Bolt: I Know What You Did Last Summer… In the Cloud

Christina Delimitrou[1] and Christos Kozyrakis[2]

[1]Cornell University, [2]Stanford University

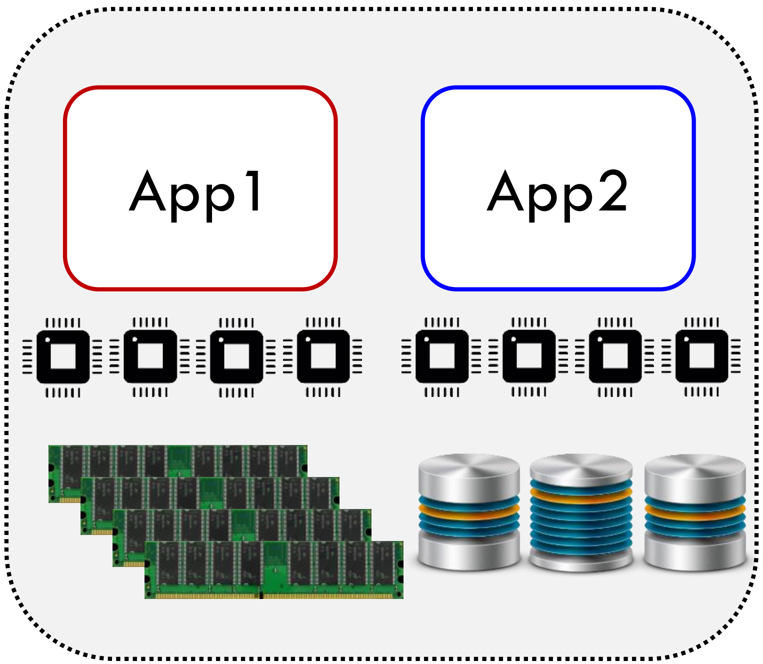
ASPLOS – April 12th 2017

Executive Summary

- Problem: cloud resource sharing hides security vulnerabilities
  - Interference from co-scheduled apps → leaks app characteristics
  - Enables severe performance attacks

- Bolt: adversarial runtime in public clouds
  - Transparent app detection (5-10sec)
  - Leverages practical machine learning techniques
  - DoS → 140x increase in latency
  - User study: 88% correctly identified applications
  - Resource partitioning is helpful but insufficient

Motivation

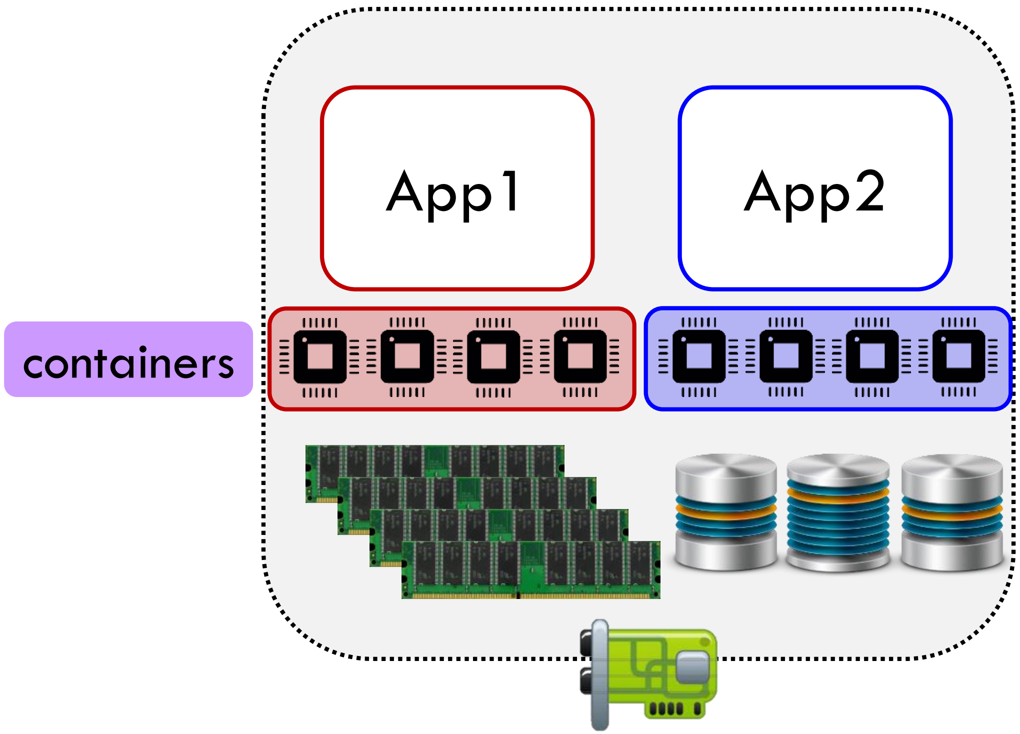


amazon web services™

Windows Azure™

Google Cloud Platform

Motivation

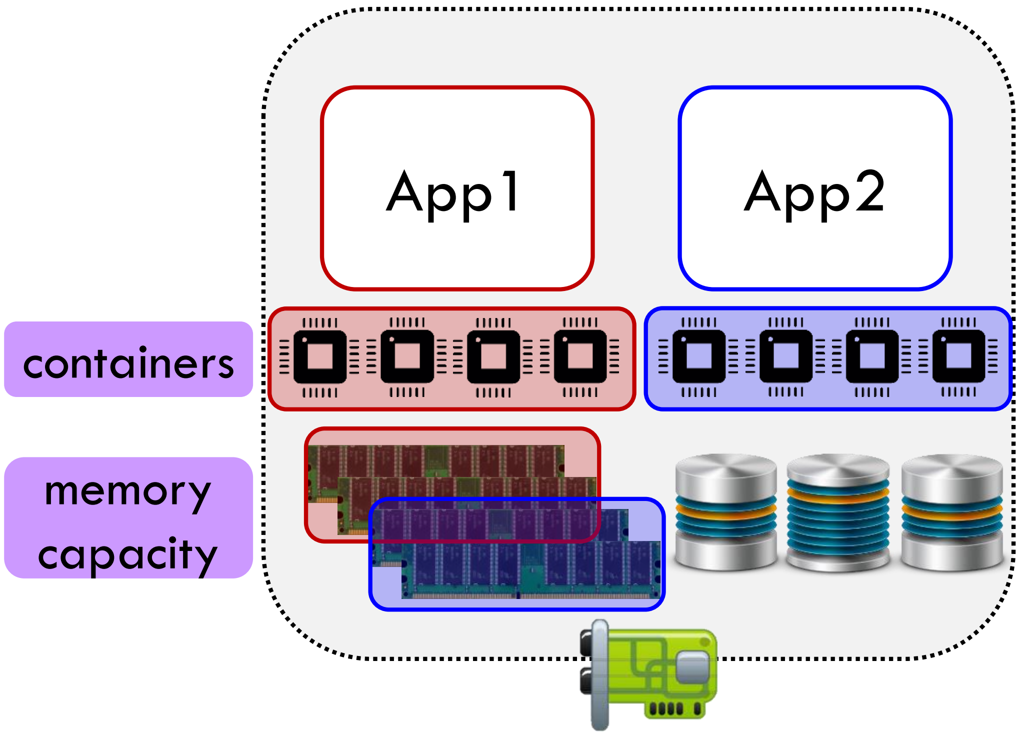


amazon web services™

Windows Azure

Google Cloud Platform

Motivation

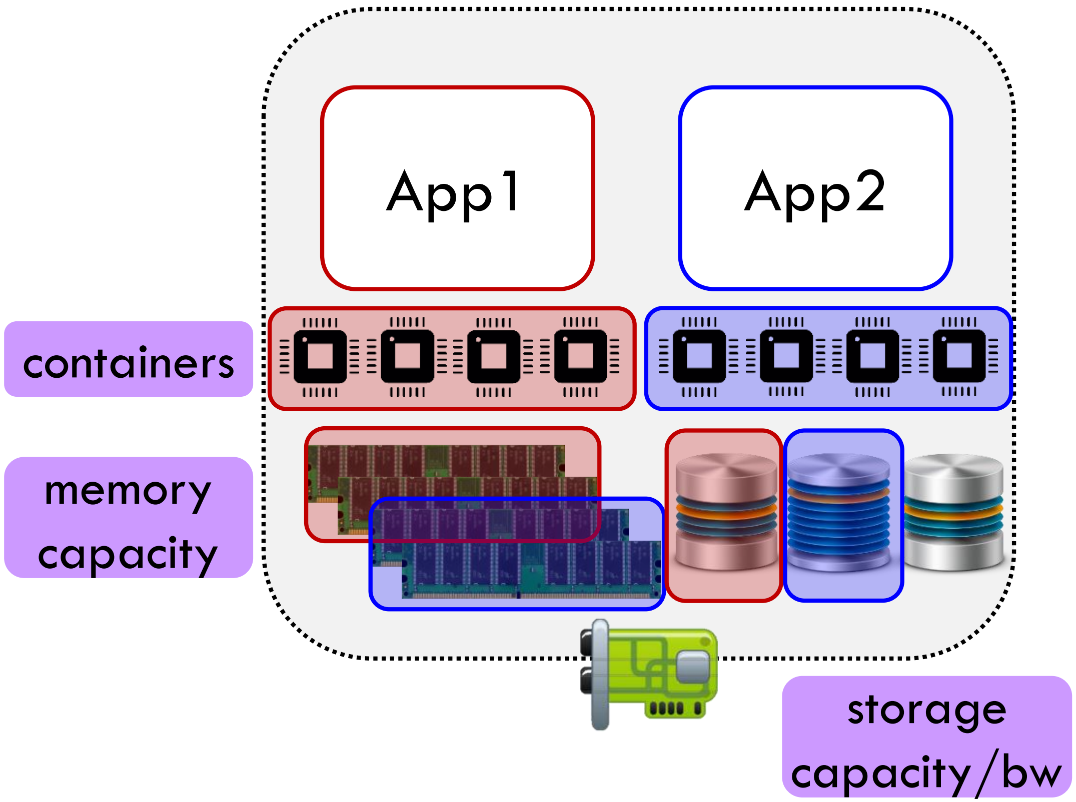


amazon web services™

Windows Azure

Google Cloud Platform

Motivation

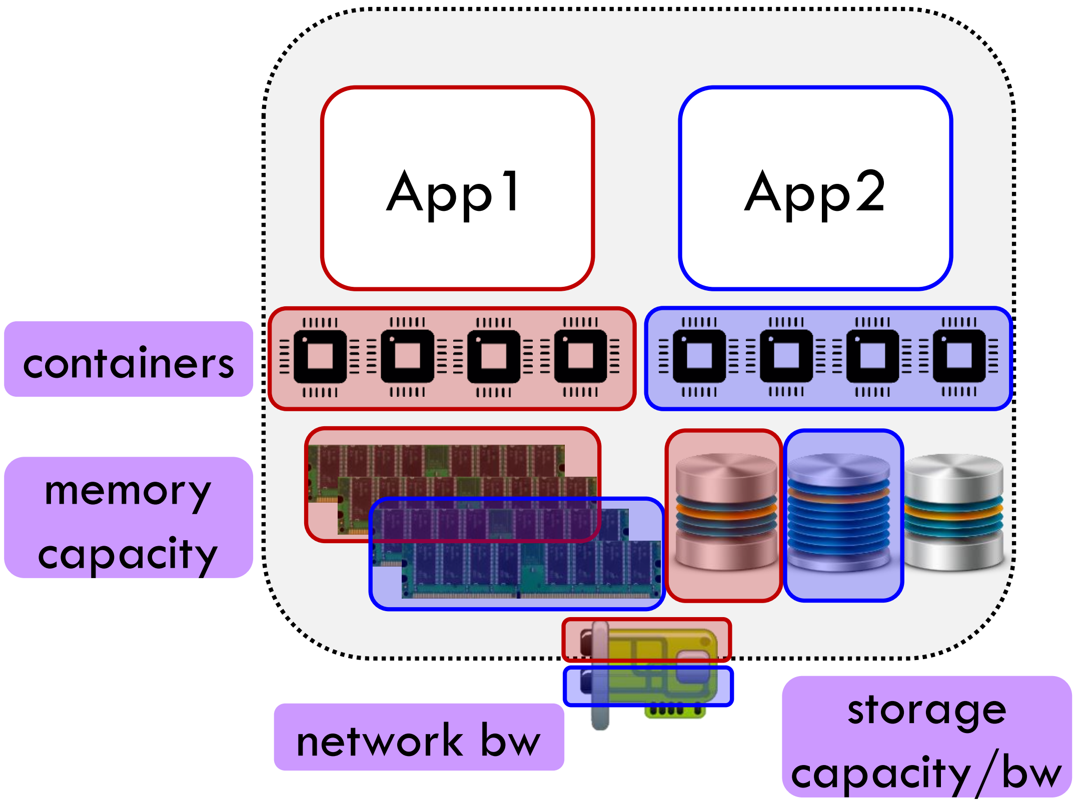


amazon web services™

Windows Azure™

Google Cloud Platform

Motivation

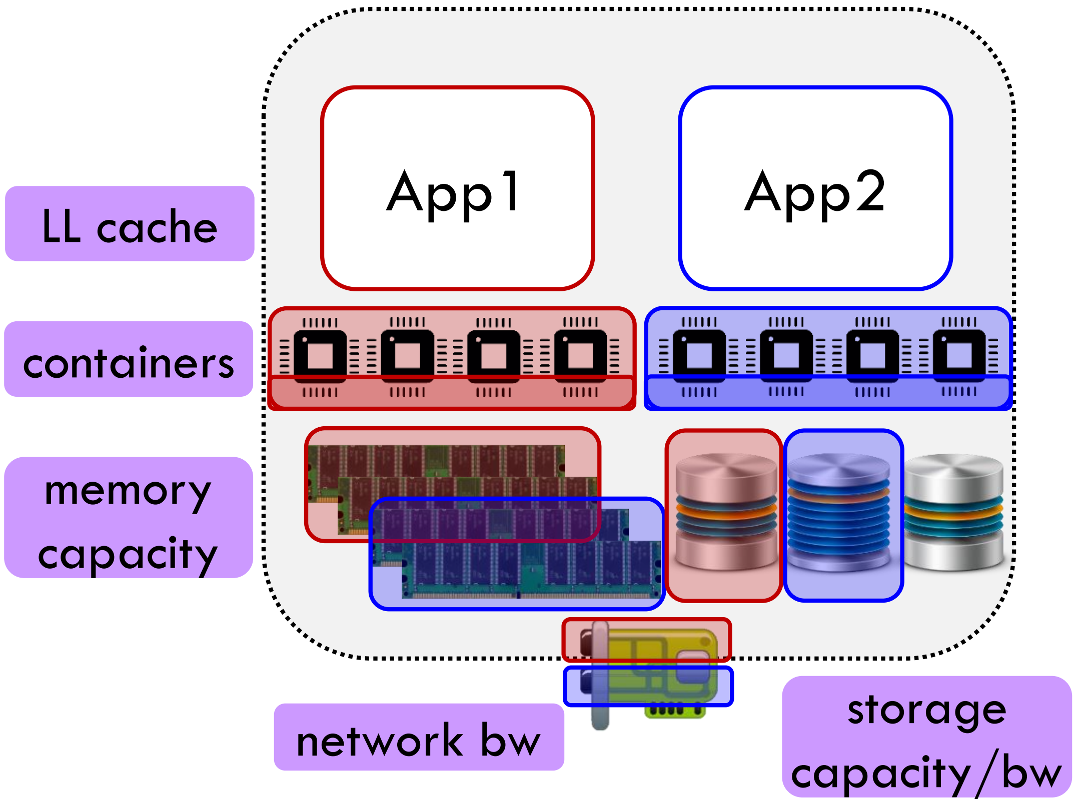


amazon web services™

Windows Azure™

Google Cloud Platform

Motivation

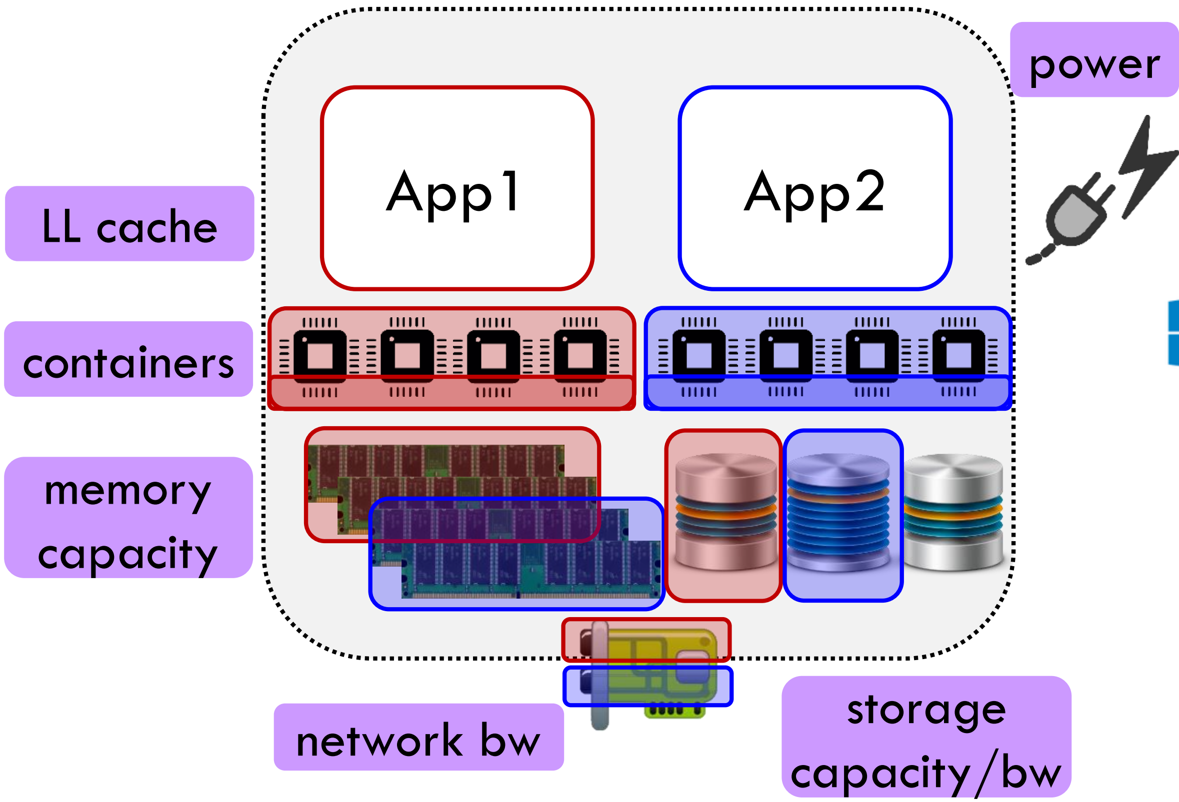


amazon web services™

Windows Azure

Google Cloud Platform

Motivation

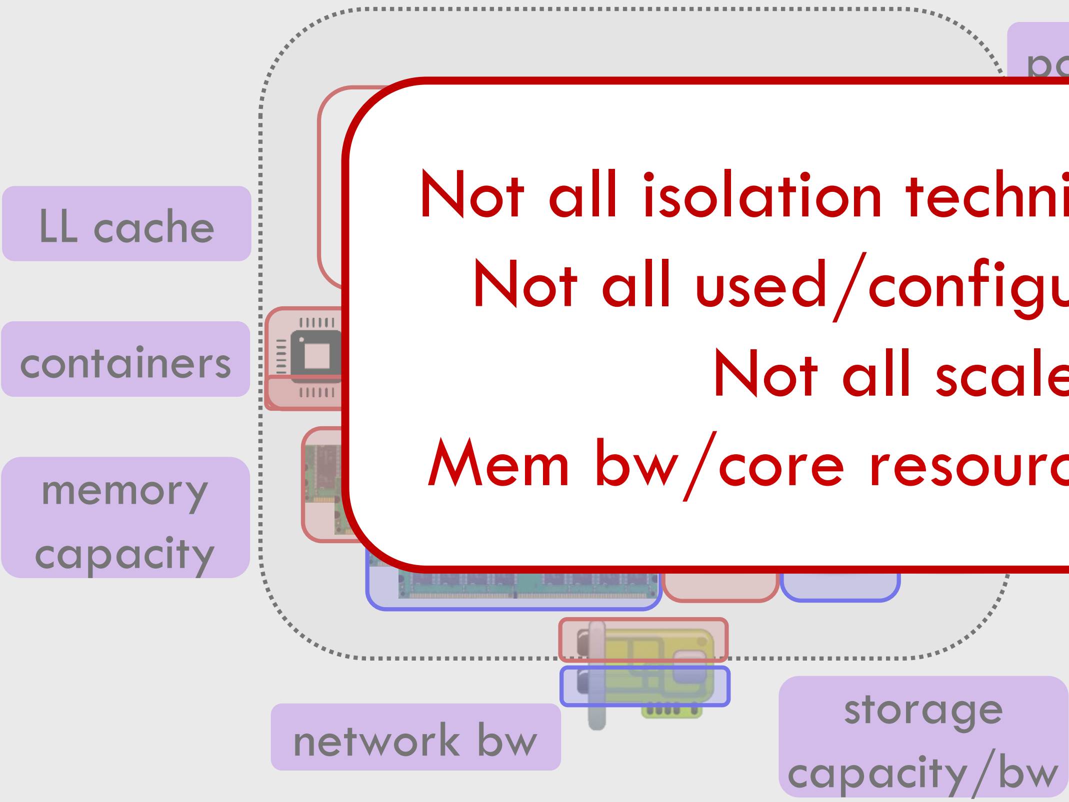


amazon web services™

Windows Azure

Google Cloud Platform

Motivation



Not all isolation techniques available

Not all used/configured correctly

Not all scale well

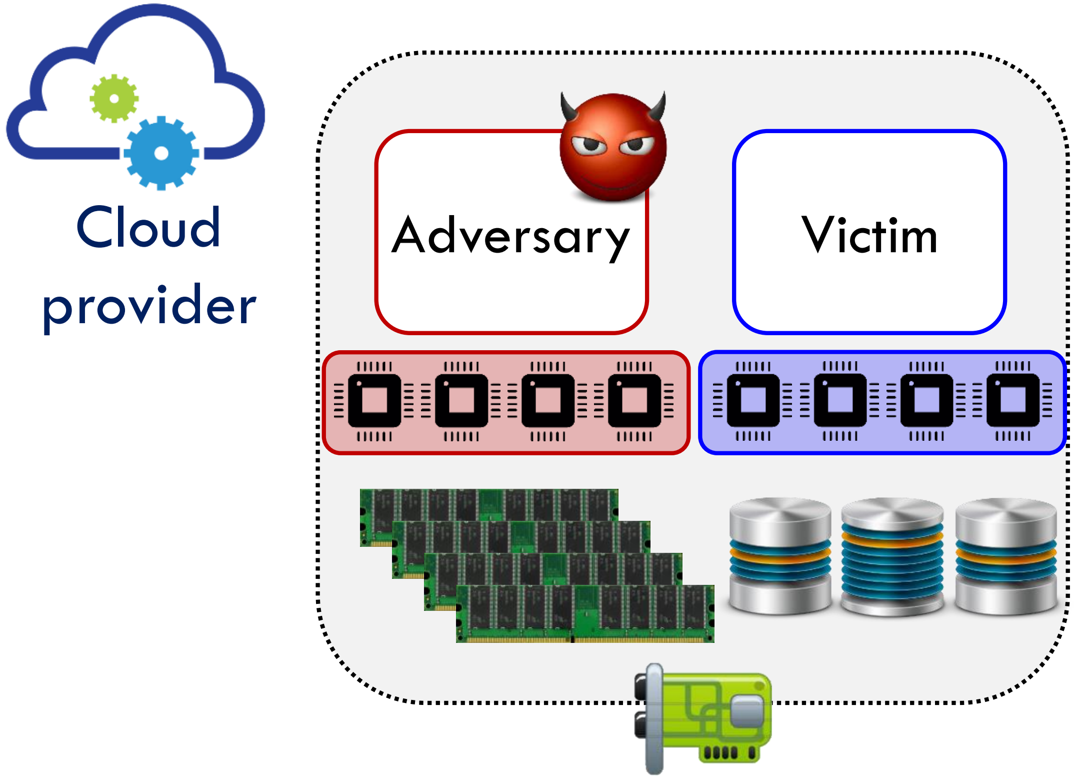
Mem bw/core resources not isolated

Bolt

- Key idea: Leverage lack of isolation in public clouds to infer application characteristics
  - Programming framework, algorithm, load characteristics

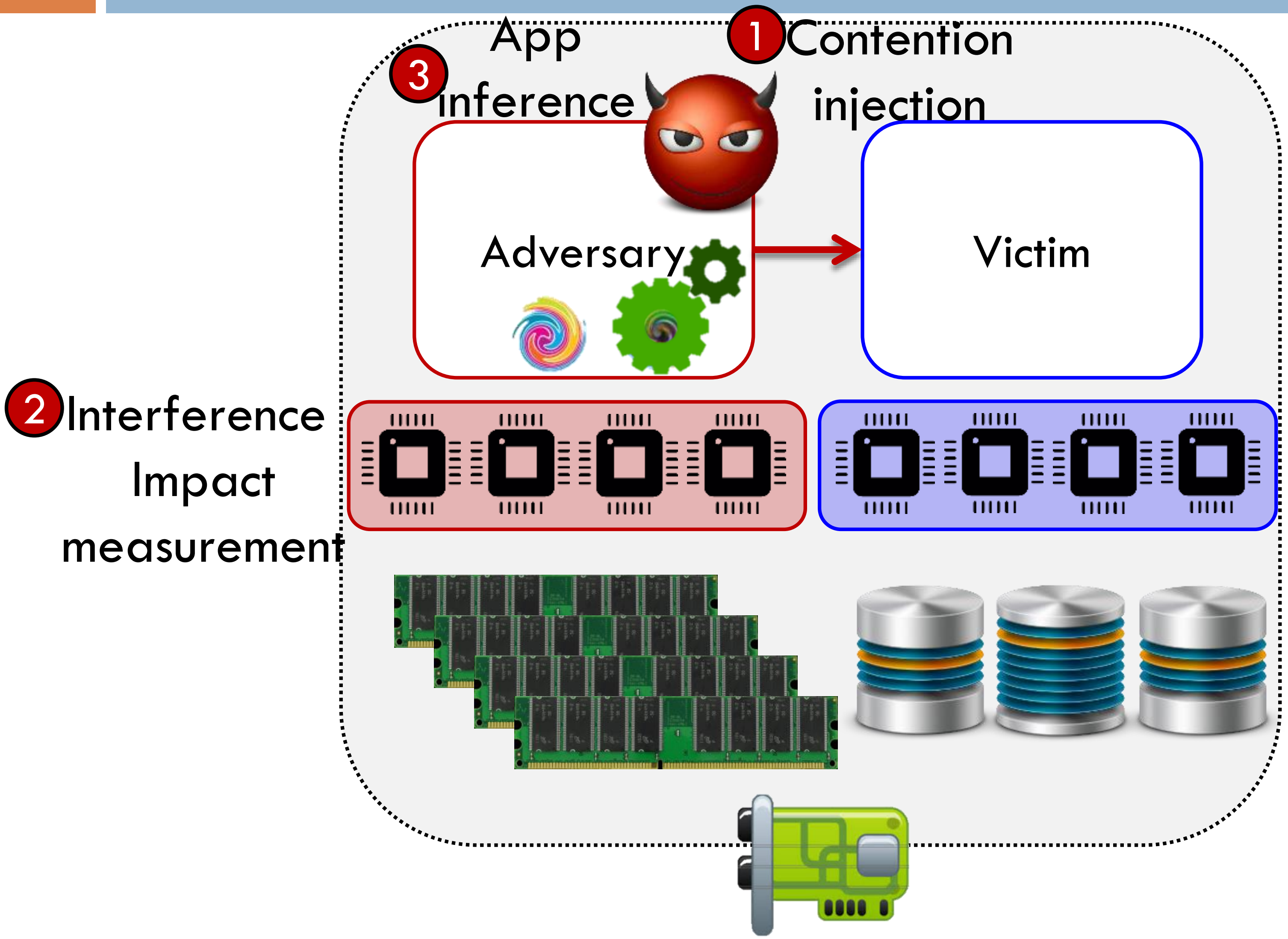
- Exploit: enable practical, effective, and hard-to-detect performance attacks
  - DoS, RFA, VM pinpointing
  - Use app characteristics (sensitive resource) against it
  - Avoid CPU saturation → hard to detect

Threat Model

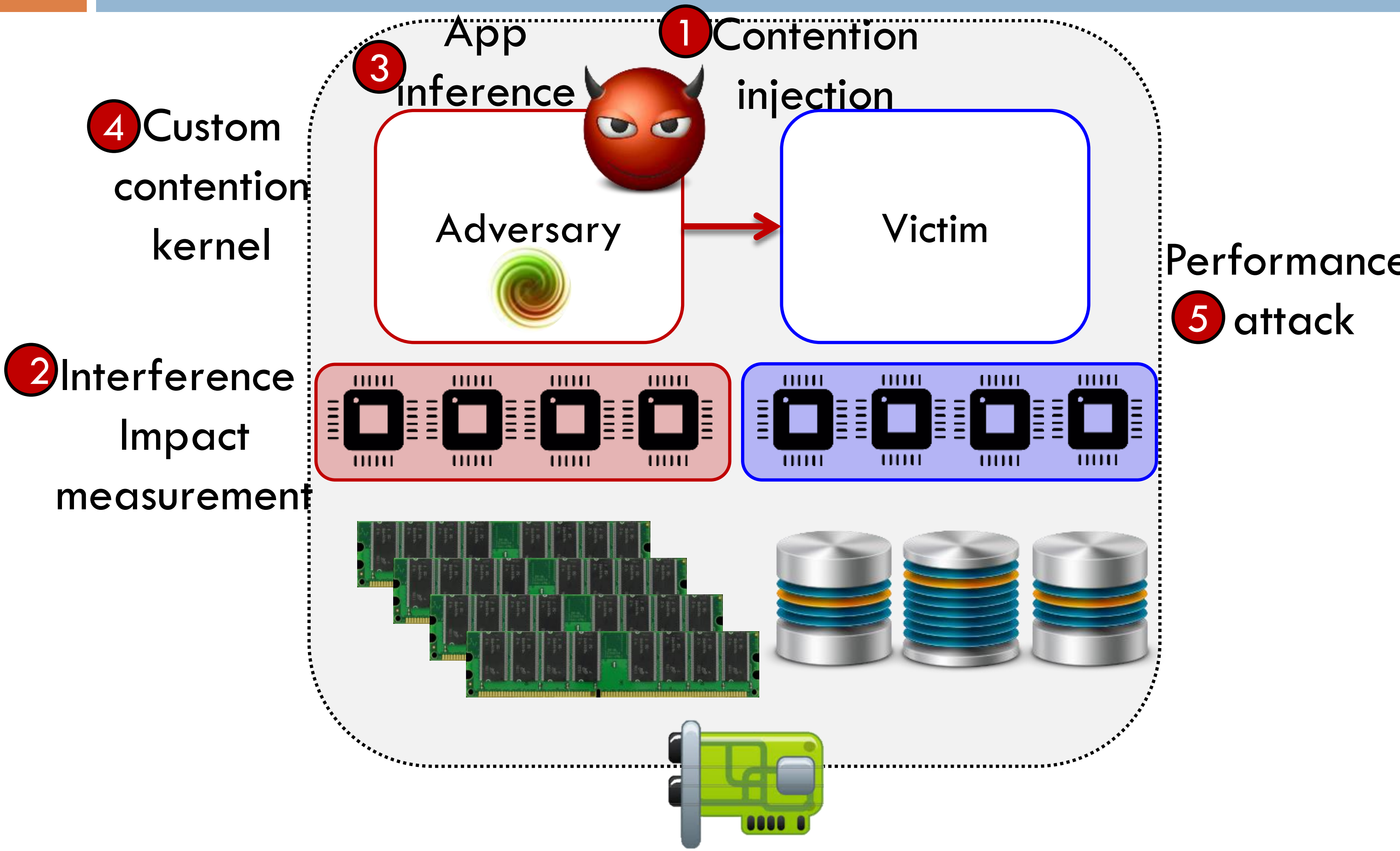


- Impartial, neutral cloud provider
- Active adversary but no control over VM placement

Bolt


3
App inference
1
Contention injection
Adversary
Victim
2
Interference Impact measurement

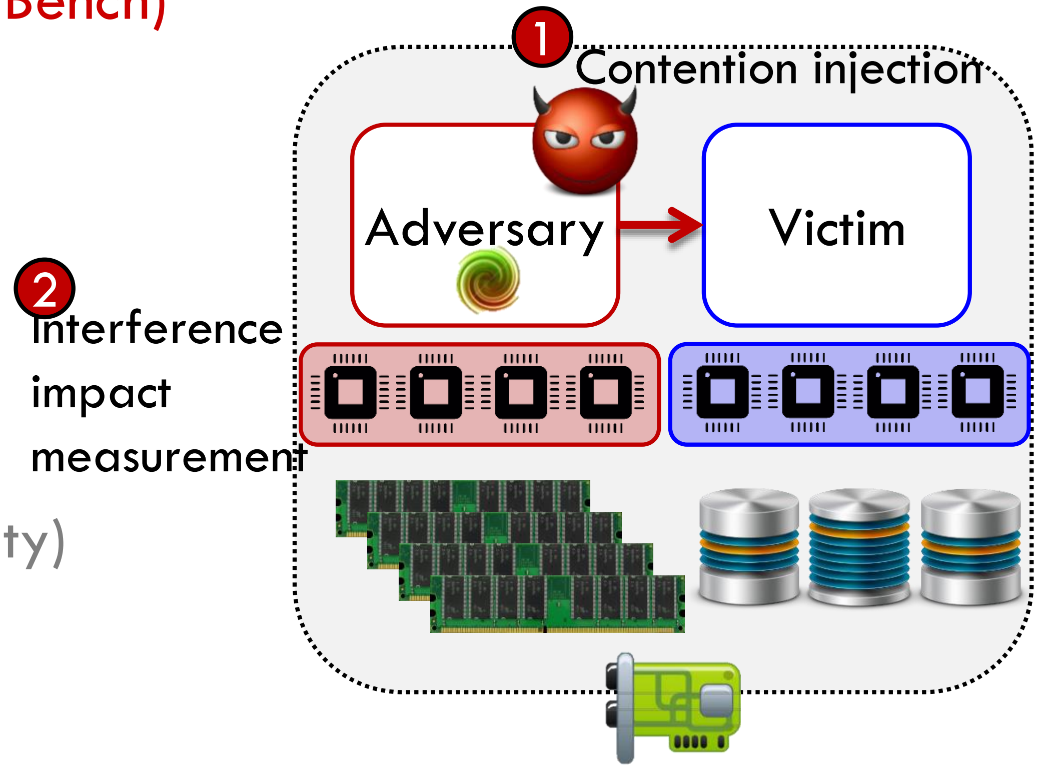
Bolt



3 App inference
1 Contention injection
4 Custom contention kernel
Adversary
Victim
Performance 5 attack
2 Interference Impact measurement

1. Contention Measurement

- Set of contentious kernels (iBench)
  - Compute
  - L1/L2/L3
  - Memory bw
  - Storage bw
  - Network bw
  - (Memory/Storage capacity)
- Sample 2-3 kernels, run in adversarial VM
- Measure impact on performance of kernels vs. isolation



2. Practical App Inference

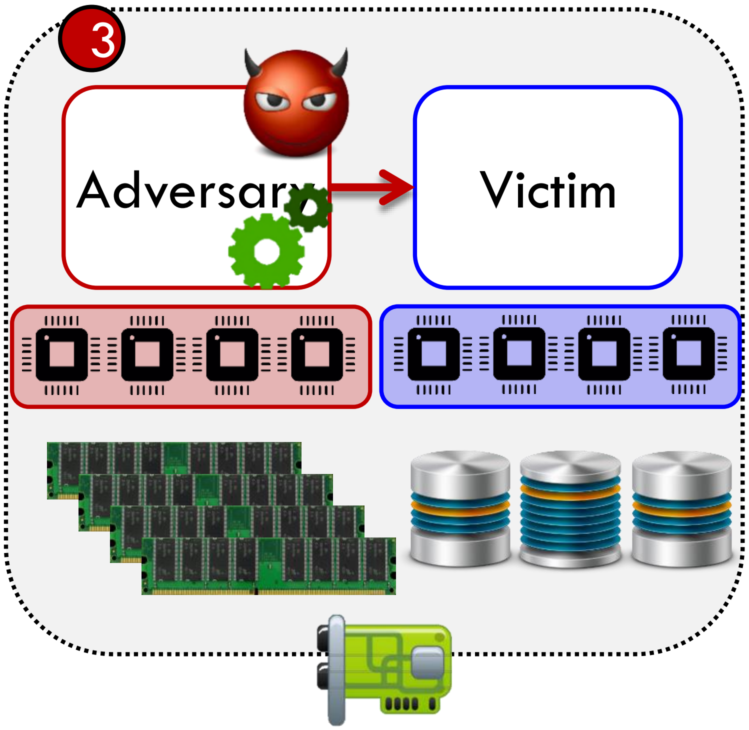
- Infer resource pressure in non-profiled resources
  - Sparse → dense information
  - SGD (Collaborative filtering)

+

- Classify unknown victim based on previously-seen applications
  - Label & determine resource sensitivity
  - Content-based recommendation

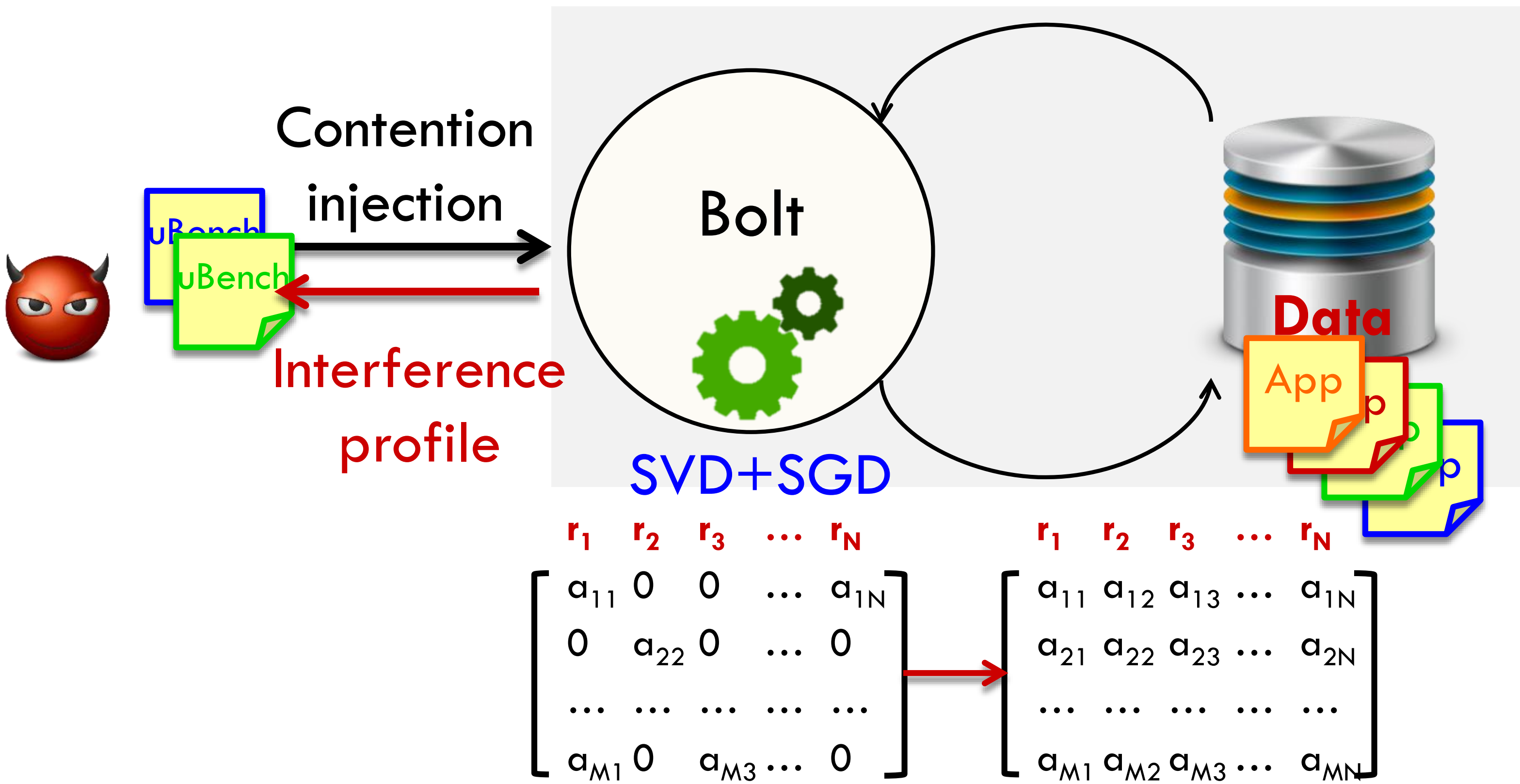
=

Hybrid recommender



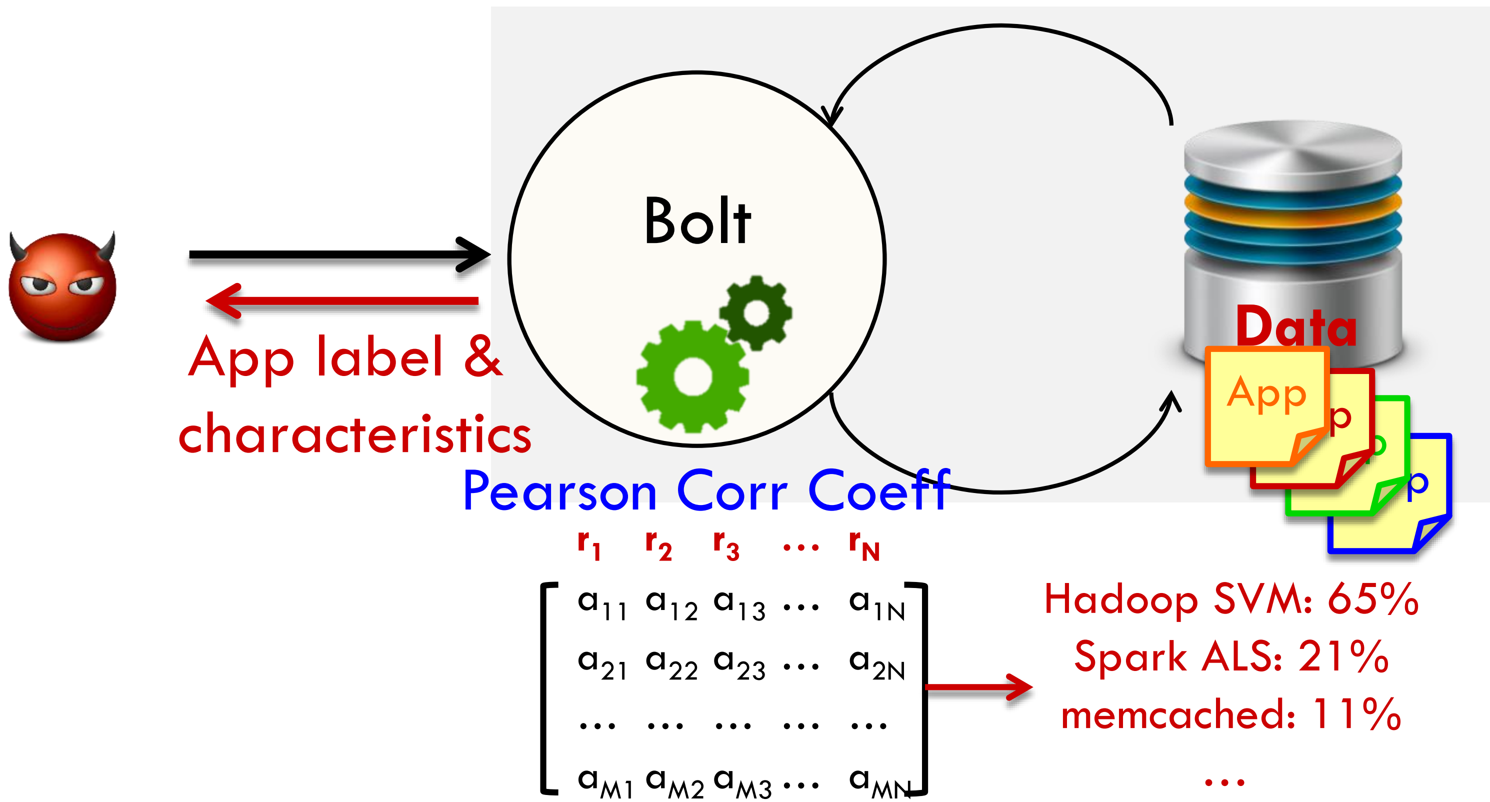
Big Data to the Rescue

1. Infer pressure in non-profiled resources
   - Reconstruct sparse information
   - Stochastic Gradient Descent (SGD), O(mpk)



Big Data to the Rescue

2. Classify and label victims
   - Weighted Pearson Correlation Coefficients
   - Output: distribution of similarity scores to app classes



Inference Accuracy

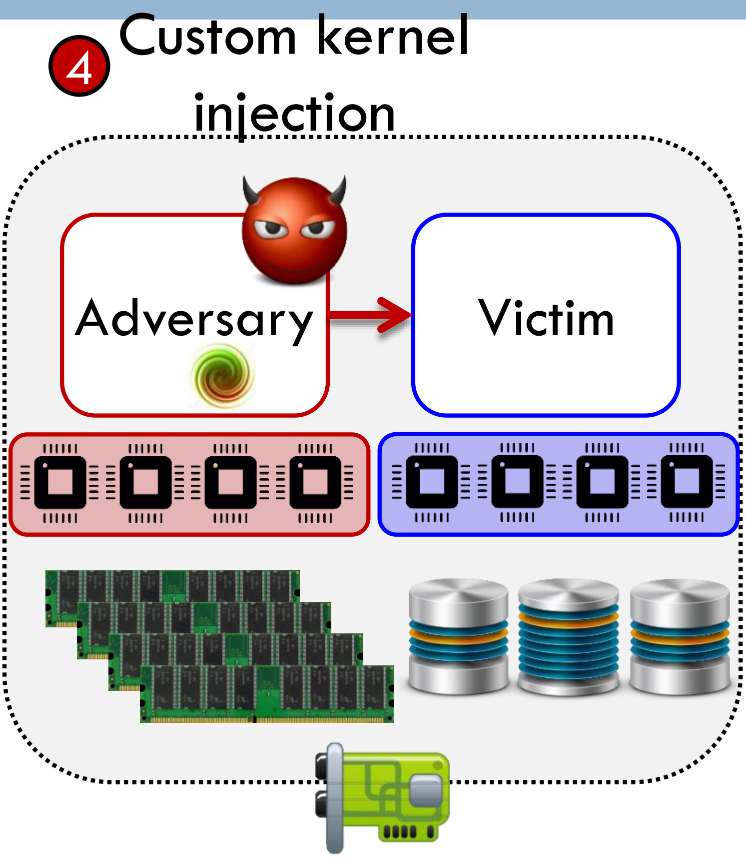
- 40 machine cluster (420 cores)
- Training apps: 120 jobs (analytics, databases, webservers, in-memory caching, scientific, js) → high coverage of resource space
- Testing apps: 108 latency-critical webapps, analytics
- No overlap in algorithms/datasets between training and testing sets

Application class	Detection accuracy (%)
In-memory caching (memcached)	80%
Persistent databases (Cassandra, MongoDB)	89%
Hadoop jobs	92%
Spark jobs	86%
Webservers	91%
Aggregate	***89%***

3. Practical Performance Attacks

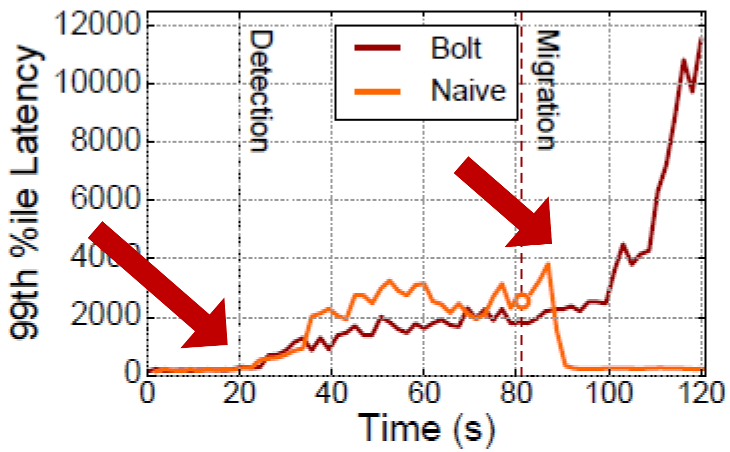
1. Determine the resource bottleneck of the victim
2. Create custom contentious kernel that targets critical resource(s)
3. Inject kernel in Bolt

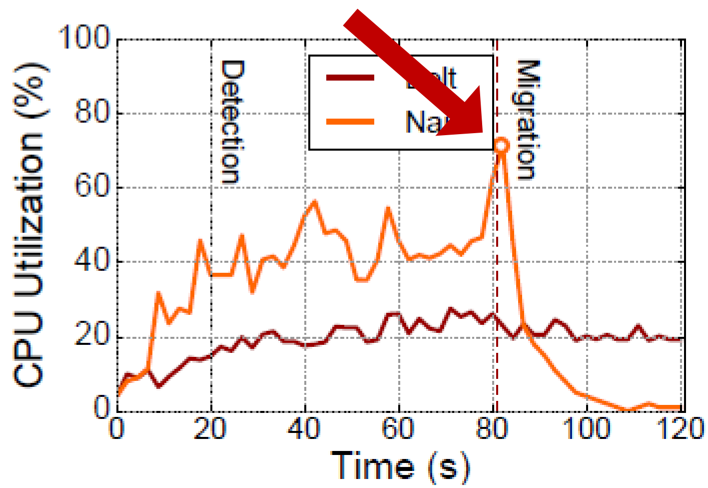
- Several performance attacks (DoS, RFAs, VM pinpointing)
- Target specific, critical resource → low CPU pressure



3. Practical DoS Attacks

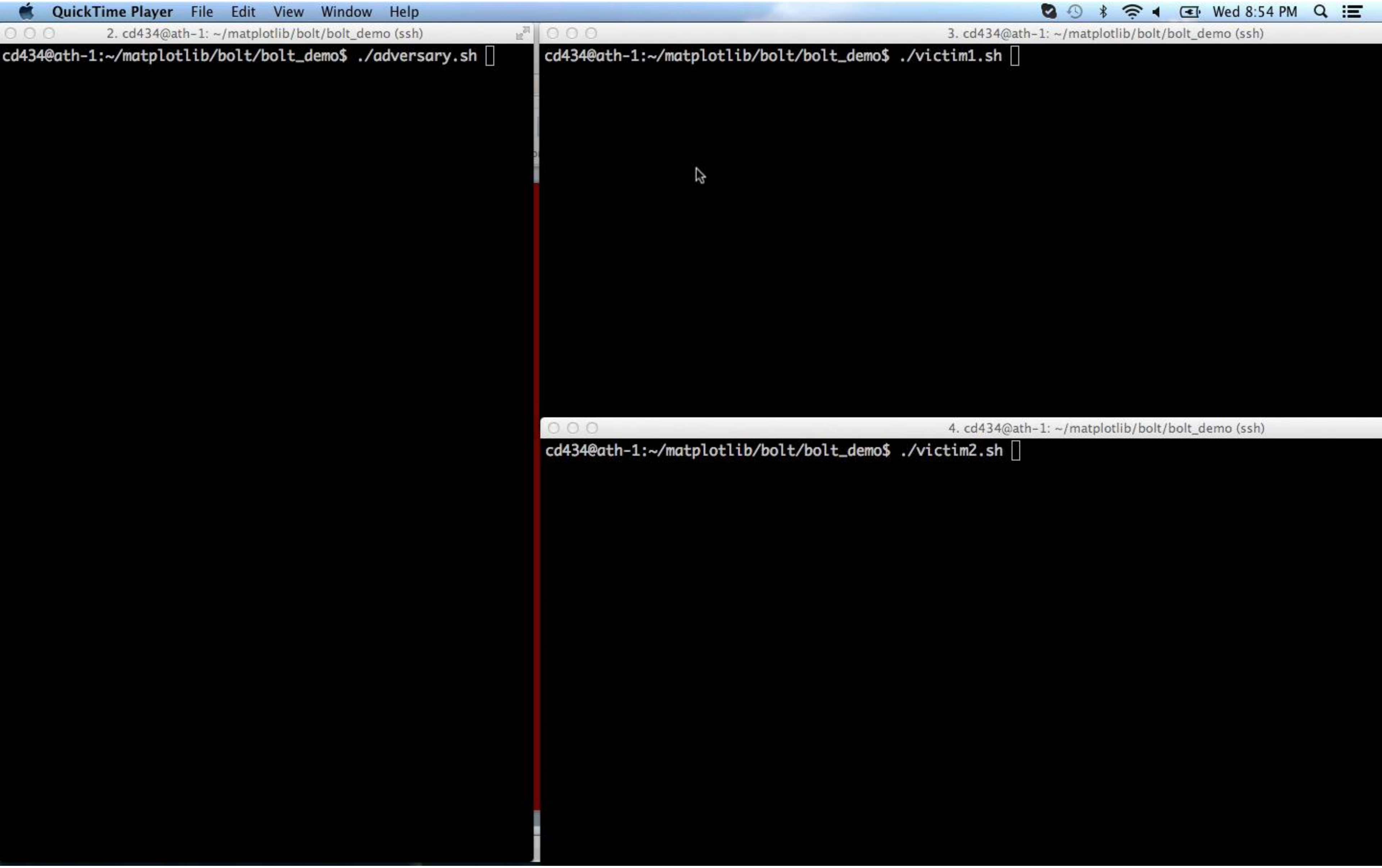
- Launched against same 108 applications as before
- On average 2.2x higher execution time and up to 9.8x
- For interactive services, on average 42x increase in tail latency and up to 140x





- Bolt does not saturate CPU
- Naïve attacker gets migrated

Demo

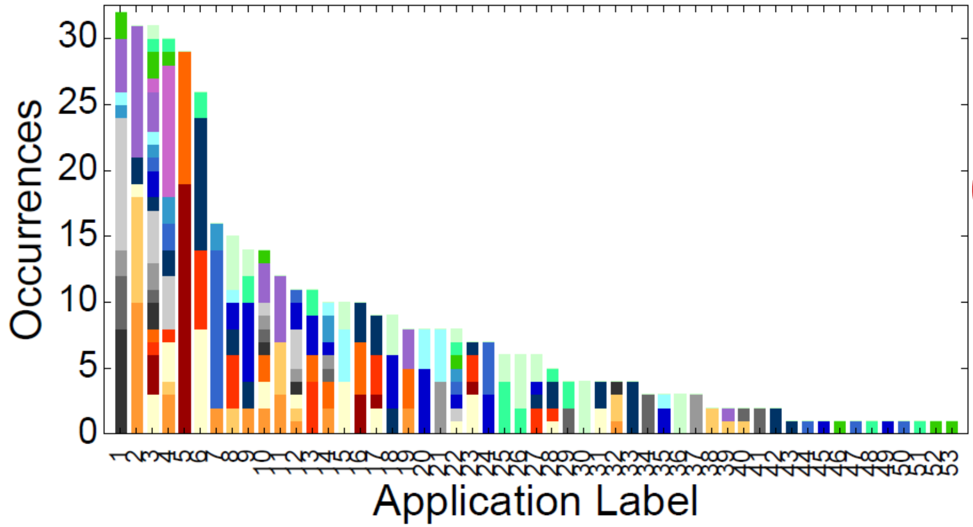

QuickTime Player File Edit View Window Help
Wed 8:54 PM
2. cd434@ath-1: ~/matplotlib/bolt/bolt_demo (ssh)
cd434@ath-1:~/matplotlib/bolt/bolt_demo$ ./adversary.sh
3. cd434@ath-1: ~/matplotlib/bolt/bolt_demo (ssh)
cd434@ath-1:~/matplotlib/bolt/bolt_demo$ ./victim1.sh
4. cd434@ath-1: ~/matplotlib/bolt/bolt_demo (ssh)
cd434@ath-1:~/matplotlib/bolt/bolt_demo$ ./victim2.sh

User Study

- 20 independent users from Stanford and Cornell
- Cluster
  - 200 EC2 servers, c3.8xlarge (32vCPUs, 60GB memory)
- Rules:
  - 4vCPUs per machine for Bolt
  - All users have equal priority
  - Users use thread pinning
  - Users can select specific instances
  - Training set: 120 apps incl. analytics, webapps, scientific, etc.

Accuracy of App Labeling

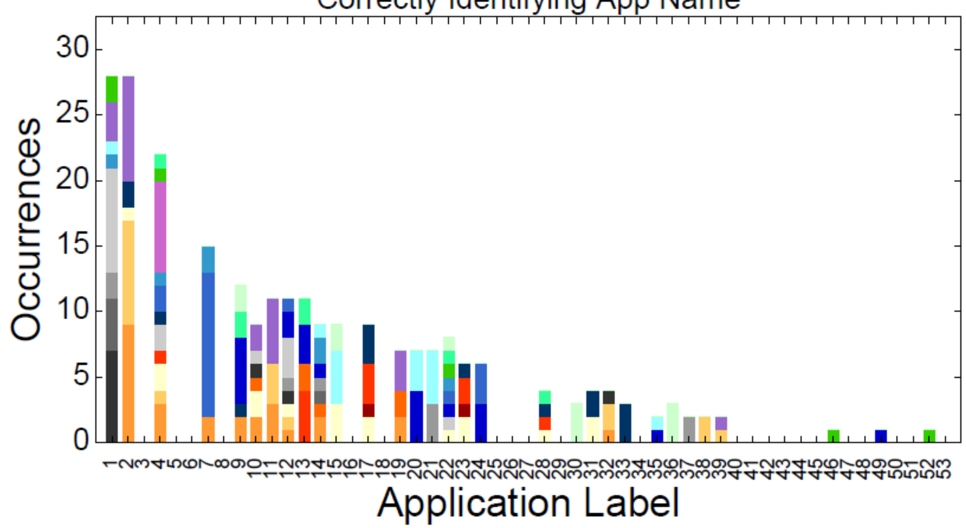
Ground Truth



53 app classes

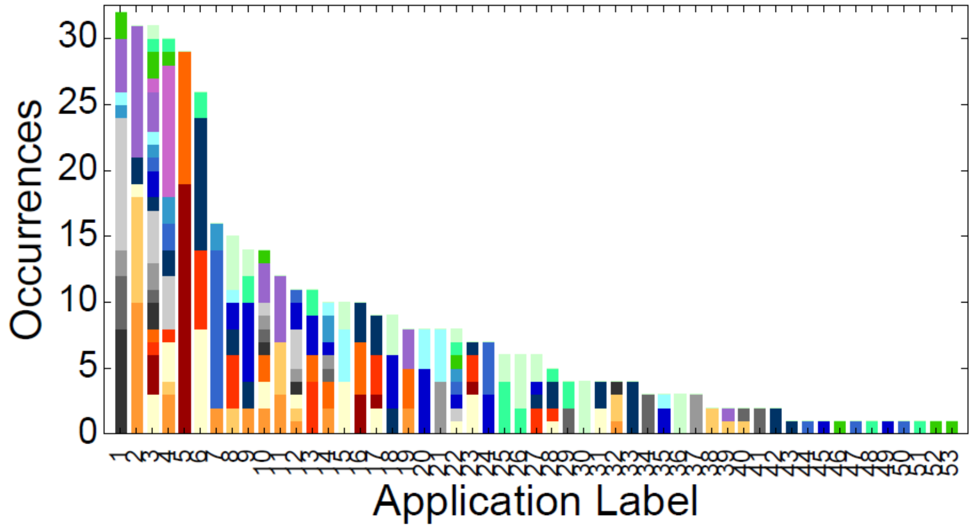
(analytics, webapps, FS/OS, HLS/sim, other…)

Correct app labels 63%

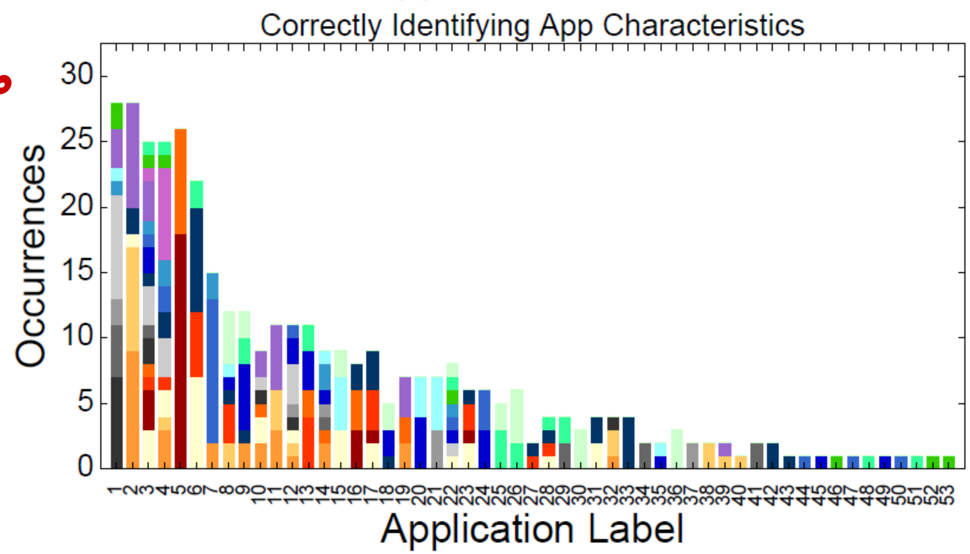


Accuracy of App Characterization

Ground Truth

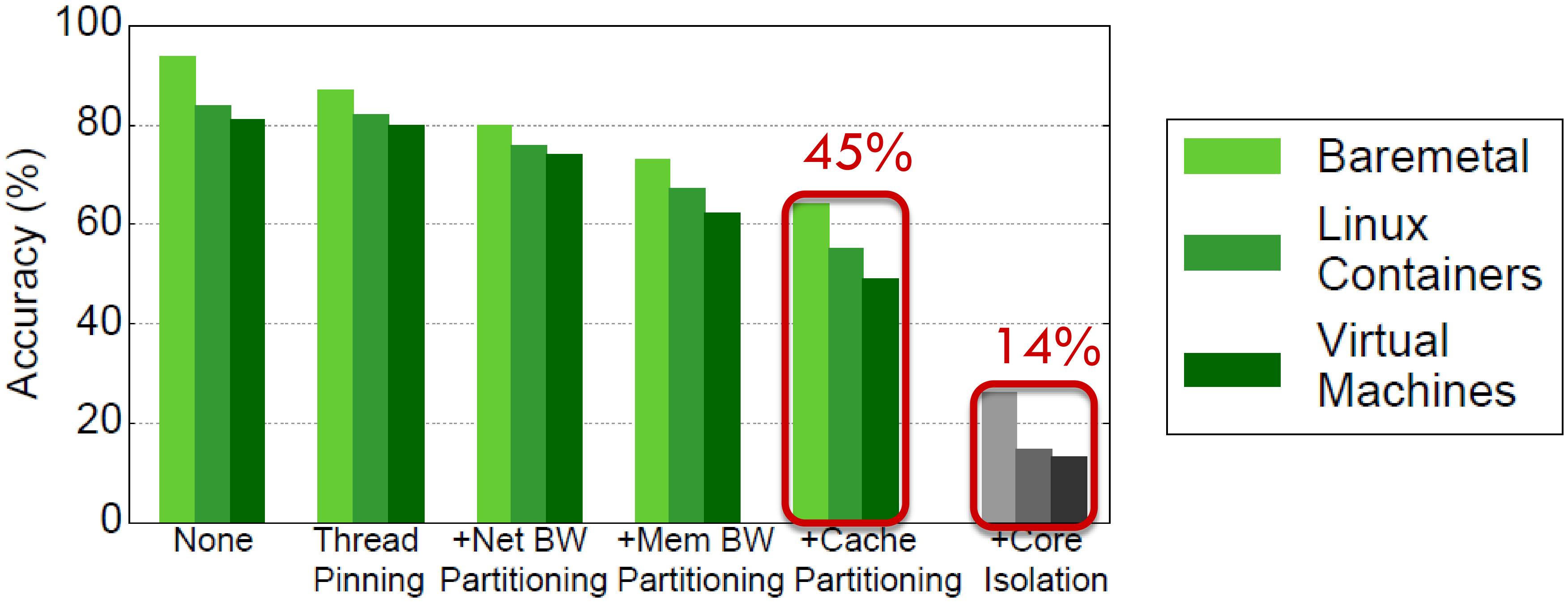


Correct app characteristics 88%



Performance attack results in the paper

The Value of Isolation



- Need more scalable, fine-grain, and complete isolation techniques

Conclusions

- Bolt: highlight the security vulnerabilities from lack of isolation
  - Fast detection using online data mining techniques
  - Practical, hard-to-detect performance attacks
  - Current isolation helpful but insufficient

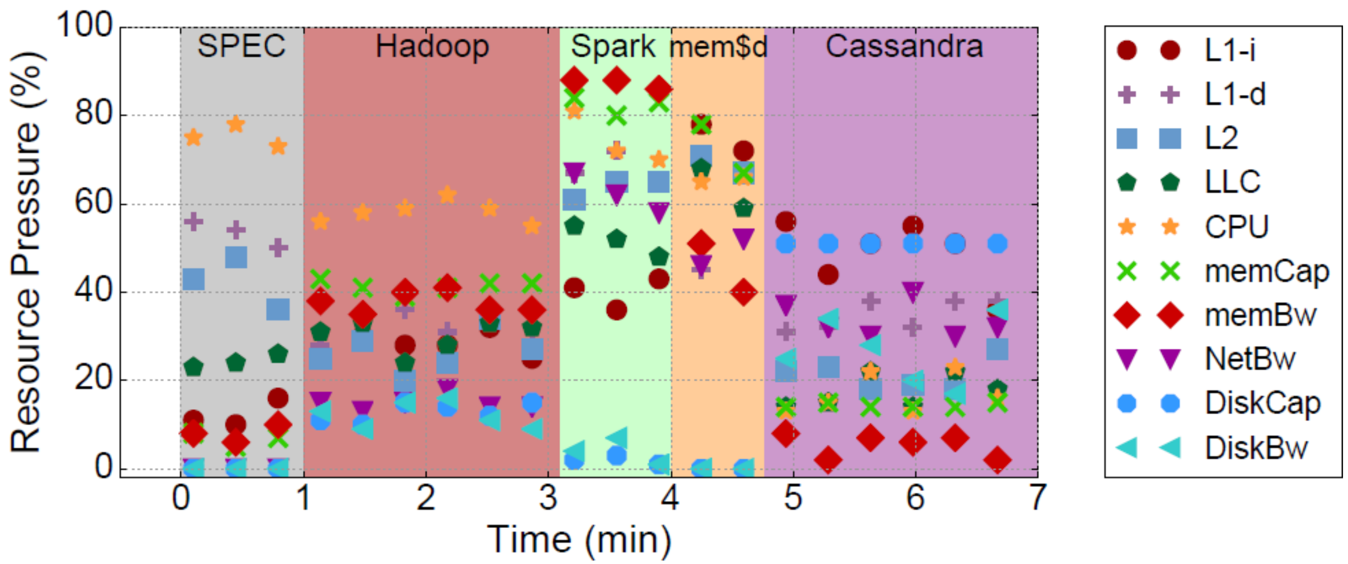
- In the paper:
  - Sensitivity to Bolt parameters
  - Sensitivity to applications and platform parameters
  - User study details
  - More performance attacks (resource freeing, VM pinpointing)

Questions?

- Bolt: highlight the security vulnerabilities from lack of isolation
  - Fast detection using online data mining techniques
  - Practical, hard-to-detect performance attacks
  - Current isolation helpful but insufficient

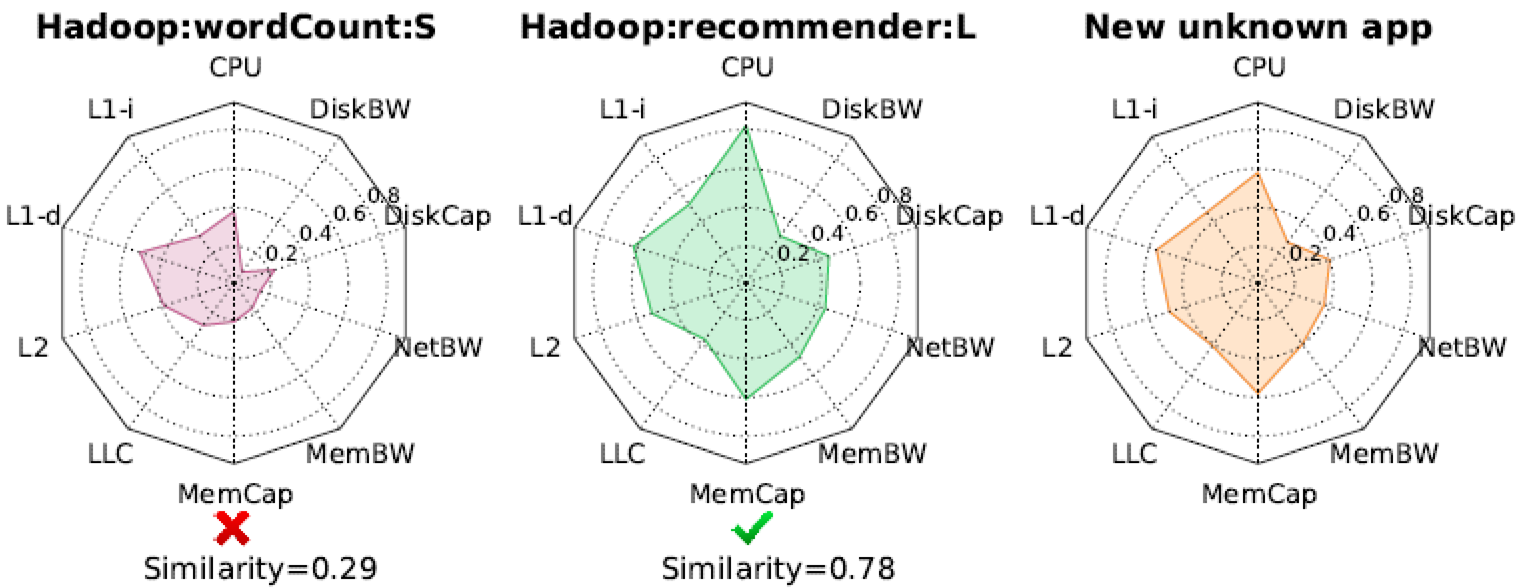
- In the paper:
  - Sensitivity to Bolt parameters
  - Sensitivity to applications and platform parameters
  - User study details
  - More performance attacks (resource freeing, VM pinpointing)

Evolving Applications



- Cloud applications change behavior
- Users use the same cloud resources for several apps over time
- Bolt periodically wakes up, checks if app profile has changed; if so, reprofile & reclassify

Inference Within a Framework



- Within a framework, dataset and choice of algorithm affect resource requirements
- Bolt matches a new unknown application to apps in a framework by distinguishing their resource needs