# Impact of Memory DoS Attacks on Cloud Applications and Real-Time Detection Schemes

Zhuozhao Li*, University of Chicago

Tanmoy Sen and Haiying Shen, University of Virginia

Mooi Choo Chuah, Lehigh University

*Work done when Zhuozhao Li was at the University of Virginia

THE UNIVERSITY OF CHICAGO

UNIVERSITY of VIRGINIA

LEHIGH UNIVERSITY

# Cloud resources are shared among multi-tenants

- Cloud providers
  - E.g., Amazon AWS, Google Cloud, Microsoft Azure

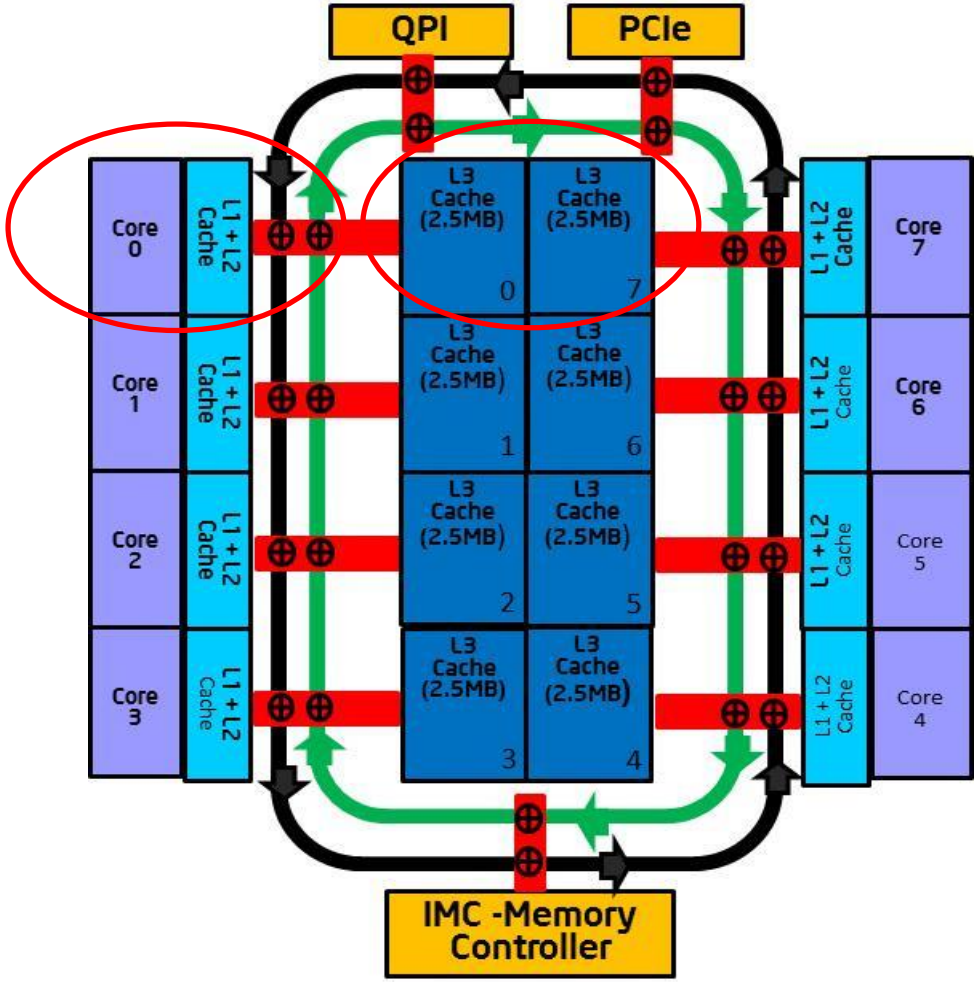

- Infrastructure-as-a-Service (IaaS)
  - Virtualization technique, e.g., hypervisor
    - Virtual machines (VMs)



  - Well isolated resources: CPU, memory pages, etc.
  - Shared among all VMs: hardware memory resources

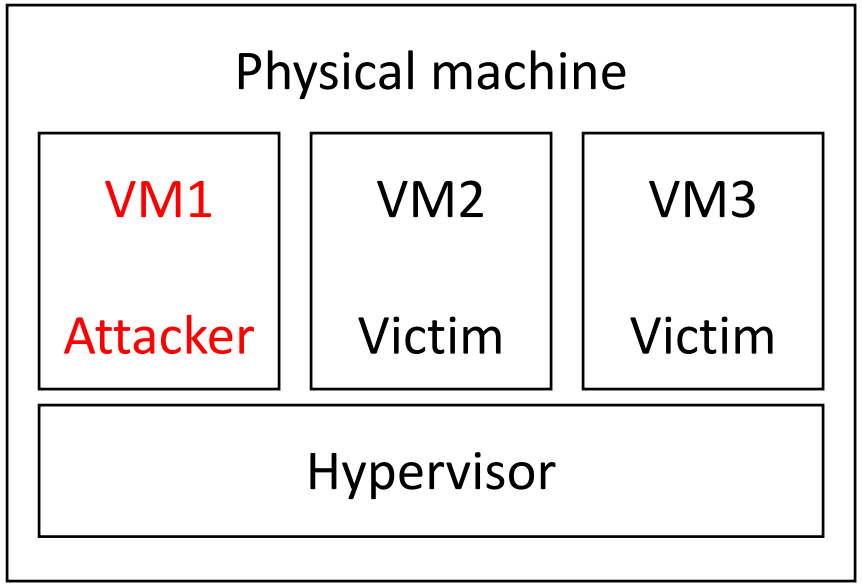

# Not all hardware memory resources are well isolated

- Dedicated cache per core, E.g.,
  - L1 and L2 cache
- Cache shared among all the cores, E.g.,
  - Last-level cache (LLC)
  - Ring-based bus to interconnect multiple memory resources



# Memory DoS attacks

- Severe resource contention on the shared memory resource
  - Memory Denial-of-Service (DoS) attack
- Intentional VM co-location with victim VM on the same physical machine (PM)
  - Achieved using several previous studies in minutes [1]
  - Low cost – less than $8



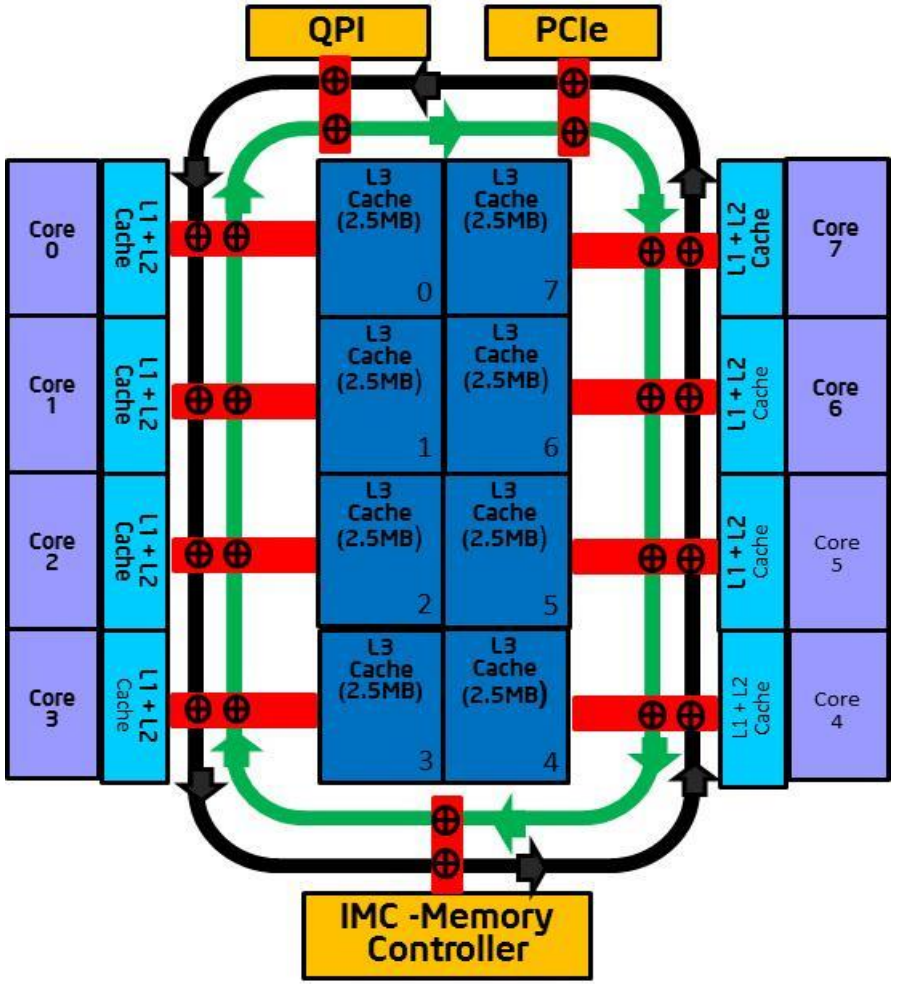

[1] Zhang Xu, Haining Wang, and Zhenyu Wu. A Measurement Study on Coresidence Threat inside the Cloud. In Proceedings of USENIX Security Symposium. 929–944, 2015

# Threat model

- Multi-tenancy public clouds
  - Memory Denial-of-Service (DoS) attack
- VM co-location with victim VM on the same physical machine (PM)
- The VMs from different tenants on the same machine share one LLC and several memory buses even with today's hypervisor techniques

# Memory DoS attacks

- LLC cleansing attack
  - Evict LLC lines of other VMs
  - Could be worse for inclusive CPUs
- Bus locking attack
  - Exotic atomic operations
  - Bus lock to block access
- Slowdown distributed applications (e.g., Hadoop MapReduce) up to 3.7 times [2]



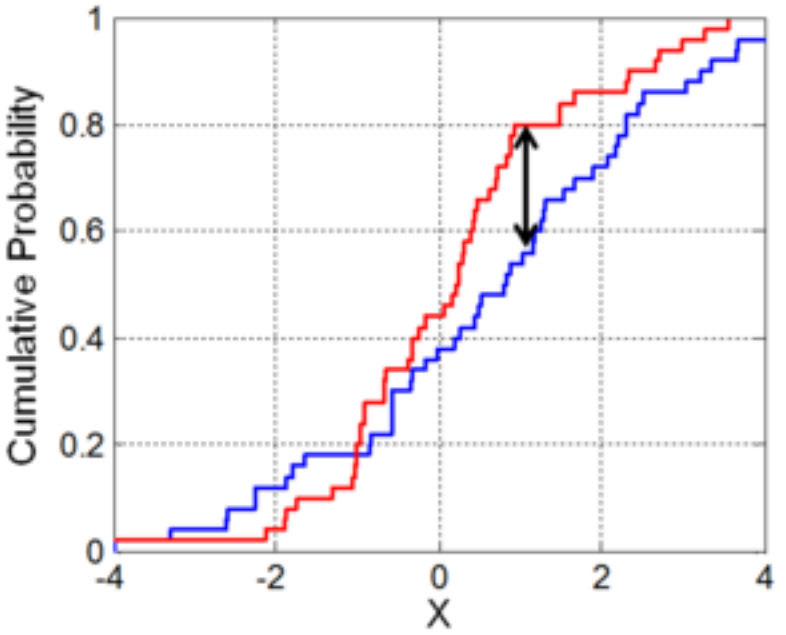

[2] Zhang, Tianwei, Yinqian Zhang, and Ruby B. Lee. "Dos attacks on your memory in cloud." Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. 2017

# Existing solutions

- Monitor cache statistics [2]

- Two-sample Kolmogorov-Smirnov test (KStest)
  - Determine if two statistics follow the same probability distribution
  - real-time statistics (with attack) vs. referenced statistics (no attack)
  - referenced statistics: throttle all other applications running on a machine

- Assumption: follow certain probability distribution at different times---Not true for all applications

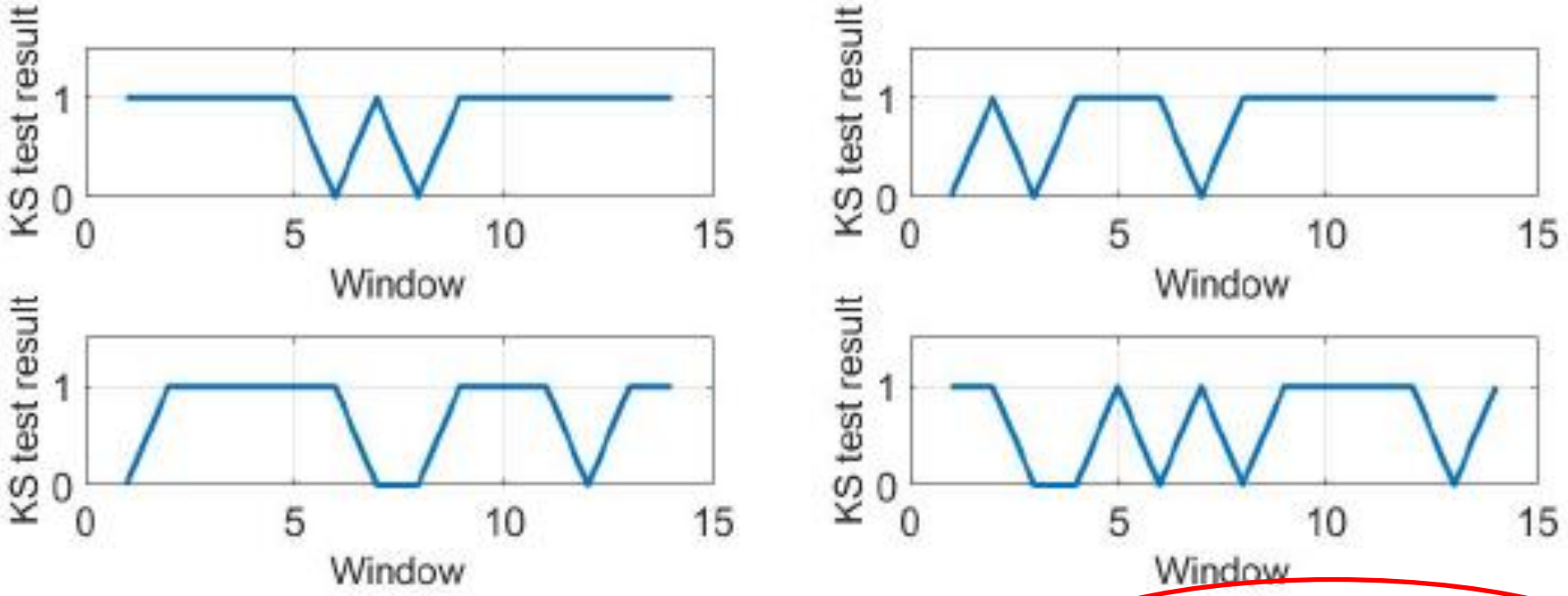



Two-sample Kolmogorov-Smirnov test

Source:https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test

[2] Zhang, Tianwei, Yinqian Zhang, and Ruby B. Lee. "Dos attacks on your memory in cloud." Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security. 2017.

# KStest is insufficient for all applications




**1: Do not follow**

**0: Follow**

**Figure 1: KStest results of TeraSort – no attack launched.**

Even when there is no attack, the application may not follow the same probability distribution

# Existing solutions

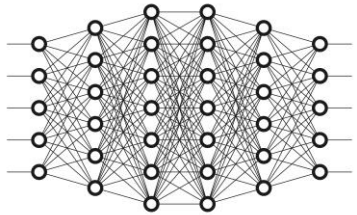
- VM migration
  - Easily co-locate with the victim VM again
- Hardware or software LLC partition
  - Waste the LLC resources significantly
  - Cannot defeat the memory bus locking attacks
- **Focus on attack detection in this paper**

# Contributions

- A measurement study of memory DoS attacks
  - How do the attacks impact different applications?
- Design of detection schemes
- Performance evaluation to show effectiveness

# Applications and Metrics

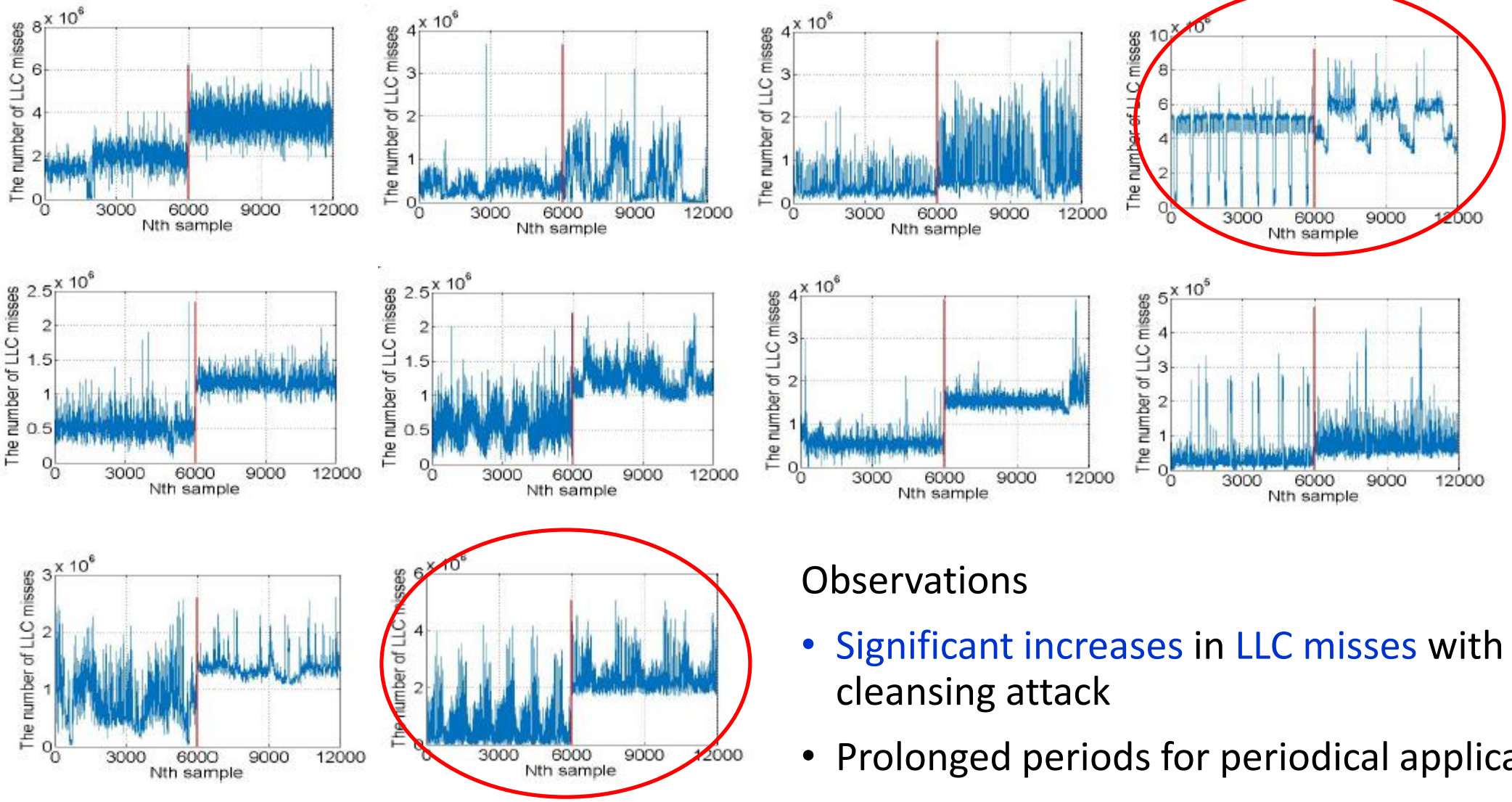
- Applications
  - Database
  - Machine learning and deep learning
  - Data-intensive
  - Web search





- Metrics
  - Collect statistics with Processor Counter Monitor (PCM) every interval
  - The number of LLC accesses
  - The number of LLC misses

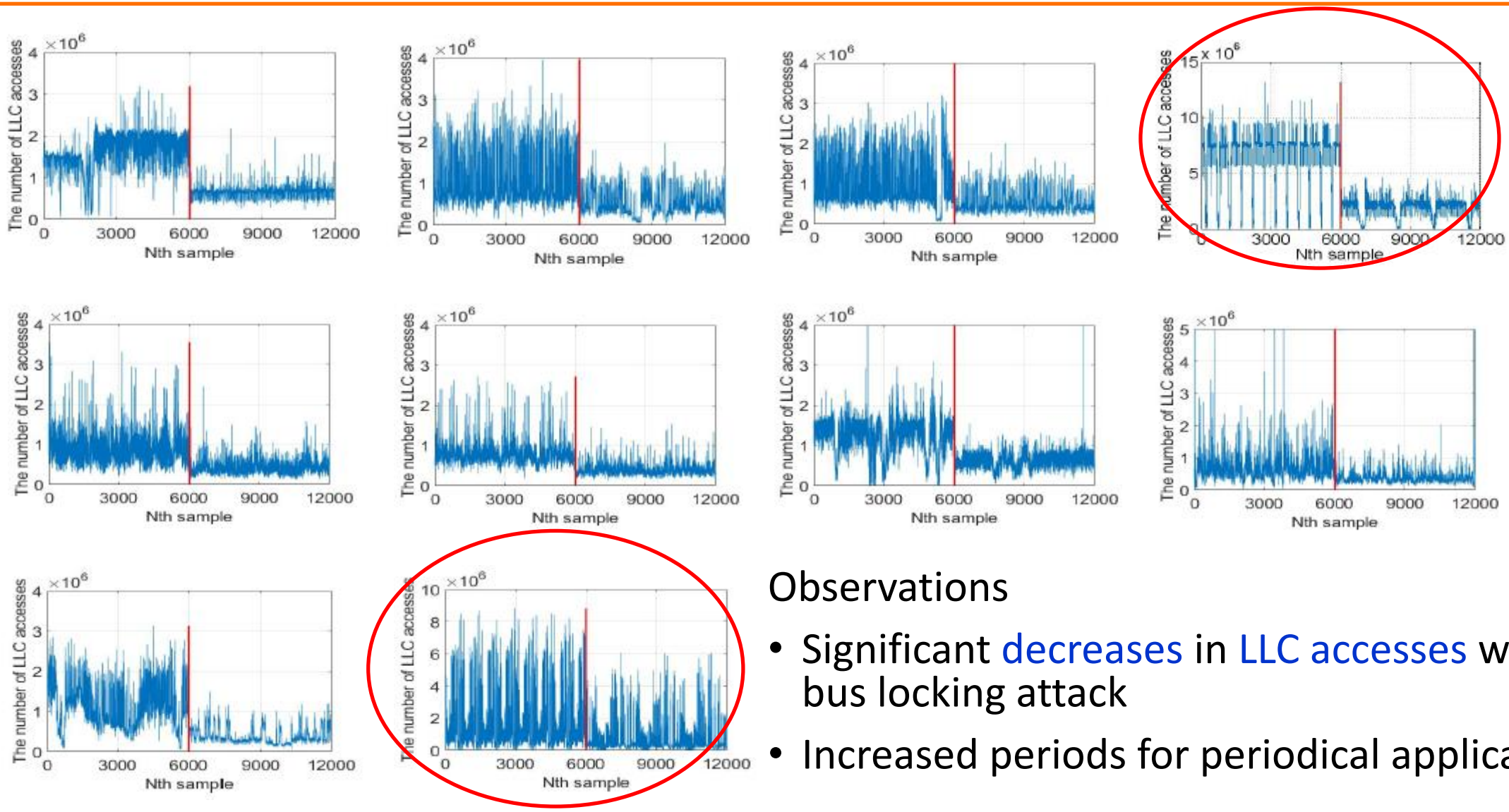
# Measurement studies – **LLC cleansing attack**




Observations

- Significant increases in LLC misses with LLC cleansing attack
- Prolonged periods for periodical application

# Measurement studies – **Bus locking attack**




Observations

- Significant decreases in LLC accesses with bus locking attack
- Increased periods for periodical application

# Design goals

- Irrespective of applications---regardless of statistics distribution
  - High accuracy
- Lightweight---low overhead
- Responsive---low detection delay

# Design considerations

- Overall design of the detection scheme:
  - Collect real-time cache statistics with processor counter monitor
    - Responsive and low overhead
  - Use moving average algorithm to smooth the collected sample data
    - Handle fluctuations of cache related statistics
  - Use a simple and efficient approach to analyze data in real-time
    - Low overhead

# General for all applications

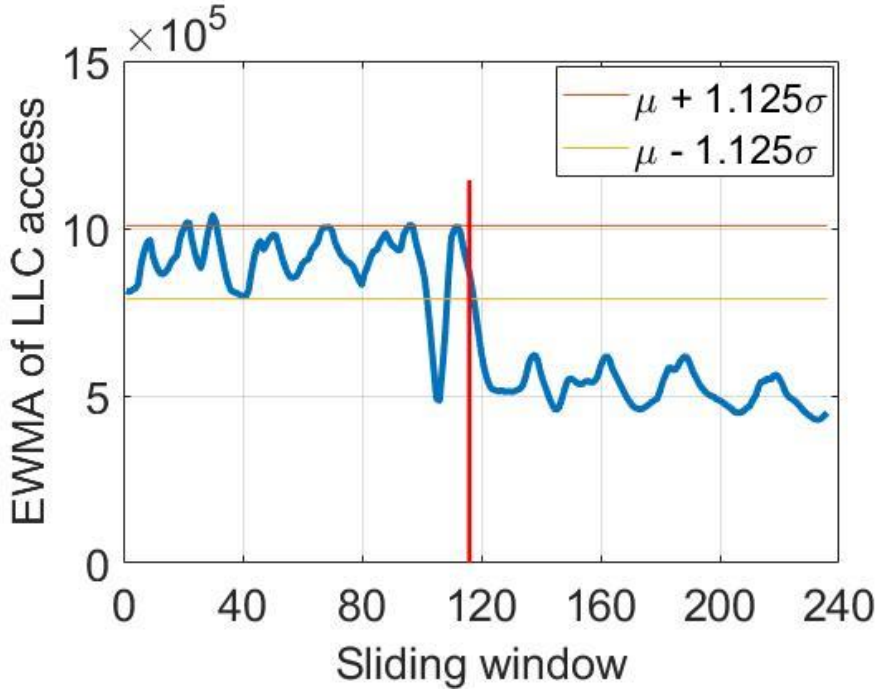
- Model the probability distributions of cache related statistics
  - E.g., Gaussian Distribution
  - Confidence level
  - **Problem:** not general enough for all applications

- Solution: use a model-independent approach
  - Chebyshev's inequality, applied to any probability distributions
  - $\mu$ is the expected value, $\sigma$ is the standard deviation

$$Pr(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}.$$

- The probability that any sample point is greater than the expected value by $\pm k\sigma$ is lower than $\frac{1}{k^2}$

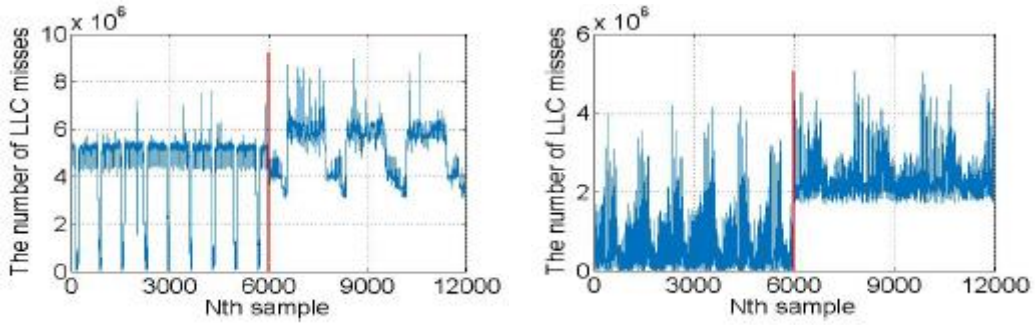
# Key rationales



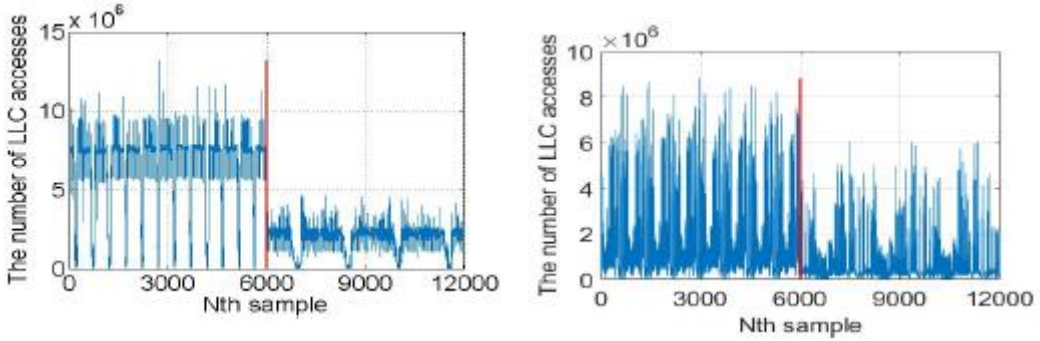

- Multiple consecutive outliners (e.g., 30) is likely to be attack
- Tune k based on confidence level and sensitivity

- Rationale: the memory DoS attacks need to change the cache related statistics to some degree to degrade the performance

# Enhancing detection accuracy for periodical applications

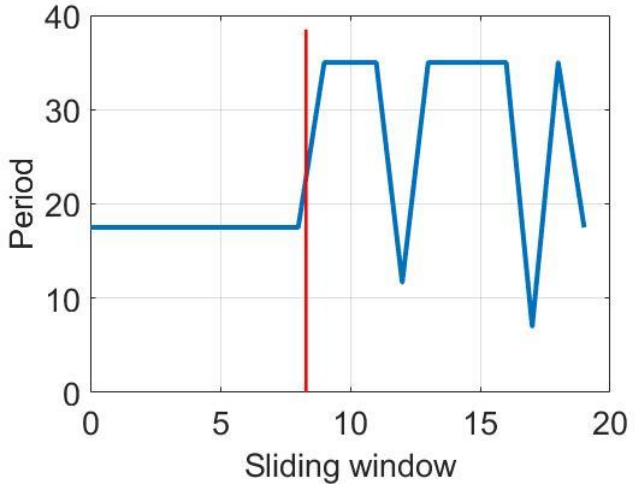



LLC cleansing attack




Bus locking attack

- Observation: prolonged periods for periodical applications
- Period detection
  - Discrete Fourier Transform
  - Auto Correlation Function




Period detection

# Evaluation

- Implementation on a server with an Intel CPU---14 cores, 35MB LLC
- KVM hypervisor, 9 VMs: 1 victim, 1 attacker, and 7 benign VMs
- Baseline comparison: KStest
- Metrics
  - Accuracy
  - Detection delay
  - Performance overhead
  - Sensitivity analysis

# Accuracy – True positive

- Recall: ability to correctly detect an attack
- All approaches show high recall
- High true positives and few false negatives

Our approach: SDS = SDS/B + SDS/P

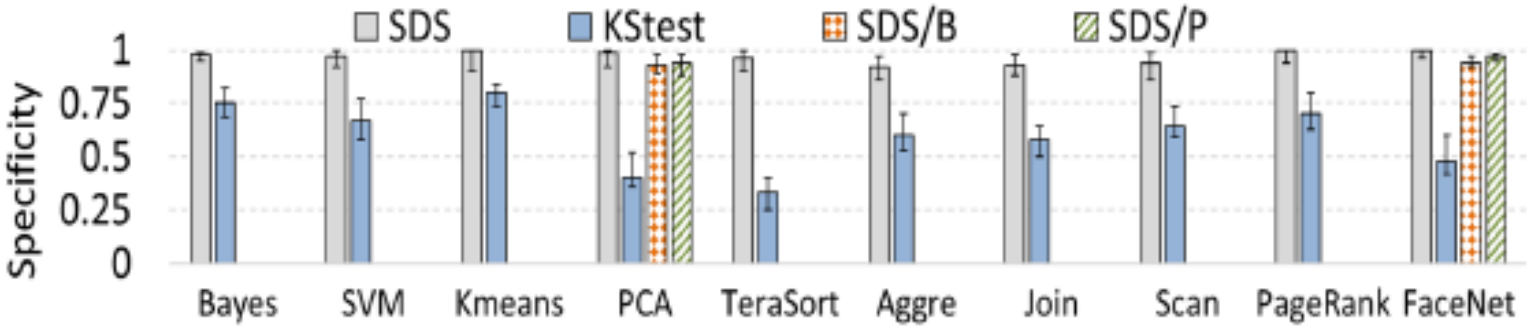

Recall for bus locking attack


Recall for LLC cleansing attack

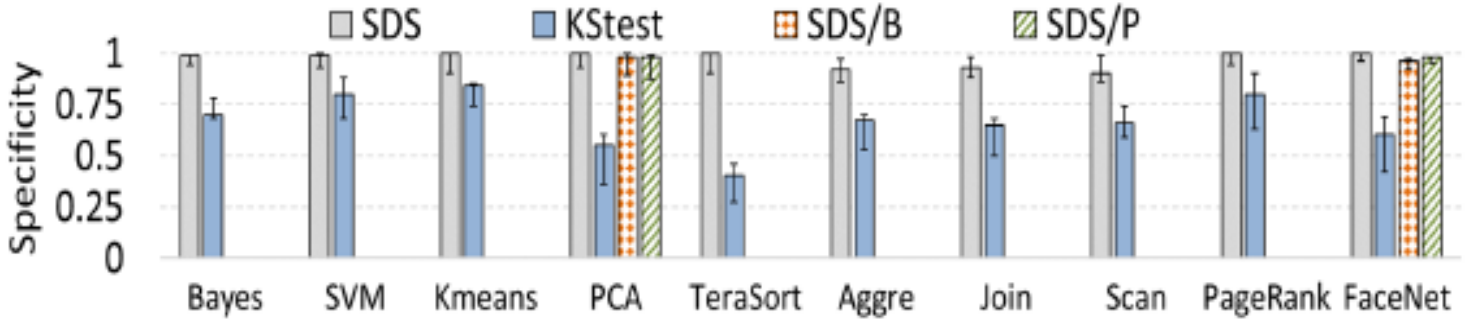
# Accuracy – False negative

- Specificity: ability to correctly infer no attack
- Our approach outperforms KStest on some applications by 20-65%
- High true negatives and few false positives

Our approach: SDS = SDS/B + SDS/P




Specificity for bus locking attack



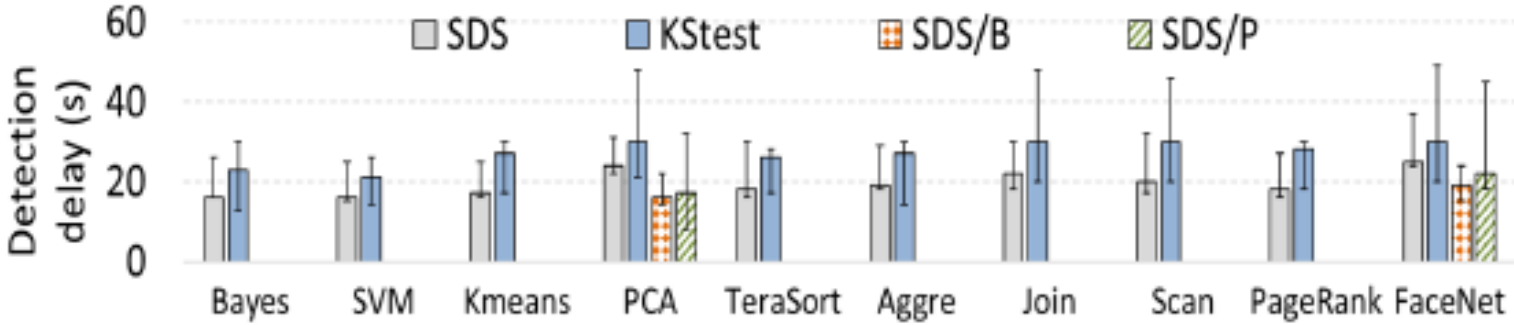

Specificity for LLC cleansing attack

# Detection delay

- Detection delay: the time to detect an attack
- SDS outperforms KStest by 3-20 seconds (5-40%)

Our approach: SDS = SDS/B + SDS/P


Detection delay for bus locking attack




Detection delay for LLC cleansing attack

# Conclusions

- Analyze the insufficiency of previous approaches to detect memory DoS attacks
- Conduct measurement studies on how memory DoS attacks impact the cloud applications
- Design lightweight, statistics-based detection schemes to detect memory DoS attacks accurately and responsively
- Future work: more complex attack scenarios

# Questions?

Zhuozhao Li
Postdoctoral Scholar
University of Chicago
zhuozhao@uchicago.edu