Introduction to Cloud Computing (CS 524)

(Homework 2)

Prof. Igor Faynberg

Student Name: **Paras Garg**

Course Section: **CS 524-A**

Homework 1.1 –
Complete reading Chapter 3 of the textbook and the lecture materials. **Please note the errata: The references to [19] on p. 56 of the book should be replaced with references to [20]**! Please also read [20] (available free) at https://www.kernel.org/doc/ols/2007/ols2007v2-pages-87-96.pdf.

Solution 1.1 –
I read the materials.

(Reference: Cloud Computing: Business Trends and Technologies)

Homework 1.2 –
Explain the advantage that *paravirtualization* provides for handling timers in virtual machines.

Solution 1.2 –
All modern operating systems, even idle virtual machine rely on clock interrupts to maintain their internal timers, a feature that is particularly essential for real-time media processing. With *paravirtaulization*, the virtual machine code is changed to request a notification at the specified time. The hypervisor without the *paravirtualization* would need to schedule timer interrupts back-to-back for idle machines when the guest operating system is scheduled back to run, which would not be considered as a reliable or scalable way of virtualization.

(Reference: Cloud Computing: Business Trends and Technologies)

Homework 1.3 –
Explain how paravirtualization helps in minimizing access to APIC.

Solution 1.3 –
An operating system deals with multiple CPUs in the same way it deals with one; with modular design, it is just scheduler and the interrupt handlers that need to be fully aware of the differences. Note that, x86 based multi-processor architectures use the *Advanced Programmable Interrupt Controller (APIC)* for interrupt redirection in support of Symmetric Multiprocessing (SMP). Accessing APIC in virtual mode is expensive because of the transitions into and out of the hypervisor. With paravirtualizaion, which has the full view of the code, the multiple APIC access request can be replaced with a single hyper call.

(Reference: Cloud Computing: Business Trends and Technologies)

Homework 1.4 –
Find out if *Linux* (like *Unix*) has both the user-mode and system-mode stacks for each process it runs.

Solution 1.4 –
Yes, in Linux like Unix, has both the user-mode and system-mode (or supervisory mode or kernel-mode) stacks for each process it runs.

The CPU may have more than one set of identical registers. As a minimum, one register set is reserved for the user mode in which application program executes and the other for the system mode in which only the operating system software executes. The switch from user mode to system mode is not done automatically by the CPU. The first thing that happens is CPU gets interrupted by interrupts (like timers, keyboards, etc.) and when these interrupts occur, the current running execution is stopped by the CPU and switch to system mode to execute interrupt handler which activates the system stack pointer. This handler saves the state of the CPU by performing its operations, then restore the state and returns to user mode.

(Reference: Cloud Computing: Business Trends and Technologies)

Homework 1.5 –
Find out what "unscrambled" means in the description of the *Intel LSL* instruction (you can, for example, use the Intel manual referenced in the lecture).

Solution 1.5 –
As per Intel manual, section 5.10.3, "Unscrambled" in *Intel Load Segment Limit (LSL)* instruction means the limit scaled according to the setting of the G flag in the segment descriptor. The unscrambled limit is loaded when the privilege level and type checks pass into destination register and set a 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 cleans the ZF flag.

When the processor access any segment it performs a limit check to make sure that the offset is within the limit of the segment. The software can perform the limit checking using the LSL (Load Segment Limit) instruction. The LSL instruction specifies the segment selector for the segment descriptor whose limit is to be checked and a destination register. Depending on the G flag. The limits are interpreted in a different ways. When the G flag is clear, the effective limit is the value of the 20-bit limit in the segment descriptor. Here, the limit ranges from 0 to 1MB when the G flag is set 4KB page granularity, the processor scales the value in the limit field by a factor of 212 (4KBytes). In this case, the effective limit ranges from 4KB to 4GB.

(References: http://www.intel.com/Assets/ja_JP/PDF/manual/253668.pdf , http://www.nacad.ufrj.br/online/intel/vtune/users_guide/mergedProjects/analyzer_ec/mergedProjects/reference_olh/mergedProjects/instructions/instruct32_hh/vc163.htm)

Homework 1.6 –
Read the following two papers:

- Carl Waldspurger and Rosenblum, M. (2012) *I/O Virtualization*. Communications of the ACM, vol. 55, No 1. January 2012. Pages 66-72; and
- Muli Ben-Yehuda; Xenidis, J.; Ostrowski, M.; Rister, K.; Bruemmer, A.; Van Doorn, L. (2007). *The Price of Safety: Evaluating IOMMU Performance*. Proceedings of the Linux Symposium in June 27th–30th, 2007. Ottawa, Ontario. Pages 225-230.

1. Explain the advantages and disadvantages of using I/O MMU by citing the appropriate text from the paper;
2. Research the Web to find what is meant by "carrier-grade hypervisors". What products are available?

Solution 1.6 –

1. <u>Advantages of I/O MMU</u>
    - I/O MMU translates the I/O virtual memory address to corresponding physical memory, making direct access by devices safe and 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.
    - The large region of memory can be allocated without the need to be contiguous in physical memory – the IOMMU maps contiguous virtual addresses to the underlying fragmented physical addresses.
    - Devices that do not support memory addresses long enough to address the entire physical memory can still address the entire memory through the IOMMU, avoiding overheads associated with copying buffers to and from the peripheral's addressable memory space.
    - Decoupling enables time and space multiplexing of I/O devices, allowing multiple logical devices to be implemented by a smaller number of physical devices.
    - The ability to multiplex logical I/O devices onto physical ones allows both administrators and automated systems to drive I/O devices at higher utilization and achieve better hardware efficiency. Much about virtualization rapid adoption over the past decade can be attributed to the significant cost savings resulting from such basic partitioning and server consolidation.
    - Decoupling provides for flexible mappings between logical and physical devices, facilitating seamless portability. By supporting mappings of logical I/O devices to physical devices with different yet

semantically compatible interfaces, virtualization makes VMs portable, even across heterogeneous systems.

- Decoupling also enables popular VM features such as the ability to suspend and resume a VM and the ability to move a running VM between physical machines, known as live migration.
- This virtualization layer may also change mappings to physical devices, even when the VM itself does not move. For example, by changing mappings while copying, storage contents, a VM's virtual disk can be migrated transparently between network storage units, even while remaining in active use by the VM.
- The same capability can be used to improve availability or balance load across different I/O channels.
- I/O virtualization provides a foothold for many innovative and beneficial enhancements of the logical I/O devices.
- One useful capability enabled by I/O virtualization is device aggregation, where multiple physical devices can be combined into a single more capable logical device that is exported to the VM. Examples include combining multiple disk storage devices exported as a single larger disk.
- New features can be added to existing systems by interposing and transforming virtual I/O requests, transparently enhancing unmodified software with new capabilities. For example, a disk write can be transformed into replicated writes to multiple disks, so that the system can tolerate disk-device failures.

Disadvantages of I/O MMU

- Some degradation of performance of translation and management overhead (e.g., page table walks).
- 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.

(References: I/O Virtualization by Carl Waldspurger, Mendel Roseblum,
https://en.wikipedia.org/wiki/Input%E2%80%93output_memory_management_unit,
http://www.linuxpundit.com/documents/CGV_WP_Final_FN.pdf)

2. Carrier Grade can be defined as virtualization services that fulfil 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:

   - Availability
   - High performance scaling
   - Small error recovery domains
   - Real-time behavior
   - Upgrade capabilities
   - Configurable security
   - Efficient and Uniform management services

   Products available are:

   - VirtualLogix Carrier Grade Hypervisors
   - Bare-metal Xen Hypervisor
   - NEC’s Carrier Grade Cloud Platform
   - Oracle Solaris

   (Reference: http://www.linuxpundit.com/documents/CGV_WP_Final_FN.pdf)

Homework 1.7 –
Find out what hypervisors *Amazon* is using in EC2, and describe their major characteristics.

Solution 1.7 –
Amazon EC2 uses bare-metal hypervisors in Xen. Major characteristics are:

a. Live VM Migration: It supports virtual machine live migration from one host to another allows workload balancing and the avoidance of downtime.
b. Live Storage Migration: Move live running virtual machines and their associated virtual disk image within and across resource pools leveraging local and shared storage.
c. 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.
d. 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.
e. Memory Overcommit: Reduce costs and improve application performance and protection by sharing unused server memory between VMs on the host server.
f. Site Recovery: Provides site-to-site disaster recovery planning and services for virtual environments. Site recovery is easy to set up, fast to recover, and has the ability to frequently test to ensure disaster recovery plans remain valid.

(References: https://en.wikipedia.org/wiki/Amazon_Elastic_Compute_Cloud, https://en.wikipedia.org/wiki/Xen, http://xenserver.org/overview-xenserver-open-source-virtualization/open-source-virtualization-features.html)

Homework 1.8 –
Examine the *Amazon* EC2 VM offer capabilities and answer the following questions:

a. How (i.e., in what units) does EC2 measure the CPU power of a virtual machine and how is the unit in question translated into the power of the physical processors?
b. What kinds of machine instances are there as characterized by the power of their respective CPUs, platform (i.e., 32-bit or 64-bit), memory, storage, etc.? Please list all the instances in the nomenclature along with their respective characteristics;
c. Which operating systems are available on the above systems?

Solution 1.8 –

a. EC2 measures the CPU power of a virtual machine in ECU (EC2 Computing Units). ECU equates to a certain amount of computing cycles in a way that is purportedly independent of the actual 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.

   (References: https://aws.amazon.com/ec2/faqs/#What_is_an_EC2_Compute_Unit_and_why_did_you_introduce_it, https://www.datadoghq.com/blog/are-all-aws-ecu-created-equal/)

b. Amazon EC2 gives the option of choosing between different instance types and provides the flexibility to choose the combination of instance to meet the computing need most appropriately and these sets of instance combinations can be changed later depending upon change in business need. Each instance, provides a predictable amount of dedicated compute capacity and is charged per instance hour consumed. The various types of instances are:

   General Purpose: This instance family includes T2, M3, and M4 which is often the first choice because of variety of CPU size range. Also the balance of resources makes them ideal for running small and mid-size databases, more memory-hungry data processing tasks, caching fleets, and backend servers.

   - **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 (examples web servers, development environments and databases).

Features:

- High frequency Intel Xeon processors
- Burstable CPU, governed by CPU credits, and consistent baseline performance
- Lowest-cost general purpose instance type, and Free Tier eligible (t2.micro only)
- Balance of computing, memory and network resources

- **M3** instances provide balance of compute, memory and network resources, and it is a good choice for many applications.

  Features:
  - High frequency Intel Xeon E5-2670 v2 (Ivy Bridge) processors
  - SSD-based instance storage for fast I/O performance
  - Balance of computing, memory, and network resources

- **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:
  - 2.3 GHz Intel Xeon E5-2686 v4 (Broadwell) processors or 2.4 GHz Intel Xeon E5-2676 (Haswell) processors
  - EBS-optimized by default at no additional cost
  - Support for Enhanced Networking
  - Balance of computing, memory, and network resources

<u>Compute-Optimized</u>: This instance family includes the C3 and C4 instance types, and is geared towards applications that benefit from high compute power.

- **C3** have the best price/ECU compared to all other instances and also provides features those are not available in C1 and CC2.

  Features:
  - High frequency Intel Xeon E5-2680 v2 (Ivy Bridge) processors
  - Support for Enhanced Networking
  - Support for Clustering
  - SSD backed instance storage

- **C4** instances are the latest generation of computing optimized instances; provide high performance processors and the lowest cost for CPU performance.

  Features:
  - High frequency Intel Xeon E5-2666 v3 (Haswell) processors optimized specifically for EC2
  - EBS-optimized by default and at no additional cost
  - Ability to control processor C-state and P-state configuration on the c4.8xlarge instance type
  - Support for networking and clustering

<u>Memory Optimized</u>: 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.

- **X1** instances are optimized for large-scale, enterprise-class, in-memory applications and have the lowest price per GiB of RAM of all other instance types.

  Features:
  - High frequency Intel Xeon E7-8880 v3 (Haswell) processors
  - Lowest price per GiB of RAM
  - Up to 1,952 GiB of DDR4 based instance memory
  - SSD Storage and EBS-optimized by default and at no additional cost
  - Ability to control processor C-state and P-state configuration

- **R3** instances are optimized for memory intensive applications and offer lower price per GiB of RAM.

  Features:
  - High frequency Intel Xeon E5-2670 v2 (Ivy Bridge) processors
  - SSD Storage
  - Support Enhanced Networking

- **R4** instances are optimized for memory intensive applications and offer lower price per GiB of RAM than R3.

Features:
- 2.3 GHz Intel Xeon E5-2686 v4 (Broadwell) processors
- DDR4 Memory
- Support Enhanced Networking

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.

- **P2** instances are intended for general purpose GPU compute applications

  Features:
  - High frequency Intel Xeon E7-2686 v4 (Broawell) processors
  - High performance NVIDIA K80 GPUs, each with 2,496 parallel processing cores and 12GiB of GPU memory
  - Support, GPU Direct (peer-to-peer GPU Communication)
  - Provides Enhanced Networking using Amazon EC2 Elastic Network Adaptor with up to 20Gbps of aggregate network bandwidth within a Placement Group
  - EBS-optimized by default at no additional cost

- **G2** instances are optimized for graphics intensive applications.

  Features:
  - High frequency Intel Xeon E5-2670 v2 (Sandy Bridge) processors
  - High-performance NVIDIA GPUs, each with 1,536 CUDA cores and 4GB of video memory
  - Each GPU features an on-board hardware video encoder designed to support up to eight real-time HD video streams (720p@30fps) or up to four real-time full HD video streams (1080p@30fps)
  - Support for low-latency, frame capture and encoding for either the full operating system or select render targets, enabling high-quality interactive streaming experiences

- **F1** instances offer customizable hardware acceleration with field programmable arrays (FPGAs)

  Features:
  - High Frequency Intel Xeon E5-2686 v4 (Broadwell) Processors with a base frequency of 2.3 GHz
  - NVMe SSD Storage
  - Support for Amazon EC2 Enhanced Networking

  FPGA Features:
  - Xilinx UltraScale+ VU9P FPGAs
  - 64GiB of ECC-protected memory on 4x DDR4
  - Dedicated PCI-Express x16 interface
  - Approximately 2.5 million logic elements
  - Approximately 6,800 Digital Signal Processing (DSP) engines
  - FPGA Developer AMI

Storage Optimized: This instance family includes the I2 and D2 instance types, and provides you with direct-attached storage options optimized for applications with specific disk I/O and storage capacity requirements. Currently there are two types of storage-optimized instances.

- **I2 – High I/O** instances includes the High Storage Instances that provide very fast SSD-backed instance storage optimized for very high random I/O performance, and provide high IOPS at a low cost.

  Features:
  - High frequency Intel Xeon E5-2670 v2 (Ivy Bridge) processors
  - SSD Storage support for TRIM
  - Support for Enhanced Networking
  - High Random I/O performance and High Sequential Read throughput

- **D2 – Dense Storage** instances feature up to 48TB of HDD-based local storage, deliver high disk throughput, and offer the lowest price per disk throughput performance on Amazon EC2.

  Features:
  - High-frequency Intel Xeon E5-2676v3 (Haswell) processors

- HDD storage
- Consistent high performance at launch time
- High disk throughput
- Support for Amazon EC2 Enhanced Networking

(References: https://aws.amazon.com/blogs/aws/choosing-the-right-ec2-instance-type-for-your-application/, https://aws.amazon.com/ec2/instance-types/, https://aws.amazon.com/ec2/previous-generation/)

c. Amazon EC2 currently supports a variety of operating systems including: Amazon Linux, CentOS, CoreOS, Debian, Fedora, FreeBSD, Gentoo, Genymotion, Oracle Linux, RancherOS, Red Hat Enterprise Linux (RHEL), SUSE, SUSE Linux Enterprise Server, TurnKey Core, Windows Server, and Ubuntu Server 14.04 LTS (HVM)

(References: http://docs.aws.amazon.com/opsworks/latest/userguide/workinginstances-os-linux.html, https://aws.amazon.com/marketplace/b/2649367011?ref_=header_nav_category_2649367011)

Homework 1.9 –
Find out about the pricing of the EC2 platforms and provide a few examples.

Solution 1.9 –
The Amazon EC2 platform is available for free to try. There are no hidden or minimum charges and the user have to pay according to the need and the usage of the services. And to calculate the estimated monthly bill, a monthly calculator available.

There are four ways to pay for Amazon EC2 instances: On-Demand, Reserved Instances, Spot Instances and Dedicated Hosts.

1. On-Demand Instance is useful for unpredicted or short-term workload for this user has to pay for compute capacity by the hour with no long-term commitments or upfront payments.
2. Reserved Instances are useful for predictable workload to reserve the capacity and available with significant discounts up to 75% off as compared to On-Demand Instance. All the standard reserved instances are available always (i.e. 24x7) and allow launching the reserved instances at the time of need.
3. Spot Instances are helpful for urgent computing needs for large amounts of additional capacity and allow to bid on spare computing capacity for up to 90% off the On-Demand Instances price.
4. Dedicated Hosts are helpful to meet compliance requirements to reduce costs by allowing using existing server-bound software licenses (subject to user's license terms). 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 price.

Few examples are as follows:

T2.nano	STANDARD 1 YEAR TERM				
Payment Option	Upfront	Monthly	Effective Hourly	Saving over on-Demand	On-Demand Hourly
No Upfront	$0	$3.29	$0.005	24%	$0.0059 (per hour)
Partial Upfront	$18	$1.46	$0.004	32%	
All Upfront	$34	$0	$0.004	34%	
M4.large	STANDARD 1 YEAR TERM				
Payment Option	Upfront	Monthly	Effective Hourly	Saving over on-Demand	On-Demand Hourly
No Upfront	$0	$53.81	$0.074	31%	$0.108 (per hour)
Partial Upfront	$276	$23.00	$0.063	42%	
All Upfront	$541	$0	$0.062	43%	
C4.large	STANDARD 1 YEAR TERM				
Payment Option	Upfront	Monthly	Effective Hourly	Saving over on-Demand	On-Demand Hourly
No Upfront	$0	$51.25	$0.070	30%	$0.1 (per hour)
Partial Upfront	$263	$21.90	$0.060	40%	
All Upfront	$515	$0	$0.059	41%	

(References: https://aws.amazon.com/ec2/pricing/)

Homework 1.10 –

From the above exercise, you will learn that it is possible to create a free machine instance. Please, do the following:

a. Find out and document the essence of the respective *Service Level Agreement;* in particular write down what one needs to do in order to maintain this service **free;**

b. Describe the process (i.e., what exactly one needs to do) to create a free machine instance that could be used as a server. (**Do not,** however, create anything yet!)

c. Can you create a machine instance equivalent to your own PC and then transfer your own PC image there? If so, how would you achieve that?

Solution 1.10 –

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 sides. The most common component of SLA is that the services should be provided to the customer as agreed upon in the contract.

   In order to maintain free services of Amazon EC2, one needs to sign up under the Free Tier, to get hands on experience for 12 month period. Then the one need to create an account and use the services provided under certain usage limits. The need to follow the steps:

   i. Sign up for an AWS account,
   ii. Have to provide credit card information and billing address. Until the free usage exceeds the limits, you would not be charged for the services.
   iii. Get started with AWS Cloud services by choosing any of the products listed under the Free Tier service.

(References: http://searchcloudcomputing.techtarget.com/essentialguide/Breaking-down-whats-in-your-cloud-SLA, https://www.paloaltonetworks.com/documentation/glossary/what-is-a-service-level-agreement-sla, https://en.wikipedia.org/wiki/Service-level_agreement, https://aws.amazon.com/ec2/sla/)

b. The process what exactly one need to do to create a free machine instance, that could be used as a server are followed:

   i. First, have to create an instance of Amazon EC2 which can be used as a server for hosting an application on the cloud.
   ii. Then have to 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. In the last, user can associate or use a name with your web application.

(References: http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIs.html)

c. Yes, we can create a machine instance equivalent to my own PC and then transfer our own PC image there. All of 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.

(References: https://aws.amazon.com/premiumsupport/knowledge-center/)