Design and Verification of High-Performance Computing Systems

Shivani Tripathy* (A16EE09004) Supervisor: Dr. Manoranjan Satpathy

*Email id: st15@iitbbs.ac.in

School of Electrical Sciences (Computer Science)

Access nplification

INTRODUCTION

- A high-performance computing system consists of several types of computing elements like CPU cores and GPU. The computing elements depend on a multi-level memory hierarchy including SRAM/DRAM caches, main memory, and secondary storage like SSD.
- Memory is often the bottleneck in achieving optimum performance.
- Reliablity of a system depend on functional correctness of the memory system.
- Fairness of the system can be affected by the memory system design.

OBJECTIVES

- Minimizing the number of memory accesses.
- 2. Finding out a pattern in memory accesses.
- 3. Modeling and verification of memory systems.
- 4. Maximizing fairness in servicing memory requests.

RESULTS AFTER REDUCING ACCESSES optimal

Thread Thread Block Block (0,0) (1,0) Block Warp 0 (0,1) (1,1) Block Warp 2 (1,2) Thread Thread Block Block CPU code (0,3) (1,3) thread 31 thread 1

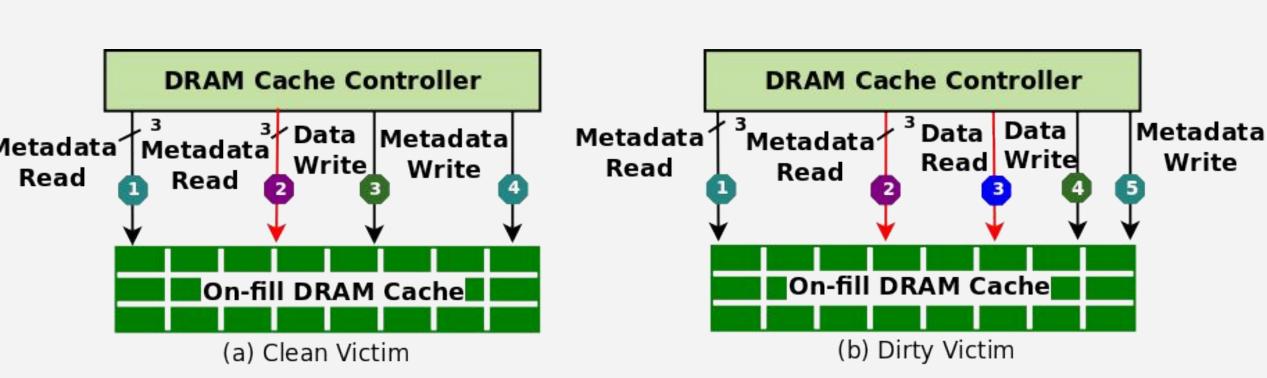
2. MEMORY ADDRESS PREDICTION IN GPUS

 Consecutive thread blocks can be assigned to different SMs.

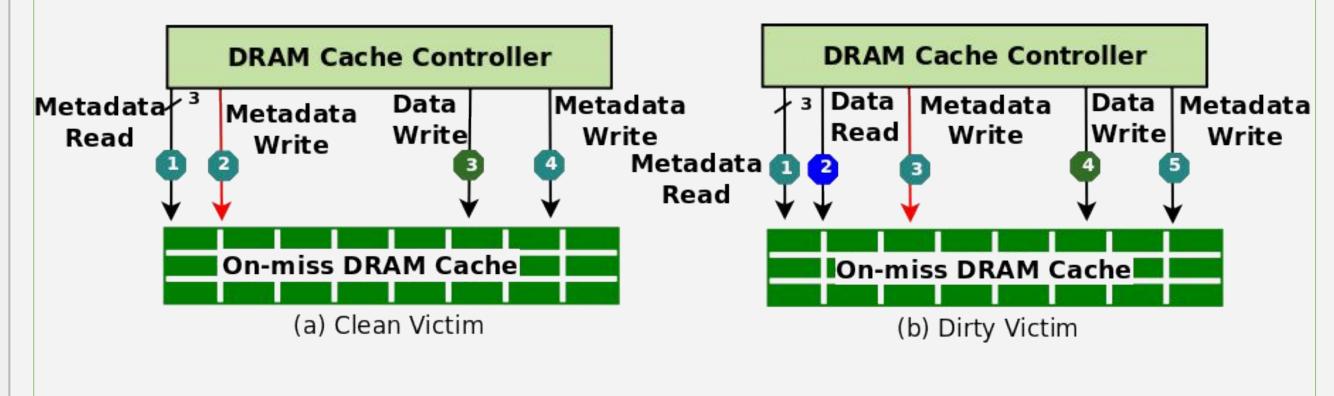
b2

- Strided memory access pattern in warps in a thread block.
- For multidimensional grid, strides among thread blocks along different dimensions are different.
- Consideration of thread block id can help in predicting memory access.

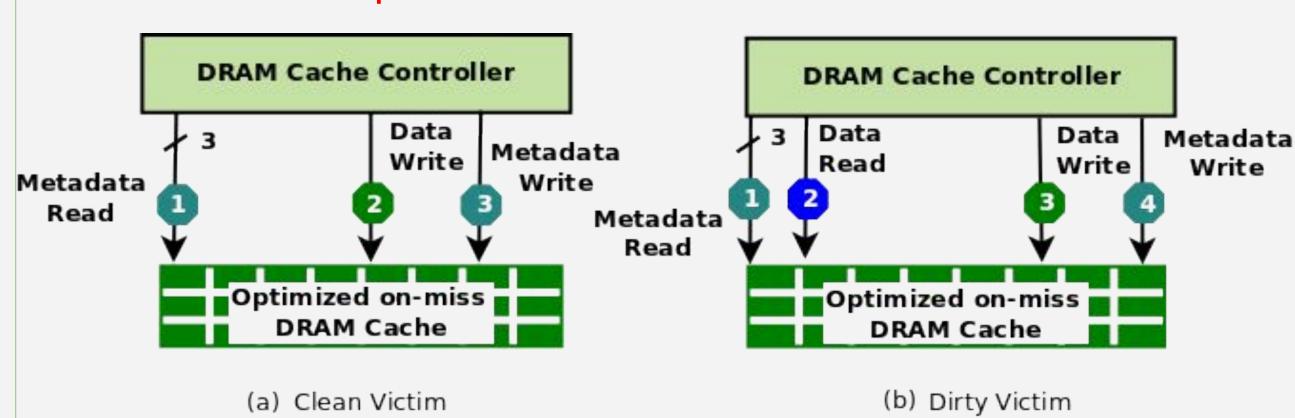
1. REDUCING CACHE ACCESS Translates **Metadata Write** Data Read **Metadata Read** Read Access Access Request Access Translates **Metadata Write Data Write** Write Metadata Read Access Request Access Access Translates **Data Write Metadata Write** Data Read Fill **Metadata Read** Access Access Access Request Access



- Extra meta data access at the time of fill.
- Delayed fill due to victim data read.

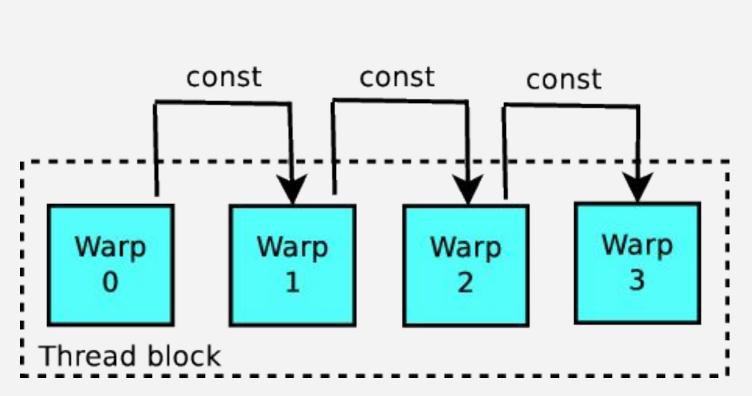


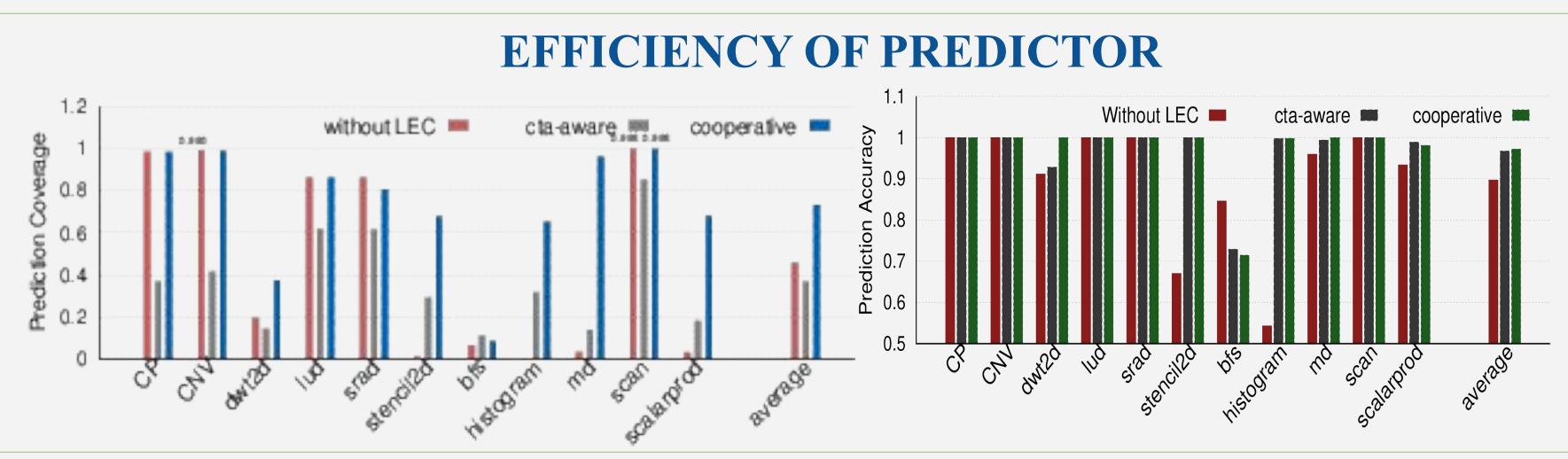
- Extra meta data write to mark invalid.
- Inefficient space utilization.



- Line locking the victim tag block.
- Servicing read requests to the victim block from the cache.

MEMORY ADDRESS PREDICTION stridex stridex **Threadblock Threadblock Threadblock** (1,0)(2,0)stride W0 W1 W2 W3 W0 W1 W2 W3 **Threadblock Threadblock Threadblock** (2,1)(0,1)(1,1)stridev W0 W1 W2 W3 **Threadblock Threadblock Threadblock** (1,2)(2,2)Grid





3. FORMAL MODELING AND VERIFICATION OF NAND FLASH (Ready-to-output & SinglePlane & Singledie ReadWait ReadOut | R-CRC & SinglePlane RD-OP & (Ready-to-output | R-CRC) & addr-transferred Cache read op & SinglePlane & Singledie / | BKGRND-PG-RE addr-transferred & ReadCache ReadCache Flash-RD-DONE Flash-RD-DONE Out Wait cachereg (Reg-Done & transferred-to-cachereg (R-CRC-Enhanced | (R-CRC & BKGRND-PG-RD new-cmd-diffpage) SelPlane = Plane) / Initialize DR (R-CRC-Enhanced | (R-CRC /SelPlane = Plane & SelPlane = Plane)) Cache / Initialize CR Valid Cache Transferred-to-CR Valid RD-OUT-DONE / -**Initialize DR** Invalid READ-OUT-DONE R-PM-CONF / Initialize CR Sache program op / addr transferred / · Input-done & (!RD-CACHE-MODE) PM-CACHE & Input-Done addr transferre !cache / -/ tpage = page WriteCache We prove 91 properties associated with a functionally correct device.

CONCLUSION AND FUTURE WORK

- Line locking with on-miss allocation policy can reduce the number of DRAM cache accesses.
- Thread block id and stride along grid dimensions can help in predicting memory address references.
- Formal modeling and verification technique can be used to design a functionally correct memory system.
- Going forward, we intend to improve fairness of the memory system (objective 4).

REFERENCES

- 1. Shivani Tripathy, D. Sahoo, M. Satpathy and S. Pinisetty, 2019. Formak Modeling and verification of NAND Flash Memory Supporting Advanced Operations. ICCD. IEEE.
- Shivani Tripathy, D. Sahoo and M. Satpathy, 2019. Multidimensional Grid Aware Address Prediction for GPGPUs. VLSID. IEEE. (among top 18 papers).
- Shivani Tripathy, D. Sahoo and M. Satpathy, 2018. Work-in-Progress: DRAM Cache Access Optimization leveraging Line Locking in Tag Cache. CASES. IEEE.

