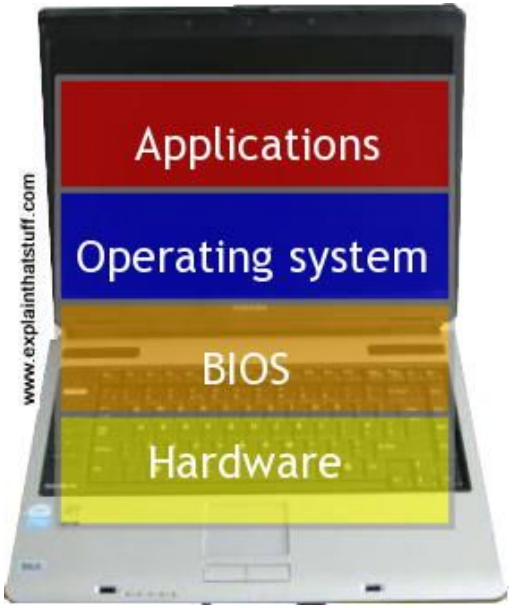
CS531: Memory System and Architecture

Course Instructor:

Dr. Shirshendu Das
Assistant Professor,
Department of CSE,
IIT Ropar.

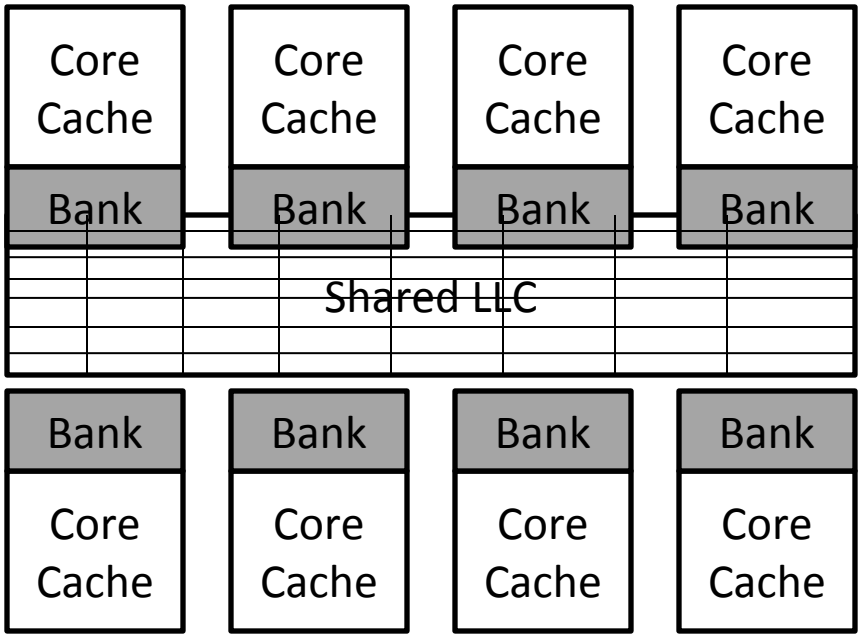
shirshendu@iitrpr.ac.in
http://cse.iitrpr.ac.in/shirshendu/shirshendu.html

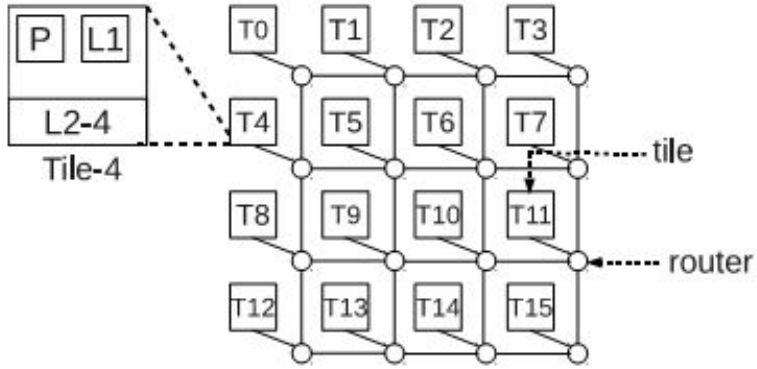


Topic: Utilisation Issues in Last Level Cache (LLC)

Introduction

- Though the LLC has larger size it cannot utilize the whole storage properly.
- Better utilization of LLC reduces the miss rate and hence improve performance.

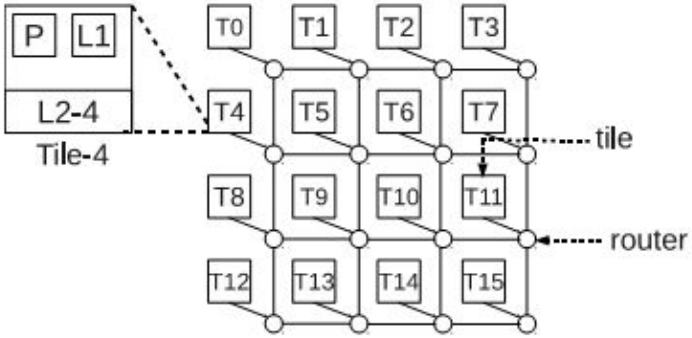




TCMP

Utilisation Issues in LLC

- **Though the LLC has larger size it cannot utilize the whole storage properly.**
- **Better utilization of LLC reduces the miss rate and hence improve performance.**



- **Current LLC utilization issues:**
  - Local issue (*local to each bank*)
    - The sets within a bank are not used uniformly [1].
  - Global issue (*considering all the banks*)
    - The banks are not loaded uniformly.

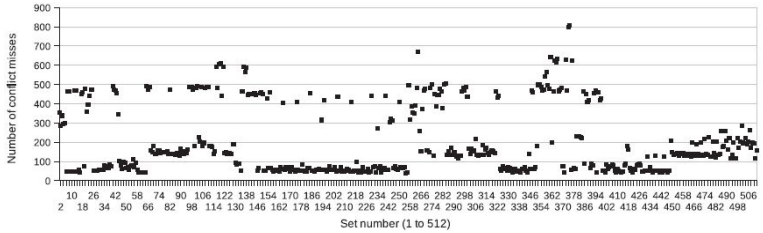


Figure 2: Non uniform load distribution within a bank.

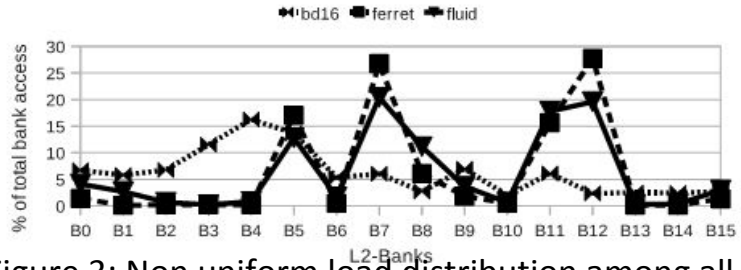
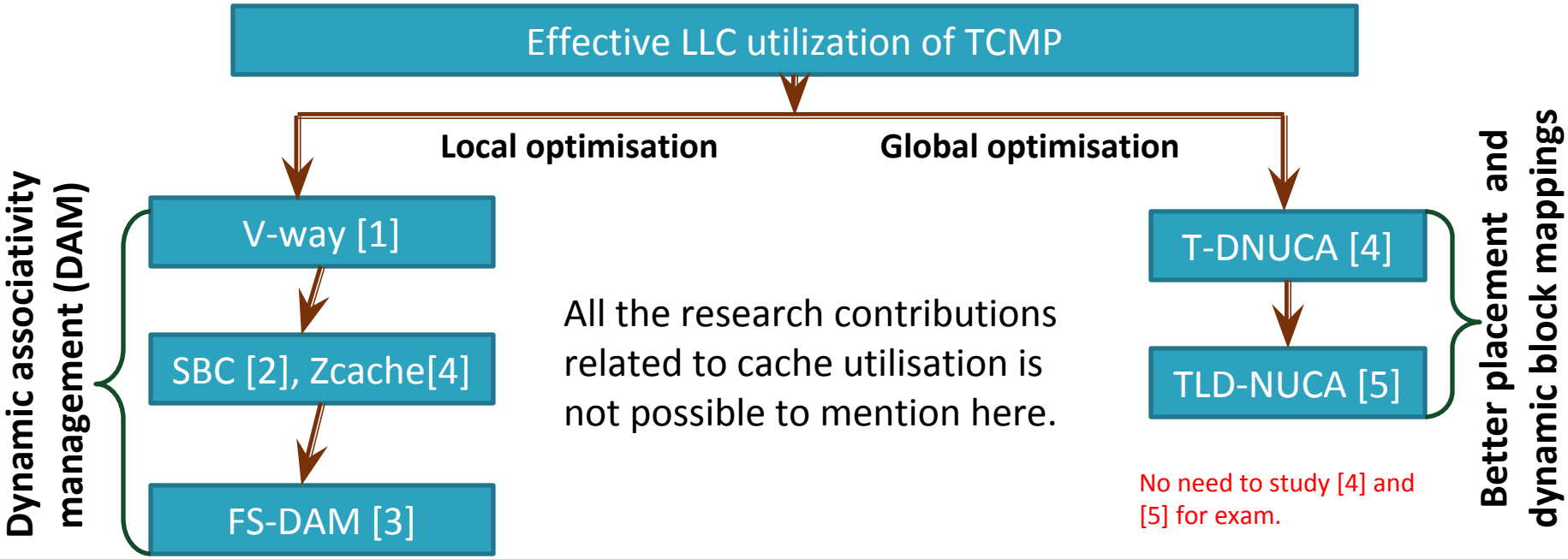


Figure 3: Non uniform load distribution among all the banks.

[1] M. K. Qureshi, D. Thompson, and Y. N. Patt, "The V-Way Cache: Demand Based Associativity via Global Replacement," *ACM SIGARCH Computer Architecture News*, vol. 33, no. 2, pp. 544–555, May. 2005.

Works for better LLC Utilization



[1] M. K. Qureshi, D. Thompson, and Y. N. Patt, "The V-Way Cache: Demand Based Associativity via Global Replacement," *ACM SIGARCH Computer Architecture News*, vol. 33, no. 2, pp. 544–555, May. 2005.

[2] D. Rolan, B. B. Fraguela, and R. Doallo, "Adaptive Line Placement ´ with the Set Balancing Cache," in Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2009, pp. 529–540.

[3] **Shirshendu Das** and H. K. Kapoor, "Dynamic Associativity Management in Tiled CMPs by Runtime Adaptation of Fellow Sets," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 28(8), 2017.

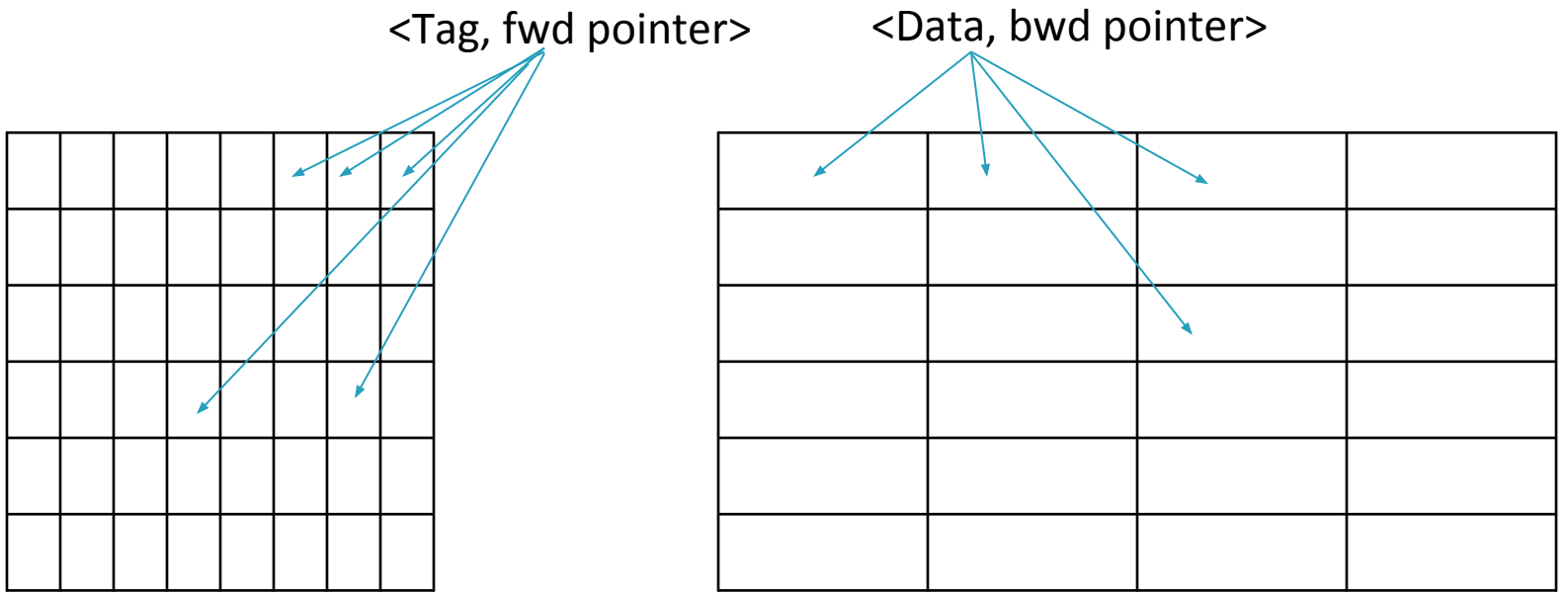
[4] D. Sanchez and C. Kozyrakis, "The ZCache: Decoupling Ways and Associativity," in Proceedings of the 43rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2010, pp. 187–198.

[5] **Shirshendu Das** and Hemangee K. Kapoor, "Exploration of Migration and Replacement Policies for Dynamic NUCA over Tiled CMPs," *28th International Conference on VLSI Design-2015 (VLSID 2015),* Bangalore, India.

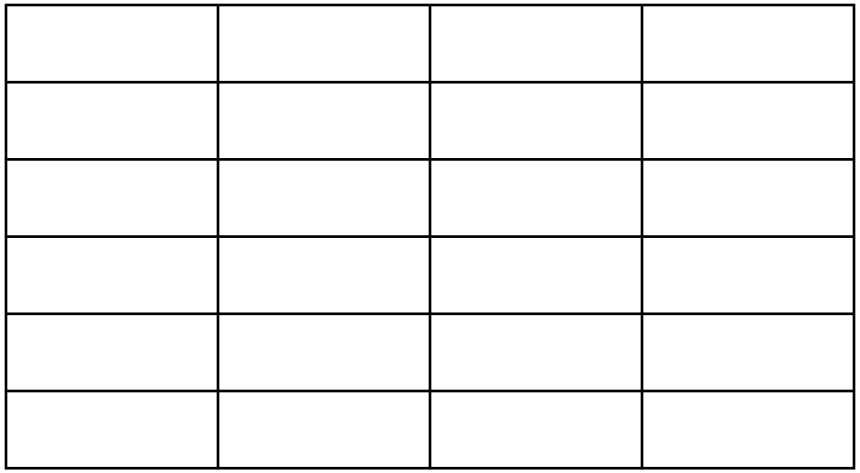
[6] **Shirshendu Das** and H. K. Kapoor, "A Framework for Block Placement, Migration and Fast Searching in Tiled-DNUCA Architecture," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 22(1), 2016

V-Way

- V-way is a DAM based technique proposed for LLC. In V-Way tag storage and data storage are separated.
- The associativity of tag storage is double compared to the associatovity of data storage.
- The traditional one-to-one mapping between tag and data storage is replaced with forward and backward pointers.
  - ✔ Forward pointer points a location in data storage.
  - ✔ Backward pointer ponits the corresponding location in tag storage.



SBC



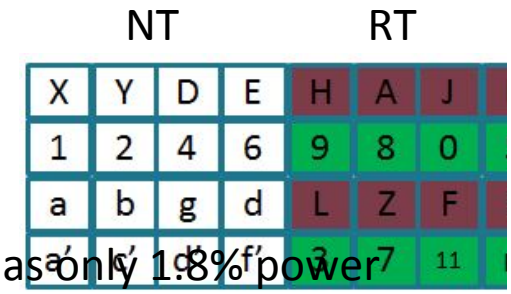
I will share a PDF where V-way and SBC are explained briefly.

CMP-SVR*

- **In CMP-SVR the sets are divided into two sections: NT and RT.**
- **It also divides the sets into some groups called fellow-groups and all fellow-groups are non-overlapped, i.e., a set cannot be in more than one fellow-group.**

- **A heavily used set can use the RT section of another fellow-set.**
- **To reduce the additional hardware overhead we use an additional tag array (SA-TGS) to handle RT.**

- **Each entry in TGS has a corresponding location in RT.**



Figur

- CMP-SVR has only 1.8% power overhead as compared to baseline.
- It improves performance by 6%.

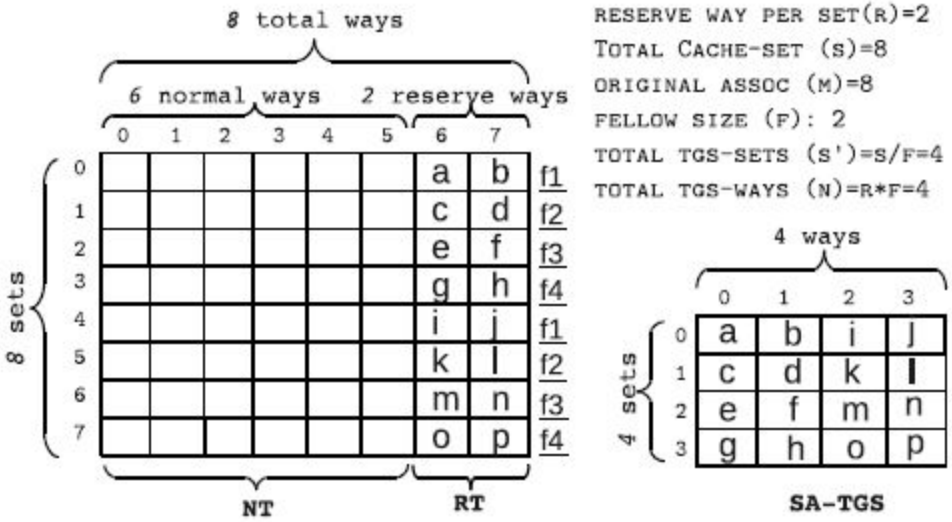


Figure 6: CMP-SVR: way distribution, fellow sets and associative mapping into SA-TGS.

FS-DAM*

- **For better performance with less hardware overhead we extend the concept of CMP-SVR to FS-DAM.**

- **Motivation of FS-DAM:**
  - Non uniform fellow-group usage in CMP-SVR.
  - CMP-SVR is unable to adjust with the changing set behavior.

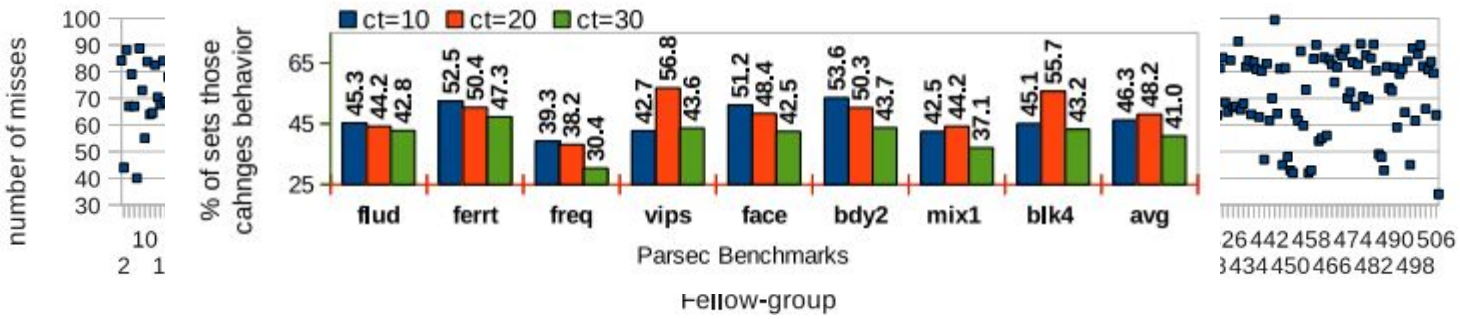


Figure 7 : Example of non-uniform fellow-group usage.

Figure 8: The percentage of sets changing category dynamically after every calculation interval (*ct*)

* **Shirshendu Das** and H. K. Kapoor, "Dynamic Associativity Management in Tiled CMPs by Runtime Adaptation of Fellow Sets," *IEEE Transactions on Parallel and Distributed Systems (TPDS) , accepted in January 2017*

FS-DAM

- The fellow-groups are created based on the current loads.
- The lightly loaded sets are equally distributed to all the groups.

- The groups are reorganized after a fixed cycles of execution.
- Some additional mechanisms are required to maintain one-to-one mapping between additional tag array and RT locations.

- The hardware overheads of the additional mechanisms is assumed to be negligible.

- **FS-DAM has three major operations:**
  - Initialization.
  - Normal Execution.
  - Re-grouping.

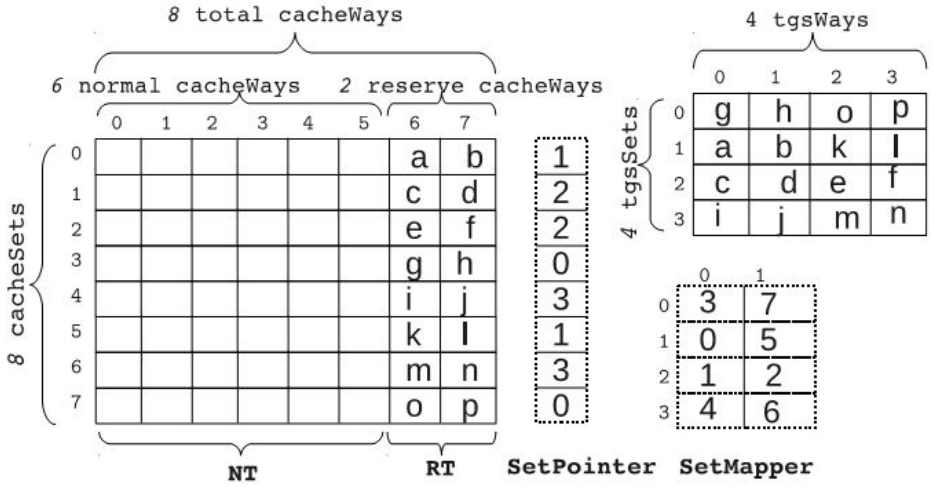
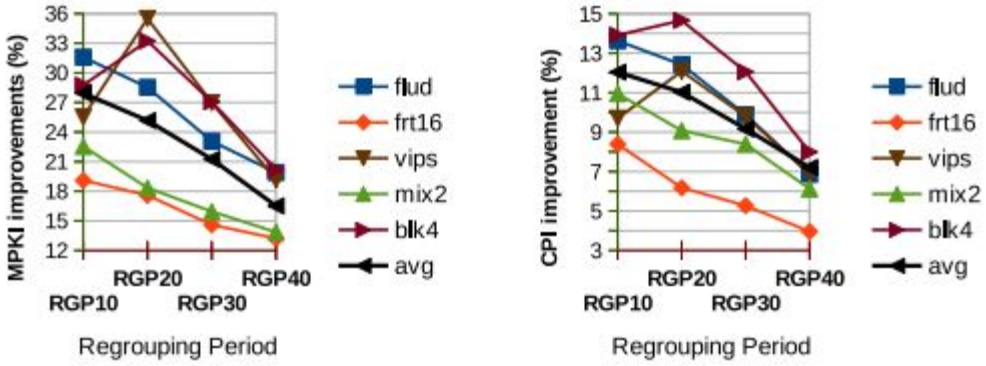


Figure 9*: Example of FS-DAM.*

FS-DAM: Regrouping Process



(a) MPKI (**C1**). (b) CPI (**C1**).

Figure 10 : Improvements of FS-DAM over baseline for different values of RGP.

Cache Size	Assoc	F	*rg-time* in % of total exec time	MPKI	CPI
2MB	4	2	0.23	26.35	10.82
		4	0.30	27.94	12.04
		8	0.48	28.97	12.58
4MB	4	2	0.47	29.78	12.14
		4	0.57	32.56	13.47
		8	0.92	33.13	13.79

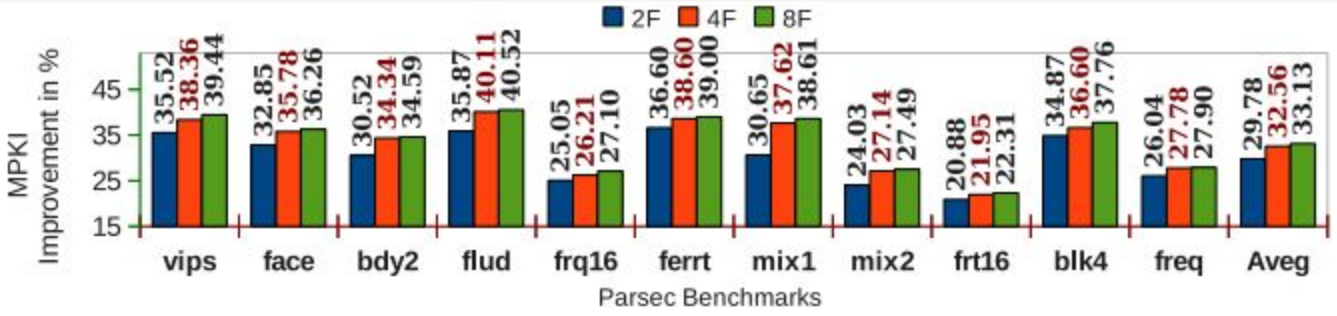
Table 1: Time required for re-grouping in FS-DAM.

FS-DAM: Experimental Analysis

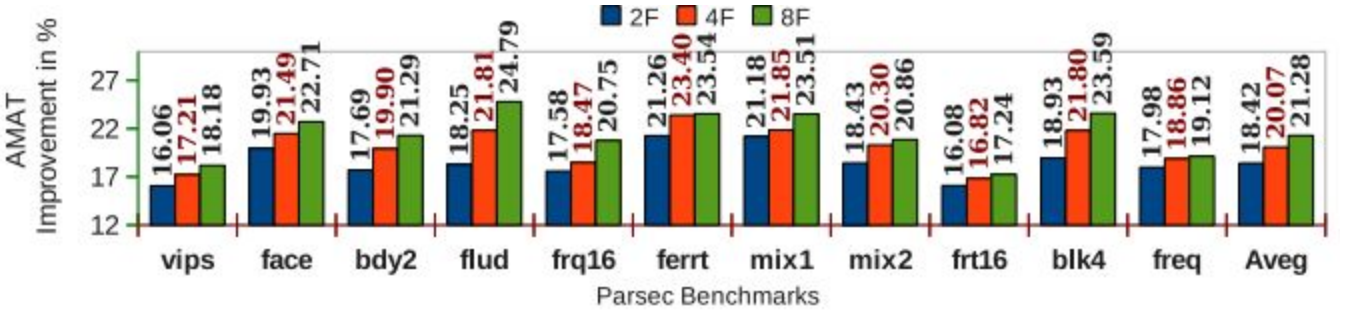
Configurations:

C1: 2MB LLC
128KB banks
4-way associative

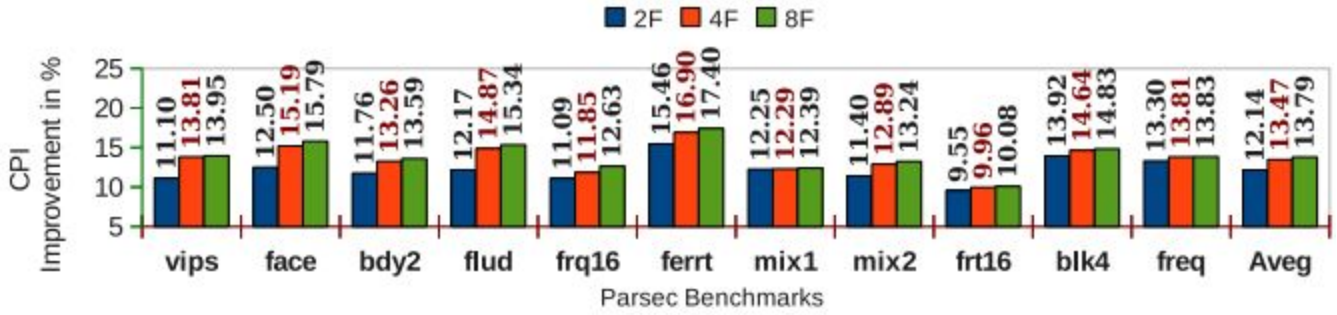
C2: 4MB LLC
256KB banks
4-way associative



(a) MPKI (**C2**).



(b) AMAT (**C2**).



(c) CPI (**C2**).

Figure 14 : Improvements in FS-DAM over baseline TCMP. *x*F means FS-DAM with fellow-group size *x*.

FS-DAM: Experimental Analysis

- **FS-DAM is also compared with V-Way [1], Z-Cache [3] and SBC [2].**

Improvements over	Improvements in 2MB LLC (C1)			Improvements in 4MB LLC (C2)		
	MPKI (in %)	*AMAT (in %)*	*CPI (in %)*	*MPKI (in %)*	*AMAT (in %)*	*CPI (in %)*
Baseline	27.93	16.53	12.04	32.56	20.06	13.47
CMP-SVR	13.56	09.78	05.95	16.74	10.34	06.62
V-Way	14.23	8.90	06.31	17.56	11.01	07.39
Z-Cache	12.74	7.80	05.28	14.65	9.93	06.21
SBC	12.56	7.65	05.41	14.83	9.63	06.07

Table 2: Improvements (in %) of FS-DAM over baseline design and the other existing techniques: CMP-SVR, V-Way, Z-Cache and SBC.

[1] M. K. Qureshi, D. Thompson, and Y. N. Patt, "The V-Way Cache: Demand Based Associativity via Global Replacement," ACM SIGARCH Computer Architecture News, vol. 33, no. 2, pp. 544–555, may 2005.

[2] D. Rolan, B. B. Fraguela, and R. Doallo, "Adaptive Line Placement ´ with the Set Balancing Cache," in Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2009, pp. 529–540.

[3] D. Sanchez and C. Kozyrakis, "The ZCache: Decoupling Ways and Associativity," in Proceedings of the 43rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2010, pp. 187–198.

FS-DAM: Experimental Analysis

❖ Hardware Analysis:

	vips	face	bdy2	flud	frq16	ferrt	mix1	mix2	frt16	blk4	freq	Average
Improvement over main memory access (MMA)	30.01	29.15	30.10	32.97	21.84	30.79	32.72	20.27	16.77	31.68	21.36	26.43
Improvement ovr dy. energy consumption	28.73	27.08	20.62	17.15	20.46	27.62	28.73	19.18	15.97	14.95	20.03	21.30

Table 6: The energy overhead of FS-DAM over the baseline design.

Cache Configuration with bank size	Storage overhead (in %)						Area overhead (in %)					
	R25			R50			R25			R50		
	F=2	F=4	F=8	F=2	F=4	F=8	F=2	F=4	F=8	F=2	F=4	F=8
C1(128KB, 4-way)	1.89	3.03	5.39	3.03	5.34	10.11	1.64	1.83	1.83	3.13	3.50	3.55
C2(256KB, 4-way)	1.98	3.11	5.48	3.11	5.43	10.20	1.77	1.77	1.78	3.38	3.40	3.48

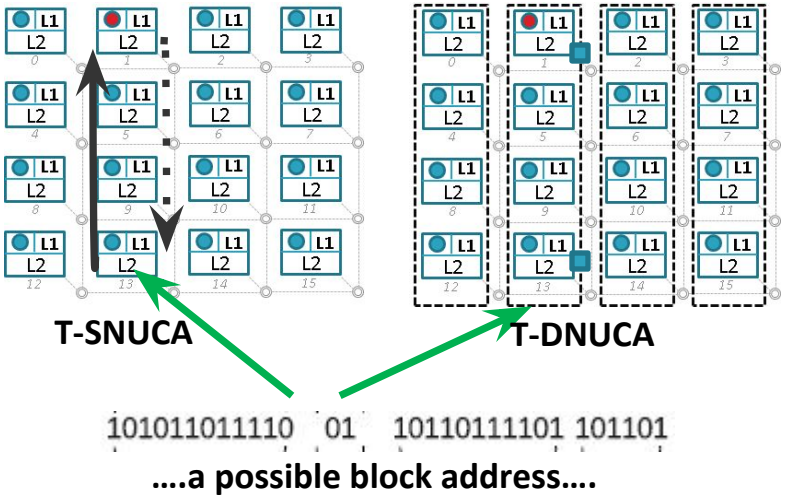
Table 7: Storage and Area overhead of FS-DAM over baseline.

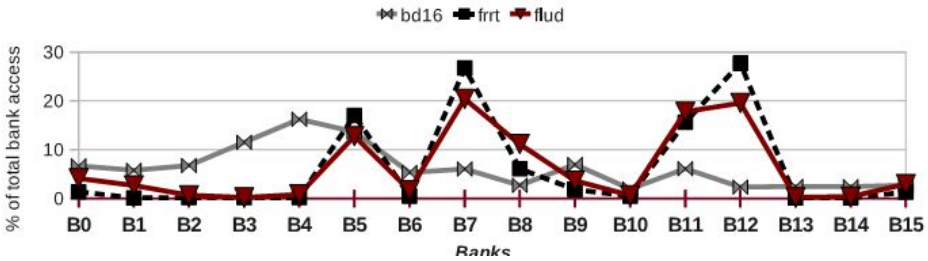
❖ Energy and area overheads are calculated by CACTI 6.0 [1].

[1] N. Muralimanohar, R. Balasubramonian, and N. Jouppi, “Optimizing NUCA Organizations and Wiring Alternatives for Large Caches with CACTI 6.0,” in Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2007, pp. 3–14.

Global policies for better LLC utilization [not for exam]

- **Both Static NUCA (SNUCA) and Dynamic NUCA (DNUCA) can be implemented on TCMP.**





Comparison of bank usages in T-SNUCA.

- **DNUCA performs better than SNUCA if efficient searching mechanism can be implemented [1].**
- **The loads can also uniformly distributed among the banks to increase the utilization.**
- **T-SNUCA is well explored but not T-DNUCA.**
- **Hence, we are motivated to design a DNUCA based TCMP (T-DNUCA) for better LLC utilization as well as performance.**

[1] R. Balasubramonian, N. P. Jouppi, and N. Muralimanohar, Multi-Core Cache Hierarchies. *Morgan and Claypool Publishers*, 2011.

T-DNUCA [not for exam]

- **In T-DNUCA the tiles are divided into multiple banksets.**
  - A block can be placed in any bank within the bankset.
  - Loads can be distributed among the multiple banks for better utilization.

- **Proper block searching mechanism is introduced to reduce search time.**
- **Existing DNUCA techniques cannot be directly applied to T-DNUCA.**

- **T-DNUCA has the following major operations:**
  - Block search in bankset
  - Block placement
  - Block replacement
  - Block migration

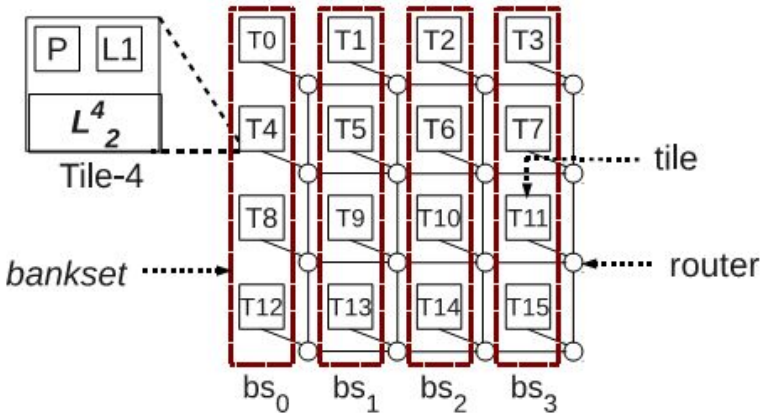
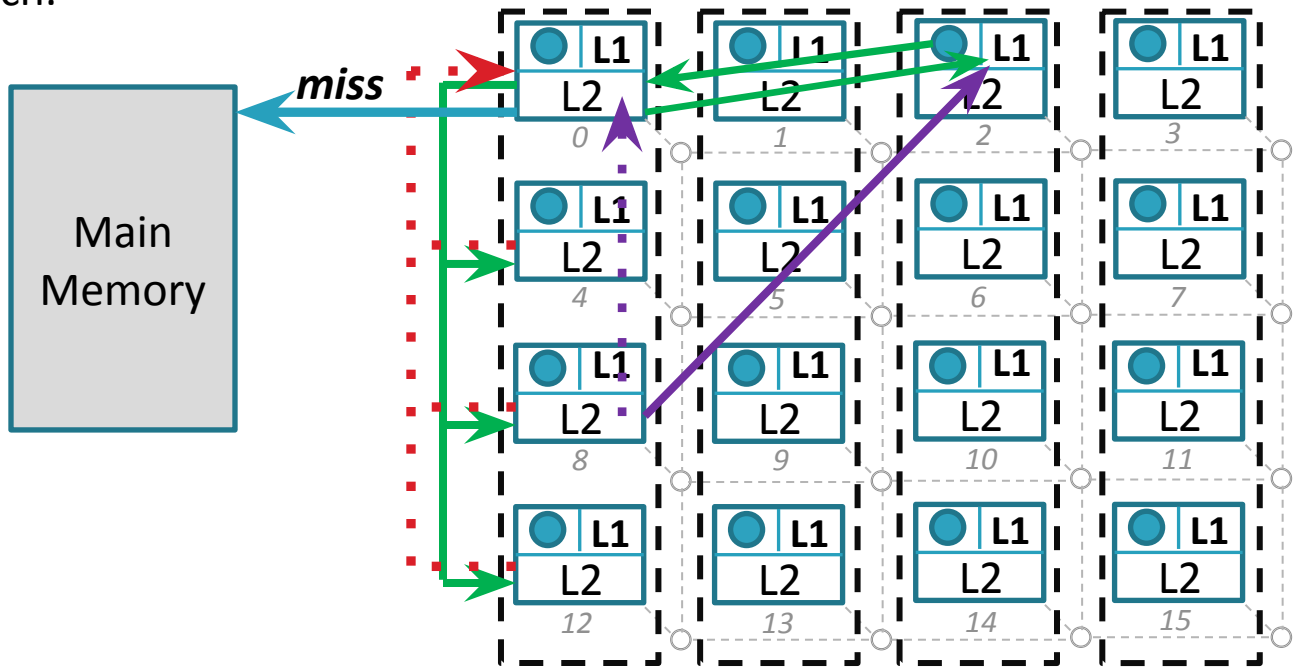


Figure 19: T-DNUCA.

* **Shirshendu Das** and Hemangee K. Kapoor, "Exploration of Migration and Replacement Policies for Dynamic NUCA over Tiled CMPs," *28th International Conference on VLSI Design-2015 (VLSID 2015),* Bangalore, India.

T-DNUCA [not for exam]

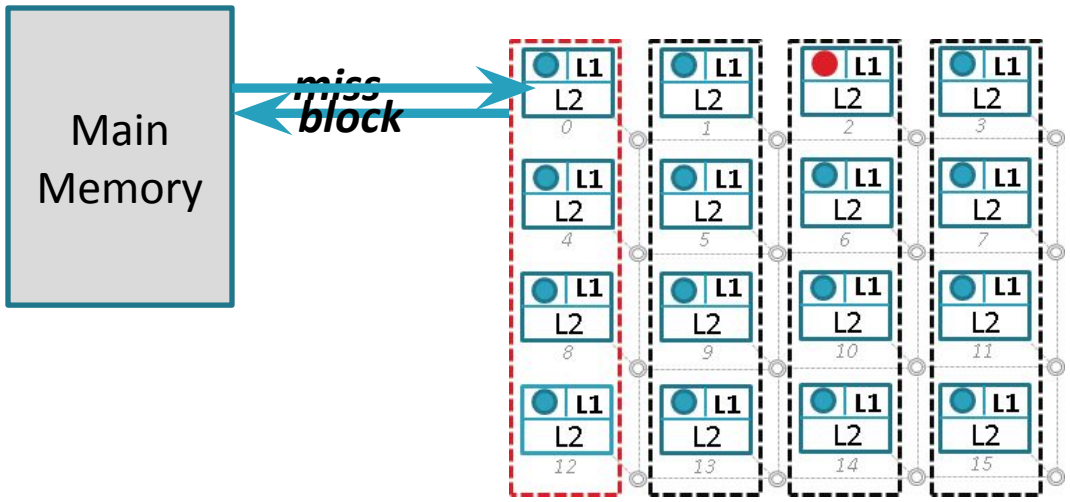
- **T-DNUCA Block Search:**
  - Local Search.
  - Multicast Search.



- **Multicast search is expensive in terms of network bandwidth and power consumption.**
- **We proposed a mechanism to minimize such multicast searches.**

T-DNUCA: Initial Block Placement [not for exam]

- **The newly incoming block from main memory are placed in its *home_bank*.**



- **Most of the blocks are found in the home-bank .**

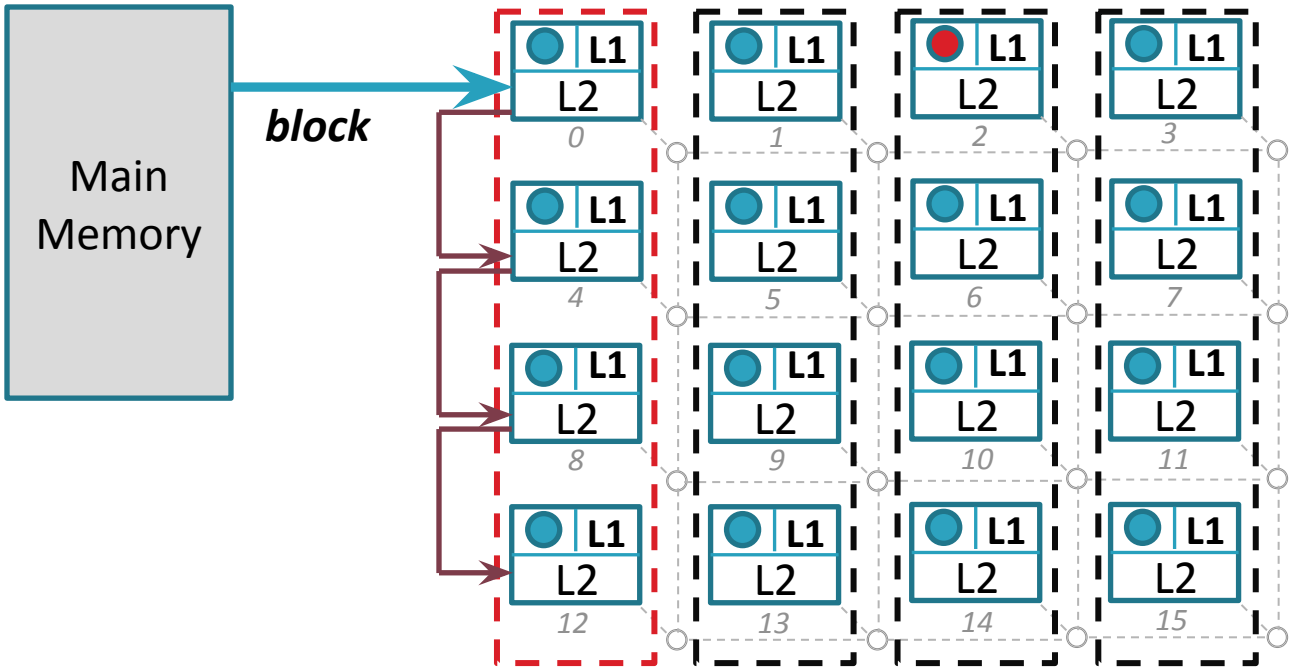
- **Hence less number of multicast searches are affordable without using any smart-search techniques as used in previous DNUCA designs [1, 2].**

[1] J. Huh, C. Kim, H. Shafi, L. Zhang, D. Burger, and S. W. Keckler, "A NUCA substrate for flexible CMP cache sharing," *in Proceedings of the 19th annual international conference on Supercomputing (ICS)*, 2005, pp. 31–40.

[2] B. M. Beckmann and D. A. Wood, "Managing Wire Delay in Large ChipMultiprocessor Caches," *in Proceedings of the 37th annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2004, pp. 319–330.

T-DNUCA: Replacement Policy [not for exam]

- **T-DNUCA has two types of replacement policies:**
  - Local replacement (LRU): local to each bank.
  - Cascading replacement: global to entire bankset.

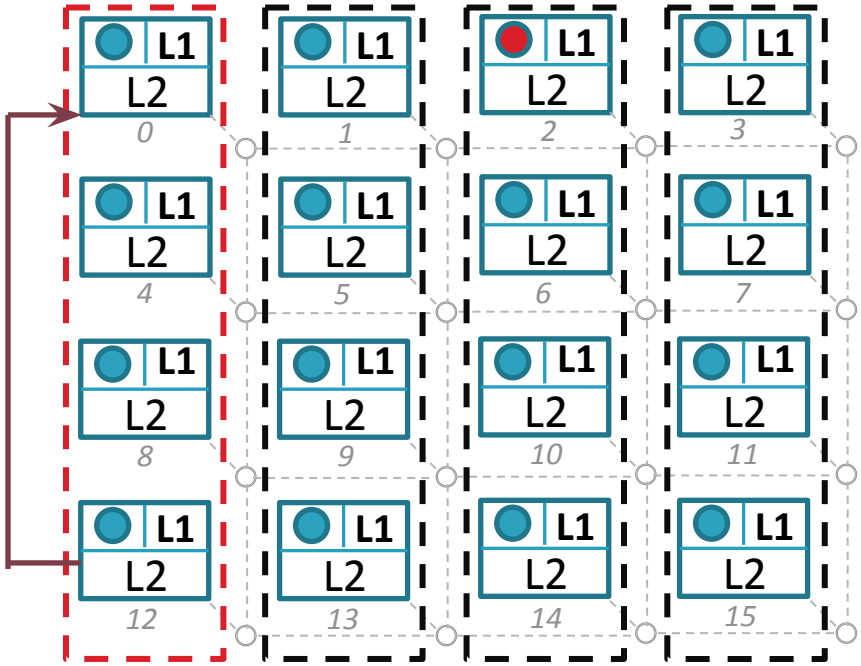


- **Cascading replacement helps to distribute the loads of heavily used banks with other peer banks.**
- **But cascading replacement requires block movement , which consumes some additional power.**

T-DNUCA: Block Migration [not for exam]

- **Gradual block migration as proposed in DNUCA is not effective here because in TCMP each bank is associated with a core.**

- ***T-DNUCA migrate a block directly to the local bank of the requesting core.***



- **Migration may evict a victim block (V), which can be:**
  - removed from the cache.
  - moved using cascading replacement.
  - swapped with sender.

- **The requirement of migration is less in T-DNUCA for our proposed placement policy.**

Extension of T-DNUCA: TLD-NUCA* [not for exam]

- **The improvements possible in T-DNUCA:**
  - Reducing the mandatory home-bank search time:
    - In T-DNUCA the home-bank can be far from the requesting core.
    - Making the local-bank as home-bank can reduce the local search time.
  - Better bank utilization:
    - T-DNUCA allows load distribution within bankset.
    - Better utilization possible if all the banks are allowed to share loads.

- **Therefore we proposed an extension of T-DNUCA called TLD-NUCA*.**

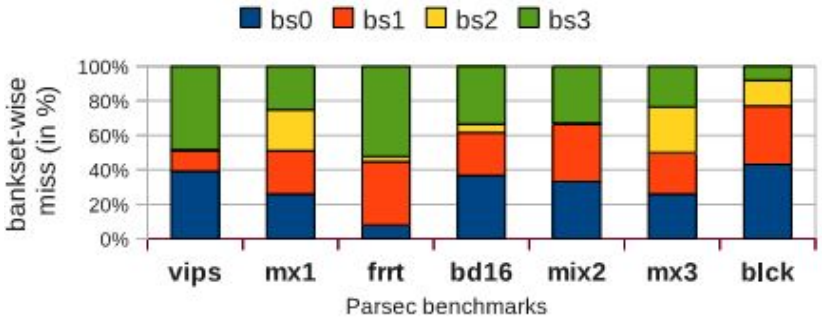


Figure 24: Load distribution among the banksets of T-DNUCA.

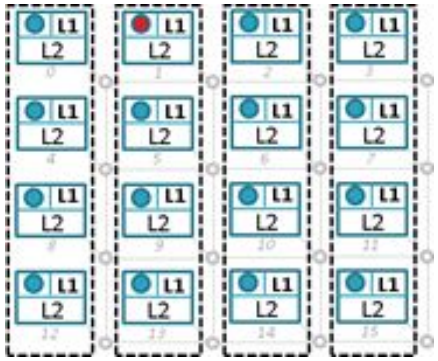


Fig 25: T-DNUCA.

* **Shirshendu Das** and H. K. Kapoor, "A Framework for Block Placement, Migration and Fast Searching in Tiled-DNUCA Architecture," *ACM Transactions on Design Automation of Electronic Systems (TODAES)* , 22(1), 2016.

Extension of T-DNUCA: TLD-NUCA [not for exam]

- **TLD-NUCA:**
  - Only one bankset.
  - The loads can be distributed among all the banks through cascaded replacement.
  - The *local-bank* becomes *home-bank* for each core.
  - Initial placement policy places the block in the *local-bank*.
  - Reduction in multicast search time:
    - There is a center tag directory storing the tag and the pointer of the bank where the block is.
    - The center tag is called Tag Lookup Directory (TLD).

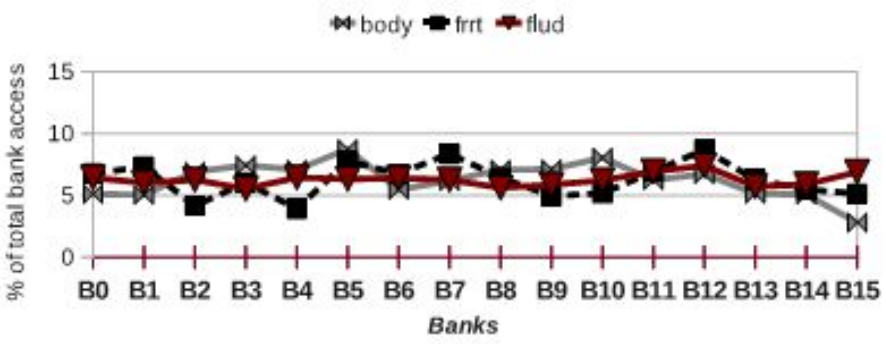


Figure 28: Bank usage in TLD-NUCA.

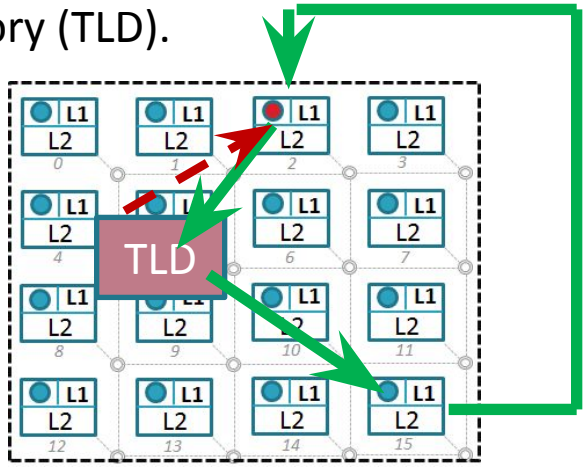
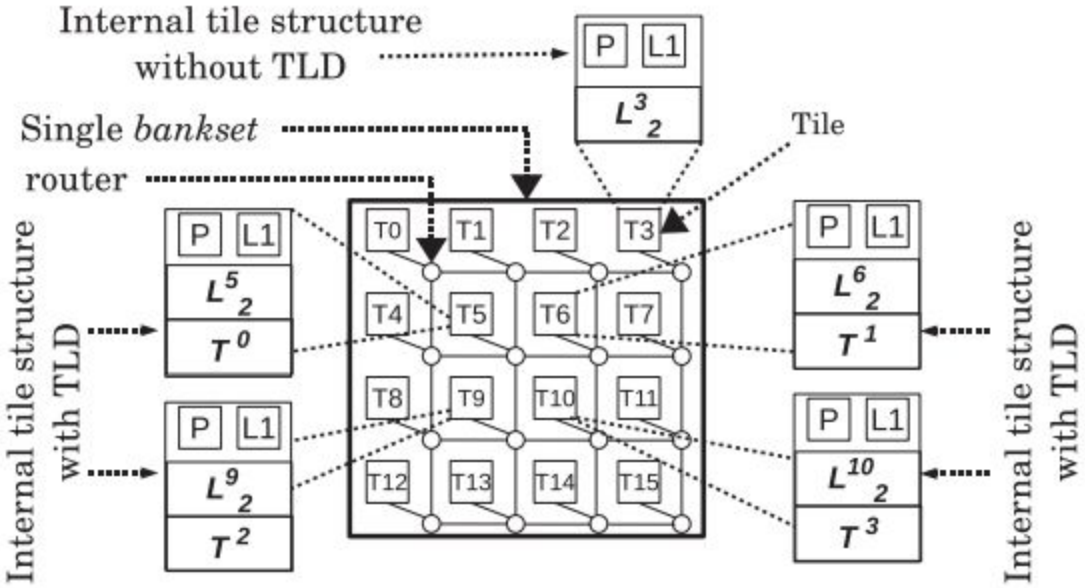
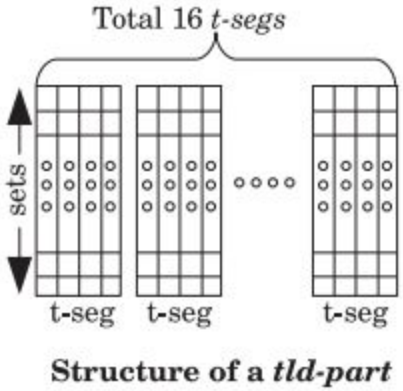


Figure 29: TLD-NUCA.

TLD-DNUCA [not for exam]





Structure of a *tld-part*

Thank You