Detection and Mitigation of Performance Attacks in Multi-Tenant Cloud Computing

Carlos Cardenas and Rajendra V. Boppana
Computer Science Department
and
Institute for Cyber Security
University of Texas at San Antonio

Issues in Cloud Computing

Top 3 problems:

- Confidentiality of data and computing activities
- Availability and accessibility to data
- Dependable performance of computing

Features of Current Cloud Stacks

Offer allocation of main resources

- Allow CPU affinity and priority
- IP QoS
- Memory and Disk Quotas
- Do not readily offer
  - Management of shared, not directly visible, resources
  - Monitoring
  - Enforcement

RoQ Attacks in Multi-Tenant Computing

Reduction of Quality (RoQ) Attacks - attacks to reduce the availability of resources

- LLC polluting
- Interrupt storm

- With trial and error, an attacker can co-locate with multiple VMs with an intended victim [Ristenpart et al. CCS-09]

Attack Scenarios

- **Cache**

  Pollute Shared Cache: Tends to be L3 (LLC) on current CPUs

- **Disk**

  Perform large number of reads, writes, or both to render disk cache ineffective

- **Network**

  Increase number of packets transferred: increases number of interrupts generated and thus number of preemptions done by the kernel

Attack Types

- **NonColluding**

Multiple VMs attack independently

- **Colluding**

Multiple VMs launch attacks in a coordinated manner to avoid detection

Attack Types cont.

- **Direct**

Reduce effectiveness or availability of shared resource by using the resource abusively (LLC polluter)

- **Indirect**

Reduce effectiveness or availability of shared resource by causing other events (sending/ receiving large number of small packets causes scheduler to handle increased number of interrupts from the NIC by preempting some other running VMs)

Experimental Setup

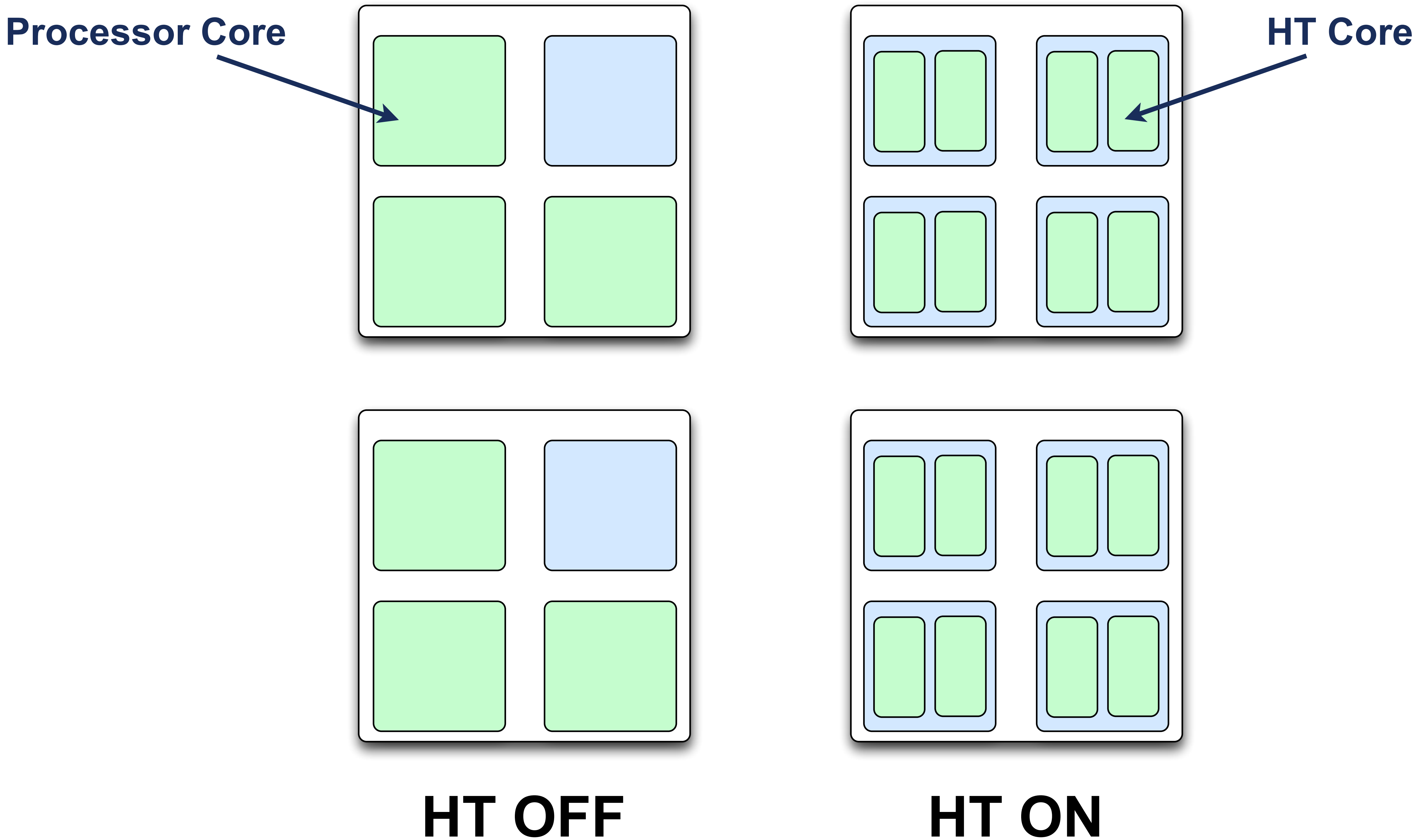
- 3 x Dell R710 (2 x Intel Xeon E5630, 4 cores per processor, 12MB L3 Shared Cache)
- OpenIndiana OS: CPU Affinity Case (pin VMs to cores, No HyperThreading or Turbo)
- SmartOS
  - HyperThreading + Turbo
  - No HyperThreading or Turbo

Experimental Setup cont.

- Victim Program: Parallel Floyd's shortest path algorithm in MPI
  - Size of the graph in number of nodes determines the computation time
- Attack Program: Simple Cache Polluter

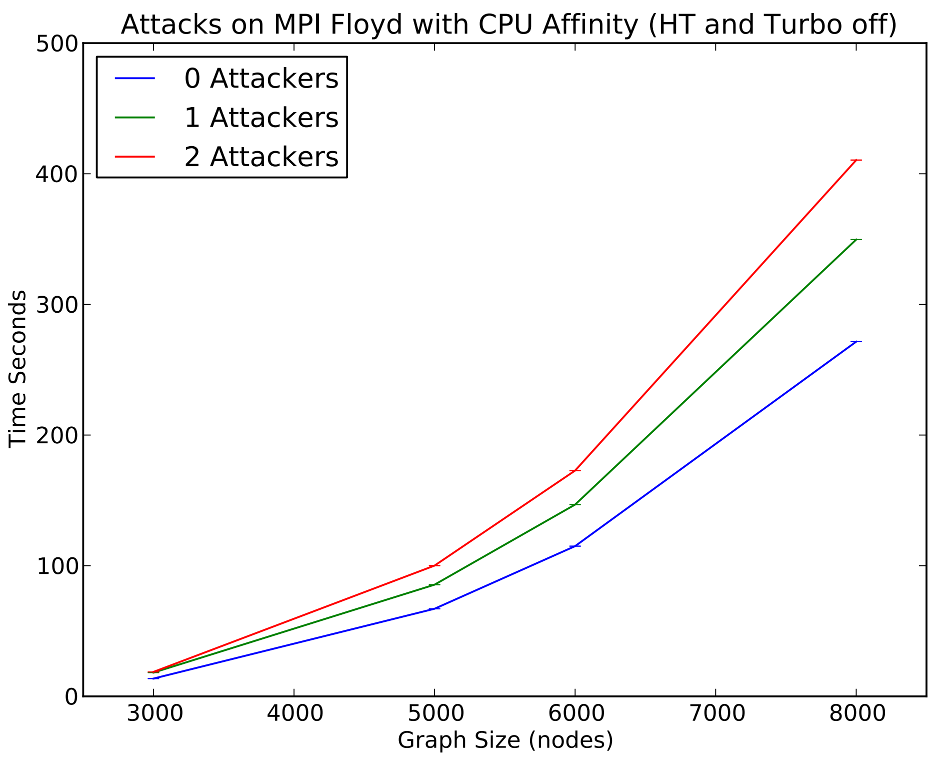
```
//array is an array of Prefetch_Degree*L3_Size
//stride is Prefetch_Degree*L3_LineSize_in_reals
//f is a floating point constant
while true
    for (i=0; i < array.length; i += stride)
        array[i] = array[i] * f;
```

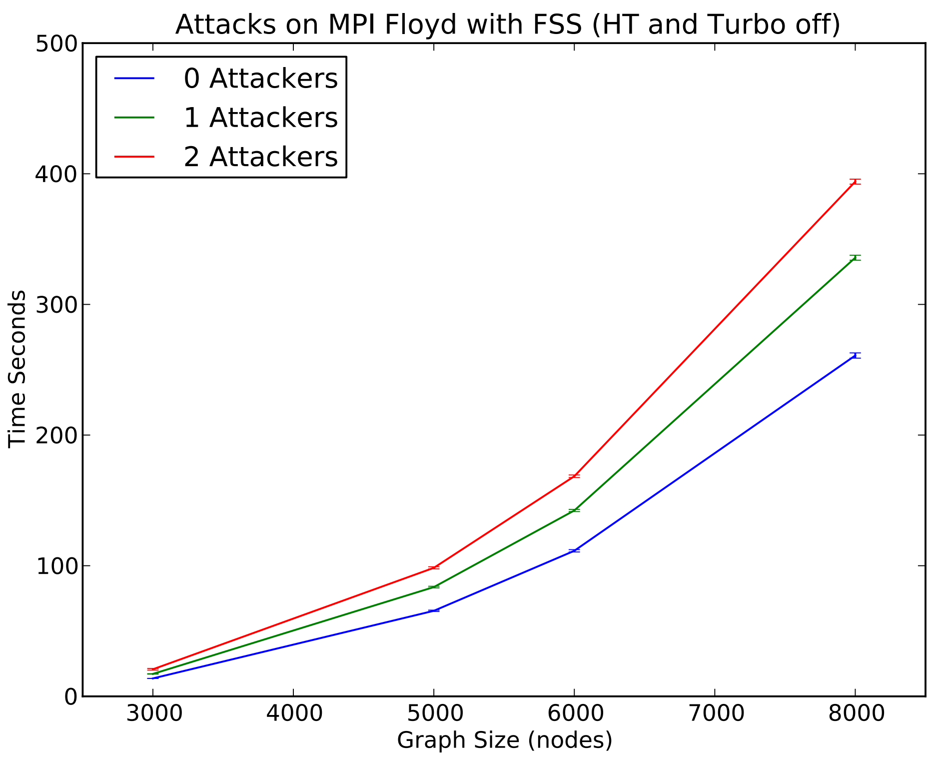
Processor Layout

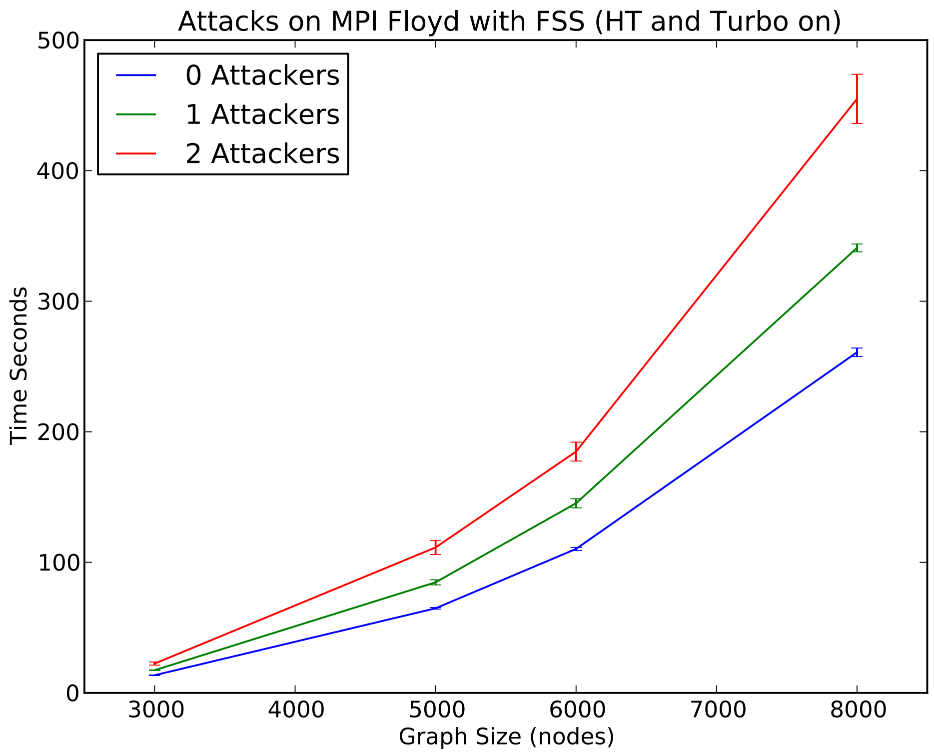


- MPI Floyd: 4 processes on 4 cores
- Attacker: 2 VMs (user zones)

Impact of Attacks







- 51-74% increase with 2 attackers
- HT On is worse than HT Off

Monitoring and Mitigation

- Resource Monitor
  - Used DTrace to record L3 cache (LLC) accesses and misses of all processes in 5 second intervals.
- Detection and Mitigation Logic
  - Used NodeJS to call DTrace and analyze the data from monitoring. Able to perform additional statistics and termination of processes within NodeJS.

Detection Logic
Interval Duration

- Experimented with 1, 3, 5, 10, and 30 seconds
  - 30 seconds does not provide enough resolution
  - 10 seconds is still not enough
  - 1 seconds provides excellent resolution but is too costly in CPU overhead (about 6%)
  - 3 and 5 seconds provides balance between resolution and CPU overhead (< 1%)

Detection Logic
Consecutive Intervals above Threshold

- Experimented with n = 3, 4, and 5 to try to achieve low false-positive rate (with 1, 3, 5, 10, and 30 second intervals)
- Number of consecutive intervals is tied to sampling period (3 consecutive, 1 second intervals, etc...)
- **Rule of Thumb**: about 25 - 30 seconds for good window of observation balances false-positive rate and monitoring overhead

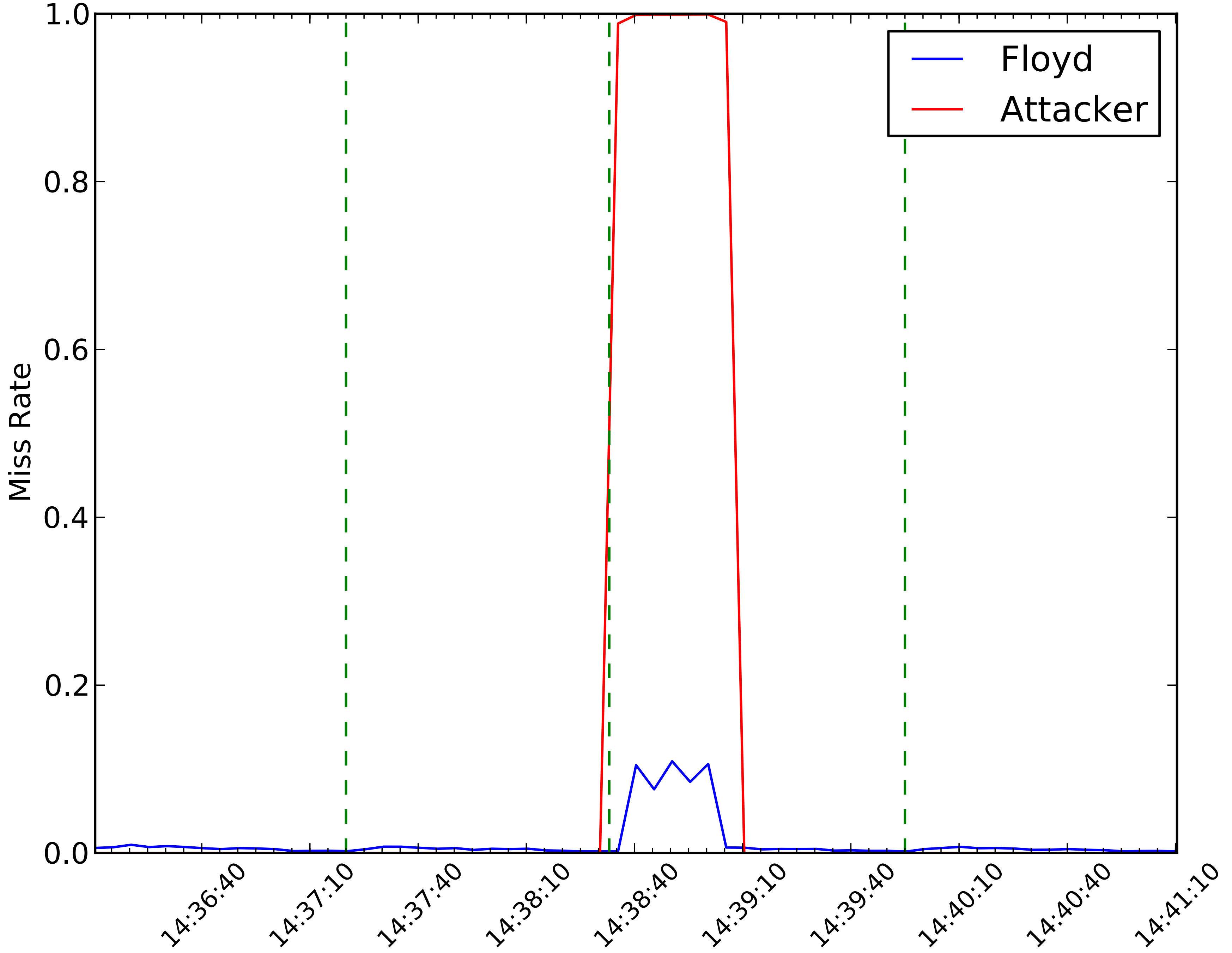
Detection Logic Thresholds

- Used DTrace to profile workload
- Experimented with various Static Thresholds, starting at 50%
  - 50% LLC miss rate, still possible to have false-positives
  - 80% miss rate yielded 3% false-positive rate
- Analyzed 3% false-positives and noticed they never went above 10^5 misses

Resource Monitor Implementation

- If L3 cache miss rate is well above the “norm” for 5 consecutive 5-second intervals, process is considered potential polluter
- Static Threshold
  - Miss Rate >= .80
  - LLC miss count > 10^6 per interval
- Terminate Process

Resource Monitor In Action



Parallel Floyd, 5000-node graph, 4 MPI Processes (OpenIndiana)

Related Work

- Dependable performance of computing in cloud. [Schad et al. VLDB 2010, Weng et al. HPDC 2011, Chen et al. UCB TR 2010]
- Co-locate multiple attackers in the cloud [Ristenpart CCS 2010]
- LLC Optimizations to resolve inter and intra cache interference [Wu et al. MICRO 2011, ISPASS 2011]

Summary and Future Work

- Investigated the effects of a malicious user has on others in Mult-Tenant Computing
- Showed impact of shared cache polluting attacks
- Designed and implemented monitoring utility in NodeJS using DTrace to detect and mitigate with low overhead (< 1%)
- Future Work
  - For faster response time and to handle multiple scenarios, it is best to have an adaptive threshold to defeat attacks