AWS Nitro

Presented by
Lavanya Karthikeyan
Agenda

• Virtualization Recap
• AWS System Before Nitro
• AWS Nitro Development (Chronologically)
• AWS Nitro Present State
• Other Infrastructure Trends
• Confidential Computing
• Summary
Virtualization Recap

• Software Virtualization
  • Binary Translation
  • Shadow Page Table
  • Port Emulation

• Hardware Support for Virtualization
  • Intel EPT, VT-x
  • SR-IOV

• Paravirtualization
  • Methods in Xen
Nitro in three parts

Nitro Cards
- VPC Networking
- Amazon Elastic Block Store (Amazon EBS)
- Instance Storage
- System Controller

Nitro Security Chip
- Integrated into motherboard
- Protects hardware resources
- Hardware Root of Trust

Nitro Hypervisor
- Lightweight Hypervisor
- Memory and CPU allocation
- Bare Metal-like performance
AWS EC2 Virtualization (Before Nitro)

Largely reliant on Software Hypervisor (Xen)
AWS EC2 Virtualization (Before Nitro)

• Largely reliant on Software Hypervisor
• Key Hypervisor components
  • Virtual Machine Monitor (VMM)
  • Device Models
  • Scheduler
  • Memory manager
  • Logging

Which of the above do you think has the most overhead?
Offloading Network Stack
C3 (early Nitro)

Hardware

Nov 2013

Software

Amazon Linux

Instance Storage

EBS Volumes

Enhanced Networking

c3.8xlarge
Nitro Cards

Network Card

- Network accelerator is connected to a standard NIC
- CPU directly talks to a network accelerator
Offloading Storage Workload
Nitro Cards

Storage Card (EBS store)

• Acquired Annapurna Labs for making Custom chips
• A chip that exposes remote stores as NVMe store to software
• Further reduction in workloads competing for CPU
Nitro Cards

Instance Store Card

• Higher performance available with NVMe and SSD.

• I3 platform offered more than 3 Million IOPS peak performance
Nitro Cards (Network Card present state)

**Nitro Card for VPC**

**ENA Controller**
- Drivers available for all major operating systems
- Independent of fabric

**VPC Data Plane**
- Encapsulation
- Security Groups
- Limiters
- Routing
Nitro Cards (EBS Card present state)

Nitro Card for EBS

NVMe Controller
Standard drivers broadly available

EBS Data Plane
Encryption support
NVM to remote storage protocol

Amazon Elastic Block Store
Nitro Cards (Instance Store Card present state)

Nitro Card for Instance Storage

NVMe Controller
Standard drivers broadly available

Instance Storage Data Plane
Transparent Encryption
Limiters
Drive monitoring
Nitro Cards

Nitro Card Controller

• Coordinates with other Nitro Cards
• Coordinates with Nitro Hypervisor
• Coordinates with Nitro Security Chip
Nitro Cards

Nitro Card Controller

System control
- Provides passive API endpoint
- Coordinates all other Nitro Cards
- Coordinates with Nitro Hypervisor
- Coordinates with Nitro Security Chip

Hardware Root of Trust
- Provides measurement and attestation
Nitro Security Chip

• Custom microcontroller that traps all I/O to non-volatile storage

• Controllable from the Nitro Controller to hold system boot

• For utility memory, if there has been any changes after an instance ends, it restores it to previous state and only then allows the next instance to start

• Provides a simple, hardware-based root of trust
Nitro Security Chip

Nitro hardware Root of Trust

Radical simplification enabled by Nitro Cards

All write access to non-volatile storage is blocked in hardware

Simple to understand security due to lack of legacy
Moving mgmt to a Nitro card

EBS Volumes

Software
Nitro Hypervisor

c5.18xlarge

Enhanced Networking
Nitro Hypervisor

• Xen to Custom KVM
• Lightweight Hypervisor
• Management Tasks moved to various Nitro Cards

Why KVM?
EC2 Bare Metal

Nov 2017

Hardware

EBS Volumes

i3.metal

Instance Storage

Enhanced Networking

© 2018, Amazon Web Services, Inc. or its Affiliates. All rights reserved.
Why Bare Metal?

• For applications that cannot or not allowed to run on virtualized systems
• Allow the user to use custom hypervisor
• Example: VMWare services migrated to AWS
Discussion Questions

1. Potential disadvantages of hardware offloading method

2. With the much lighter hypervisor, cross-VM security attacks are expected to be more difficult due to smaller stack surface. What are some attacks that can still happen in Nitro system?

3. Nitro system is expected to save costs as resources are not wasted in Dom0, but they are adding multiple custom hardware, how is it still cost efficient?
Nitro performance
## Better Performance and Price

<table>
<thead>
<tr>
<th>AWS Instance</th>
<th>vCPU</th>
<th>RAM (GiB)</th>
<th>CPU:RAM</th>
<th>Price ($/Hour)</th>
</tr>
</thead>
<tbody>
<tr>
<td>W/ Nitro c5.large</td>
<td>2</td>
<td>4</td>
<td>1:2</td>
<td>0.0085</td>
</tr>
<tr>
<td>W/o Nitro c4.large</td>
<td>2</td>
<td>3.75</td>
<td>1:1.875</td>
<td>0.1</td>
</tr>
<tr>
<td>W/ Nitro c5.4xlarge</td>
<td>16</td>
<td>32</td>
<td>1:2</td>
<td>0.68</td>
</tr>
<tr>
<td>W/o Nitro c4.4xlarge</td>
<td>16</td>
<td>30</td>
<td>1:1.875</td>
<td>0.796</td>
</tr>
<tr>
<td>W/ Nitro c5.24xlarge</td>
<td>96</td>
<td>192</td>
<td>1:2</td>
<td>4.08</td>
</tr>
<tr>
<td>W/o Nitro c4.8xlarge</td>
<td>36</td>
<td>60</td>
<td>1:1.875</td>
<td>1.591</td>
</tr>
</tbody>
</table>

Source: https://aws.amazon.com/ec2/pricing/on-demand/
Summary

**Nitro System goals**

- Reduce resource usage by hypervisor’s Dom0 and other management tasks
  - Network Card
  - EBS Storage card
  - Instance Store Card
  - Controller Card
  - Lightweight hypervisor

- Improve security
  - Hardware root-of-trust with Nitro Security chip
  - With nitro cards, goal is to make all communication encrypted
  - Only the Nitro controller has access to the physical Amazon EC2 network

- With separate hardware, more modularity hence easy to innovate
  - Bare-Metal instances

- Save costs
Other Infrastructure Trends
Data Center of the Future

General Purpose Compute Services:
- General Purpose Compute
- General Purpose Compute

Infrastructure Processing Unit

Computational Storage Services:
- Infrastructure Processing Unit
- Shared Storage

ML/AI Services:
- General Purpose Compute
- XPUs
- XPUs

Infrastructure Processing Unit

Acceleration Services:
- XPUs
- XPUs
- XPUs

Data Center Intelligent Network Fabric

intel
The Infrastructure Processing Unit
Providing new data center value

- Highly intelligent **infrastructure acceleration**
- System-level **security, control, and isolation**
- **Common** software frameworks
- HW and SW **programmable**, built to customer needs
Start-ups in this domain (Fungible Inc)

NIC vs Smart NIC vs Programmable NIC

1. NIC: Standard Network Interface, data-centric compute done on CPU outside NIC
2. Smart NIC: Consists of cores and accelerators on it. Relatively hardwired, useful for accelerating hardwired paths (Still cheaper than NIC+CPU as cheaper Arm cores are used)
3. FPGA: Soften the hardwired portions. But less programmable than CPU, worse performance than fully hard-wired solution
4. Programmable NIC (DPU etc): Develop protocol along with chip so that the custom chip (with cores and accelerators) is more programmable. OS capability.
NIC vs Smart NIC vs Programmable NIC

- Capability and complexity is not as linear as in the above picture.
- Intel IPUs (Xeon + FPGA)
- Nvidia Bluefield (ARM cores + GPU capability + Mellanox interconnect)
- Fungible DPU (cores + Accelerators)
- Cloud providers
  - Microsoft (FPGA solution)
  - Amazon (Custom chip)
Confidential Computing
Confidential Computing

• High-level idea is to untrust the service provider, although different providers have different definitions. Two commands issues are:
  • Mistrust cloud provider
  • Mistrust third-party service provider running on cloud (This is an issue outside of cloud as well)

• One approach: establish a secure “enclave” that runs user application and plain-text user data.
  • Use a hardware root of trust which “attests” the software running on it
  • Encrypts data whenever it leaves the hardware (e.g., leaves the CPU chip)
  • Checks the integrity of data
  • Example: Intel SGX, AMD SEV (Cloud offering: Azure Confidential Computing)

• Another approach: partition all hardware into a safe world and an unsafe world (e.g., ARM Trustzone)
Intel SGX

- At runtime, Intel SGX instructions build and execute the enclave into a special encrypted memory region with restricted entry/exit location defined by the developer.
- Enclave code and data inside the CPU perimeter runs in the clear, and enclave data written to memory is encrypted and its integrity checked.
- The encryption key is also stored within enclave.

**Vulnerable to speculative execution attacks**

1. Attack Software mis-trains the branch predictor so that the branch which checks the condition for enclave entry is mis-predicted.
2. Attack software can now speculatively execute enclave code.
3. Gain access to speculatively executed secret through Prime+Probe and other cache side channels.
4. Steal keys from enclaves and decrypt enclave data outside enclave.

Ref: SgxPectre Attacks: Stealing Intel Secrets from SGX Enclaves via Speculative Execution
Arm TrustZone

- SoC wide implementation
- Arm Cortex processors with TrustZone run a secure operating system (OS) and a normal OS simultaneously from a single core.
- Execution happens in a time sliced manner (Secure <-> Normal) with non-secure software blocked from accessing secure resources directly.
- TrustZone checked at uncore components such as interrupt controllers, memory to isolate resources.

71 Reported CVE
Nitro Enclaves

- AWS Nitro Enclaves has its own kernel that is separated from the parent instance’s kernel. The kernel of your parent instance has no access to the enclave.
- AWS Nitro Enclaves does not accept inbound connections.
- Applications used for processing sensitive data are embedded into the enclave.
- Cryptographic attestation to protect from accidental data manipulation.
- For protecting user application and data from cloud provider.
- Speculative attacks difficult due to full CPU isolation. Still shared L3 cache leaves it vulnerable to side channels.
Nitro TPM

- TPM (Trusted Platform Module) can be a cryptographic coprocessor used for encryption and generating keys or an equivalent firmware
- Nitro TPM is a firmware run on Nitro Cards
- Network cards can generate, store and use secret keys which the EC2 host directly accessing the key
Azure Enclaves

Built using Intel SGX
Summary

Infrastructure Trends
• Smart NICs evolved into programmable NICs with a full OS support

Confidential computing
• Cause
  • Mistrust cloud provider
  • Mistrust third party service provider
• Solutions
  • Intel SGX
  • ARM TrustZone: For SoCs
  • Nitro Enclave: Hardened isolated VM which can only be entered by Vsock
  • Azure enclave (implemented with Intel SGX)
Thank you