INFX 443 Exam 1

Types of cloud services:

- Public: infrastructure made available to the general public, owned by the organization selling cloud services
- Private: Infrastructure operated solely for an organization
- Community cloud: infrastructure shared by several organizations and supports a community that shares concerns.
- Hybrid: composition of two or more clouds (any of the above) as unique entities but bound by standardized tech that enables data and app portability

Cloud Delivery Models:

- SaaS (Software as a Service): not suitable for applications which require real time response or

those when data is not allowed to be hosted externally.

Good for:

- many user using the same product (i.e. email)
- software with demand peaks
- need for web and mobile access
- short term needs

- ### PaaS (Platform as a Service): Allows a cloud user to deploy acquired applications using languages and tolls supported by the service provider.

  User:
  - has control over the deployed applications and hosting environment configs
  - does not manage underlying infrastructure Not useful when:
  - application must be portable
  - Proprietary languages are used
  - hardware and software must be customized

- ### IaaS (Infrastructure as a Service):
  - user is able to deploy and run arbitrary software, which can include operating systems and apps
  - user does not manage or control the underlying cloud infrastructure but has control over os, storage, apps, and possibly some networking components
  - Services offered by this include : server hosting, web servers, storage, computing hardware, operating systems, virtual instances, load balancing, internet access, bandwidth provisioning.

AWS

What is AWS?

- started: 2006
- what it offers: web services, cloud solutions on a metered pay-as-you-go basis
- region vs availability zone: region is where aws physically clusters data centers. An availability zone is one or more discreet data centers with redundant power, network and connectivity in a region.
- EC2 (Elastic Compute Cloud): allows users to rent virtual computers on which to run their own computer applications.
- S3 (Simple Storage Service): provides object storage through a web service interface.
- SQS (Simple Queue Service): supports programmatic sending of messages via web service applications as a way to communicate over the Internet.
- VPC (Virtual Private Cloud): Networking Layer for EC2, a virtual network dedicated to your AWS account.

Concurrency

What is Concurrency?

Leslie Lampart: “What I call concurrency has gone by many names, including parallel computing, concurrent programming, and mul- tiprogramming. I regard distributed computing to be part of the more general topic of concurrency.”

- Concurrency describes the necessity that multiple activities take place at the same time
- Distributed Computing is the execution of concurrency on multiple systems.
- execution of multiple activities in parallel can proceed either quasi-independently or tightly coordinated with an explicit communication pattern
- coordination complicates the description of a complex activity as it has to characterize the work done by individual entities working in concert.

Concurrency and Communication:

- Two sides of concurrency:
    - Algorithmic or logical concurrency
    - Physical concurrency discovered and exploited by the software and the

hardware of the computing substrate

- Barrier synchronization: computation consists of multiple stages when concurrently running threads can not continue to the next stage until all of them have finished the current one.
- Communication speed is considerably slower than computation speed.
- Intensive communication can slow down the concurrent thread of an application considerably

Coarse and Fine Grained parallelization

- Coarse-grained parallelism: large blocks of code are executed before concurrent threads communicate
- Fine-grained parallelism: short bursts of computations alternate with relatively long periods when a thread waits for messages from other threads.

Data parallelism vs task parallelism

- Data parallelism: input data of an application is distributed to multiple cores running concurrently
- Task parallelism: tasks are distributed to multiple processors

Deadlocks and Speedup

What is Deadlock?

- A deadlock is a state in which each member of a group is waiting for another member, including itself, to take action.

What is Speedup?

- Speedup is the improvement in speed of execution of a task executed on two similar architectures with different resources.

Messages and Communication channels

- Message: a structured unit of information
- Communication channel: provides the means for processes or threads to communicate with one another and coordinate their actions by exchanging messages. Done only by means of send(m) and receive(m) events.
- Protocol: a finite set of messages exchanged among processes to help them coordinate their actions.

Space-time diagrams

Display local and communication events during a process lifetime. Local events are small black circles. Communication events in different processes are connected by lines from the send event and to the receive event.

Logical clocks

- Logical clock: an abstraction necessary to ensure the clock condition in the absence of a global clock
- A process maps events to positive integers. LC(e)
- Each process time-stamps each message m it sends with the value of the logical clock at the time of sending: TS(m) = LC(send(m))

Message delivery rules; causal delivery:

- The communication channel abstraction makes no assumptions about the order of messages ; a real-life network might reorder messages.
- First-in-first-out (FIFO) delivery: messages are delivered in the same order they are sent.
- Even if the communication channel does not guarantee FIFO delivery, FIFO delivery can be enforced by attaching a sequence number to each message sent.

The sequence numbers are also used to reassemble messages out of individual packets.

Runs and cuts:

- Run: a total ordering of all the events in the global history of a distributed computation consistent with the local history of each participant process; implies a sequence of events as well as a sequence of global states.
- Cut: a subset of the local history of all processes.
- Frontier of the cut in the global history of n processes: an n-tuple consisting of the last event of every process included in the cut.
- Cuts provide the intuition to generate global states based on an exchange of messages between a monitor and a group or processes. The cut represents the instance when requests to report individual state are received by the members of the group.

Critical Sections

- Concurrency requires a rigorous discipline when threads access shared resources
- Only one thread should be allowed to modify shared data at any given time and other threads should only be allowed to read or write this data item only after the first one has finished.
- This process, called serialization, applies to segments of code called critical sections that need to be protected by control mechanisms called locks permitting access to one and only one thread at a time.

Atomic Actions:

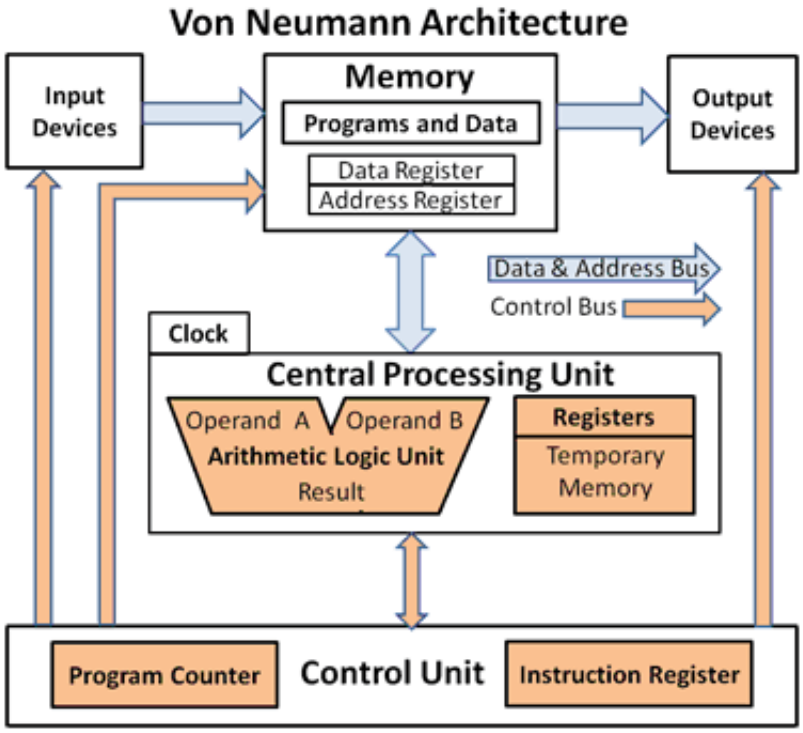
- Parallel and distributed applications must take special precautions for handling shared resources.

- Atomic operation: a multi-step operation should be allowed to proceed to completion without any interruptions and should not expose the state of the system until the action is completed.
- Atomicity requires hardware support:
    - Test-and-Set: Instruction which writes to a memory location and returns the old content of that memory cell as uninterruptible.
    - Compare-and-Swap: instruction which compares the contents of a memory location to a given value and, if the two values are the same, modifies the contents of that memory location to a given new value.

All-or-nothing Atomicity:

- Either the entire atomic action is carried out, or the system is left in the same state it was before the atomic action was attempted; a transaction is either carried out successfully, or the record targeted by the transaction is returned to its original state.
- Two Phases:
    - Pre-commit : it is possible to back up from this without leaving a trace. All steps necessary to prepare for post-commit must be done in this phase. No results should be exposed and no actions irreversible should be carried out.
    - Post-commit: should be able to run to completion. Shared resources allocated during pre-commit cannot be released until after the commit point.

Von Neumann Architecture:



Parallel computer architecure:

- Bit level parallelism: 64 bit architecture has reduced the number of instructions needed to process larger size operands and allowed a significant performance improvement. Also increased allowing instructions to reference a larger address space.
- instruction-level parallelism : today's computers use multi-stage processing pipelines to speed up execution.
- data parallelism: program loops can be processed in parallel
- Task parallelism: The problem can be decomposed into tasks that can be carried out concurrently.

Pipelining

- pipeline: a set of data processing elements connected in series, where the output of one element is the input of the next one
- data hazards: instructions that exhibit data dependence modify data in different

stages of a pipeline.

    - WAR (write after read): (j) tries to write destination before it is read by (i), (i) reads new value.
    - RAW (read after write): (j) tries to read source before (i) writes to it, (j) reads old value.
    - WAW (write after write): (j) tries to write before (i) writes to it. Writes are performed in wrong order. leaving value written by (i) instead of (j).

- structural hazards: a planned instruction cannot execute in the proper clock cycle, or clock tick, because the hardware it is running on cannot support the combination of instructions that are set to execute in the given clock cycle, causing resource conflicts.
- control hazards: pipeline makes wrong decisions on branch prediction and therefore brings instructions into the pipeline that must subsequently be discarded. The term branch hazard also refers to a control hazard.

GPU

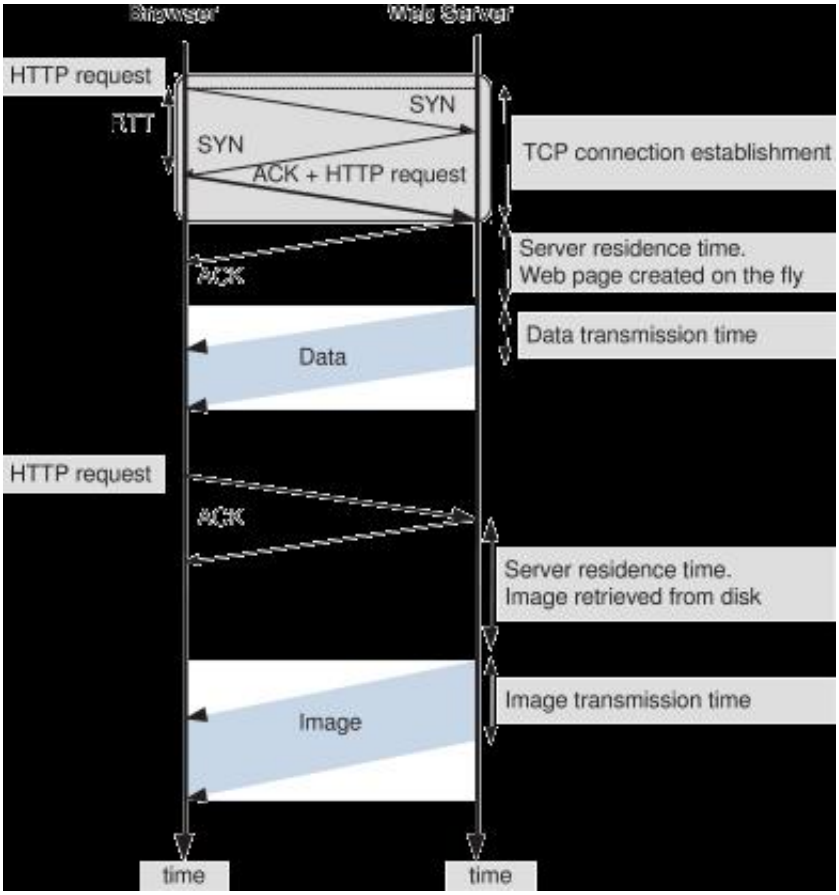
- based on heterogeneous execution model with a cpu acting as the host connected with a gpu.
- Steps of execution:
    - CPU copies input data from main RAM to GPU RAM.
    - CPU intersects the GPU to start processing data.
    - GPU uses literally thousands of cores to execute the parallel code.
    - GPU copies the result to main RAM.

Multicore processor speedup

- design space of multicore processors should be driven by cost-performance considerations.
- cost depends on number of cores and complexity.
- cost-effective design: the speedup achieved exceeds the cost up

Client-server communication for WWW

- Three-way handshake involves the first three messages exchanged between the client browser and the server.
- Once the TCP connection is established, the HTTP server takes its time to construct the page to respond to the first request. To satisfy the second request the HTTP server must retrieve an image from the disk.
- Response time components:
    - RTT (round-trip time)
    - Server residence time
    - data transmission time



Communication protocol layering

- The internet protocol stack:

    - Physical layer: accommodate drivers physical communication channels carrying electromagnetic, optical, or acoustic signals.
    - Data link layer: address the problem to transport bits, not signals between two systems directly connected to one another by a communication channel.
    - Network layer: packets carrying bits have to traverse a chain of intermediate nodes from a source to the destination.
    - Transport layer: the source and the recipient of packets are outside the network. This layer guarantees the delivery from source to destination.
    - Application layer: data sent and received by the hosts at the network periphery has a meaning only in the context of an application.
- Virtualization abstracts the underlying physical resources of a system and simplifies its use, isolates user from one another, and supports replication, which increases system elasticity and reliability.
- virtualization simulates the interface to a physical object.
    - multiplexing: create multiple objects from one instance of a physical object.
    - aggregation: create one virtual object from multiple physical objects
    - emulation: construct a virtual object from a different type of a physical object.
    - multiplexing and emulation: virtual memory with paging multiplexes real memory and disk and a virtual address emulates a real addres; the TCP protocol emulates a reliable bit pipe and multiplexes a physical communication channel and a processor.
- virtualization is critical aspect of cloud computing:
    - System security: allows isolation of services running on the same hardware.
    - Performance isolation: allows devs to optimize applications and cloud service providers to exploit multi-tenancy
    - performance and reliability: allows applications to migrate from one platform to another
    - facilitates development and management of services offered by a provider
- a hypervisor runs on the physical hardware and exports hardware-level abstractions to one or more guest operating systems.
- a guest os interacts with the virtual hardware in the same manner it would

interact with the physical hardware, but under the watchful eye of the hypervisor which traps all privileged operations and mediates the interactions of the guest os with the hardware.