CSCI 5105

Instructor: Abhishek Chandra

Today

- Cloud Computing
- Edge Computing

Cloud Computing

- On-demand access of computing and storage resources, or services
- User is not responsible for management of resources

Cloud Computing Properties

- "Infinite" computing resources
- Location Transparency
- On-demand access
- Elasticity
- Pay-as-you-go

Enabling Technologies

- Virtualization
  - Multi-tenancy and resource consolidation
  - Dynamic scaling
  - Encapsulation, isolation, and personalization
- Distributed computing and storage
  - Data warehousing and distributed file systems
  - Distributed and data-intensive computing
  - Fault tolerance and scalability
- Good connectivity
  - User-to-cloud
  - CDNs for accessing cloud services

5

Cloud Computing Services

- Software as a Service (SaaS)
  - Provide applications, data, services
  - E.g.: SalesForce, Gmail
- Platform as a Service (PaaS)
  - Provide development platform
  - E.g.: Google AppEngine, MS Azure
- Infrastructure as a Service (IaaS)
  - Provide low-level compute, storage (mostly virtualized)
  - E.g.: Amazon AWS, Rackspace

6

Cloud Deployment Models

- Public clouds: Accessible to public for rent over the Internet
  - E.g.: Amazon AWS
  - Pay-as-you-go, contracts, spot instances
- Private clouds: In-house enterprise data centers with similar scale and technologies
- Hybrid clouds: Public+private cloud
  - Cloud bursting to public cloud
  - Separate functionalities in public (e.g., user front end app) vs. private (e.g., data)

7

Cloud Challenges

- Data mobility
  - Data transfer to/from cloud
  - Data lock-in
- Security and privacy
  - Users do not have control over data/computation
- Programmability
  - Lack of standardization, need to rewrite apps
- Reliability
  - What if cloud goes down/is inaccessible?
- Administrative and legal issues
  - E.g.: s/w licensing, resource ownership

8

Multi-Data Center Cloud

- A Cloud provider with multiple data centers
  - E.g.: Amazon, Microsoft, Google
- Why have multiple DCs?
- Goals:
  - Allow users to access "nearest" DC
  - Different resources, functions in different DCs

9

Edge Computing

- Computing done on the edge of network
- Away from centralized clouds or data centers
- Near the users and devices

10

Edge Computing: Benefits

- Capacity: Large number of resources on the edge
- Locality: Close to data, users
- Resilience: No single point of failure

11

Edge Computing: Examples

- UMN Nebula: Decentralized edge cloud
- Cloudlets: Localized edge computing for mobile devices

12

Nebula: Decentralized Edge Cloud

- Exploit edge computing and storage capacity
  - Uses volunteer nodes
- Support distributed data-intensive applications
  - Strong Data-compute interaction
  - Locality-awareness

Nebula Services

- Nebula Central:
  - Front end to applications and resources
- DataStore: Storage layer
  - Consists of Data nodes and DataStore Master
  - Provides locality-aware *put* and *get* operations
- ComputePool: Computation layer
  - Consists of Compute nodes and ComputePool Master
  - Uses Chrome NaCl for sandboxing
  - Supports locality-aware scheduler
- Applications utilize both services together

Nebula: Locality Awareness

- Challenge: Network may be bottleneck
- DataStore: Locality-aware put/get
  - Data nodes ordered by their locality (b/w, latency, etc.) w.r.t. client
- ComputePool: Locality-aware scheduler
  - Estimate runtime for a task on a node based on both data transfer and computation time
  - Compute nodes ordered for each task based on runtime estimates

Nebula: Fault Tolerance

- Challenge: Node/link failures, node churn
- DataStore:
  - Replication of input and intermediate/output data
  - Maintains a desired replication factor in the presence of failures
- ComputePool:
  - Soft failures: Compute node retries task again
  - Hard failures: Re-execution of tasks on other nodes

Nebula: Usage Scenarios

- In-situ data processing:
  - Exploit locality of data and computation
- On-ramp to centralized cloud:
  - Edge filtering, aggregation, sampling, …
- Low cost cloud:
  - If built using volunteer resources

Cloudlets: Localized Edge Clouds

- Consist of local resources on the edge
  - E.g.: nearby server
  - Designed for latency-sensitive apps
- Useful for mobile application offloading

18

Mobile Computing: Challenges

- Devices are resource-constrained:
  - Lower CPU, memory, storage
  - Network: low b/w, high latency, intermittent connectivity
  - Battery life
- Apps may be resource-intensive:
  - May require higher performance, fidelity, features
- Mobility of user
  - Environment, context, network may change

19

Mobile Application Offloading

- Offload resource-intensive computation to external resources
- Tradeoff:
  - Communication vs. computation overhead
- Questions:
  - What type of computation to offload?
  - Where to offload it?

20

Offloading to the Cloud

- Apps that rely on centralized services
  - E.g.: Web, email, social networking, etc.
  - Amount of data transfer is low
  - Not latency-sensitive
  - Can be used when connectivity is good

21

Offloading to the Edge

- Apps that are latency-sensitive
  - E.g.: Speech recognition, computer vision, augmented reality
  - Amount of data transfer may be high
  - Local computation on device may be slow, energy-heavy
  - May not rely on strong connectivity
- Cloudlet offloading done by VM synthesis
  - Mobile device delivers a VM overlay
  - Applied to a base VM for offloaded computation

22

Edge Computing: Challenges

- Business model:
  - Who pays for infrastructure?
  - How are users charged?
- Resource provisioning and management:
  - How much resources to provision?
  - How to resize dynamically?
- Trust, security, and privacy:
  - Can the user trust the edge with data/compute?
  - How to protect sensitive/personal information?

23