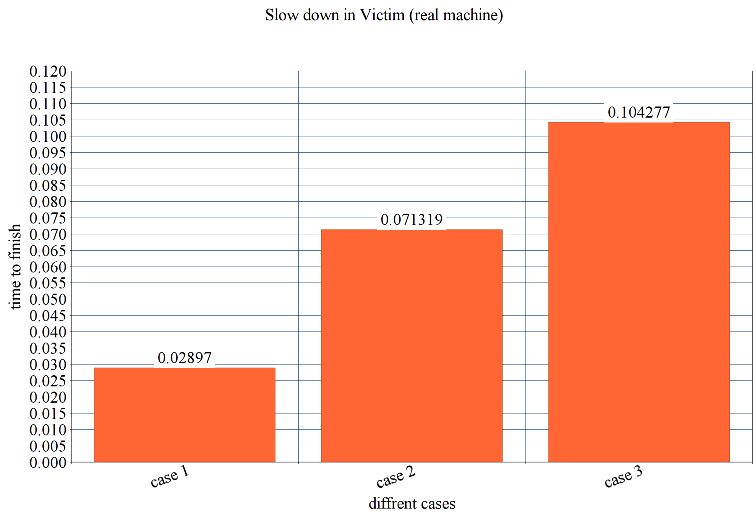
Indian Institute of Technology Kanpur
CS665 Secure Memory Systems (Sem I, 2018-19)

QUESTION 1

Assignment Number: 3
Student Name: Arun KP, Nabhiraj Jain
Roll Number: 18111263, 18111407
Date: November 27, 2018

TASK-A

DOS attack on the victim application



1. Case-1: victim is alone.
2. Case-2: victim and attacker is running on two separate physical core and attacker is running only on one of the virtual core of its corresponding physical core.
3. Case-3: victim and attacker is running on two separate physical core and attacker is running on both the virtual core of the corresponding physical core.

Attack Logic

- Today most of the DRAM controllers use FR-FCFS like scheduling policy.
- FR-FCFS policy behaves in following way.
  - It gives highest priority to the request which results in a row hit.
  - If two or more requests result in row hit, then priority amoung them is decided in FCFS order.
  - If none of the requests in the queue results in row hit then highest priority is given to the oldest request.
- We have mounted DRAM DOS attack with the assumption of FR-FCFS policy in the system.

- The attacker process continuously accesses a char array of size 512MB with a stride of 64 bytes.
- Due to this nature of attacker process, it fills the memory read queue in a way which leads to high number of row hits for the attacker process.
  Hence victim process's memory requests do not get priority due to FR-FCFS, this will create a denial of service attack on the victim.

Files Included

1. Attacker.cpp

Indian Institute of Technology Kanpur
CS665 Secure Memory Systems (Sem I, 2018-19)

Assignment Number: 3
Student Name: Arun KP, Nabhiraj Jain
Roll Number: 18111263, 18111407
Date: November 27, 2018

QUESTION 2

TASK-B
DRAM-DOS attack Mitigation in ChampSim

2-cores

Branch: bimodal
L1D_Prefetcher: no
L2C_Prefetcher: no
n_warm: 0
n_sim: 90000000
cpu-0: Attacker
cpu-1: Victim

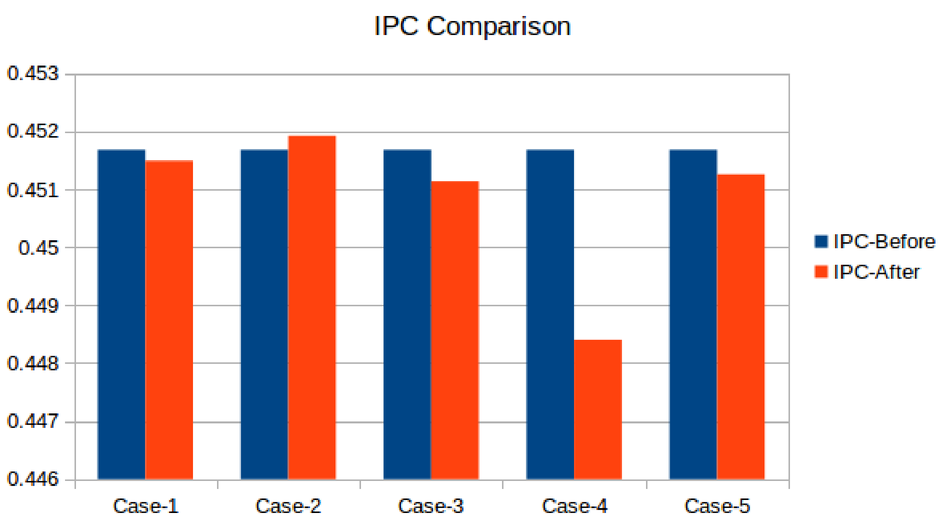


Figure 1: IPC Before and After Mitigation

1. Case-1: Bank Partitioning.
   (a) 8 banks are equally parititioned between cpus.
   (b) Each cpu gets 4 banks each.
   (c) Process running in cpu-0 is assigned to banks 0-3 and in cpu-1 is assigned to banks 4-7.
2. Case-2: Rank Partitioning.

   (a) 8 ranks are equally parititioned between cpus.
   (b) Each cpu gets 4 ranks each.
   (c) Process running in cpu-0 is assigned to ranks 0-3 and in cpu-1 is assigned to ranks 4-7.

3. Case-3: Address Scheme(RoBaRaCoCh)
   (a) DRAM address scheme is changed to Row,Bank,Rank,Column,Channel.
4. Case-4: Address Scheme(BaRaCoRoCh)
   (a) DRAM address scheme is changed to Bank,Rank,Column,Row,Channel.
5. Case-5: Non-Open Page Policy(FCFS)
   (a) DRAM schedule policy is changed to Non-Open Page.
   (b) Request are scheduled on FCFS basis rather than looking for open row.
   (c) Each requests is considered as no matching open row (row buffer miss)

Observations

1. Rank paritioning helped in mitigating DOS attack and improved IPC slighlty.
2. BaRaCoRoCh scheme degraded the IPC since rows are opened and closed frequently.
3. Bank Partitioning did not give much benefit since requests competed for limited banks.

Validation Against Stream Access

1. The migitation schemes are further evaluated against stream access based attack.
2. The attack is formalized based on stream access with fixed offset inside a large array(1; 2).
3. The pattern is same as figure 1, and it confirms that, stream access based attacks can be mitigated by rank paritioning.

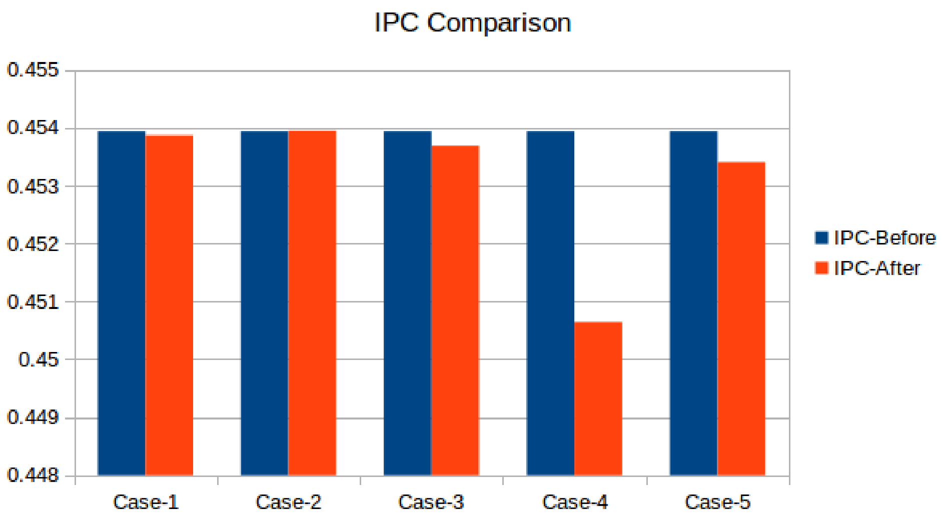


Figure 2: IPC Before and After Mitigation

Perf Statistics

Binary	cycles	instructions	cache-references	LLC-load-misses	LLC-loads	LLC-store-misses	LLC-stores
Victim	5,19,83,305	5,87,82,318	2,76,114	83.69%	71,477	16.31%	not counted
Attacker	29,44,30,97,858	13,15,73,29,854	1,08,14,06,726	75.01%	1,07,86,49,291	24.99%	24,089

Files Included

1. bank.patch
2. BaRaCoRoCh.patch
3. nonopenpage.patch
4. rank.patch
5. RoBaRaCoCh.patch

Division of Work

- Nabhiraj Jain
  - TASK A
- Arun KP
  - TASK B
- Arun KP & Nabhiraj
  - Non-Open Page Policy(FCFS)

References

[1] Moscibroda, Thomas, and Onur Mutlu.. *Memory performance attacks: Denial of memory service in multi-core systems.*. Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium. USENIX Association, 2007.

[2] J. D. McCalpin. *STREAM: Sustainable memory bandwidth in high performance computers.* `http://www.cs.virginia.edu/stream/`.