REPTTACK: EXPLOITING CLOUD SCHEDULERS TO GUIDE CO-LOCATION ATTACKS

April 26, 2022

Chongzhou Fang, Han Wang, Najmeh Nazari, Behnam Omidi, Avesta Sasan, Khaled N. Khasawneh, Setareh Rafatirad,and Houman Homayoun

University of California, Davis
George Mason University

UCDAVIS
ELECTRICAL AND COMPUTER ENGINEERING

Introduction

Micro-architectural Attacks

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

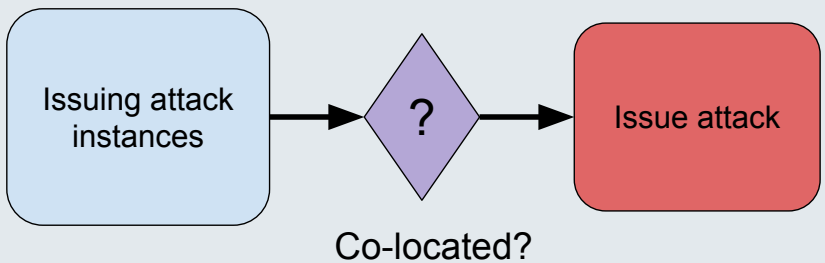
Micro-architectural attacks have become a threat to cloud users!

1. Side-channel attack.
2. Transient execution attack.
3. Rowhammer attack.
4. Faults attack.
5. ...

Prerequisite of Micro-architectural Attacks

Workflow of Attack (Ristenpart et al., 2009)

1. Submit attack program to the cloud.
2. Determine if victim is co-located.
3. Start stealing information / interfere with victim program.

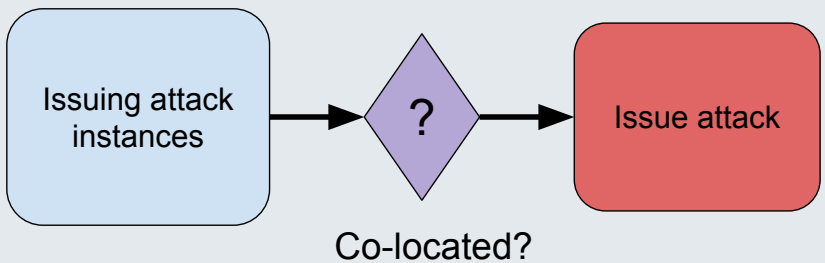


Prerequisite of Micro-architectural Attacks

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Workflow of Attack (Ristenpart et al., 2009)

1. Submit attack program to the cloud.
2. Determine if victim is co-located.
3. Start stealing information / interfere with victim program.



Before attack, achieving co-location is required.

Motivation

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

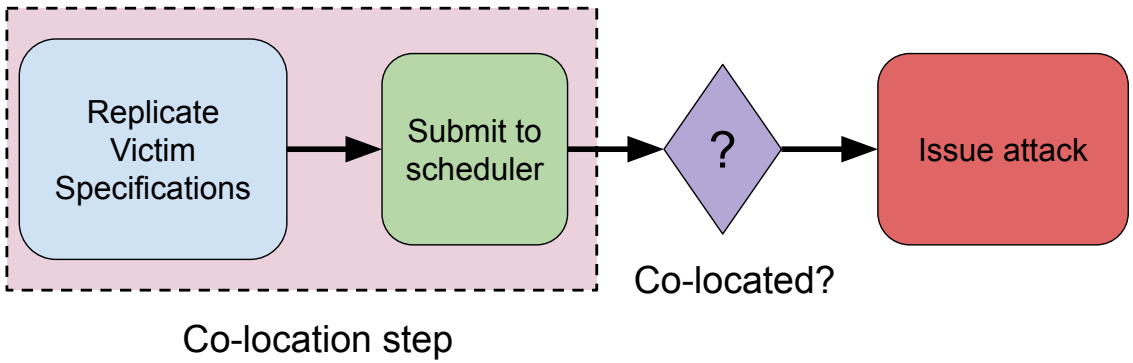
Important to study how to achieve co-location

Brute-force issuing can be easy to defend.

- For attackers: without co-location strategies, subsequent attacks are impossible
- For defenders: more efficient to defend and patch at scheduler level

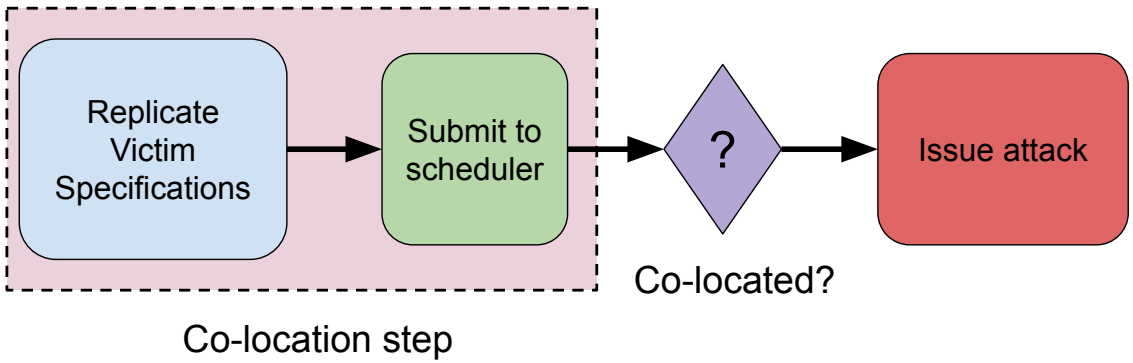
Vulnerabilities in the scheduler should be studied.

Focus of this work



- We focus on co-location step.

Focus of this work



- We focus on co-location step.
- We don't consider how the attacker obtains location status.
- We don't consider how a specific type of attack works.

Threat Model

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Cloud providers

- Trusted, do not assist attackers
- Treat all users (malicious and non-malicious) equally

Threat Model

Cloud providers

- Trusted, do not assist attackers
- Treat all users (malicious and non-malicious) equally

Users

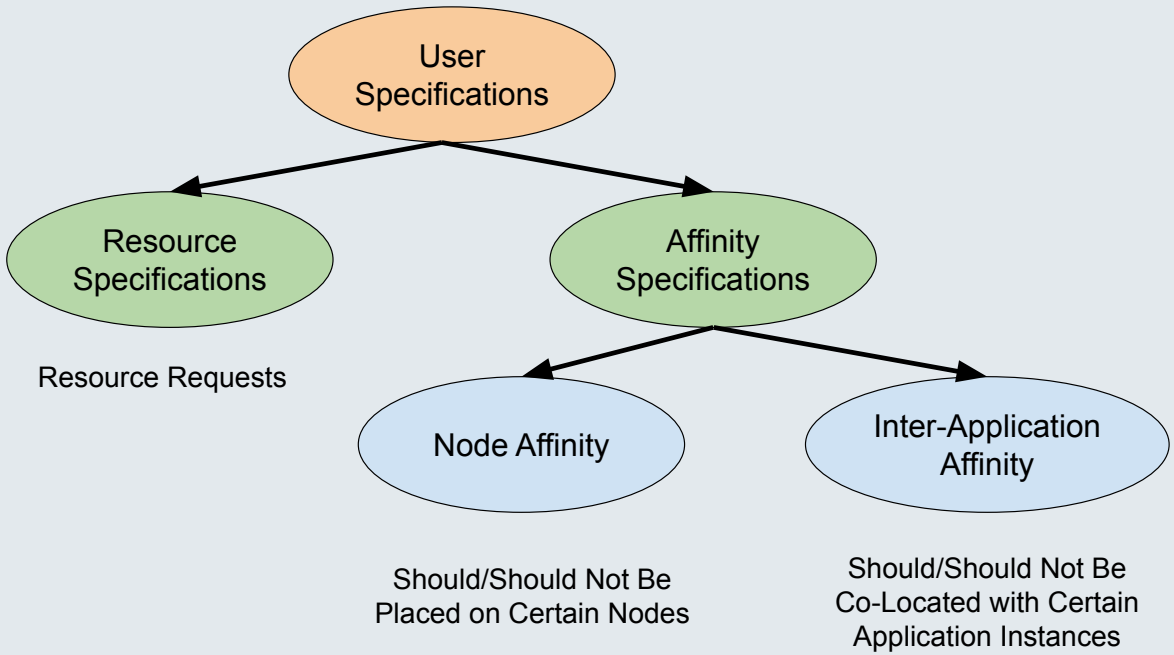
- All users have the same privilege and can only access their own allocated resources.
- Attackers knows about victim applications.
- Non-malicious users always try to optimize the scheduling outcome.

Method

Targeted Scheduler

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

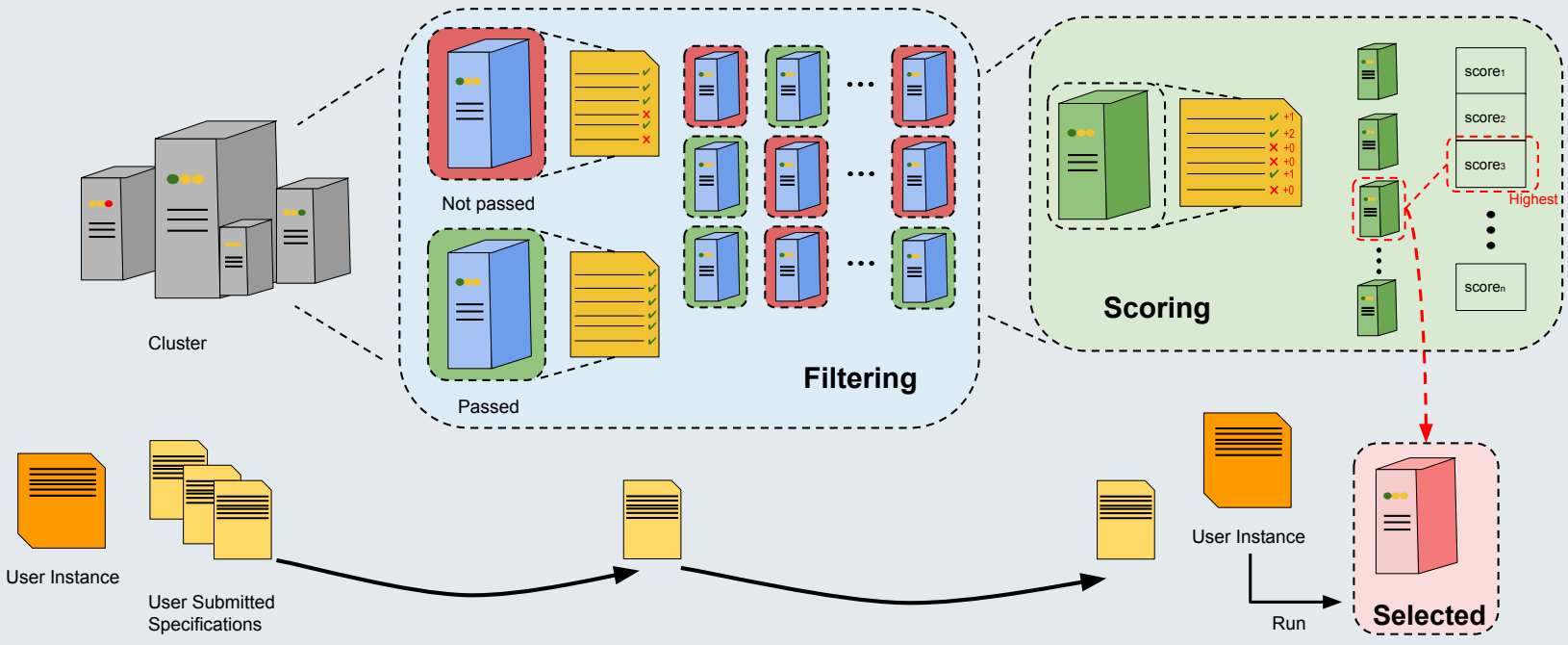
User submitted requirements



Targeted Scheduler

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Filter-score scheduler

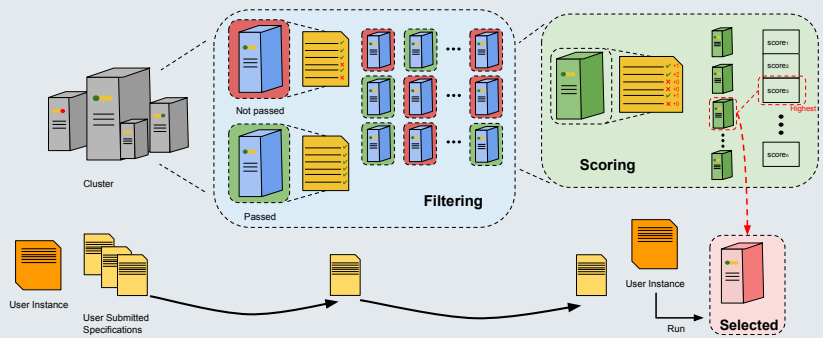


Targeted Scheduler

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Filter-score scheduler

Widely used type of scheduling pattern ("Kubernetes," 2021; "OpenStack," 2021).

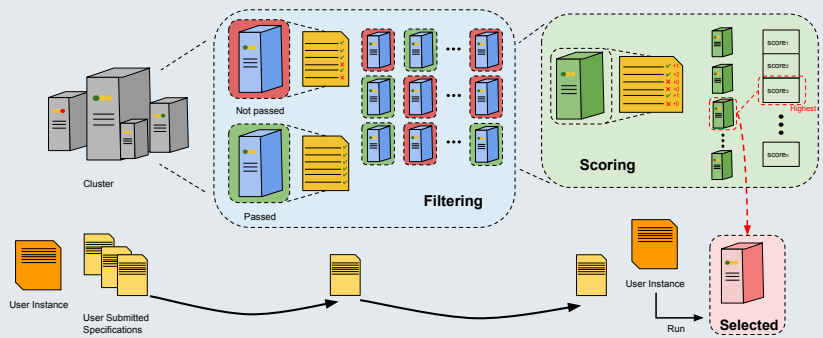


- Filtering and scoring based on user specifications

Targeted Scheduler

Filter-score scheduler

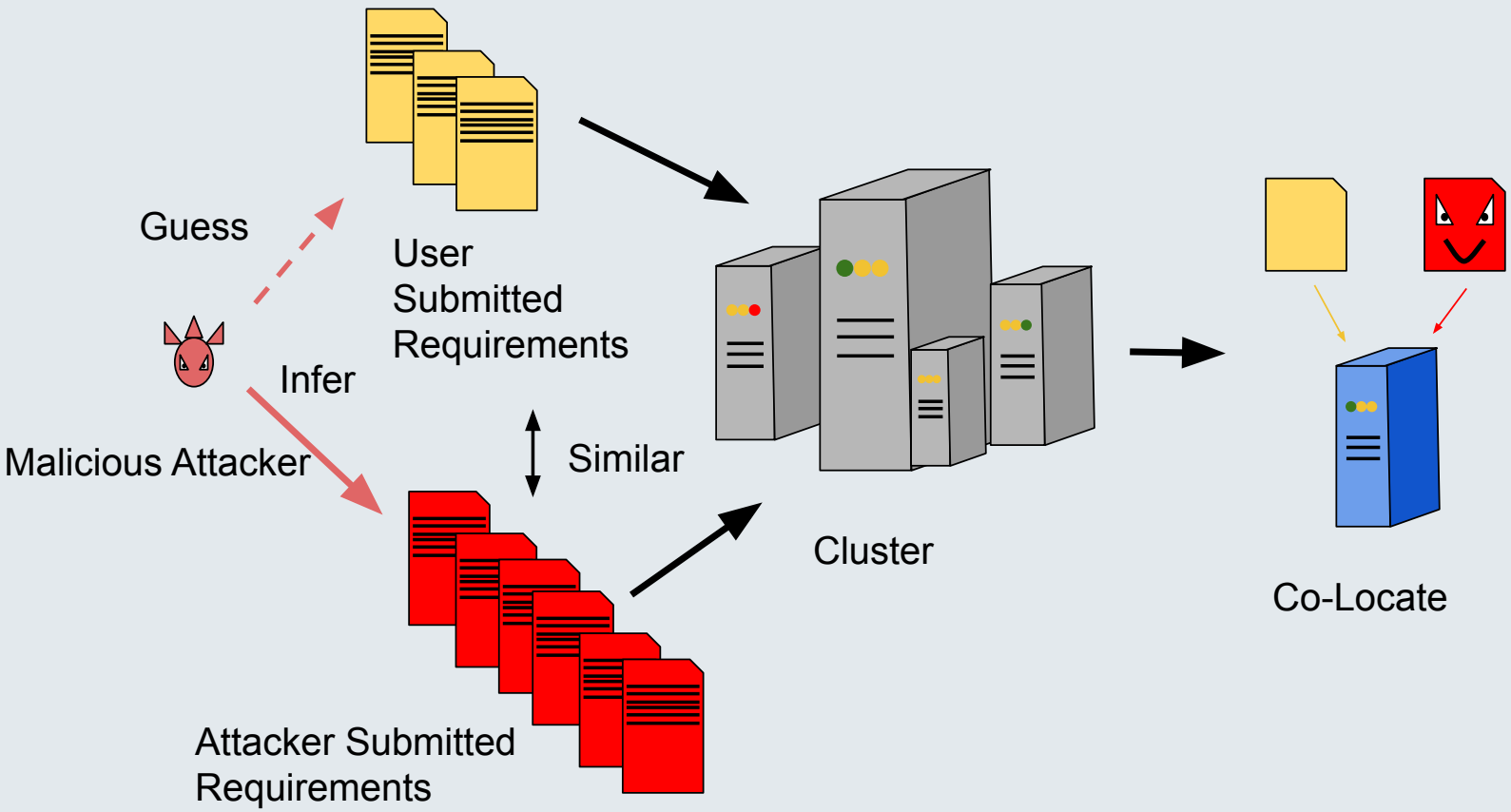
Widely used type of scheduling pattern ("Kubernetes," 2021; "OpenStack," 2021).



- Filtering and scoring based on user specifications
- Filtering: Find a list of candidates that satisfy user needs
- Scoring: Rate every candidate and select the one with highest score

Attack Strategy

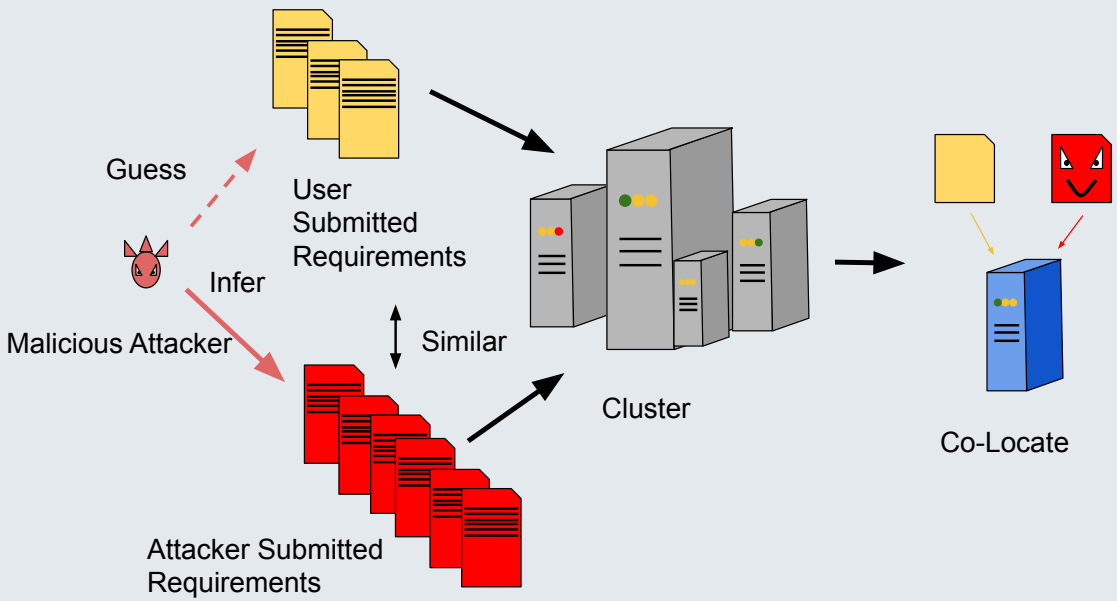
Replicating user specifications



Attack Strategy

Replicating user specifications

Exploit scheduler features.

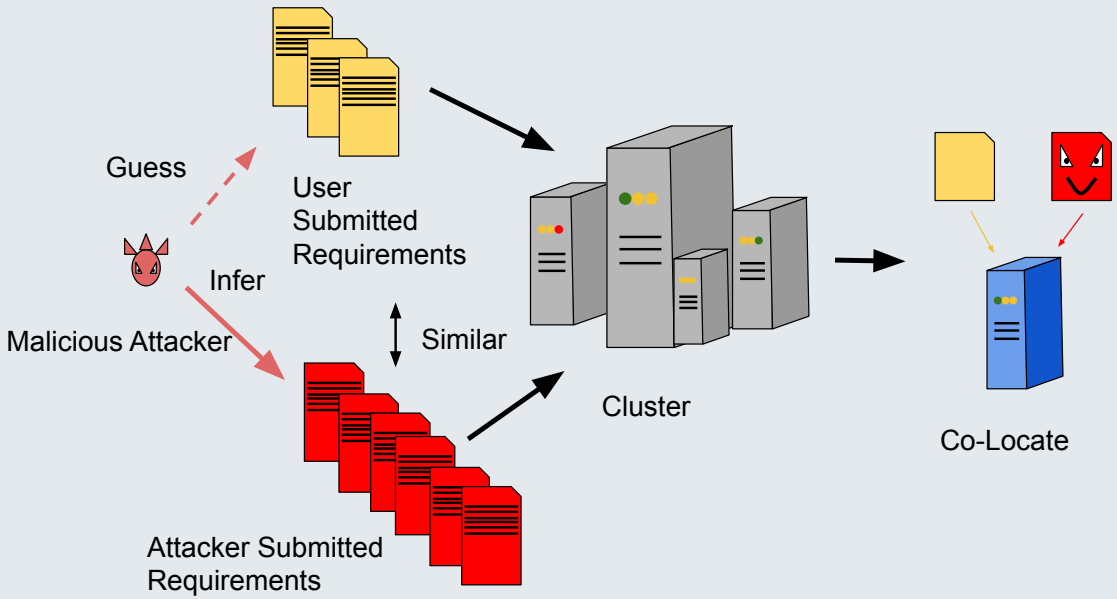


- Infer victim submitted requirements/preferences

Attack Strategy

Replicating user specifications

Exploit scheduler features.



- Infer victim submitted requirements/preferences
- Replicate these specifications to the scheduler

Evaluation

Experiment Setup

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Simulation

- Python behaviorial simulator, implementation based on Kubernetes ("Kubernetes," 2021)[1].

[1] It has been re-written in C++ and will be released in the future.

Experiment Setup

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Simulation

- Python behaviorial simulator, implementation based on Kubernetes ("Kubernetes," 2021)[1].
- Server configurations: generated randomly.
- Applications: generated randomly.

[1] It has been re-written in C++ and will be released in the future.

Experiment Setup

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Simulation

Cluster

- Experiment conducted on Kubernetes deployed on CloudLab (Duplyakin et al., 2019).

Experiment Setup

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Simulation

Cluster

- Experiment conducted on Kubernetes deployed on CloudLab (Duplyakin et al., 2019).
- Server configurations: heterogeneous hardware features, generated randomly.
- Applications: randomly selected from popular docker apps, user specifications generated randomly.

Simulation Results

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Single-instance attack

- What are the factors that affect attack success rate?

Simulation Results

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Single-instance attack

- What are the factors that affect attack success rate?
- How high can the co-location rate reach?

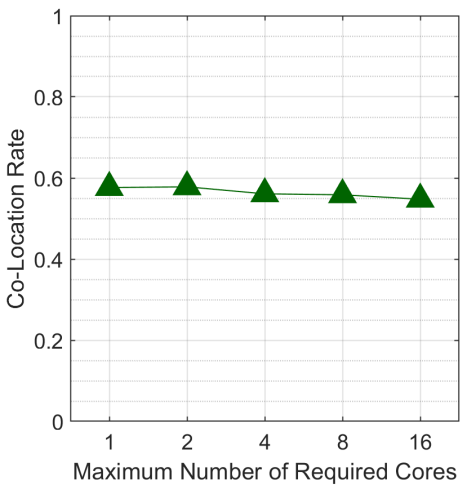
Simulation Results

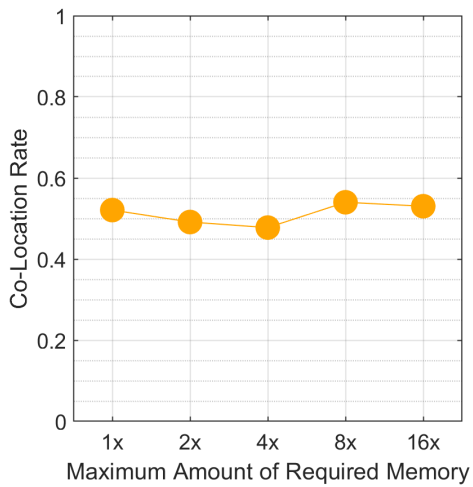
UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

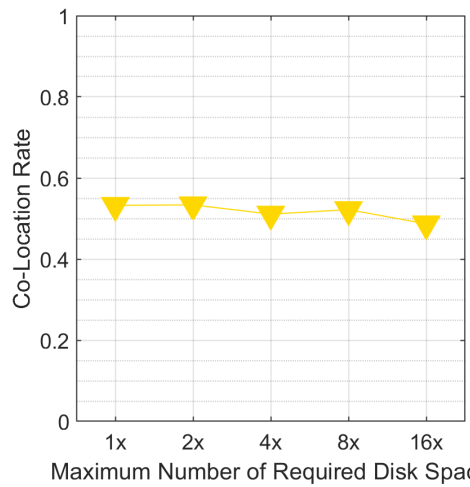
Single-instance attack

- What are the factors that affect attack success rate?
- How high can the co-location rate reach?

Resource requirements:





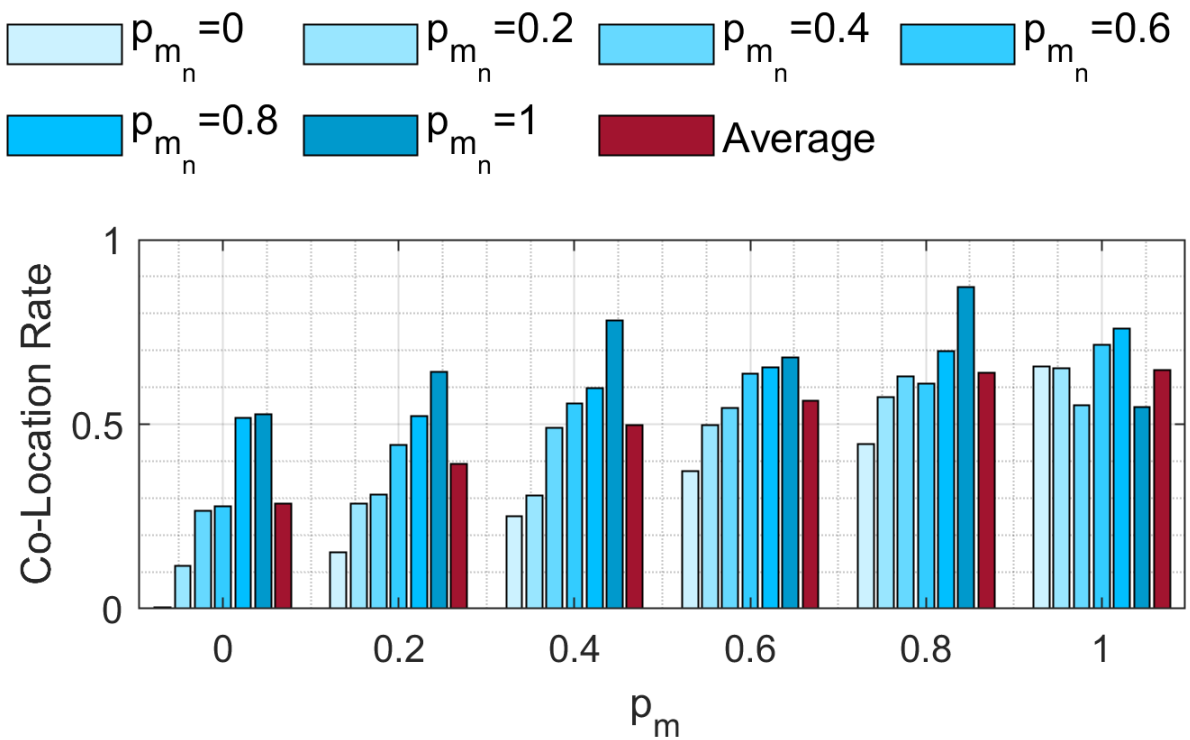


Simulation Results

Single-instance attack

- What are the factors that affect attack success rate?
- How high can the co-location rate reach?

Affinity:



Simulation Results

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Multi-instance attack

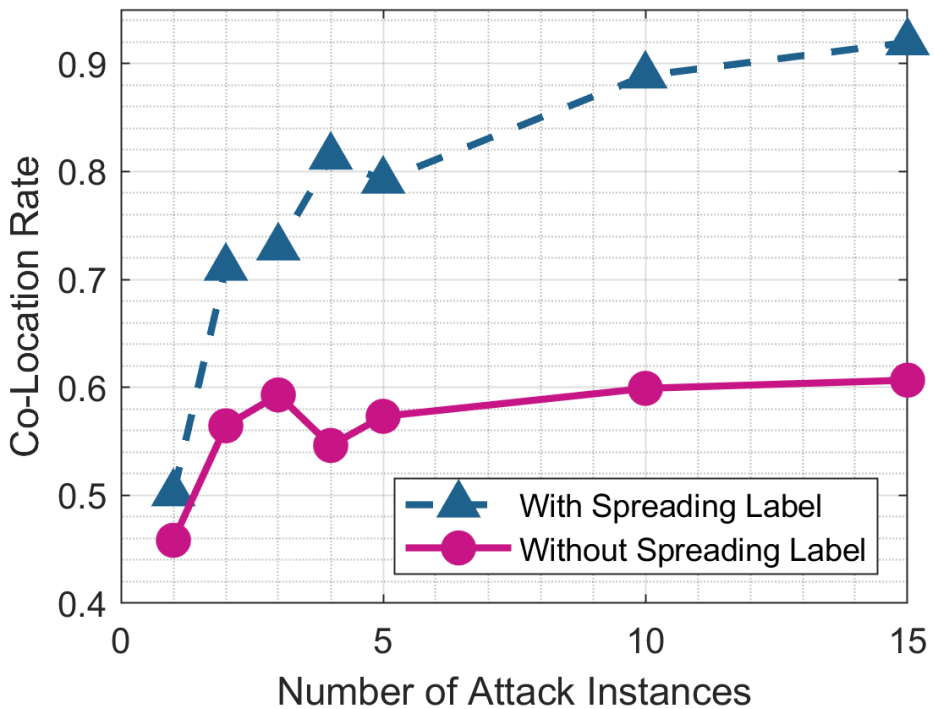
- Does increasing number of attack instances improve attack success rate?

Simulation Results

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Multi-instance attack

- Does increasing number of attack instances improve attack success rate?

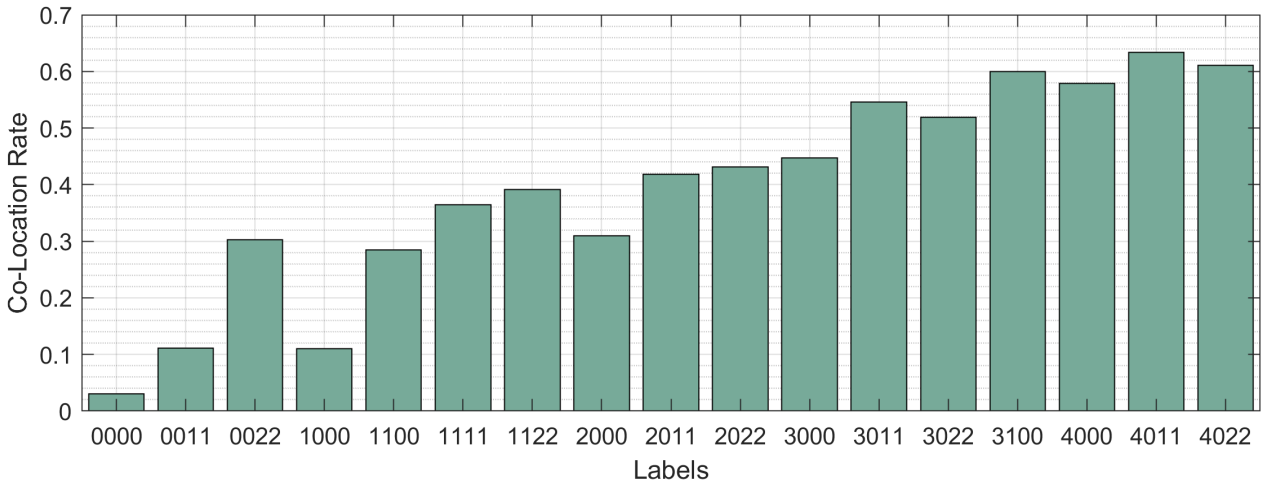


Cluster Experiment Results

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Single-instance attack

- What are the factors that affect attack success rate?
- How high can the co-location rate reach?



Notation: 1 2 3 4

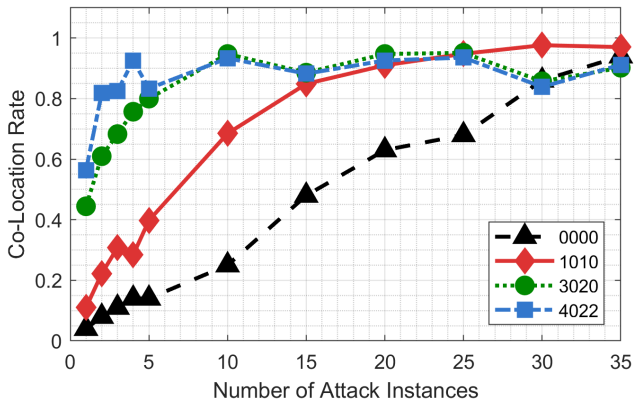
- 1 #. of Required Node Affinity
- 2 #. of Preferred Node Affinity
- 3 #. of Required Inter-Application Affinity
- 4 #. of Preferred Inter-Application Affinity

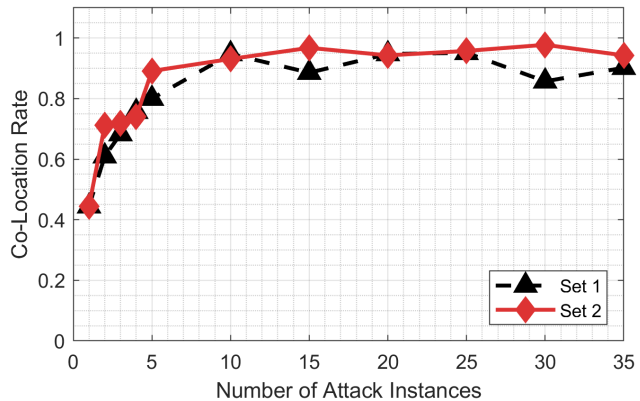
Cluster Experiment Results

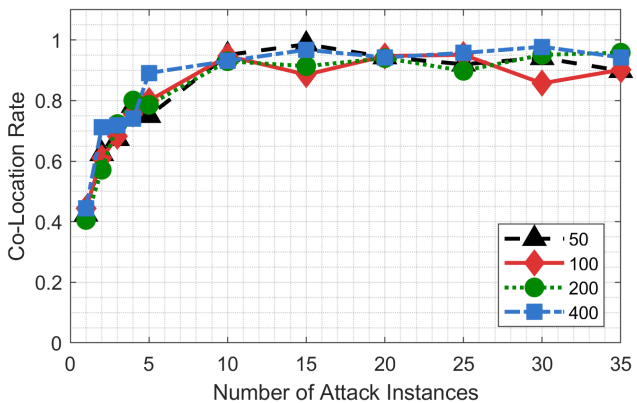
UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

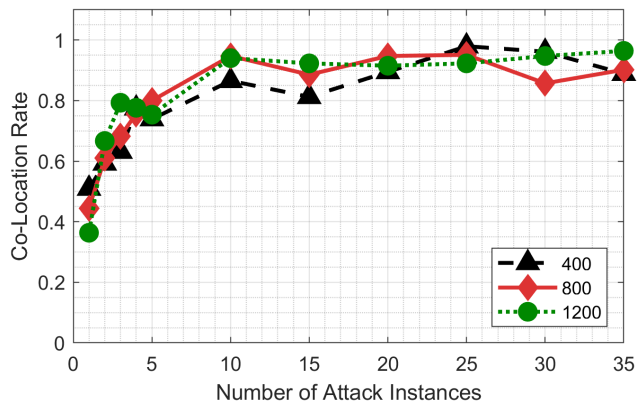
Multi-instance attack

- Does increasing number of attack instances improve attack success rate?



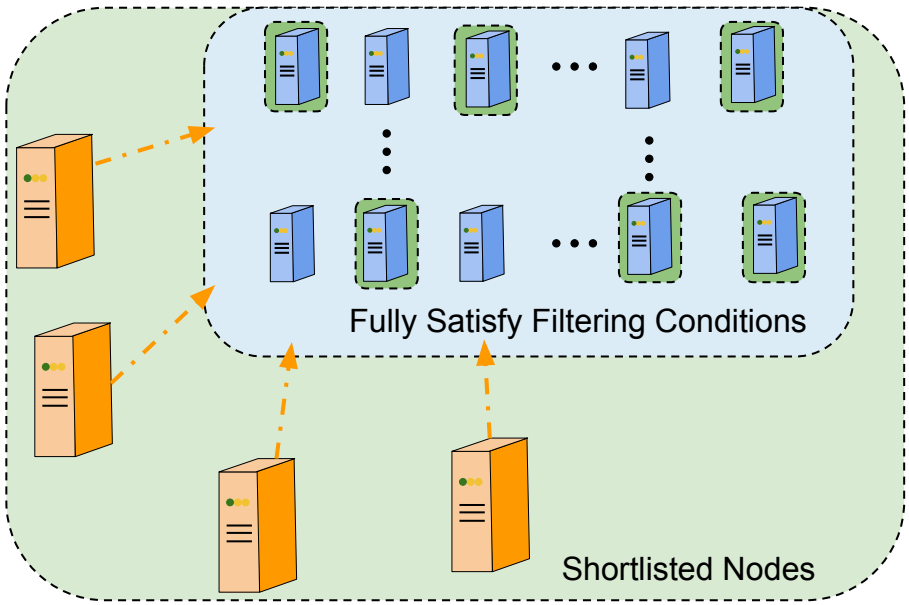






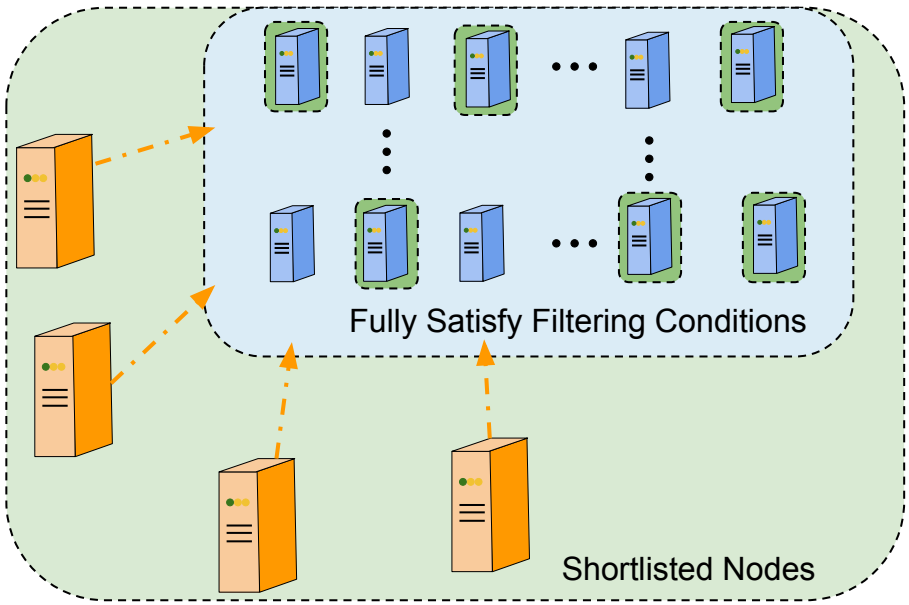
Mitigation

Mitigation Strategy



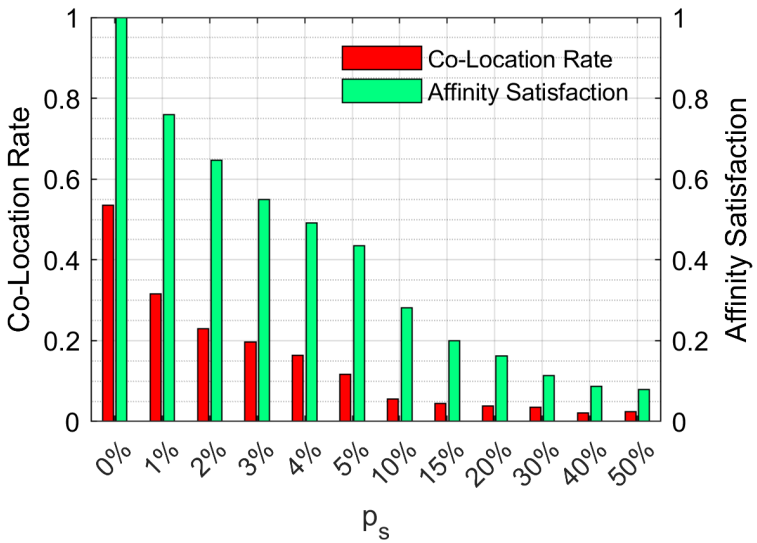
- Randomly skip affinity check during filtering.

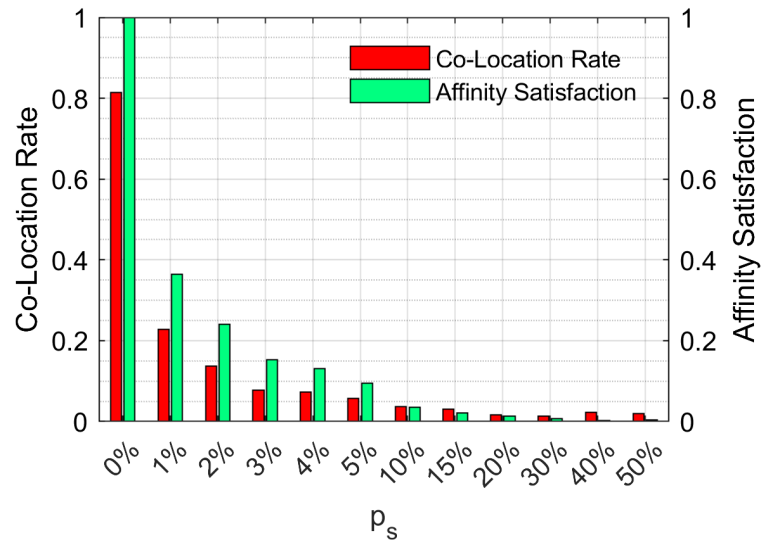
Mitigation Strategy



- Randomly skip affinity check during filtering.
- **Adding randomness!**

Mitigation





Cost: measured by average number of violated specifications

p_{m_n}, p_{m_a}	$p_s = 0\%$	$p_s = 1\%$	$p_s = 2\%$	$p_s = 3\%$	$p_s = 4\%$	$p_s = 5\%$	$p_s = 10\%$	$p_s = 15\%$	$p_s = 20\%$
0.5	0.00	0.45	0.68	0.88	1.07	1.19	1.68	2.00	2.17
0.9	0.00	1.65	2.29	2.78	3.02	3.33	4.04	4.40	4.57

Discussion

Trade-off

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Trade-off between security and performance

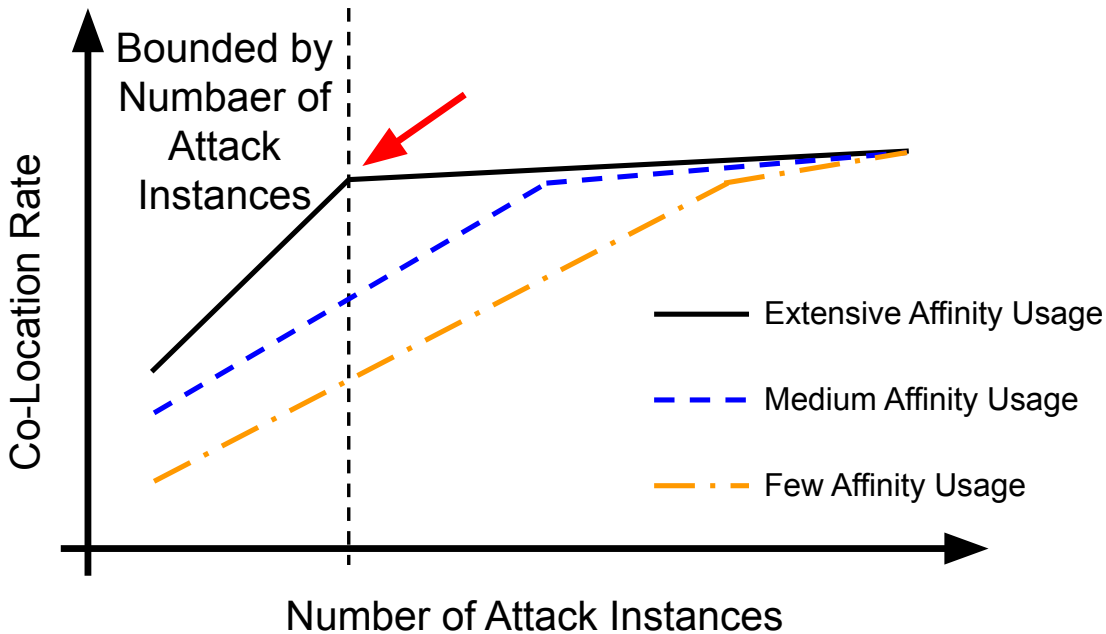
Let users have control over scheduling outcomes

- Better performance: can run on more suitable machines
- Worse security: location in the cloud can be relatively accurately determined

Roofline Model

UCDAVIS ELECTRICAL AND COMPUTER ENGINEERING

Optimum trade-off point exists



Suggestions

For cloud managers

- Expose heterogeneity as little as possible
- Bring randomness to scheduling process

Suggestions

For cloud managers

- Expose heterogeneity as little as possible
- Bring randomness to scheduling process

For users

- Utilize heterogeneity as little as possible
- Keep scheduling specifications confidential

Suggestions

For attackers

- Study target applications
- Use multiple attack instances with different possible specifications to increase coverage
- Be aware of the trade-off point of attack instance: optimize for cost of attack

Conclusion

Conclusion

Our contributions

- Affinity feature in filter-score schedulers are prone to be exploited
- Repttack: an attack method to increase the chance of achieving co-location in a heterogeneous cluster
- Mitigation technology
- Guidelines for cloud managers and users

References

Duplyakin, D., Ricci, R., Maricq, A., Wong, G., Duerig, J., Eide, E., Stoller, L., Hibler, M., Johnson, D., Webb, K., Akella, A., Wang, K., Ricart, G., Landweber, L., Elliott, C., Zink, M., Cecchet, E., Kar, S., & Mishra, P. (2019). The design and operation of CloudLab. *Proceedings of the USENIX Annual Technical Conference (ATC)*, 1–14. https://www.flux.utah.edu/paper/duplyakin-atc19

Kubernetes [[Online; accessed 1 May 2021]]. (2021).

OpenStack [[Online; accessed 1 May 2021]]. (2021).

Ristenpart, T., Tromer, E., Shacham, H., & Savage, S. (2009). Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. *Proceedings of the ACM conference on Computer and communications security*, 199–212.