Midterm Review Guide – TCSS 562

Version 0.1

Midterm Date: Wednesday May 9th

The midterm exam will test basic knowledge and awareness of core technologies and concepts relating to cloud computing through mostly short answer questions. If questions require writing, length may be limited.

The midterm exam is open book and open notes. Feel free to bring any paper-based resources for the exam. Calculators are allowed, but laptop computers and cell phones are not permitted. The midterm will be completed as individual work during the class session on Wednesday May 9th.

Potential content includes material from the Cloud Computing Concepts, Technology, and Architecture book. Material from the slides posted online, and/or from chapters 3, 4, 5, 6, 7, and 8 could be covered. Additionally, questions relating to tutorials 1, 2, and 3 may be included as well as questions regarding various AWS cloud technology based on in-class demonstrations.

Some core concepts include:

- Motivations for cloud computing
- What is Grid Computing?
- What is Cluster Computing?
- What is virtualization?
- What are virtual machines?
- What are operating system containers?
- What is the difference between virtual machines and operating system containers?
- What is horizontal scaling?
- What is vertical scaling?
- What is the difference between a cloud provider and a cloud consumer?
- Can a cloud consumer also be a provider?
- What factors (list a few) make ensuring service level agreements “on the cloud” difficult?
- What are some risks associated with cloud adoption?
- What is multitenancy?
- What is resource elasticity in the cloud?
- Types of clouds: Public, Private, Hybrid, Federated, Community
- Know some differences between a type 1 and type 2 hypervisor used for hosting virtual machines
- What is an elastic block store volume?

- For AWS, fundamentally how is performance different for local disks (instance store volumes) vs. network disks (elastic block store volumes)?
- Know about credit-based resource sharing models. Examples:
    - t2 series VM instances – the CPU is shared based on a “credits” system
    - General purpose 2, EBS volumes – I/O operations can burst at a higher rate until credits are exhausted
- Know about spot instances
- What’s the difference between a paravirtual instance and an hvm instance on AWS? Why use paravirtual? Why use hvm?
- What’s the difference between a EBS-backed vs. an instance store volume?
- What instance operations are supported on instance-store VMs?
- What instance operations are supported on paravirtual VMs?
- What’s the difference between a snapshot, an AMI, and an EBS volume?
- Workload distribution architecture
- Active/passive mode for high availability
- Active/active mode for high availability
- What is a warm replica?
- Dynamic scalability architecture
- Service load balancing architecture
    - What is round-robin load balancing?
    - What is least-connection load balancing?
    - What is greedy load-balancing?
- Cloud storage models
    - Elastic block storage
    - Blob/object storage
    - Local disk storage (ephemeral / instance storage)
    - Thin provisioning of disk space
    - Thick provisioning of disk space
- Overprovisioning of cloud resources
- Difference between operating system container, and application container
- Differences between containers and virtual machines
- When to use a container? When to use a VM?
- Isolation limitations of VMs, containers
- Basics of Linux cgroups
- Basics of Linux namespaces
- What is a container orchestration framework?
- What are the motivations for serverless computing? Serverless platforms include Integrated support for:
    - high availability
    - fault tolerance

    - automatic scaling and load balancing
- Other advantages of serverless computing
    - No server configuration
    - Pay only for actual service execution time
- Know some of the challenges for leveraging serverless computing
    - obfuscated billing models
        - The pricing is hard to estimate
        - Reliance on so many external cloud services really "distributes" the cost across a variety of services, complicating cost accounting and bill reconcilation.
    - knowing how much "memory" to reserve
    - Software disaggregation into individual functions impacts hosting costs – there is a case here for consolidating functions, which may be counter-intuitive to the recommended software engineering practices of building many light-weight independent microservices
    - Provisioning variation of infrastructure can lead to performance variability
    - Infrastructure initialization provides 20x slower performance
- What is function-as-a-service?
- What is a COLD call to a serverless computing platform (e.g. AWS Lambda)?
- What is a WARM call to a serverless computing platform (e.g. AWS Lambda)?
- What is the infrastructure freeze/thaw cycle of serverless computing?
- When to use Infrastructure-as-a-Service cloud?
- When to use Platform-as-a-Service cloud?
- When to use Software-as-a-Service cloud?
- When to use Function-as-a-Service cloud? (Serverless Computing)

Pay attention to lecture slides that have questions directly on them. These could be adapted into midterm questions.

It is important to know, and be able to distinguish between the core classifications used for cloud services and what use cases are appropriate for each: Infrastructure-as-a-Service, Platform-as-a-Service, and Software-as-a-Service, Function-as-a-Service

1 Change History

Version	Date	Change
0.1	05/09/2018	Original Version