	60	8	61	6	61
pangloss	1.61216	0.021%	3317021	67	145	37	0	37
sangam++	1.61216	0.021%	3364093	68	65	72	0	72
t-skid	1.61217	0.022%	2626876	71	0	71	0	72
IPCP++	1.61219	0.023%	3899974	76	104	78	0	78

Table 18: Simulations for the trace 648.exchange2_s-1699B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	0.59993	–	1859436	132339	132342	87845	87690	87805
berti	1.44999	141.69%	2068843	132585	239551	87929	102068	87880
bingo	1.41465	135.81%	2066274	133381	134556	87955	87757	87870
bouquet	1.46373	143.98%	3359113	132848	431927	87969	87736	87886
enhancing	1.45905	143.21%	2726912	132786	272098	88220	88415	87995
multi-lop	1.46326	143.91%	2072321	135646	135475	88261	87966	87984
pangloss	1.46372	143.98%	5227275	134068	1244341	88918	88415	88256
sangam++	1.46455	144.12%	2414483	134142	290754	88442	88115	87981
t-skid	1.44433	140.75%	2067766	134534	220999	87928	150336	87912
IPCP++	1.46432	144.09%	3405826	132960	340669	88283	87931	88010

Table 19: Simulations for the trace 649.fotonik3d_s-1176B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	1.04191	–	1680278	603	231	603	0	603
berti	1.04486	0.283%	1681894	608	540	612	335	612
bingo	1.04480	0.277%	1681756	591	241	595	0	595
bouquet	1.04486	0.283%	1935436	706	1547	698	0	698
enhancing	1.04485	0.282%	2231070	610	616	611	9	610
multi-lop	1.04475	0.273%	1681607	815	494	720	0	720
pangloss	1.04483	0.280%	3735654	628	4232	642	0	642
sangam++	1.04487	0.284%	1900881	614	971	615	0	615
t-skid	1.04486	0.283%	1681849	675	566	617	2805	629
IPCP++	1.04486	0.283%	1983009	695	1202	694	0	694

Table 20: Simulations for the trace 654.roms_s-842B.champsimtrace

Prefetchers	IPC	Speedup	L1D		L2C		LLC	
			Hits	*Misses*	*Hits*	*Misses*	*Hits*	*Misses*
default	0.758470	–	1679541	46393	64038	18390	10798	16183
berti	0.785037	3.50%	1706907	53603	75945	21569	20112	26419
bingo	0.783408	3.29%	1703705	59659	108925	51878	31873	36781
bouquet	0.785740	3.59%	2040422	103868	143862	50337	30340	34822
enhancing	0.781891	3.09%	2058685	53562	81779	24222	19597	25991
multi-lop	0.779525	2.78%	1684052	54062	91135	33212	21455	25379
pangloss	0.780476	2.90%	2388990	85689	297676	141142	67872	93086
sangam++	0.788157	3.91%	2124020	104811	152563	56142	32762	37897
t-skid	0.784730	3.46%	1683922	151607	153116	51677	34700	40239
IPCP++	0.786002	3.63%	2112233	116003	140586	48573	30191	34889

Table 21: Simulations for the trace 657.xz_s-3167B.champsimtrace