Throwhammer

Rowhammer Attacks over the Network and Defenses

Andrei Tatar

Radhesh Krishnan Konoth Elias Athanasopoulos

Cristiano Giuffrida Herbert Bos Kaveh Razavi

VU VRIJE UNIVERSITEIT AMSTERDAM

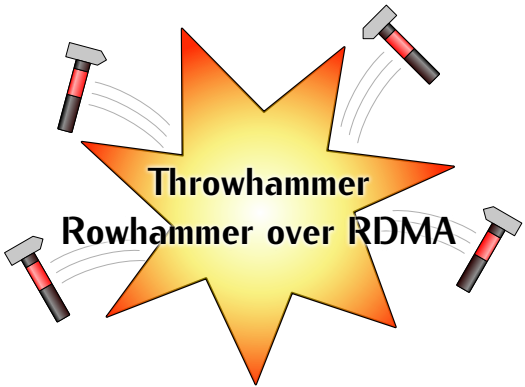
University of Cyprus

What is this about?

- Fast networks
- Smart NICs (RDMA)
- Dense DRAM

What is this about?

- Fast networks
- Smart NICs (RDMA)
- Dense DRAM



What is this about?

Throwhammer:

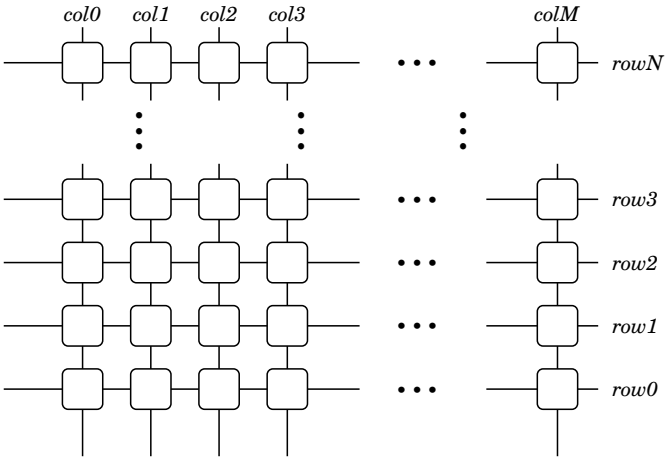
- Rowhammer attack *over the network*
- Requires *no* code execution on victim
- Vanilla network stack

What is this about?

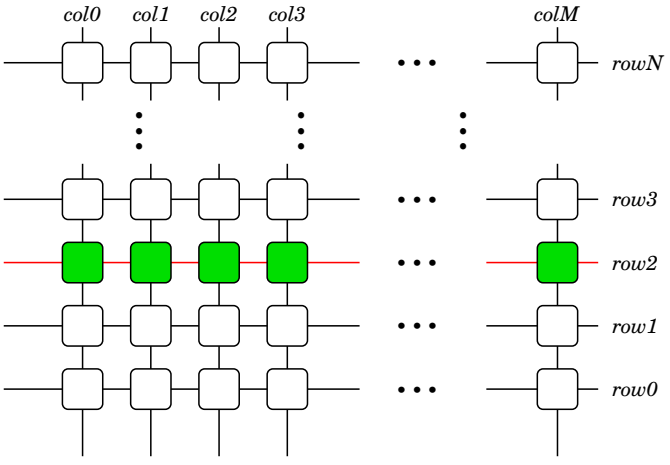
Defending against Throwhammer:

- Main idea: isolate RDMA buffers
- Challenges of isolation — not trivial
- Solution: **ALIS**

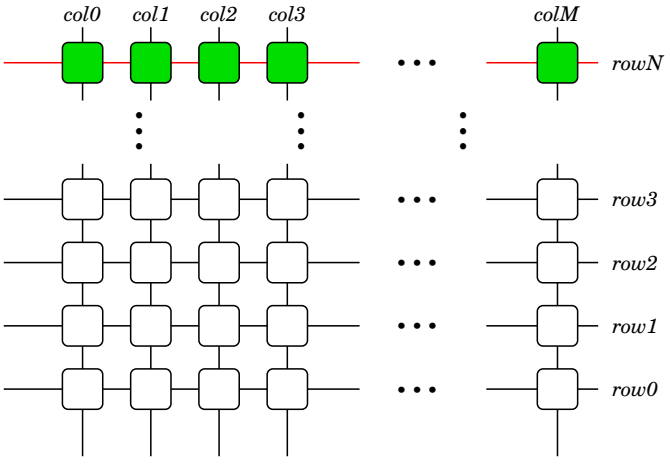
Inside DRAM



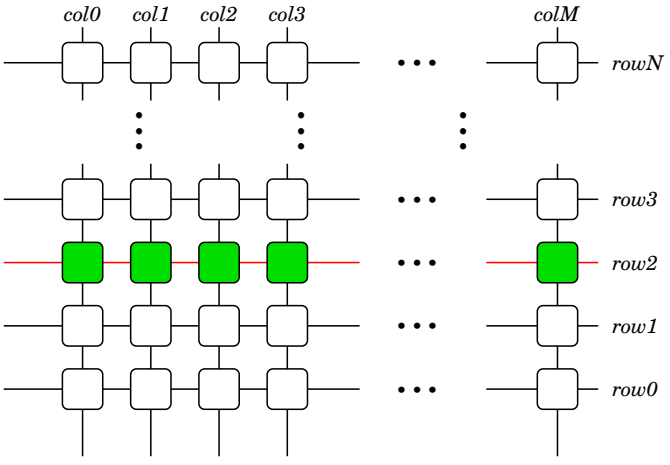
Accessing DRAM



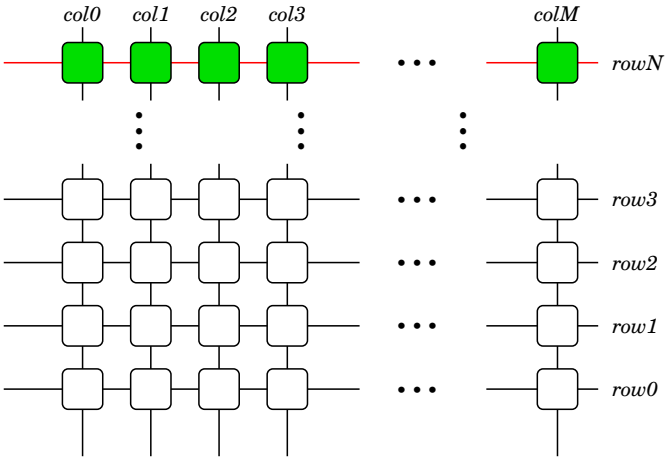
Accessing DRAM



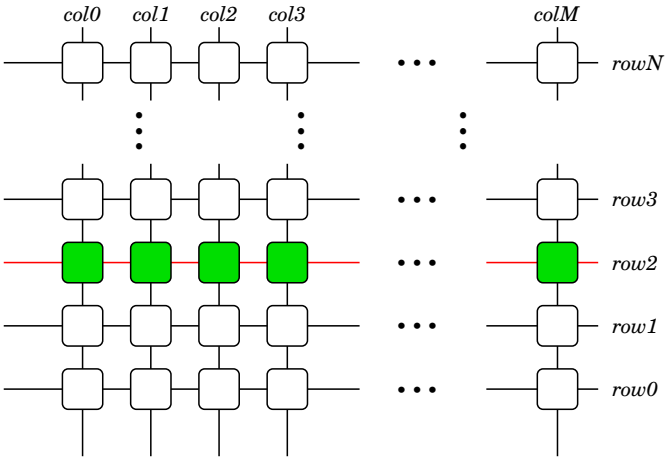
Leaking Charge: Rowhammer



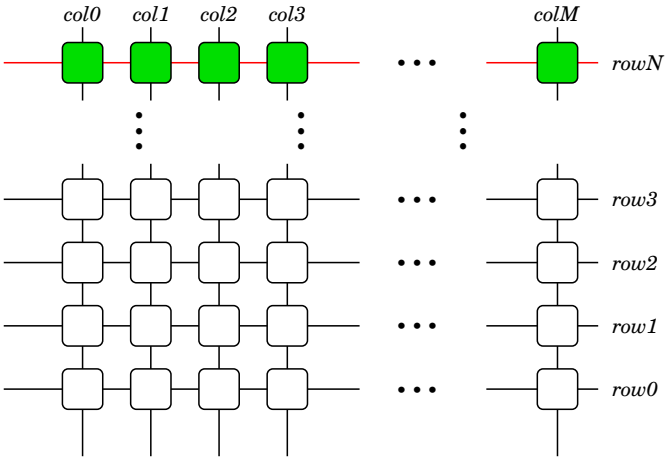
Leaking Charge: Rowhammer



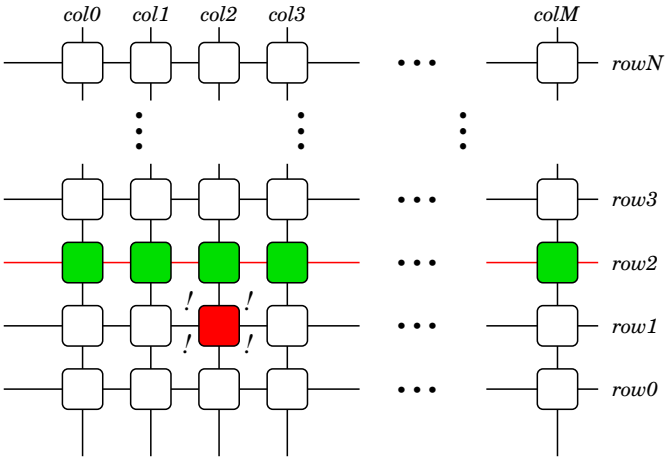
Leaking Charge: Rowhammer



Leaking Charge: Rowhammer



Leaking Charge: Rowhammer



Rowhammer: The Basics

- First study published in 2014
- Present in DDR3 DRAM and newer
- Affected devices in all price ranges and form factors
- Vulnerable locations are specific to particular chips

Rowhammer: The Basics

- First study published in 2014
- Present in DDR3 DRAM and newer
- Affected devices in all price ranges and form factors
- Vulnerable locations are specific to particular chips
- … and are reproducible ⇒ attackers can choose their flip

Triggering Rowhammer

- ≈ 500k activations / refresh interval (64 ms)
- High spatial and temporal locality

Triggering Rowhammer

- ≈ 500k activations / refresh interval (64 ms)
- High spatial and temporal locality
- … caches get in the way of accidental triggering

Triggering Rowhammer

Can be triggered on purpose:

- Cache flushing (`clflush` on x86)
- Cache eviction strategies
- Uncached memory

Exploiting Rowhammer

Has been exploited from:

- Unprivileged native code (e.g., co-hosted VMs)
- Interpreted/Sandboxed languages (e.g., Javascript)
- WebGL code running on the integrated GPU (GLitch)

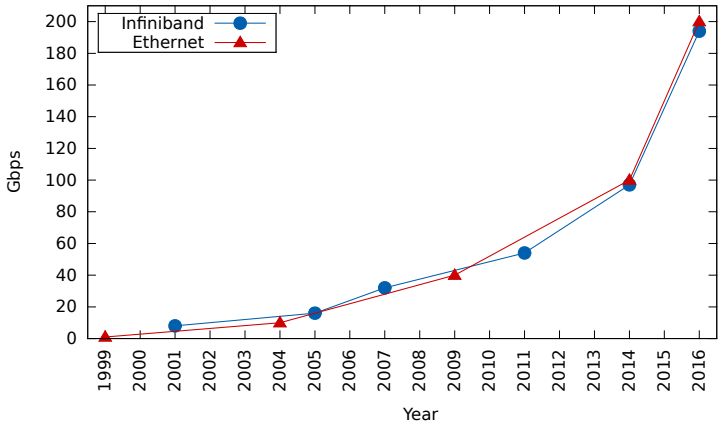
However

- All previous attacks relied on code execution on the victim machine
- … a privilege most attackers don’t have
- … and one that can be revoked if necessary

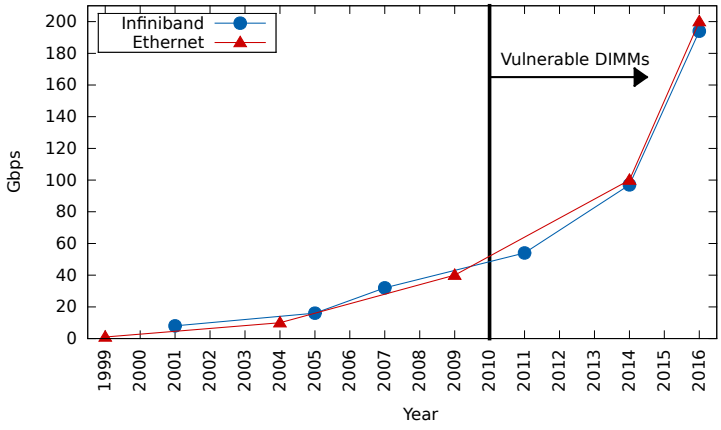
However

So are we safe?

A Step Back: Network Speeds Over Time



A Step Back: Network Speeds Over Time



A Step Back: Triggering Rowhammer

- 500k accesses / 64ms

A Step Back: Triggering Rowhammer

- 500k accesses / 64ms ≈ 8M accesses / sec

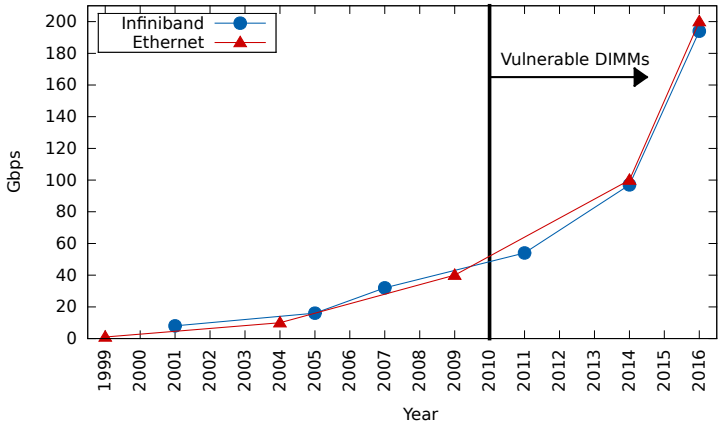
A Step Back: Triggering Rowhammer

- 500k accesses / 64ms $\approx$ 8M accesses / sec
- $=$ 8M packets / sec

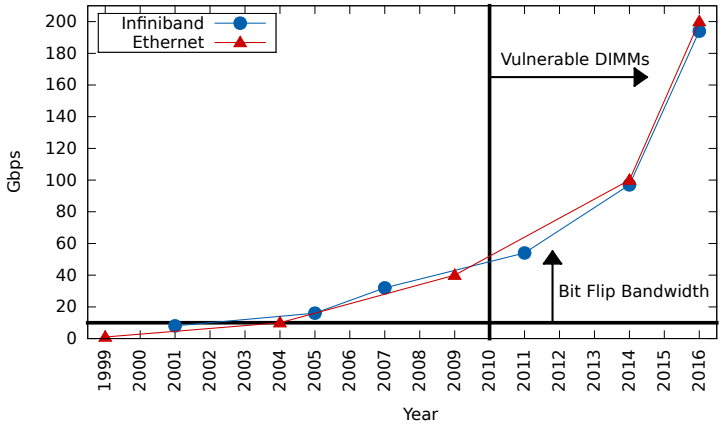
A Step Back: Triggering Rowhammer

- 500k accesses / 64ms $\approx$ 8M accesses / sec
- $=$ 8M packets / sec … feasible with 10Gbps

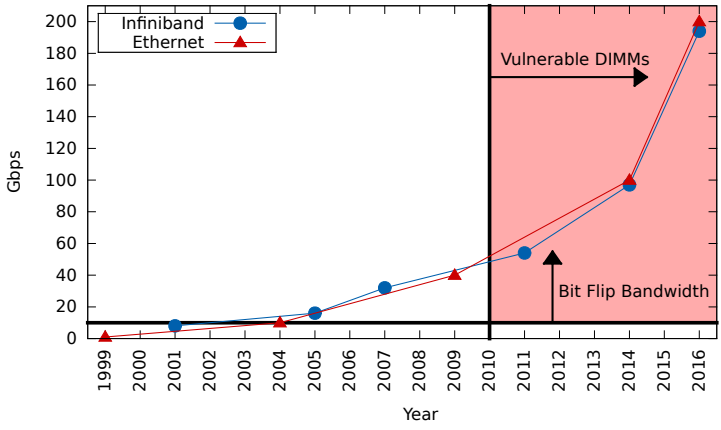
Vulnerable Networks?



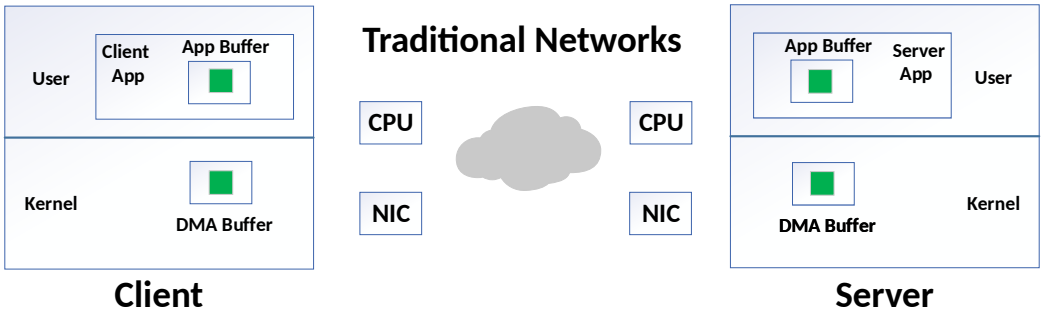
Vulnerable Networks?



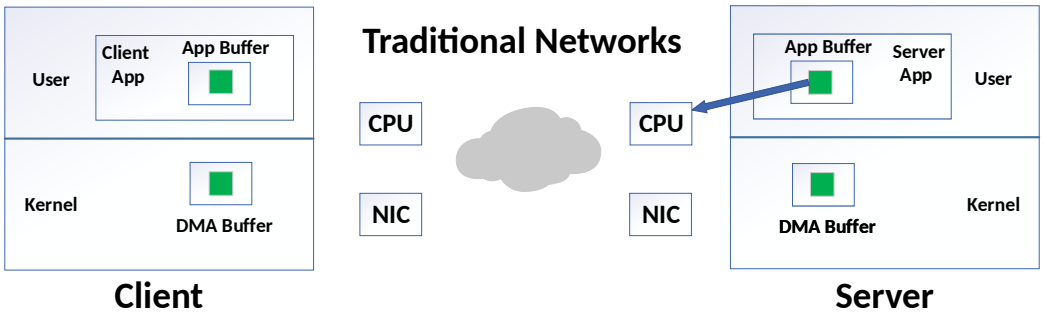
Vulnerable Networks?



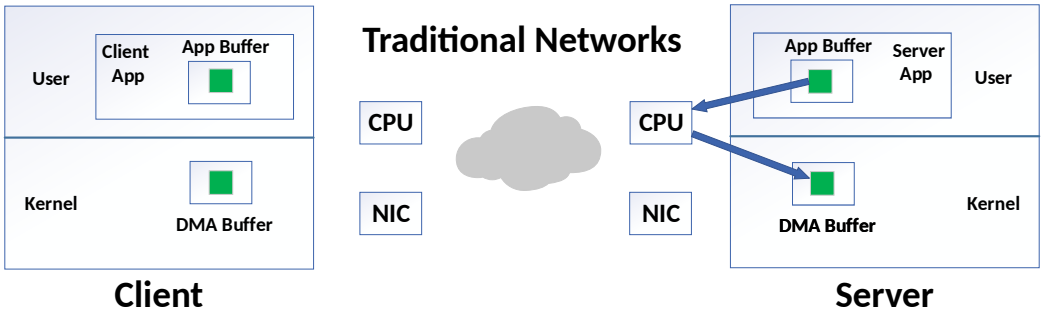
Not So Fast


User
Client App
App Buffer
Kernel
DMA Buffer
Client
Traditional Networks
CPU
NIC
CPU
NIC
App Buffer
Server App
User
Kernel
DMA Buffer
Server

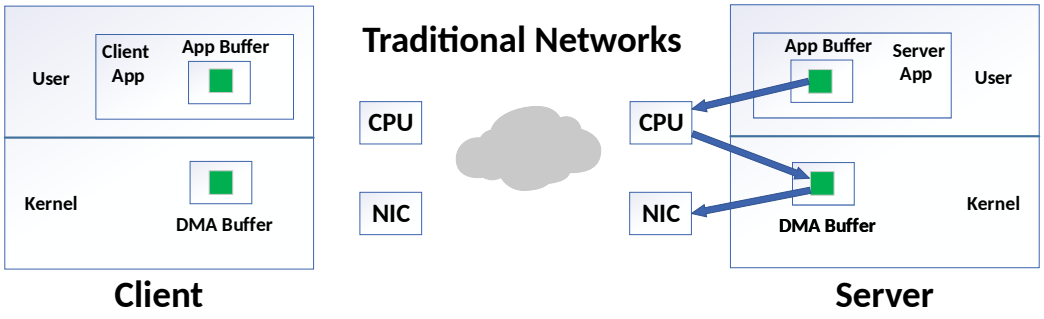
Not So Fast


Traditional Networks
User
Client App
App Buffer
Kernel
DMA Buffer
Client
CPU
NIC
CPU
NIC
App Buffer
Server App
User
Kernel
DMA Buffer
Server

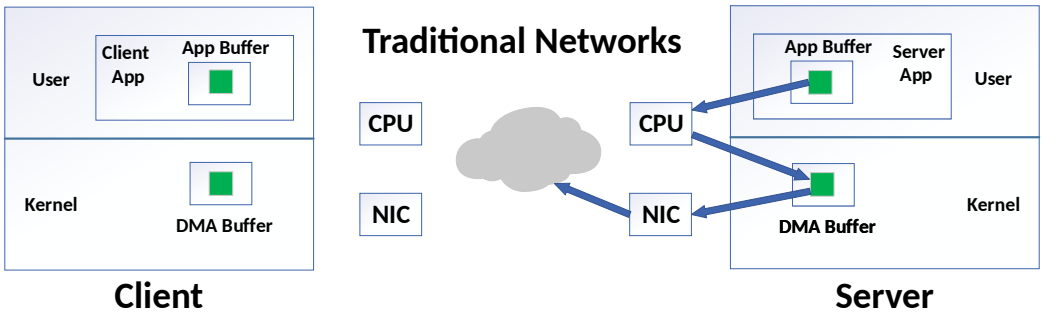
Not So Fast


Traditional Networks
User
Client App
App Buffer
Kernel
DMA Buffer
Client
CPU
NIC
CPU
NIC
App Buffer
Server App
User
Kernel
DMA Buffer
Server

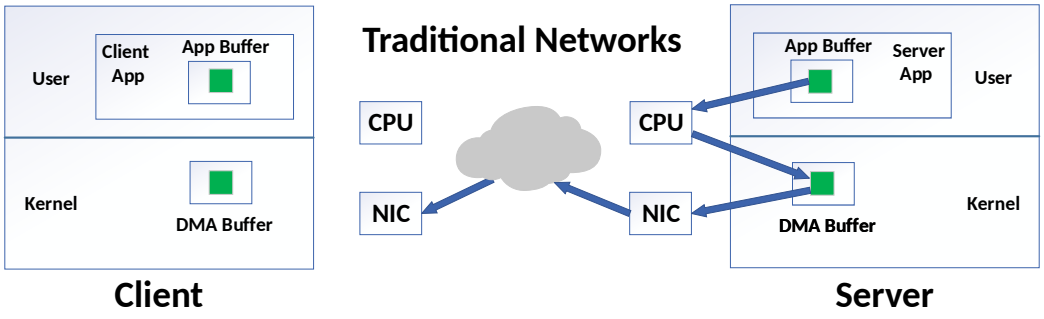
Not So Fast


User
Client App
App Buffer
Kernel
DMA Buffer
Client
Traditional Networks
CPU
NIC
CPU
NIC
App Buffer
Server App
User
Kernel
DMA Buffer
Server

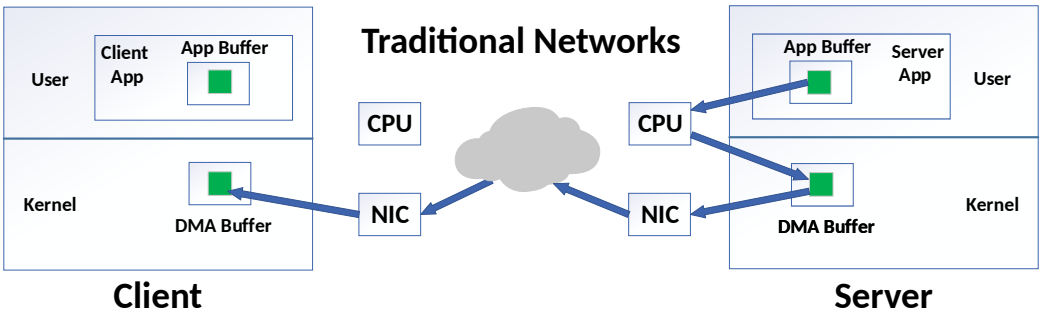
Not So Fast


Traditional Networks
User
Client App
App Buffer
Kernel
DMA Buffer
Client
CPU
NIC
CPU
NIC
App Buffer
Server App
User
Kernel
DMA Buffer
Server

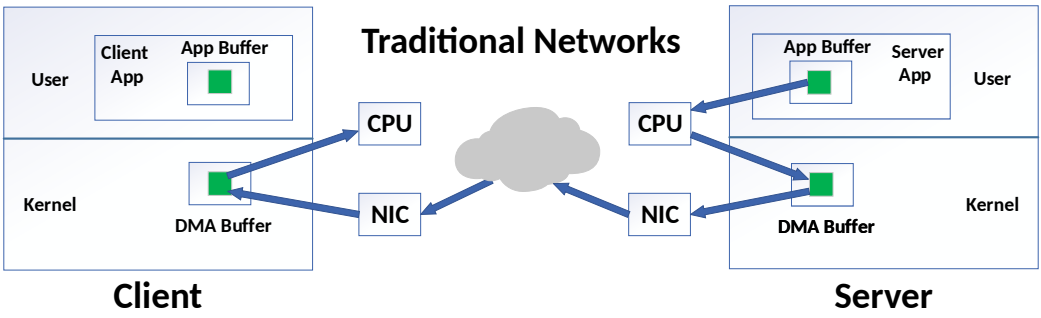
Not So Fast


Traditional Networks
User
Client App
App Buffer
Kernel
DMA Buffer
Client
CPU
NIC
CPU
NIC
App Buffer
Server App
User
Kernel
DMA Buffer
Server

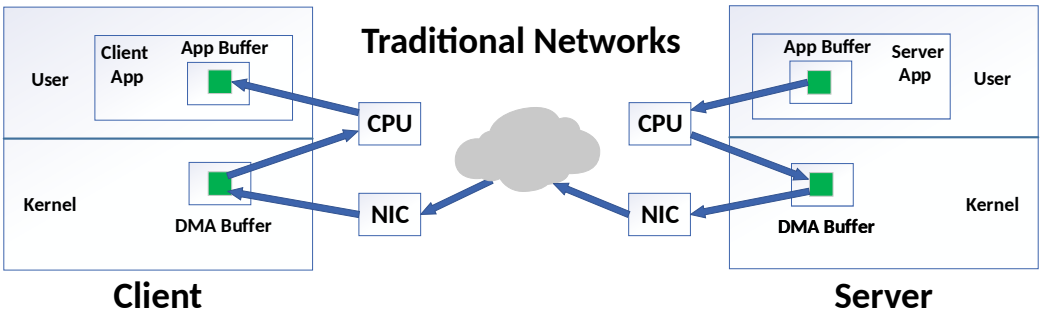
Not So Fast


Traditional Networks
User
Client App
App Buffer
Kernel
DMA Buffer
CPU
NIC
CPU
NIC
App Buffer
Server App
User
Kernel
DMA Buffer
Client
Server

Not So Fast


Traditional Networks
User
Client App
App Buffer
Kernel
DMA Buffer
CPU
NIC
Client
CPU
NIC
App Buffer
Server App
User
Kernel
DMA Buffer
Server

Not So Fast


Traditional Networks
User
Client App
App Buffer
Kernel
DMA Buffer
CPU
NIC
CPU
NIC
App Buffer
Server App
User
Kernel
DMA Buffer
Client
Server

Not So Fast

- Network I/O entails many memory accesses per packet
- … DMA from the NIC
- … memory copies within the network stack
- … userspace handling of packet data

Not So Fast

- Network I/O entails many memory accesses per packet
- … DMA from the NIC
- … memory copies within the network stack
- … userspace handling of packet data
- Reduces control of targeted memory accesses

Not So Fast

- Network I/O entails many memory accesses per packet
- … DMA from the NIC
- … memory copies within the network stack
- … userspace handling of packet data
- Reduces control of targeted memory accesses
- … and involves the cache

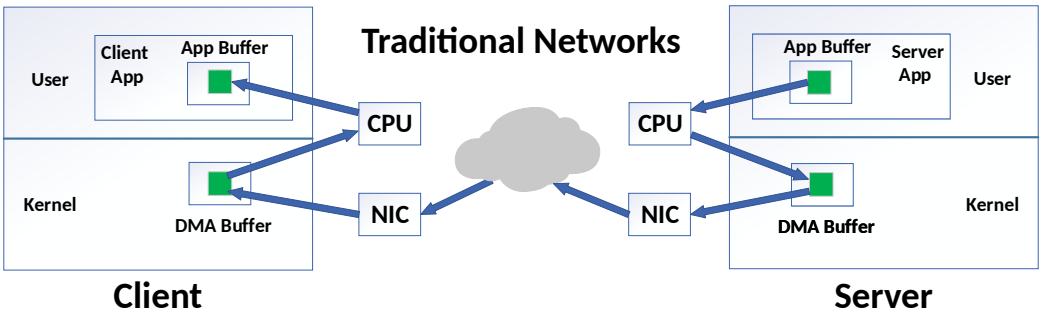
Not So Fast

Are we safe *now*?

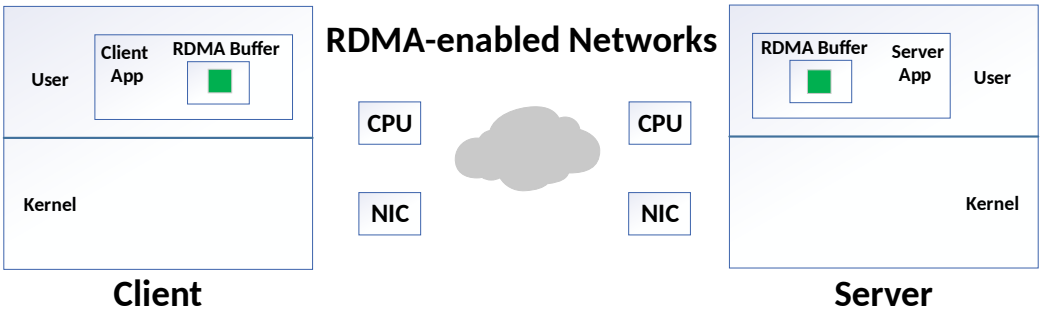
Enter RDMA

- Short for **Remote Direct Memory Access**
- High speed, low latency networking
- Supported by an real world network daemons (e.g., NFS, SMB)
- Used extensively in data centers
- Available in public clouds (e.g., Azure)

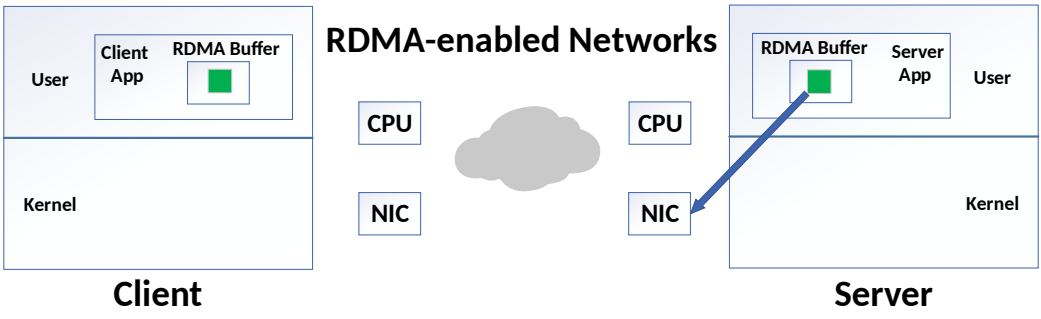
Enter RDMA


Traditional Networks
User
Client App
App Buffer
Kernel
DMA Buffer
CPU
NIC
Client
CPU
NIC
App Buffer
Server App
User
Kernel
DMA Buffer
Server

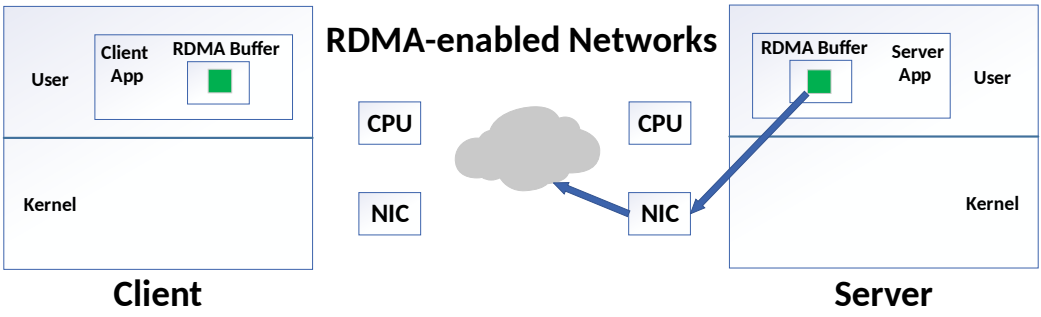
Enter RDMA


User
Client App
RDMA Buffer
Kernel
Client
RDMA-enabled Networks
CPU
NIC
CPU
NIC
RDMA Buffer
Server App
User
Kernel
Server

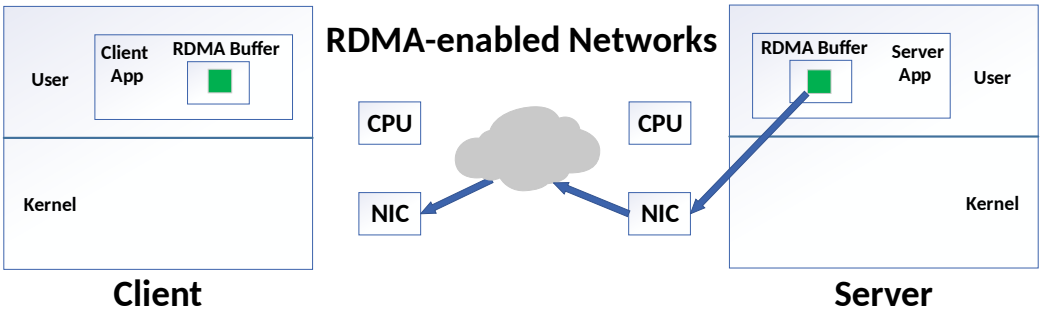
Enter RDMA


User
Client App
RDMA Buffer
Kernel
Client
RDMA-enabled Networks
CPU
NIC
CPU
NIC
RDMA Buffer
Server App
User
Kernel
Server

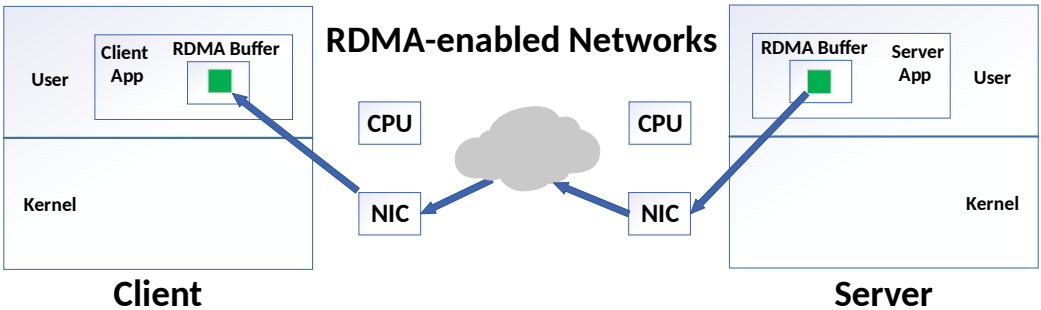
Enter RDMA


User
Client App
RDMA Buffer
Kernel
Client
RDMA-enabled Networks
CPU
NIC
CPU
NIC
RDMA Buffer
Server App
User
Kernel
Server

Enter RDMA


User
Client App
RDMA Buffer
Kernel
Client
RDMA-enabled Networks
CPU
CPU
NIC
NIC
RDMA Buffer
Server App
User
Kernel
Server

Enter RDMA


RDMA-enabled Networks
User
Client App
RDMA Buffer
Kernel
CPU
NIC
CPU
NIC
RDMA Buffer
Server App
User
Kernel
Client
Server

Putting It All Together

With RDMA we have a client requesting data from a server …

Putting It All Together

With RDMA we have a client requesting data from a server …

- … requests that each directly determine a memory location

Putting It All Together

With RDMA we have a client requesting data from a server …

- … requests that each directly determine a memory location
- … from which the NIC does *uncached* reads

Putting It All Together

With RDMA we have a client requesting data from a server …

- … requests that each directly determine a memory location
- … from which the NIC does *uncached* reads
- … at a rate of millions every second

Putting It All Together

With RDMA we have a client requesting data from a server …

- … requests that each directly determine a memory location
- … from which the NIC does *uncached* reads
- … at a rate of millions every second

Can we trigger Rowhammer?

Throwhammer: Remote Rowhammer

Yes.

Throwhammer: Remote Rowhammer

Yes.

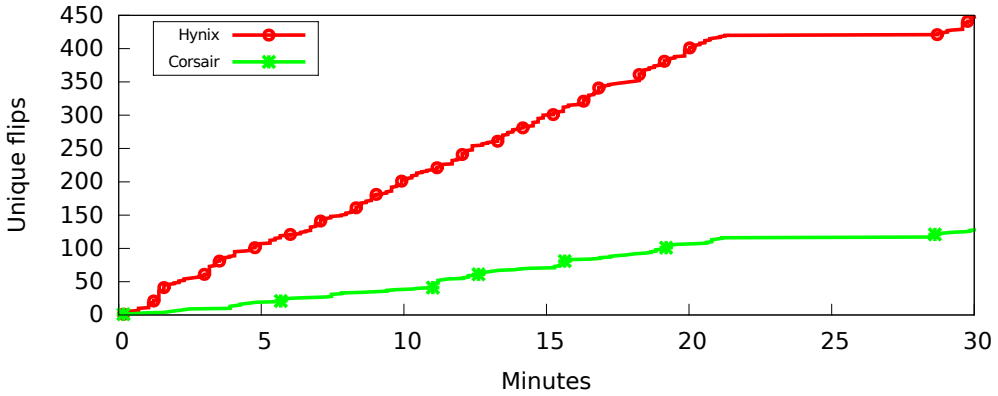
Rowhammer can be triggered remotely through RDMA networks.

Throwhammer: Remote Rowhammer

Our setup:

- 2 x Mellanox ConnectX-4 NICs @ 40Gbps
- Client repeatedly requests data from locations of its choosing

Throwhammer: Remote Rowhammer



Throwhammer: Lower Speeds

We throttled the packet rate to test lower network speeds.

- Similar bit flip success rate as for 40Gbps down to 26 Gbps
- Found bit flips with speeds as low as 10Gbps

Exploiting Remote Bit Flips

Threat model:

- Malicious client connected to victim via RDMA
- Client can repeatedly induce memory errors outside RDMA buffers
- All server software is bug-free

Exploiting Remote Bit Flips

RDMA Memcached

- RDMA-enabled fork of memcached

Memcached

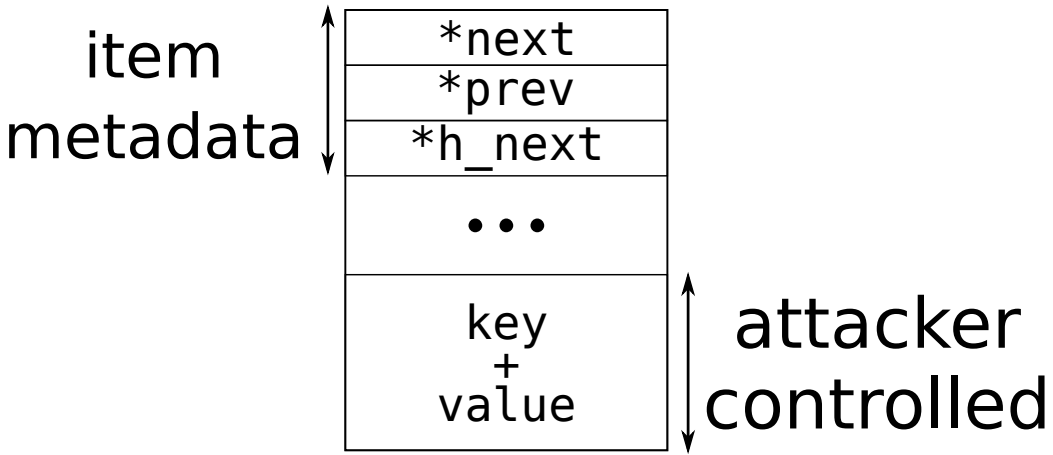
- In-memory key/value store commonly used as a backend DB cache
- Retrieves values using a hash table

Exploiting Remote Bit Flips

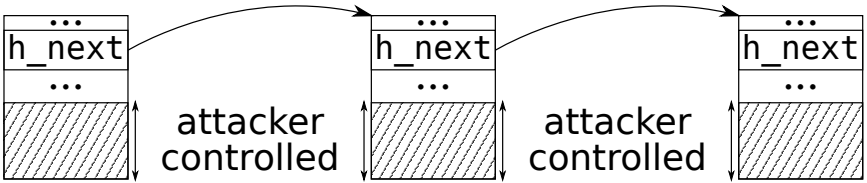
In a nutshell:

- Template memory to find which bit flips we can use
- Spray memory with hash colliding target items
- Use bit flip to add counterfeit item to collision chain
- Escalate!

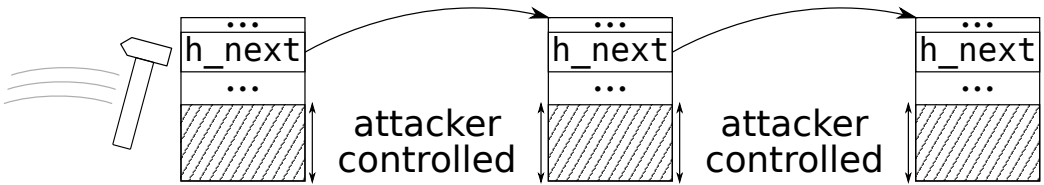
Exploiting Remote Bit Flips



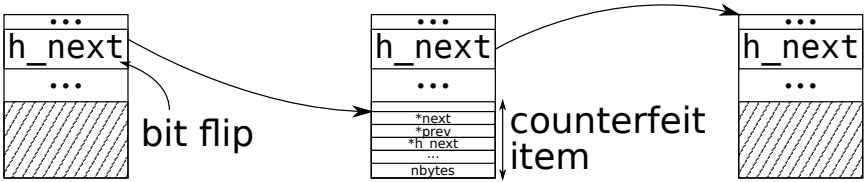
Exploiting Remote Bit Flips



Exploiting Remote Bit Flips



Exploiting Remote Bit Flips



Defending Against Throwhammer

Defending Against Throwhammer

- Don’t Use RDMA

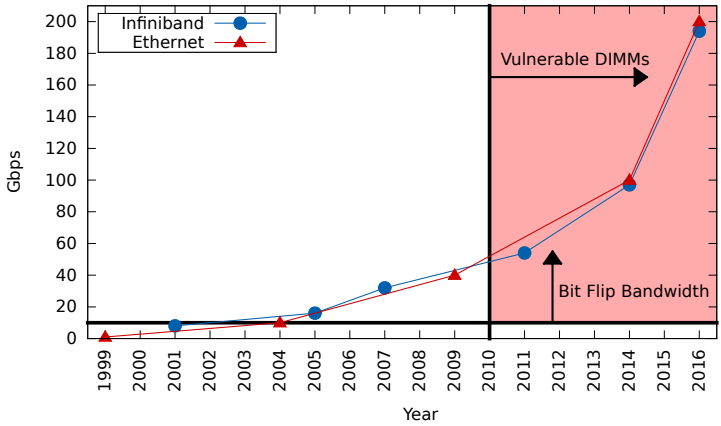
Defending Against Throwhammer

- Don’t Use RDMA
- … and pay the price of a traditional network stack

Defending Against Throwhammer

- Don’t Use RDMA
- … and pay the price of a traditional network stack
- … and even then you may not be 100% safe

Defending Against Throwhammer



Defending Against Throwhammer

- Don’t Use RDMA
- … and pay the price of a traditional network stack
- … and even then you may not be 100% safe

Defending Against Throwhammer

- Don’t Use RDMA
- … and pay the price of a traditional network stack
- … and even then you may not be 100% safe
- Isolate “untrusted” buffers

Defending Against Throwhammer

- Don’t Use RDMA
- … and pay the price of a traditional network stack
- … and even then you may not be 100% safe
- Isolate “untrusted” buffers
- … not *quite* trivial

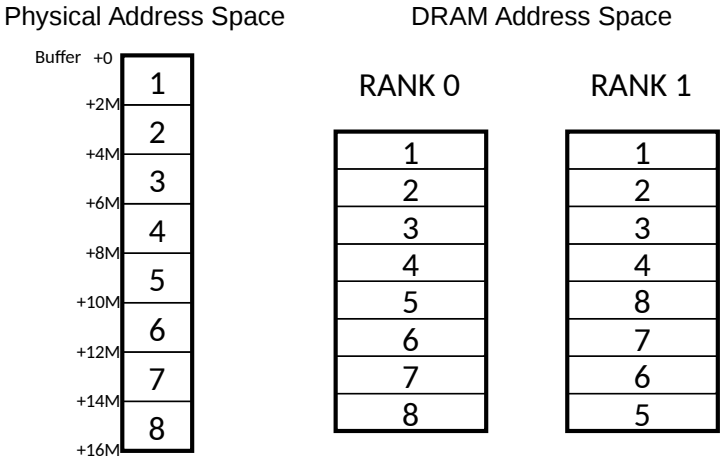
Challenges of Memory Isolation

- Isolation must be done at DRAM row level
- Software “sees” at most physical addresses
- Memory Controller has much freedom in mapping phys $\leftrightarrow$ DRAM addresses

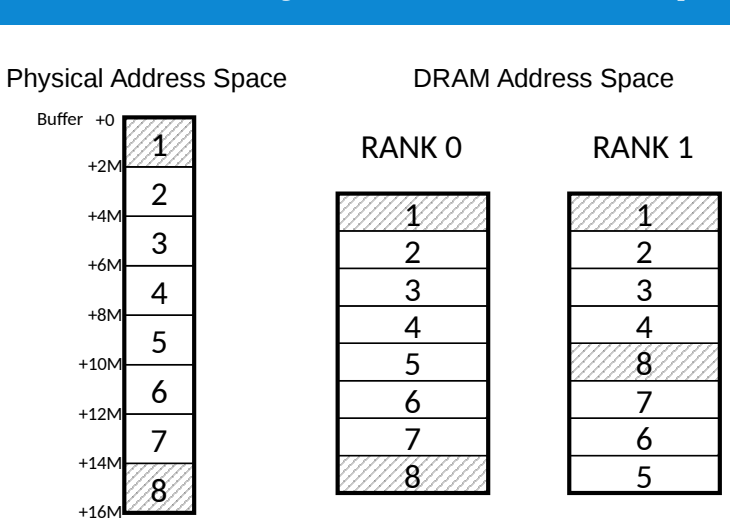
Challenges of Memory Isolation

- Isolation must be done at DRAM row level
- Software “sees” at most physical addresses
- Memory Controller has much freedom in mapping phys $\leftrightarrow$ DRAM addresses
- $\Rightarrow$ heuristics based on physical addresses simply won’t do

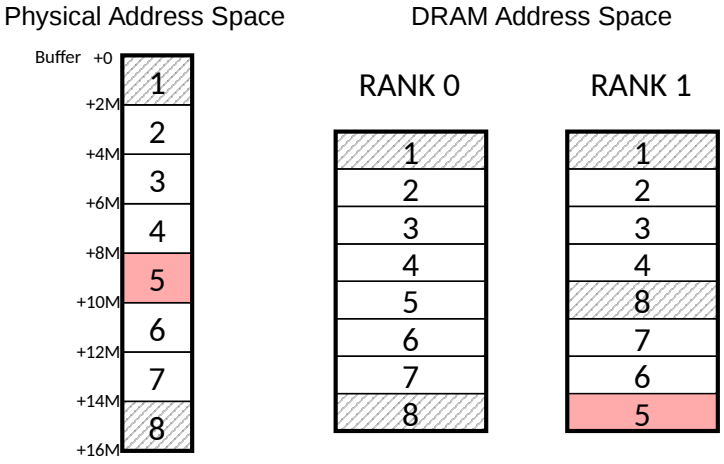
Challenges of Memory Isolation: Example



Challenges of Memory Isolation: Example



Challenges of Memory Isolation: Example



ALIS

- Generic, policy-driven isolation algorithm for arbitrary memory mappings
- Supports common memory controllers (documented or reverse-engineered)
- Easily extensible and adaptable

ALIS

Now we're safe.

Protecting Applications

We implemented ALIS as a userspace library along with drop-in changes for:

- **HERD** — in-memory key/value store designed for RDMA
- **rdma-memcached**

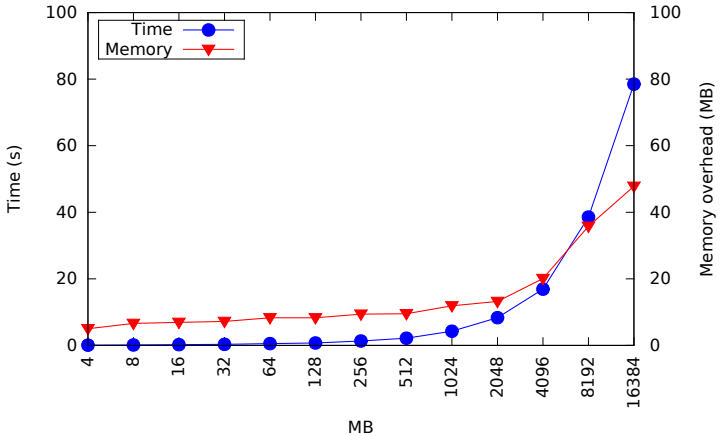
Protecting Applications

We implemented ALIS as a userspace library along with drop-in changes for:

- **HERD** — in-memory key/value store designed for RDMA
- **rdma-memcached**

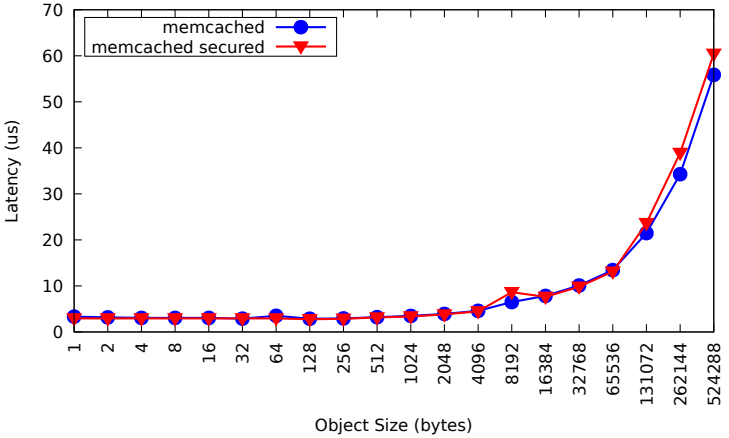
- HERD: 10 line diff
- rdma-memcached: binary patch

Protecting Applications: Allocation Cost



Four 8 GB (two ranks, dual channel)

Protecting Applications: Runtime Overhead



Conclusions

- New attack vector: Rowhammer over the network
- RDMA buffers are viable attack surfaces in real binaries
- ALIS: generic low-overhead isolation for untrusted memory areas
- Code for Throwhammer and ALIS available open-source
  `https://github.com/vusec`

