**HOMEWORK - 1**

2)

All modern operating systems including idle virtual machines depend on clock interrupts to maintain their internal timers. Using paravirtualization, the virtual machine code is changed to request a notification at the specified time. This precludes the need for an hypervisor to schedule timer interrupts which is not scalable. Hence we observe scalability advantages when using the paravirtualization to handle timers.

*Source: Textbook*

3)

Using modular design, an operating system deals with multiple CPUs in the same way it deals with one. Here only the scheduler and the interrupt handlers need to be fully aware of the differences. For eg., x86 based multiprocessor architectures use the Advanced Programmable Interrupt Controller (APIC) for interrupt redirection in support of Symmetric Multiprocessing (SMP). However, accessing APIC in virtual mode is costly because of the transitions into and out of the hypervisor. Hence we prefer paravirtualization, using which several APIC request can be replaced with a single hyper call.

*Source: Textbook*

4)

Yes, Linux also has stacks for both the user-mode and kernel-mode for each process it runs. The kernel stack and the user stack are implemented using the stack data structure and, taken together, serve as a call stack. In the user space, the user stack grows downward to lower addresses. The kernel stack is part of the kernel space. Hence, it is not directly accessible from a user process. Whenever a user process uses a syscall, the CPU mode switches to kernel mode. During the syscall, the kernel stack of the running process is used.

The CPU reserves one register set for the user mode) for application processes) and the other for the kernel(system) mode (for O.S processes). Whenever switch between the modes take place, interrupts are sent to the CPU, which stops the current execution and switched to the system mode to execute the interrupt handler routine which activates the system stack pointer. The handler saves the state of CPU and restores the state returning to the user mode.

*Sources: Kernel Stack and User space Stack(* [*https://www.baeldung.com/linux/kernel-stack-and-user-space-stack*](https://www.baeldung.com/linux/kernel-stack-and-user-space-stack) *), Textbook*

5)

“Unscrambled” in Intel Load Segment Limit (LSL) instruction means the limit scaled as per the setting of the G flag in the segment descriptor. Basically, it's due to the fact that the limit field is spread across several bits within the GDT entry. The unscrambled limit is set when the privilege levels and type checks and loaded in the destination register and set the ZF flag in the EFLAGS register. If the segment selector is not visible at the current privilege level or is an invalid type for the LSL instruction, the instruction does not modify the destination register and clears the ZF flag. Depending on the G flag, the limit is interpreted in a different way. When the G flag is clear (byte granularity), the effective limit is the value of the 20-bit limit field in the segment descriptor. Here, the limit ranges from 0 to FFFFFH (1 MByte). When the G flag is set (4-KByte page granularity), the processor scales the value in the limit field by a factor of 212 (4 KBytes). In this case, the effective limit ranges from FFFH (4 KBytes) to FFFFFFFFH (4 GBytes).

*Sources: Intel Manual(* [*Intel® 64 and IA-32 Architectures Software Developer’s Manual Volume 2 (2A, 2B, 2C & 2D): Instruction Set Reference, A-Z*](https://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-instruction-set-reference-manual-325383.pdf) *), Quora (*[*https://stackoverflow.com/questions/22005051/what-does-unscrambled-mean-in-this-context*](https://stackoverflow.com/questions/22005051/what-does-unscrambled-mean-in-this-context) *), GDT entry(* [*https://wiki.osdev.org/File:GDT\_Entry.png*](https://wiki.osdev.org/File:GDT_Entry.png) *)*

6)

i) Advantages of I/O MMU:

a) The need to have contiguous memory locations in physical memory are overcome by I/O MMU which maps continuous virtual addresses to disjointed physical address

b) I/O virtual memory address is translated to corresponding physical memory by using I/O MMU. This makes direct access by devices safer and more efficient and allows the driver in the VM to program device DMA using its virtualized notion of memory address, while allowing the hypervisor to decide where VM memory is actually located.

c) It ensures the division of labour that enable time and space multiplexing of I/O devices. This in turn leads multiple devices being implemented by small number of physical devices.

d) Allows devices that cannot access large memory addresses to address the entire memory using IOMMU. This avoids the need for buffer and higher overhead.

e) Flexible mappings between physical and logical devices is provided by decoupling. This enables improved portability.

f) Allows logical I/O devices to be multiplexed upon physical ones. This gives administrators higher utilization and efficiency to drive I/O devices.

g) Common VM functions like ability to suspend and resume a VM, and live migration of VM is facilitated.

h) I/O virtualization provides the groundwork for a lot of innovative improvements of the logical I/O devices.

i) There is a possibility to add new features to existing systems by interposing and transforming virtual I/O requests, transparently enhancing unmodified software with new capabilities. For eg., a disk write can be transformed into replicated writes to multiple disks.

j) Multiple physical devices can be aggregated into single logical device that is exported to the VM.

Disadvantages of I/O MMU:-

1. Consumption of physical memory for the added I/O page (translation) tables. This can be mitigated if the tables can be shared with the processor.
2. Drop in performance of translation and management overhead.

*Sources: I/O Virtualization Communication of the ACM paper(* [*https://dl.acm.org/doi/pdf/10.1145/2063176.2063194*](https://dl.acm.org/doi/pdf/10.1145/2063176.2063194) *)*

ii) Carrier Grade is the virtualization services that satisfy some or all expected properties existing in carrier grade solution. Carrier Grade Virtualization reduces the cost and complexity of maintaining carrier grad properties in edge and core network elements such as IP Multimedia Systems (IMS) nodes. Also networking and telecommunication OEMs can reuse existing investments in their carrier grade system while gaining the benefits of using real-time virtualization software.

Some of its features are:

* High performance scaling
* Availability
* Upgrade capabilities
* Real-time responsivity
* Better upgrade capabilities
* Configurable security
* Efficient and Uniform management services

Products available are:

• Bare-metal Xen Hypervisor

• Oracle Solaris

• Virtual Logix Carrier Grade Hypervisors

*Source: Carrier Grade Virtualization*([*http://www.linuxpundit.com/documents/CGV\_WP\_Final\_FN.pdf*](http://www.linuxpundit.com/documents/CGV_WP_Final_FN.pdf)*)*

7)

Amazon EC2 uses bare-metal hypervisors in Xen. Its major characteristics are:

i) Live VM Migration: It supports virtual machine live migration from one host to another allows workload balancing and the avoidance of downtime.

ii) Live Storage Migration: Move live running virtual machines and their associated virtual disk image within and across resource pools leveraging local and shared storage.

iii) Host Failure Protection: Deliver high availability by automatically restarting virtual machines if a failure occurs at the VM, hypervisor or server level. Link aggregation bonds network interfaces for network redundancy and increased throughput.

iv) Host Power Protection: Take advantage of embedded hardware features to lower datacenter electricity consumption by dynamically consolidating VMs on fewer systems and then powering off underutilized servers as demand for services fluctuates.

v) Memory Overcommit: Reduce costs and improve application performance and protection by sharing unused server memory between VMs on the host server.

8)

i) The CPU power of a virtual machine is measured in EC2 Compute Units by the EC2. ECU is said to represent a given number of computation cycles in a fashion that is unrelated to the real hardware. So, several benchmarks and tests are used to determine how the Computing Units translate into power of the physical processor. According to the documentations, a single ECU is defined as the compute power of a 1.0 to 1.2 GHz of a 2007 server CPU capacity.

ii) General Purpose:

This instance family includes T2, M3, and M4 which is often the first choice because of variety of CPU size range.

1) T2 instances are Burstable Performance Instances that provides a baseline level of CPU performance with the ability to burst above the baseline, which is governed by CPU Credits. These instances are good choices for workloads that don’t use the full CPU often or consistently, but occasionally need to burst Features:

a) High frequency Intel Xeon processors

b) Burstable CPU, governed by CPU credits, and consistent baseline performance

2) M3 instances provide balance of compute, memory and network

resources, and it is a good choice for many applications.

Features:

i) SSD-based instance storage for fast I/O performance

ii) Balance of computing, memory, and network resources

3) M4 instances are the latest generation of General-Purpose

Instances. This provides a balance of computing, memory and

network resources, and it is a good choice for many applications

Features:

i) EBS-optimized by default at no additional cost

ii) Support for Enhanced Networking

• Memory Optimized: This instances family includes X1, R3

and R4 instance types and is designed for memory intensive

applications. Instances have the lowest cost per GiB of RAM

of all other instance types.

1) X1 instances are optimized for large-scale, enterprise-class, inmemory applications and have the lowest price per GiB of RAM of

all other instance types.

Features:

i) Lowest price per GiB of RAM

ii) Up to 1,952 GiB of DDR4 based instance memory

2) R3 instances are optimized for memory intensive applications and

offer lower price per GiB of RAM.

Features:

i) SSD Storage

ii) Support Enhanced Networking

3) R4 instances are optimized for memory intensive applications and

offer lower price per GiB of RAM than R3.

Features:

i) 2.3 GHz Intel Xeon E5-2686 v4 (Broadwell) processors

ii) DDR4 Memory

• Accelerated Computing/GPU: This instances family includes

P2, G2 and F1 instance types and allows to take advantage of

the parallel performance of NVIDIA Tesla GPU using

CUDA or OpenCL programming models for GPGPU.

1) P2 instances are intended for general purpose GPU compute

applications

Features:

i) High frequency Intel Xeon E7-2686 v4 (Broawell) processor.

ii) Support, GPU Direct (peer-to-peer GPU Communication)

2) G2 instances are optimized for graphics intensive applications

Features:

i) High frequency Intel Xeon E5-2670 v2 (Sandy Bridge)

processors

ii) High-performance NVIDIA GPUs, each with 1,536 CUDA

cores and 4GB of video memory

3) F1 instances offer customizable hardware acceleration with field

programmable arrays (FPGAs)

Features:

i) High Frequency Intel Xeon E5-2686 v4 (Broadwell)

Processors with a base frequency of 2.3 GHz.

ii) Support for Amazon EC2 Enhanced Networking

Sources: Choosing the Right EC2 Instance Type for Your Application( <https://aws.amazon.com/blogs/aws/choosing-the-right-ec2-instance-type-for-your-application/> ),

iii) Amazon EC2 currently supports a variety of operating systems including: Amazon Linux, CentOS, CoreOS, Debian, Fedora, FreeBSD, Gentoo, TurnKey Core, Windows Server, and Ubuntu Server 14.04 LTS (HVM)

*Sources: AWS marketplace(* [*https://aws.amazon.com/marketplace/b/c3bc6a75-0c3a-46ce-8fdd-498b6fd88577?category=c3bc6a75-0c3a-46ce-8fdd-498b6fd88577*](https://aws.amazon.com/marketplace/b/c3bc6a75-0c3a-46ce-8fdd-498b6fd88577?category=c3bc6a75-0c3a-46ce-8fdd-498b6fd88577) *), EC2 Instance types(* [*https://aws.amazon.com/ec2/instance-types/*](https://aws.amazon.com/ec2/instance-types/) *), Previous generation instances(* [*https://aws.amazon.com/ec2/previous-generation/*](https://aws.amazon.com/ec2/previous-generation/) *)*

iv) An Amazon Machine Image (AMI) is a special type of virtual appliance that is used to create a virtual machine within the Amazon Elastic Compute Cloud ("EC2"). It serves as the basic unit of deployment for services delivered using EC2. And it is a type of instance provided on AWS.

*Source: AWS documentation*

v) The AMI components are:

Storage for the Root Device

Determining the Root Device Type of Your AMI

Stopped State

Default Data Storage and Persistence

Boot Times

AMI Creation

9)

Amazon EC2 may be tried free of charge. There is no benchmark of charges and the user will pay as per their usage of the services. The monthly bill is calculated using monthly calculator which is used for billing purposes.

The four ways to pay for Amazon EC2 instances are:

1. *Spot Instances*: These Instances let you take advantage of unused EC2 capacity in the AWS cloud. Spot Instances are available at up to a 90% discount compared to On-Demand prices. They are useful in cases of urgent or sudden need.
2. *On-Demand* Instance: For short-term workloads this is an ideal choice as the user has to pay just for the usage with no long-term commitments.
3. *Reserved* Instances: reserved instances provide you with a significant discount (up to 72%) compared to On-Demand instance pricing. They are available 24\*7 and can be made to launch in the time of requirement.
4. *Dedicated Hosts*: These hosts are helpful to meet compliance requirements and lower costs by permitting existing server-bound software licenses. Dedicated Hosts either can be purchased On-Demand on an hourly basis or can be purchased as a reservation for up to 70% off the On-Demand pricing.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Instance name** | **On-Demand hourly rate** | **vCPU** | **Memory** | **Storage** | **Network performance** |
| a1.medium | $0.03 | 2 | 2 GiB | EBS Only | Up to 10 Gigabit |
| a1.large | $0.05 | 4 | 4 GiB | EBS Only | Up to 10 Gigabit |
| a1.xlarge | $0.10 | 8 | 8 GiB | EBS Only | Up to 10 Gigabit |
| a1.2xlarge | $0.20 | 16 | 16 GiB | EBS Only | Up to 10 Gigabit |
| a1.4xlarge | $0.41 | 72 | 32 GiB | EBS Only | Up to 10 Gigabit |
|  |  |  |  |  |  |

*Sources: EC2 pricing (* [*https://aws.amazon.com/ec2/pricing/*](https://aws.amazon.com/ec2/pricing/) *)*

10)

a)

Service Level Agreement (SLA) is a contract between a cloud provider (either internal or external) and the service user that outlines responsibilities, quality, and scope on both the sides. The most common component of SLA is that the services should be provided to the customer as agreed upon in the contract.

For free services on Amazon EC2, one needs to sign up under the Free Tier, to get services for a 12 month period. After that one need to create an account and use the services provided under certain usage limits. The following steps are needed:

i) Sign up for an AWS account,

ii) provide credit card information and billing address. Until the free usage exceeds the limits, you can use free of charge.

iii. Get started with AWS Cloud services by choosing any of the products listed under the Free Tier service.

*Sources: What is service level agreement (* [*https://www.paloaltonetworks.com/documentation/glossary/what-is-a-service-level-agreement-sla*](https://www.paloaltonetworks.com/documentation/glossary/what-is-a-service-level-agreement-sla) *), cloud SLA (* [*http://searchcloudcomputing.techtarget.com/essentialguide/Breaking-down-whats-in-your-cloud-SLA*](http://searchcloudcomputing.techtarget.com/essentialguide/Breaking-down-whats-in-your-cloud-SLA) *), Service Level Agreement(* [*https://en.wikipedia.org/wiki/Service-level-agreement*](https://en.wikipedia.org/wiki/Service-level-agreement) *), EC2 pricing (* [*https://aws.amazon.com/ec2/sla/*](https://aws.amazon.com/ec2/sla/) *)*

b)

The process to follow is describe below in the following steps:

i) create an instance of Amazon EC2

ii) create a server for the database which would be a database instance.

iii) After performing above steps, a web app can be deployed on the server.

iv) After that, load balancing and scaling needs to be done so that the traffic is distributed across the

number of servers or application servers.

v) user can associate or use a name with your web application.

*(Sources: Amazon Machine Image(* [*http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html*](http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html) *)*

c)

Yes, we can create a machine instance equivalent to our own PC and transfer our own PC image there. This can be done by creating an EC2 instance on the Amazon Cloud and host it as a server. After that, we need to connect our own PC to that server and then transfer the image.

*(Sources: AWS Knowledge center(* [*https://aws.amazon.com/premiumsupport/knowledge-center/*](https://aws.amazon.com/premiumsupport/knowledge-center/) *)*