Cloud Computing

Summary of Cloud Computing

Pelle Jakovits
Outline

• Overview of other Cloud services
  – Not yet covered in the course

• Exam information
  – Rules
  – Example questions

• Bonus content:
  – Challenges in Cloud Computing
Cloud computing

- Has emerged as the backbone of modern economy
  - Offers subscription-based services anytime, anywhere following a pay-as-you-go model
- Shorter establishment times for start-ups
- Creation of scalable global enterprise applications
- Has enabled Infrastructure-as-code (IaC) model
- Much more than just IaaS, PaaS and FaaS
  - Has enabled many specialized and provider-managed services
- Has enabled Everything as a Service (EaaS)
Cloud Providers and Services we have discussed

- **AWS**: EC2, S3, Data pipelines, EMR, Auto Scale, CloudWatch
- **Azure**: VMs, Functions, Blob Storage, Cosmos DB
- **OpenStack**
- **Heroku**
- **IBM**: Functions, Cloudant
- ...

5/17/2022
Example Cloud native architecture

https://www.radnip.com/serverless-technology-is-growing-salesforce-is-joining-up/
API Gateways

• Provides a central interface to a group of APIs
  – API authentication
  – API logging endpoint

• Allows you to restrict the use of APIs

• Allows you to dynamically add/remove API endpoints and re-configure routing

• Configure firewall rules
  – Which APIs are available from the external network, which only from the internal network
Route 53

• Cloud DNS - Domain Name Service
• Allows you to manage:
  – How domain names are resolved into IP addresses
  – Obtain Domain names on-demand and in real-time
  – Routing of requests (e.g., based on latency, geolocation)
• Cloud DNS lets you control how user requests and traffic are distributed between cloud regions and your on-premise systems
Content Delivery Networks

• Making resources available from separate *content layer*
  – Images, Videos, HTML, JS files, etc.

• Take advantage of geographical location of resources and end-users
  – Fast delivery of resources
  – Buffering
  – Reducing Internet traffic

• Cloud service do not return data directly, but instead provide a reference to data objects in the CDN
  – The end user (Client application, browser) is automatically redirected to download data from nearest location s
Google Edge Network

• Google Content Delivery Network
• Global infrastructure for delivering and caching data and cloud resources to end users
• Historically mainly used for Youtube videos
• Internet-level traffic balancing
• Caching and Content Delivery
Google Edge network

• Data Centers
• Edge Points of Presence
  – Bigger cities, smaller countries
• Edge Nodes
  – Local service providers

https://cloud.google.com/blog/products/networking/understanding-google-cloud-network-edge-points
Data Centers

https://cloud.google.com/blog/products/networking/understanding-google-cloud-network-edge-points
Edge Points of Presence

https://cloud.google.com/blog/products/networking/understanding-google-cloud-network-edge-points
Edge Nodes

https://cloud.google.com/blog/products/networking/understanding-google-cloud-network-edge-points
AWS Glue

- Scalable serverless data integration
  - No servers to manage
- Automate data integration at scale
- Visual and code-based interfaces to define data integration services
AWS Glue services

• AWS Glue Data Catalog
  – Central and persistent metadata store for data assets stored in different AWS services
  – Automatic schema discovery
  – Manage and enforce schemas for data streams
• AWS Glue DataBrew
  – Visual data preparation tool
  – 250 pre-built transformations to automate data preparation tasks
  – Filtering anomalies, converting data to standard formats, and correcting invalid values
• AWS Glue Elastic Views
  – Combine and replicate data across multiple data stores
  – Build materialized views using SQL
  – Changes to source data trigger updates to the materialized views
  – Supports Amazon DynamoDB, S3, Redshift, Elasticsearch, RDS, Aurora
AWS Glue

Data sources
Connect data directly from the data lake, data warehouses, and databases

AWS Glue DataBrew
Visual data preparation tool to clean and normalize data

Over 250 built-in transformations
Clean and normalize data without writing code

Evaluate data quality
Profile data to highlight data patterns and detect anomalies

Automate at scale
Save transformation steps and reuse them as new data comes in

Amazon S3
Publish prepared data to Amazon S3

Faster insights
Immediately use prepared data in analytics, business intelligence, and machine learning initiatives
AWS Data Exchange

- Find, subscribe to, and use datasets
- 80+ qualified data providers
  - Reuters, Foursquare, Vortexa, IMDb
- AWS Data Exchange charges customers to store data you load to the service
- Measure storage usage in “Byte-Hours”
- Providers can ask subscription fee (or provide for free)
- Providers can verify subscribers who get access and see an overview of subscribers
Machine Learning

- **TensorFlow on AWS** – On-demand performance optimised TensorFlow VMs
- **AWS managed PyTorch** -
  - TorchServe - Deploy trained PyTorch models at scale
  - TorchElastic - training large-scale deep learning models
- **Amazon Lookout**
  - Vision: Spot product defects using computer vision
  - Metrics: Detect anomalies in metrics and identify their root causes
  - Equipment: Detect abnormal behavior by analyzing sensor data
- **Amazon Lex**: Conversational AI for Chatbots
- **Amazon Forecast**: Time-series forecasting service, no machine learning experience required
- **Amazon CodeGuru**: Automate code reviews and optimize application performance with ML-powered recommendations
Amazon SageMaker

- Build, train, and deploy machine learning models
- Automated machine learning
- Data Wrangler to clean and
- Feature store for storing, updating and sharing ML features
- Clarify for detecting bias in models
- Distributed training for scaling up model training
AWS Analytics

- **AWS Athena** - Interactive analytics
- **AWS Elastic MapReduce** - Big data processing
- **Amazon Redshift** - Data warehousing
- **Amazon Kinesis** - Real-time analytics
- **Amazon Elasticsearch Service** - Operational analytics
- **Amazon Quicksight** - Dashboards and visualizations
RoboMaker

• Simulate and deploy robotic applications
• WorldForge: Generate randomized simulation worlds that mimic real-world conditions
• Run large-scale, parallel simulations
• Managed ROS/Gazebo environment
• ROS Cloud Extensions
  – Amazon Rekognition for object detection
  – Amazon Kinesis for video streaming
  – Amazon Polly for converting text to speech
  – Amazon Lex for speech recognition
• Hardware developer kits from Intel, Nvidia, Qualcomm
RoboMaker world simulation
AWS DeepRacer

- Machine learning through a cloud based 3D racing simulator
- Experiment with sensor inputs, reinforcement learning algorithms, neural network configurations and simulation to-real domain transfer methods
- Fully autonomous **Physical** 1/18th scale race car driven by reinforcement learning,
- Global racing league.
Amazon Lumberyard

- Game engine with no royalties or seat fees
- Only pay for the AWS services used (EC2, data transfer, database)
- Open Source - [https://github.com/aws/lumberyard](https://github.com/aws/lumberyard)
- Designed for live, multiplayer, community-driven games
- Based on CryEngine (Far Cry, Prey)
- No fee for single player games
  - can deploy on private servers
Amazon GameLift

- Service hosting solution that deploys, operates, and scales cloud servers for **multiplayer games**
- 21 regions - bring game servers closer to users
- Up to 200 players in a single game session
- Services to manage user matchmaking automatically
- Automatic scaling for up to hundreds of thousands of clients
- DDoS protection
- Provide real-time data on player demand, server capacity, CPU and memory utilization, and create operational alarms
- Supports Unreal Engine, Unity, Amazon Lumberyard, and custom C# and C++ game engines.
- Pay for the compute resources and bandwidth your games actually use
BlockChain

• Simplify and speed up developing blockchain and ledger applications

• **Amazon Quantum Ledger Database** (QLDB)
  – Fully managed ledger database that provides a centralized, immutable, and cryptographically verifiable transaction log
  – Centralized ownership
    • Central authority owns and manages the ledger
    • shared with parties that are working together.

• **Amazon Managed Blockchain**
  – Managed service to create and manage scalable blockchain networks
  – Decentralized ownership
  – Multiple parties can transact with one another without having to know or trust each other

• **Example:** Company wants to set up a Blockchain application to track product supply chain from raw material producer to end customers
IBM quantum computers

• 28 quantum computers available in IBM cloud
• Full Stack Quantum Software
  – Programming languages, IDE, testing, auto-completion, deployment
• Access through a job queue

# IBM quantum computers

<table>
<thead>
<tr>
<th>Name</th>
<th>Qubits</th>
<th>QV</th>
<th>Status</th>
<th>Total pending jobs</th>
<th>System type</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>ibmq_santiago</td>
<td>5</td>
<td>32</td>
<td>Online</td>
<td>6</td>
<td>Falcon r4</td>
<td>-</td>
</tr>
<tr>
<td>ibmq_athens</td>
<td>5</td>
<td>32</td>
<td>Online</td>
<td>10</td>
<td>Falcon r4</td>
<td>-</td>
</tr>
<tr>
<td>ibmq_belem</td>
<td>5</td>
<td>16</td>
<td>Online</td>
<td>43</td>
<td>Falcon r4</td>
<td>-</td>
</tr>
<tr>
<td>ibmq_quito</td>
<td>5</td>
<td>16</td>
<td>Online</td>
<td>11</td>
<td>Falcon r4</td>
<td>-</td>
</tr>
<tr>
<td>ibmq_16_melbourne</td>
<td>15</td>
<td>8</td>
<td>Online</td>
<td>83</td>
<td>Canary r1.1</td>
<td>-</td>
</tr>
<tr>
<td>ibmq_lima</td>
<td>5</td>
<td>8</td>
<td>Online</td>
<td>0</td>
<td>Falcon r4</td>
<td>-</td>
</tr>
<tr>
<td>ibmqx2</td>
<td>5</td>
<td>8</td>
<td>Paused - In use</td>
<td>3</td>
<td>Canary r1</td>
<td>-</td>
</tr>
<tr>
<td>ibmq_armonk</td>
<td>1</td>
<td>1</td>
<td>Online</td>
<td>14</td>
<td>Canary r1.2</td>
<td>-</td>
</tr>
<tr>
<td>simulator_stabilizer</td>
<td>5000</td>
<td>-</td>
<td>Online</td>
<td>1</td>
<td>Clifford simulator</td>
<td>-</td>
</tr>
<tr>
<td>simulator_mps</td>
<td>100</td>
<td>-</td>
<td>Online</td>
<td>1</td>
<td>Matrix Product State</td>
<td>-</td>
</tr>
</tbody>
</table>
IBM quantum programming

Untitled circuit

```
from qiskit import QuantumRegister, ClassicalRegister, QuantumCircuit
from numpy import pi
qreg_q = QuantumRegister(3, 'q')
creg_c = ClassicalRegister(3, 'c')
circuit = QuantumCircuit(qreg_q, creg_c)
circuit.h(qreg_q[1])
circuit.x(qreg_q[1])
circuit.t(qreg_q[1])
```
IBM quantum computers

Draw the circuit

```python
In [ ]:
circuit.draw(output='mpl')
```

Run on a simulator

```python
In [ ]:
simulator = Aer.get_backend('qasm_simulator')
result = execute(circuit, backend=simulator).result()
plot_histogram(result.get_counts(circuit))
```
Advantages of using Cloud services

- On-demand and real-time deployment of resources and services
- No up-front costs and free quotas when starting up
- Less management effort
  - Ease of use
- Many tailored services ready-to use
- Automatic scalability – often managed in the background
- Can deploy services closer to users globally
Disadvantages of using Cloud services

- Limited control over underlying resources and hardware
- May be difficult to estimate costs ahead of time
  - Cost optimization may become a complex problem
- Risk of data confidentiality
- Managing access policies for a large group of users becomes difficult because of very high granularity of control.
- Vendor lock-in (next slide)
- What happens if someone manages to get access to your cloud account?
Conclusions

• There are a very wide variety of cloud services available

• Using cloud services to build your applications has many advantages
  – But need to be careful to mitigate the potential effects of the disadvantages

• Should always consider the Vendor lock-in risk, potential effect and available alternatives
EXAM INFO
Exam

- Option 1: Tuesday 25 May - at 10:15 - 11:45
- Option 2: Tuesday 1 June - at 10:15 - 11:45
- **NB!** You must submit 80% of the lab solutions to qualify for the Exam!
- **NB!** Must collect at least 40% from exam!

- Open Book & Online Exam
- 90 minutes long, 3 questions
- [Model Examination Paper (online version)](#)

5/17/2022
Exam Rules

• Exam paper will be shared to you at the start of the exam.
• You are allowed to use external sources.
  – You are NOT allowed to copy/paste text from external resources,
  – You must write the answer yourself with your own words.
  – You MUST always provide a link to the resources you used as a footnote link.
• You are not allowed to collaborate with anyone during the exam and/or share answers.
• We will create a Zoom conference call, where you can ask questions ~15 minutes before the start of the exam and ask for clarification during the exam.

5/17/2022
Example question 1

• **Compare** Document-oriented and Column family non-relational database models. (10 Marks)
  
  – **Discuss** what needs to be considered **when choosing which model to use** when a scalable database needs to be used to store and query **very large amounts of data**.
  
  – **Describe a detailed scenario:**
    
    • B: When it is **more beneficial to use the Column family model** over the Document-oriented model?
  
    – **Discuss** and provide **supporting argumentation** for this scenario, **why** do you think the selected model is more suitable?
Example question 2

- You are being hired by a startup, which aims to design and launch a Software as a Service (SaaS) application for users to upload different types of files for temporary storing and sharing between users. (10 Marks)
  - The startup wants to reduce the initial launch costs while they have no actual users as they have not obtained significant funding, but still want to be able to easily scale the system once the number of users increases significantly.
  - Which cloud computing model would be the most suitable for such a use case: Infrastructure as a Service (IaaS) or Serverless/FaaS model?
- Discuss the benefits and disadvantages of using each of the models for this use case and provide supporting argumentation why you think the selected model is more suitable than the other two models.
Example question 3

• Describe a detailed scenario when a system deployed in Cloud needs to be scaled, but horizontal scaling is not sufficient by itself and vertical scaling needs to be applied instead. (10 Marks)
  – Discuss and provide supporting argumentation why horizontal scaling is not sufficient for the described scenario and why vertical scaling suits better.
Next lab

• No new lab task this week!
• Start preparing for exam :)
Next Courses

• DevOps: Automating Software Delivery and Operations (LTAT.06.015, 6 ECTS) - Autumn 2022
• Parallel Computing (MTAT.08.020, 6 ECTS) - Autumn 2022
• Mobile Application Development (LTAT.06.021, 6 ECTS) - Autumn 2022
• Internet of Things (LTAT.06.020, 6 ECTS) - Autumn 2022
• Mobile and Cloud Computing Seminar (MTAT.03.280, 6 ECTS) - Autumn 2022, Spring 2023
(This part was not covered in the lecture, included for additional info)

CHALLENGES
A Manifesto for Future Generation Cloud Computing: Research Directions for the Next Decade

Rajkumar Buyya*¹, Satish Narayana Srirama¹²,¹, Giuliano Casale³, Rodrigo Calheiros⁴, Yogesh Simmhan⁵, Blesson Varghese⁶, Erol Gelenbe³, Bahman Javadi⁴, Luis Miguel Vaquero⁷, Marco A. S. Netto⁸, Adel Nadjaran Toosi²¹, Maria Alejandra Rodriguez¹, Ignacio M. Llorente⁹, Sabrina De Capitani di Vimercati¹⁰, Pierangela Samarati¹⁰, Dejan Milojicic¹¹, Carlos Varela¹², Rami Bahsoon¹³, Marcos Dias de Assuncao¹⁴, Omer Rana¹⁵, Wanlei Zhou¹⁶, Hai Jin¹⁷, Wolfgang Gentzsch¹⁸, Albert Y. Zomaya¹⁹, and Haiying Shen²⁰

¹University of Melbourne, Australia
²University of Tartu, Estonia
³Imperial College London, UK
⁴Western Sydney University, Australia
⁵Indian Institute of Science, India
⁶Queen’s University Belfast, UK
⁷Dyson, UK
⁸IBM Research, Brazil
⁹Universidad Complutense de Madrid, Spain
¹⁰Universita degli Studi di Milano, Italy
¹¹Hewlett Packard Labs, USA
¹²Rensselaer Polytechnic Institute, USA
¹³University of Birmingham, UK
¹⁴INRIA, France
¹⁵Cardiff University, UK
¹⁶University of Technology Sydney, Australia
¹⁷Huazhong University of Science and Technology, China
¹⁸UberCloud, USA
¹⁹University of Sydney, Australia
²⁰University of Virginia, USA
²¹Monash University, Australia
Cloud Computing Manifesto

• Discusses major challenges in cloud computing
  – Investigates their state-of-the-art solutions
  – Investigates their limitations
• Overview of current research being done in Cloud Computing
• Emerging trends and impact areas, that further drive these challenges
• Offers comprehensive future research directions

Challenges in Cloud Computing

- Economics of Cloud Computing
- Security and Privacy
- Empowering Resource-Constrained Devices
- Interconnected Clouds
- Heterogeneity
- Sustainability
- Reliability
- Application Development and Delivery
- Data Management
- Networking
- Usability
- Scalability and Elasticity
- Resource Management and Scheduling

5/17/2022
Scalability and Elasticity

• Cloud computing promises virtually unlimited computational resources on demand

• Benefits:
  – **Scalability**: Unexpected peaks in computational demand do not break service level agreements (SLAs)
  – **Elasticity**: Users do not need to make significant upfront investments in infrastructure
    • Rather grow/shrink as their computing needs increase/decrease
Scalability and Elasticity - challenges, and open issues

• Cloud providers must embrace parallel computing hardware
  – Multi-core, clusters, accelerators (GPUs), non-traditional architectures
    • e.g., neuromorphic and future quantum
• Scalability of the Cloud is limited by the extent to which individual components scale, namely compute, storage and interconnects scale
  – End of scaling of Moore’s law? (Doubling the number of transistors every 1.5 year)
  – Research in new technologies (e.g. beyond Metal-Oxide-Semiconductor)
  – For memory, new non-volatile technologies are being explored

• Research challenges
  – Ability to accurately predict computational demand
  – Dynamic creation, mobility, and garbage collection of VMs and containers
  – Programming models that enable dynamic reconfiguration of applications
  – Decentralized scalable distributed algorithms for edge, hybrid and interclouds
Resource Management and Scheduling

- Modern CDCs contain 100,000+ computing and storage devices
- Effective resource management and scheduling policies are important to achieve high scalability and operational efficiency
- IaaS providers mostly rely on
  - Static VM provisioning policies
    - Allocate a fixed set of physical resources to VMs using bin-packing algorithms
  - Dynamic policies
    - Handle load variations through live VM migrations and other load balancing techniques
- Policies available at PaaS and SaaS level
  - Auto-scaling techniques
  - Resource throttling methods
    - To handle workload bursts, trends, control usage of preemptible VMs
  - Admission control methods
    - To handle peak load and prioritize workloads of high-value customers
Resource Management and Scheduling - Challenges

• Existing management policies tend to be intolerant to inaccurate estimates of resource requirements
  – Need to study novel tradeoffs between policy optimality and its robustness to inaccurate workload information

• Resource management policies tend to focus on optimizing specific metrics and resources
  – Lack of systematic approach to co-existence of multiple control loops

• Novel scheduling methods for the distributed and heterogeneous environments

• Serverless computing offloads computations far from the application core components
  – Network latency can visibly affect function response time
  – Novel resource management policies to decide to which extent to rely on FaaS

• By logically centralizing the network control plane, SDNs provide opportunities for more efficient management of resources
Reliability

• Cloud computing systems face a variety of reliability-related threats
  – Hardware failures, missing resource, network failures, timeout failures, flaws in software being triggered by environmental change

• A cascade of failures may be triggered leading to large-scale service disruptions with far-reaching consequences

• Reliability has a great impact on the QoS as well as on the long term reputation of the service providers

• State-of-the-art:
  – Replication of data and computation (e.g. MapReduce)
  – Erasure coding for data storage
Reliability - Challenges

- Standard fault-tolerance and reliability approaches of traditional distributed systems cannot be directly applied in Cloud computing systems
  - Cloud computing is typically more service oriented rather than resource oriented
- Scale and expected reliability are hard to analyze due to the range of inter-related characteristics
  - e.g., their massive-scale, service sharing models, wide-area network, and heterogeneous software/hardware components.
- Cloud environments lack thorough service reliability models, automatic reliability-aware service management mechanisms, and failure-aware provisioning policies
- Deep learning for failure prediction
- Failure aware resource provisioning policies
- Beyond replication for new type of applications such as IoT applications
- Employing both replications and erasure coding in cloud storage for Big Data applications
- Correlation of reliability and energy efficiency
Sustainability

- Data center deployments until recently have mainly focused on high performance and have not paid enough attention to energy consumption.
- Cloud computing worldwide consumes more energy than most countries:
  - Estimate of annual electricity usage are between 200 terawatt hours (TWh) \((\text{Jones, 2018})\) to 500 TWh \((\text{Bashroush & Lawrence, 2020})\).
  - Only 9 countries, (inc. USA, China, Russia, and Japan) surpass Clouds in electricity usage.
- State-of-the-art:
  - Primarily focused on consolidation of VMs for minimizing energy consumption of servers.
  - ML based methods for task allocations to reduce energy consumption.
  - Models for mixing renewable and non-renewable energy.
Sustainability - Future Research Directions

• Provide online automatic, or autonomic and self-aware methods to holistically manage both QoS and energy consumption of Cloud systems
• Rationing the energy supply so that clouds can manage energy vs QoS tradeoffs
• Geographically distributed data coordination, resource provisioning and energy-aware and carbon footprint-aware provisioning in data centers
  – Require novel system architectures and algorithms
• Dynamic task scheduling for energy consumption, SLA and QoS optimization
  – Online decision making and adaptive self-aware techniques
• Tradeoff between the increased energy consumption from many disparate and distribute Fog servers, and the reduced network energy consumption when the Fog servers are installed in close proximity
  – Saving energy for networking elements often disturbs other aspects such as reliability, scalability, and performance of the network
  – Design energy-aware routing algorithms
  – Global network awareness and centralized decision-making offered by SDN may provide better opportunity for creating sustainable networks
Heterogeneity

• Public Cloud infrastructure has constantly evolved in the past decade

• Heterogeneity at three levels
  – VM level: Organization of homogeneous (e.g. same processor family) resources in multiple ways and configurations (e.g. VM types)
  – Vendor level: Different hypervisors or software suites
  – Hardware architecture level: CPUs and hardware accelerators
Heterogeneity challenges

• Workload management is underpinned by benchmarking techniques that are used for workload placement and scheduling techniques
  – Current benchmarking practices are reasonably mature only for the VM level
  – Slowly developing for the vendor level
• Challenging for general-purpose Cloud platform to integrate and manage heterogeneity at all three levels
• Development of application software that is compatible with heterogeneous resources
  – Most accelerators require different (and sometimes vendor specific) programming languages, which have significant learning curve (e.g. CUDA or OpenCL for GPUs)
  – Abstracting hardware accelerators under middleware will reduce opportunities for optimizing the source code for maximizing performance
  – For IaaS, onus on developer but for PaaS and SaaS onus on cloud provider
  – Open challenge in this area is developing software that is agnostic of the underlying hardware, but can still adapt based on the available hardware
Interconnected Clouds

- Cloud providers and platforms still operate in silos
  - Their efforts for integration usually target their own portfolio of services
- Existing public Cloud providers offer proprietary mechanisms for interoperation
  - Exhibit important limitations as they are not based on standards and opensource
- Multiple efforts for standardization of federated cloud computing e.g. NIST Federated cloud, IEEE InterCloud etc.
- Application containers and configuration management tools for portability
- Middleware and library solutions for hybrid clouds (e.g. Aneka)
Interconnected Cloud challenges

• How to go beyond the minimum common denominator of services when interoperating across providers (thus enable richer cloud applications)?
• How to coordinate authorization, access, and billing across providers?
• Efficient and transparent provisioning, management and configuration of cross-site networks
• How to apply InterCloud solution in the context of fog computing?
• Novel approaches for billing and accounting, novel interconnected Cloud suitable pricing methods, along with formation of InterCloud market-places
• SDN and NFV enhanced InterCloud operations
  – SDNs and the capability to shape and optimize network traffic
    • For optimization of wide-area network traffic connecting data centers
  – Enforcing prioritization of service traffic across providers/data centers and specific security requirements
Empowering Resource-Constrained Devices

• Cloud services are also relevant for the resource constrained devices and their applications
• Mobile devices have limited battery life and when compared to desktops have limited CPU, memory and storage capacities
• Harnessing external Cloud resources from mobiles led to the emergence of Mobile Cloud paradigm
• Mobile Cloud binding models
  – Task delegation: Invokes web services from multiple Cloud providers (e.g. taking help of middlewares)
  – Mobile code offloading: Profiles and partitions the applications, and the resource-intensive methods/operations are identified and offloaded to surrogate Cloud instances
    • Open challenges with ideal decision mechanisms, adaptability issues etc.
• Cloud-centric Internet of Things
  – Limitations with privacy, autonomy, limited throughput, and latency lead to Fog computing
Empowering Resource-Constrained Devices - Future Research Directions

• Better models for multi-tenancy in Mobile Cloud applications
  – To share the costs among multiple mobile users
• Incentive mechanisms for mobile clouds and independent fog providers
  – To increase the adoption of these technologies
• Containers for fog computing
• Edge analytics for real-time data processing
  – The sensor data will be processed across the complete hierarchy of Fog topology
    • Tasks include filtering, consolidation, error detection etc.
• Efficient algorithms for self-adaptive resource provisioning and QoS-aware application module scheduling in fog topology
  – E.g. QoS includes deadline, priority, optimizing energy consumption, cost-aware etc.
  – Heuristics, meta-heuristic and evolutionary algorithms
Security and Privacy

• Security is a major concern in ICT systems and Cloud computing is no exception
• Secure and private management of data and computations in the Cloud (confidentiality, integrity, and availability)
• Confidentiality:
  – Encrypt the data before storing them at external Cloud providers
  – Limits the support of query evaluation at the provider side
    • Definition of indexes, which enable (partial) query evaluation at the provider side without the need to decrypt data
      – Definition of indexes must balance precision and privacy
    • Encryption techniques that support the execution of operations (e.g. Order Preserving Encryption)
  – Selective visibility for different users
    • In scenarios where different parties cooperate for sharing data and to perform distributed computations
  – They are exposed to attacks exploiting frequency of accesses to violate data and users privacy
    • Solution: Private information retrieval technique
Security and Privacy - continued

• Techniques such as digital signatures, Provable Data Possession and Proof Of Retrievability, detect unauthorized modifications of stored data
  – When data can change dynamically (e.g. multiple writes), they have issues
• Design of solutions completely departing from encryption
  – Based on the splitting of data among multiple providers to guarantee generic confidentiality and access/visibility constraints possibly defined by the users
• Lack of central controls in Fog-based scenarios may raise privacy and trust issues
  – Adapt earlier research of secure routing, redundant routing and trust topologies performed in the P2P context
• Risk of inferring sensitive information significantly increases in Big Data
• Tracking Big Data provenance
  – Verifying whether data came from trusted sources
  – Evaluating the quality of the Big Data
• Blockchain technology can be helpful for addressing the data provenance challenge
  – Novel privacy concerns - Since data in a blockchain cannot be changed or deleted
Economics of Cloud Computing

- Centered on a number of key aspects
  - **Pricing of cloud services**: How a cloud provider should differentiate between capabilities they offer
  - **Brokerage mechanisms**: Enable users to dynamically search for Cloud services
  - **Monitoring**: To determine user requirements are being met, identifying penalty
- Specifications (e.g. WS-Agreement) and application of SLAs (performance, memory and CPU)
- Understanding of how organizations migrate in-house or externally hosted infrastructure
  - Risks associated with uptime and availability
  - IT departments and system admins acting as potential brokers
  - Relying mainly on pre-agreed SLAs with CDC
- Sub-second billing made possible with containers and serverless computing
- Licensing solutions (e.g. annual or perpetual licensing, BYOL – bring your own license)
- Choosing the right Cloud provider
  - Commercial and research grade platforms to investigate benefit/limits of Cloud selection (e.g. RightScale, CloudMarketMaker)
- Marketplace models where users purchase services from SaaS providers that inturn procure computing resources from either PaaS or IaaS

5/17/2022
Economics of Cloud Computing - Future Research Directions

• Factors influencing economics of serverless functions
  – e.g. average vs peak transaction rates, scaling and concurrency, benchmarking on different hardware platforms

• Economics of fog and edge computing
  – Additional business models and the inclusion of additional category of providers in the Cloud marketplace
  – Micro Data Centers (MDC)
    • Dynamic MDC discovery, Pre-agreed MDC contracts, MDC federation (multiple MDCs collaborate and share workload in a particular area)
  – MDC-Cloud data center exchange
  – User contacts CDC which decides to outsource computation to MDC (e.g. to meet QoS such as latency)

• Economics under uncertainties

• Trust issues arising due to migrating in-house IT systems
  – GDPR (came into effect in May 2018) and its benefits

• Potential new players in marketplace with serverless and edge computing
  – Telco vendors are likely to form alliances with existing Cloud providers for supporting real time stream processing and edge analytics (e.g. autonomous vehicles and smart city sensing)
Application Development and Delivery

- Cloud computing empowers application developers with the ability to programmatically control infrastructure resources and platforms.
- Resource programmability:
  - A looser boundary between development and operations (DevOps).
    - Results in the ability to accelerate the delivery of changes to the production environment.
    - A variety of agile delivery tools and model-based orchestration languages (e.g., Terraform and OASIS TOSCA) are increasingly adopted in Cloud application.
      - These tools help automating lifecycle management, including continuous delivery and continuous integration, application and platform configuration, and testing.
- Platform programmability:
  - Separation of concerns has helped in tackling the complexity of software development for the Cloud and runtime management.
    - E.g. with MapReduce application developers specify Map and Reduce, and middleware layers deal with non-functional concerns, such as parallelization, data locality optimization, and fault tolerance.
  - However, there is a shortage of application delivery frameworks and programming models to deliver software spanning both the Edge and the CDC.
- Application evolution is still a major challenge.
Application Development and Delivery
- Future Research Directions

• How to continuously monitor and iteratively evolve the design and quality of Cloud applications within the continuous delivery pipelines?
• To extend existing software development and delivery methodologies with reusable abstractions
  – For designing, orchestrating and managing IoT, Fog/Edge computing, Big Data, and serverless computing technologies and platforms
• Infrastructure-as-code (IaC) is expected to grow further
• Novel architectures and patterns to tackle cloud application decomposition
  – With implications in security, performance, reliability and operations costs
    • Individual functions are easier to protect and verify than monoliths vs. greater attack surface with FaaS-based architectures
    • The benefits of function-level auto-scaling vs increased network traffic and latency experienced with FaaS
    • FaaS cheaper to use per function invocation but can incur higher network charges
• Finer-grained programming abstractions (e.g., actor model) to dynamically re-configure programs between edge resources and CDC
Data Management

• Key selling points of Cloud computing is the availability of affordable, reliable, and elastic storage, collocated with the computational infrastructure

• Several storage abstractions are offered
  – Object-based storage (e.g. Amazon S3, Azure File)
  – Block storage services (AzureBlob, Amazon EBS) of a disk volume
  – Higher-level data platforms (Relational DBs, NoSQL DBs)

• Proliferation of big data platforms
  – Batch processing (Apache Hadoop, Spark)
  – NoSQL query platforms for web and enterprise workloads
  – Distributed stream processing (Apache Storm, Heron)
Data Management – Future Research Directions

• Services for data storage have not been adequately supported by services for managing their metadata
  – Allow data to be located and used effectively
• Data increasingly being sourced from edge of the network
  – Their low-latency processing requirements
• Lack of tracking metadata describing the source and provenance of the data in “Data lakes” makes it challenging to use them
  – Scientific repositories have over a decade of experience with this
  – Blockchains for data provenance
• While legal protections exist, there are no clear audit mechanisms to show that data has not been accessed by the Cloud provider themselves
• Leveraging computing capabilities of network devices to perform in-transit processing – Fog and Edge analytics
• Efficient management of trained models and their rapid loading and switching to support online and distributed analytics applications
Usability

• The Human Computer Interface and Distributed Systems communities are still far from one another

• Usability is a key aspect to reduce costs of organizations exploring Cloud services and infrastructure

• Cloud Usability Framework highlights five aspects:
  – **Capable**: meeting Cloud consumers expectations with regard to Cloud service capabilities
  – **Personal**: Allowing users and organizations to change the look and feel of user interfaces and to customize service functionalities
  – **Reliable, secure, and valuable**: Having a system that performs its functions under state conditions, safely/protected, and that returns value to users

• Current efforts in Cloud have mostly focused on encapsulating complex services into APIs to be easily consumed by users
  – E.g. services to expose HPC applications
Usability - Future Research Directions

• It is still hard for users to know how much they will spend renting resources due to workload/resource fluctuations or characteristics
  – Tools to have better estimations would improve user experience and satisfaction
• New visualization technologies could be further explored on the different layers of Cloud environment
  – To better understand infrastructure and application behavior and highlight insights to end users
• Easier API management methodologies, tools, and standards are also necessary
  – To handle users with different levels of expertise and interests
• Users are still overloaded with resource and service types available to run their applications
  – E.g. CPUs, GPUs, network, storage, operating system flavor, and all services available in the PaaS
  – Advisory systems to also recommend how users should use Cloud more efficiently
• Change in focus at evaluation
  – Cloud computing researchers and practitioners mostly perform quantitative experiments, whereas researchers working closer to users have deep knowledge on qualitative experiments
Discussion and Summary

• There will be significant developments across all the service models (IaaS, PaaS, and SaaS) of Cloud computing

• IaaS:
  – Scope for heterogeneous hardware such as CPUs and accelerators (e.g., GPUs and TPUs) and special purpose Clouds for specific applications (e.g., HPC and deep learning)
  – Future generation Clouds should also be ready to embrace the non-traditional architectures (e.g. neuromorphic, quantum computing)
  – Emerging trends such as containerisation, SDN and Fog/Edge computing are going to expand the research scope of IaaS by leaps and bounds
  – Sustainability of CDC through utilisation of renewable energy and IoT-enabled cooling systems
  – Scope for emerging trends in IaaS, such as disaggregated data centres
  – Future research directions proposed for addressing the scalability, resource management and scheduling, heterogeneity, interconnected Clouds and networking challenges
Discussion and Summary - continued

- PaaS:
  - Should see significant advancements through future research directions in resource management and scheduling
  - Need for programming abstractions, models, languages and systems supporting scalable elastic computing and seamless use of heterogeneous resources are proposed
    • Lead to energy-efficiency, minimised application engineering cost, better portability and guaranteed level of reliability and performance
  - ML, AI and Deep learning should help in dealing with the complexity, heterogeneity, scale and load balancing applications developed through PaaS
  - Interesting future directions are proposed such as function-level QoS management and economics for serverless computing
  - Future research directions for data management and analytics are discussed in detail along with security
    • Lead to interesting applications with platform support (e.g. edge analytics for real-time stream data processing in IoT and Smart Cities domains)
Discussion and Summary - continued

• **SaaS:**
  – Should mainly see advances from the application development and delivery, and usability of Cloud services
  – A variety of agile delivery tools and Cloud standards (e.g., TOSCA) are increasingly being adopted during Cloud application development
  – Future research should focus at how to continuously monitor and iteratively evolve the design and quality of Cloud applications
  – Extend DevOps methods and define novel programming abstractions to support IoT, Edge computing, Big Data, and serverless computing
  – Novel incentive mechanisms are required for mobile Cloud adaptability as well as for designing Fog architectures

• Discussed research directions show a promising and exciting future for the Cloud computing field both technically and economically
• The manifesto calls the community for action in addressing them
References