LockedDown: Exploiting Contention on Host-GPU PCIe Bus for Fun and Profit

IEEE EuroS&P 2022

Mert Side [1] Fan Yao [2] Zhenkai Zhang [3]

[1]Texas Tech University

[2]University of Central Florida

[3]Clemson University

June 8, 2022

TEXAS TECH UNIVERSITY

UNIVERSITY OF CENTRAL FLORIDA

CLEMSON UNIVERSITY

Introduction

- Discrete GPUs have become an integral part of computer systems,
  - for performing resource-intensive visual computing tasks,
  - for performing massively parallel computing tasks.

Introduction

- Discrete GPUs have become an integral part of computer systems,
  - for performing resource-intensive visual computing tasks,
  - for performing massively parallel computing tasks.
- Cloud service providers are using GPU virtualization techniques for sharing various computing resources with multiple users.

Introduction

- Discrete GPUs have become an integral part of computer systems,
  - for performing resource-intensive visual computing tasks,
  - for performing massively parallel computing tasks.
- Cloud service providers are using GPU virtualization techniques for sharing various computing resources with multiple users.
- We show an attack surface of host-GPU communication and disclose a new side-channel vulnerability that can be exploited to mount realistic attacks.

Introduction

- Discrete GPUs have become an integral part of computer systems,
  - for performing resource-intensive visual computing tasks,
  - for performing massively parallel computing tasks.
- Cloud service providers are using GPU virtualization techniques for sharing various computing resources with multiple users.
- We show an attack surface of host-GPU communication and disclose a new side-channel vulnerability that can be exploited to mount realistic attacks.
- To demonstrate, we conducted two case studies:
  1. a covert communication channel for data exfiltration across virtual isolation boundaries,
  2. a website fingerprinting attack that can infer the web browsing activities.

Side-Channel

Observation 1

The host-GPU PCIe bus is shared among processes running in different security domains.

Side-Channel

Observation 1

The host-GPU PCIe bus is shared among processes running in different security domains.

Observation 2

The contention on the PCIe bus is measurable in the form of data transfer latencies.

Side-Channel

Observation 1

The host-GPU PCIe bus is shared among processes running in different security domains.

Observation 2

The contention on the PCIe bus is measurable in the form of data transfer latencies.

Threat

The measurable contention on this bus can be leveraged as a side-channel to leak information across strong isolation boundaries.

Topology

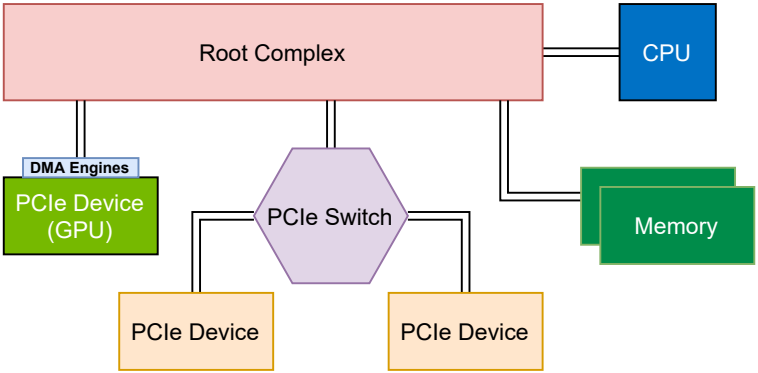
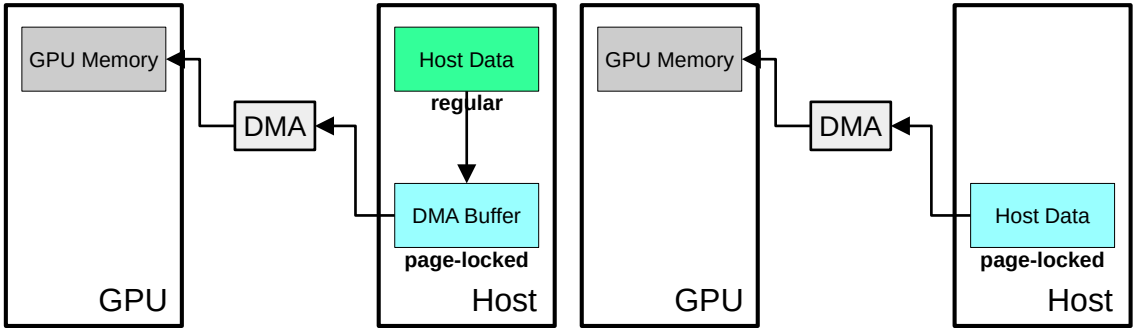


Figure 1: The PCIe topology.

Page-Locked Memory Allocation in CUDA

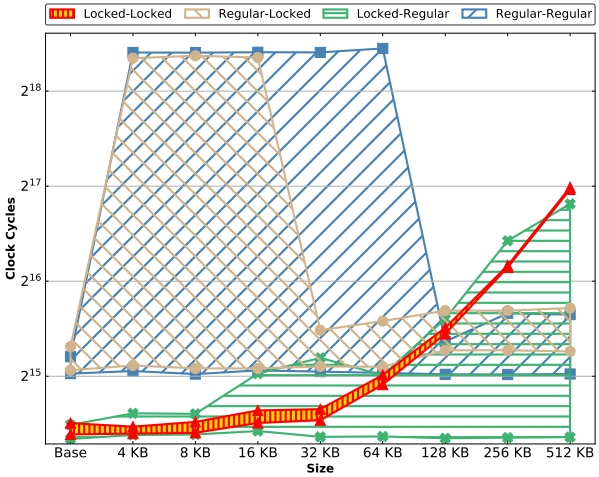


(a) Two-step data transfer.

(b) Direct data transfer.

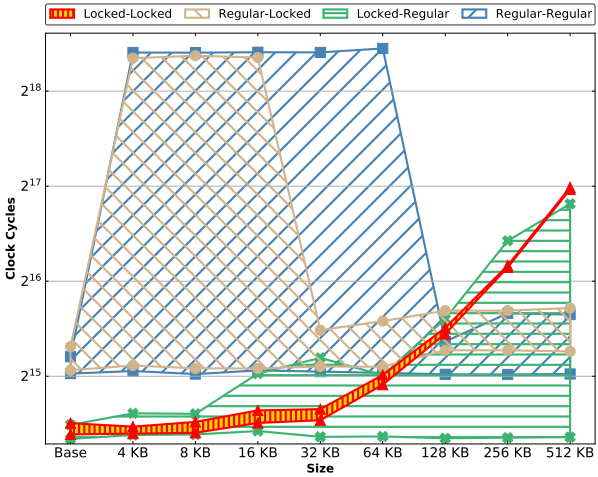
Figure 2: CUDA runtime data transfers.

Contention Measurement



Data transfer latencies measured by *Alice*. The first part of "Locked/Regular – Locked/Regular" indicates whether *Alice*'s data resides in page-locked memory or regular memory, and the second part indicates *Bob*'s.

Contention Measurement



Data transfer latencies measured by *Alice*. The first part of "Locked/Regular – Locked/Regular" indicates whether *Alice*'s data resides in page-locked memory or regular memory, and the second part indicates *Bob*'s.

Observation 3

Contention on the host-GPU PCIe bus can lead to observable and consistent increases in the host-to-GPU data transfer time if the host data resides in page-locked memory.

Cross-VM Covert Channel Attack

- We assume that there are two colluding parties, a sender and a receiver, are on the same platform but in different security domains.

Cross-VM Covert Channel Attack

- We assume that there are two colluding parties, a sender and a receiver, are on the same platform but in different security domains.

- The sender has access to some sensitive data, and it attempts to transmit this piece of data to the receiver through a covert channel.

Cross-VM Covert Channel Attack

- We assume that there are two colluding parties, a sender and a receiver, are on the same platform but in different security domains.
- The sender has access to some sensitive data, and it attempts to transmit this piece of data to the receiver through a covert channel.
- The platform is equipped with a modern GPU accessible by both the sender and receiver.

Cross-VM Covert Channel Attack

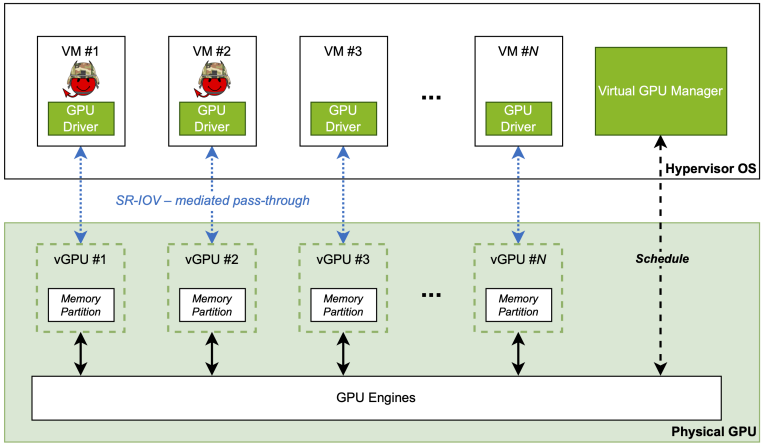


Figure 3: NVIDIA vGPU architecture.

Cross-VM Covert Channel Attack

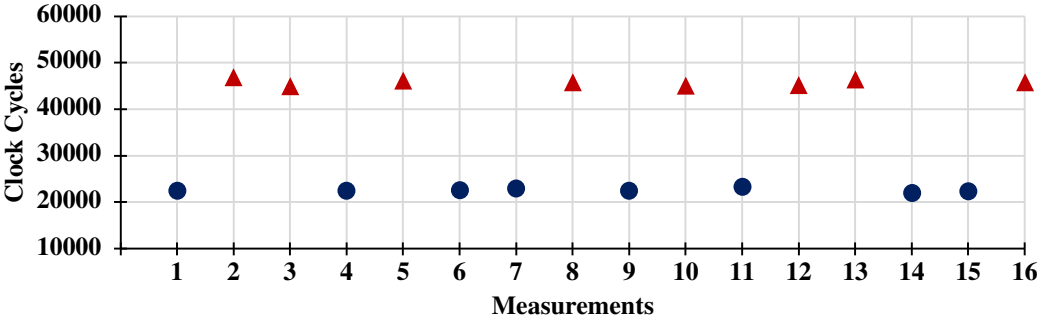


Figure 4: 16 measured data transfer latencies corresponding to bits "0110100101011001".

Cross-VM Covert Channel Attack

Table 1: Platforms of Chameleon Cloud on which the covert channel is evaluated.

System	Platform	CPU	Memory	OS	GPU	# vGPUs
A	Dell PowerEdge R740	2 × Xeon Gold 6126	12 × 16 GB DDR4-2666	CentOS 8.3	Quadro RTX 6000	6
B	Dell PowerEdge C4140	2 × Xeon Gold 6230	8 × 16 GB DDR4-2933	CentOS 8.3	Tesla V100	8

Cross-VM Covert Channel Attack

Table 1: Platforms of Chameleon Cloud on which the covert channel is evaluated.

System	Platform	CPU	Memory	OS	GPU	# vGPUs
A	Dell PowerEdge R740	2 × Xeon Gold 6126	12 × 16 GB DDR4-2666	CentOS 8.3	Quadro RTX 6000	6
B	Dell PowerEdge C4140	2 × Xeon Gold 6230	8 × 16 GB DDR4-2933	CentOS 8.3	Tesla V100	8

Table 2: Bandwidth and error rate of the covert channel in a controlled environment.

System	Bandwidth	Error Rate μ (σ)
A	64 kbps	0.0088 (0.0043)
B	20 kbps	0.0029 (0.0005)

Cross-VM Covert Channel Attack

Table 1: Platforms of Chameleon Cloud on which the covert channel is evaluated.

System	Platform	CPU	Memory	OS	GPU	# vGPUs
A	Dell PowerEdge R740	2 × Xeon Gold 6126	12 × 16 GB DDR4-2666	CentOS 8.3	Quadro RTX 6000	6
B	Dell PowerEdge C4140	2 × Xeon Gold 6230	8 × 16 GB DDR4-2933	CentOS 8.3	Tesla V100	8

Table 2: Bandwidth and error rate of the covert channel in a controlled environment.

System	Bandwidth	Error Rate μ (σ)
A	64 kbps	0.0088 (0.0043)
B	20 kbps	0.0029 (0.0005)

Table 3: Bandwidth and error rate of the covert channel in different scenarios with synchronization.

Scenario	Bandwidth	Error Rate μ (σ)
1	90 kbps	0.0140 (0.005)
2	81 kbps	0.0038 (0.006)
3	88 kbps	0.1569 (0.076)

Website Fingerprinting Attack

- We assume that there is an attacker who wants to stealthily learn information about which websites have been visited by a victim.

Website Fingerprinting Attack

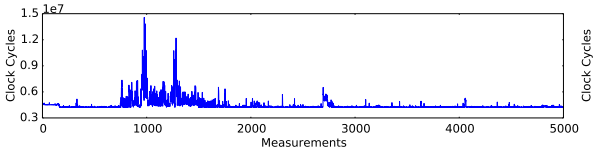
- We assume that there is an attacker who wants to stealthily learn information about which websites have been visited by a victim.

- The victim uses a personal computer to browse websites, and the computer is assumed to have a CUDA-enabled NVIDIA GPU.

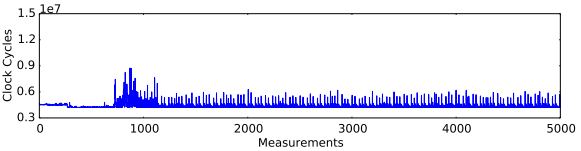
Website Fingerprinting Attack

- We assume that there is an attacker who wants to stealthily learn information about which websites have been visited by a victim.
- The victim uses a personal computer to browse websites, and the computer is assumed to have a CUDA-enabled NVIDIA GPU.
- We do not impose strong assumptions on the OS or the web browser, as long as it is has the CUDA runtime is installed and the browser uses the GPU to help render websites.

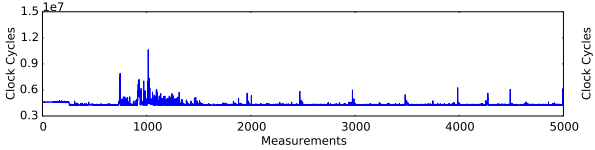
Website Fingerprinting Attack



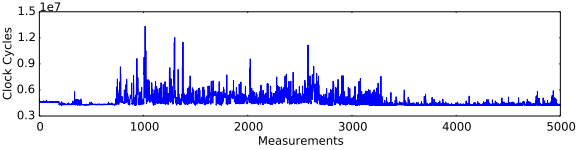
(a) Amazon.com



(b) Google.com



(c) Facebook.com



(d) Nytimes.com

Figure 5: Data transfer latency traces for visiting four websites.

Website Fingerprinting Attack

Table 4: The average and minimum precision and recall for evaluation against Google Chrome and Firefox browsers on Windows and Ubuntu Linux.

	Precision		Recall	
GPU	Mean	Min.	Mean	Min.
W-C-1080	92.8%	47.2%	91.8%	44.0%
W-C-2060	92.5%	52.8%	91.5%	44.0%
W-C-2080	95.5%	73.4%	95.2%	60.0%
W-F-1080	92.0%	56.7%	90.7%	56.0%
W-F-2060	94.0%	78.0%	93.7%	54.7%
W-F-2080	93.6%	66.9%	93.3%	60.0%
U-C-1080	91.9%	59.0%	91.0%	43.3%
U-C-2060	90.1%	46.4%	89.0%	60.7%
U-C-2080	94.2%	73.6%	93.8%	72.7%
U-F-1080	86.0%	45.9%	85.4%	42.7%
U-F-2060	89.1%	55.5%	88.5%	38.7%
U-F-2080	84.9%	50.0%	84.4%	46.0%

Table 5: The average and minimum precision and recall for evaluation against Tor browser on Windows.

	Precision		Recall	
GPU	Mean	Min.	Mean	Min.
W-T-1080	90.2%	58.5%	89.9%	52.7%
W-T-2060	90.9%	57.0%	90.6%	54.0%
W-T-2080	90.9%	45.1%	89.9%	56.0%

Website Fingerprinting Attack

Table 6: Accuracy for testing the models against traces from the same and different platforms.

		Testing cases											
		W-C-1080	W-C-2060	W-C-2080	W-F-1080	W-F-2060	W-F-2080	U-C-1080	U-C-2060	U-C-2080	U-F-1080	U-F-2060	U-F-2080
Classifier	W-C-1080	91.8%	36.2%	27.6%	4.7%	4.1%	3.5%	1.5%	2.8%	2.0%	0.6%	1.0%	1.1%
	W-C-2060	18.6%	91.5%	4.1%	4.1%	4.5%	5.4%	1.2%	4.3%	2.0%	1.1%	1.0%	1.2%
	W-C-2080	3.4%	7.0%	95.2%	2.4%	4.2%	3.7%	1.8%	1.7%	1.8%	0.8%	1.6%	1.0%
	W-F-1080	3.4%	3.1%	1.5%	90.7%	7.1%	6.7%	1.2%	2.2%	1.2%	1.4%	1.1%	1.2%
	W-F-2060	3.5%	4.1%	0.9%	8.6%	93.7%	41.5%	2.0%	1.8%	2.7%	1.4%	0.7%	1.2%
	W-F-2080	3.1%	6.6%	2.0%	12.0%	30.5%	93.3%	2.5%	2.4%	2.8%	1.5%	1.0%	1.2%
	U-C-1080	2.1%	3.2%	1.7%	3.8%	2.9%	3.4%	91.0%	16.9%	33.7%	1.2%	0.8%	1.4%
	U-C-2060	3.0%	3.6%	2.0%	2.0%	2.5%	2.2%	14.4%	89.0%	11.3%	0.9%	1.1%	1.1%
	U-C-2080	2.1%	1.7%	2.2%	2.6%	1.6%	1.8%	28.7%	7.1%	93.8%	1.6%	1.1%	1.5%
	U-F-1080	0.7%	0.9%	0.7%	1.5%	0.8%	0.8%	1.3%	0.4%	2.5%	85.4%	0.4%	0.5%
	U-F-2060	2.5%	1.6%	0.9%	1.6%	0.6%	0.2%	2.6%	3.1%	2.8%	11.4%	88.5%	50.6%
	U-F-2080	0.2%	1.1%	0.8%	1.6%	0.2%	0.2%	1.8%	2.0%	1.7%	8.2%	52.6%	84.4%

Countermeasures

- Remove the page-locked memory allocation and transfer feature from CUDA.

Countermeasures

- Remove the page-locked memory allocation and transfer feature from CUDA.
- Implement a time-division multiple access (TDMA) method that divides hardware resource usage into time-sharing slices.

Countermeasures

- Remove the page-locked memory allocation and transfer feature from CUDA.
- Implement a time-division multiple access (TDMA) method that divides hardware resource usage into time-sharing slices.
- Detection by occasionally measuring the transfer bandwidth.

Conclusion

- In this paper, we disclose a novel side-channel vulnerability on systems equipped with GPUs. Side-channels caused by contention on the PCIe bus are overlooked by manufacturers.

Conclusion

- In this paper, we disclose a novel side-channel vulnerability on systems equipped with GPUs. Side-channels caused by contention on the PCIe bus are overlooked by manufacturers.
- Motivated by the observation that heterogeneous parallel computing models on GPUs require immense amounts of data to transfer, we constructed two realistic attacks exploiting the contention on the host-GPU PCIe bus.

Conclusion

- In this paper, we disclose a novel side-channel vulnerability on systems equipped with GPUs. Side-channels caused by contention on the PCIe bus are overlooked by manufacturers.
- Motivated by the observation that heterogeneous parallel computing models on GPUs require immense amounts of data to transfer, we constructed two realistic attacks exploiting the contention on the host-GPU PCIe bus.

Thanks! Questions?

E-mail: mert.side@ttu.edu

Artifacts: https://github.com/mertside/lockeddown