Chapter 18
Cloud Computing for Multimedia Services

18.1 Cloud Computing Overview

18.2 Multimedia Cloud Computing

18.3 Multimedia Content Sharing over Cloud

18.4 Multimedia Computation Offloading

18.5 Interactive Cloud Gaming

18.6 Edge Computing and Serverless Computing for Multimedia

18.1 Cloud Computing Overview

- A new paradigm, allowing for computing resources to be purchased or shared using a pay for usage model.

- Much like traditional utility resources, such as electrical power, users are now free to purchase or lease resources as needed.

- Many existing applications, from content sharing to media streaming have been migrated to this model with great success in terms of costs and system efficiency.

- Examples include the file synchronization system DropBox and the media streaming service Netflix

Cloud Computing: Private vs Public

- Clouds a generally split into two categories private and public.
  - **Private clouds** are generally maintained on private networks and usually only utilized by a single datacenter tenant to provide services to their customers.
  - **Public Clouds** are available as a public utility where multiple tenants can purchase resources to provide services to their customers. For example, Amazon Elastic Compute Cloud (EC2).

Conceptual Overview of Cloud Computing

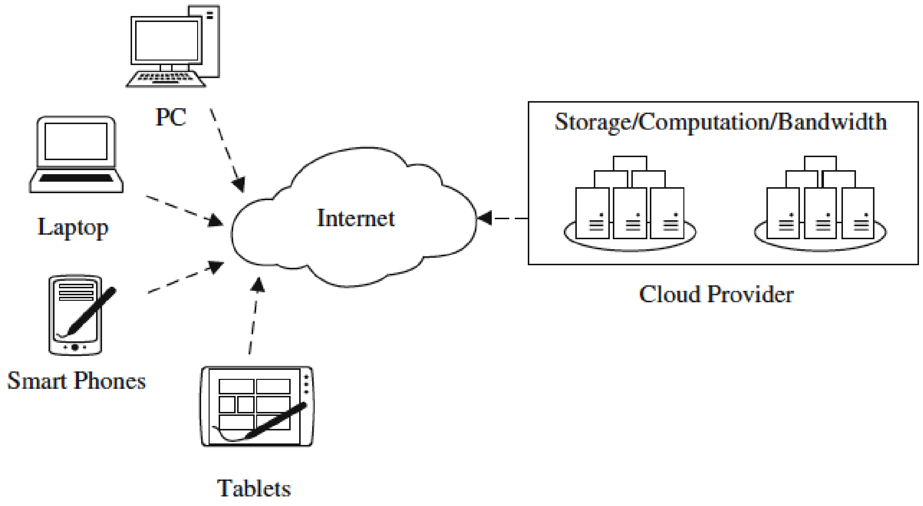


Fig 18.1: A conceptual overview of cloud computing

- Cloud users use an Internet enabled device to connect to a cloud service provider.
- The cloud provider can provide superior computing resources such as storage, computation and bandwidth.
- End users can store data in the cloud, making ubiquitous data access possible.

Cloud Computing: Characteristics

- Common set of characteristics identified by National Institute of Standards and Technology (NIST).
  - **On-Demand Self Service.**

    A user can unilaterally provision computing resources.
  - **Resource Pooling and Rapid Elasticity.**

    •Cloud resources are available for provisioning by multiple users and resources can be dynamically assigned.
  - **Measured Service.**

    •Computing resource usage (eg. computation, storage, bandwidth) can be monitored, controlled and reported to both the cloud provider and user.
  - **Broad Network Access.**

    •Persistent and high quality network access is available.

Different Cloud Service Models

- **Infrastructure as a Service (IasS)**
  - Provides a pool of resources, which the user accesses and configures for their needs. Users are usually free to install their own operating systems and applications. Examples include the virtual machine based Amazon EC2 and the storage service Amazon S3.
- **Platform as a Service (PaaS)**
  - Delivers a development environment, typically include preselected operating systems, programming environments, databases management systems, and webservers. Google’s App Engine is a typical example.
- **Software as a Service (SaaS)**
  - Allows applications to run on the infrastructure and platforms offered by the cloud. Often storing both the users data and application in the cloud. Examples include Google Drive (Formely Google Docs).

Relations Among Different Service Models

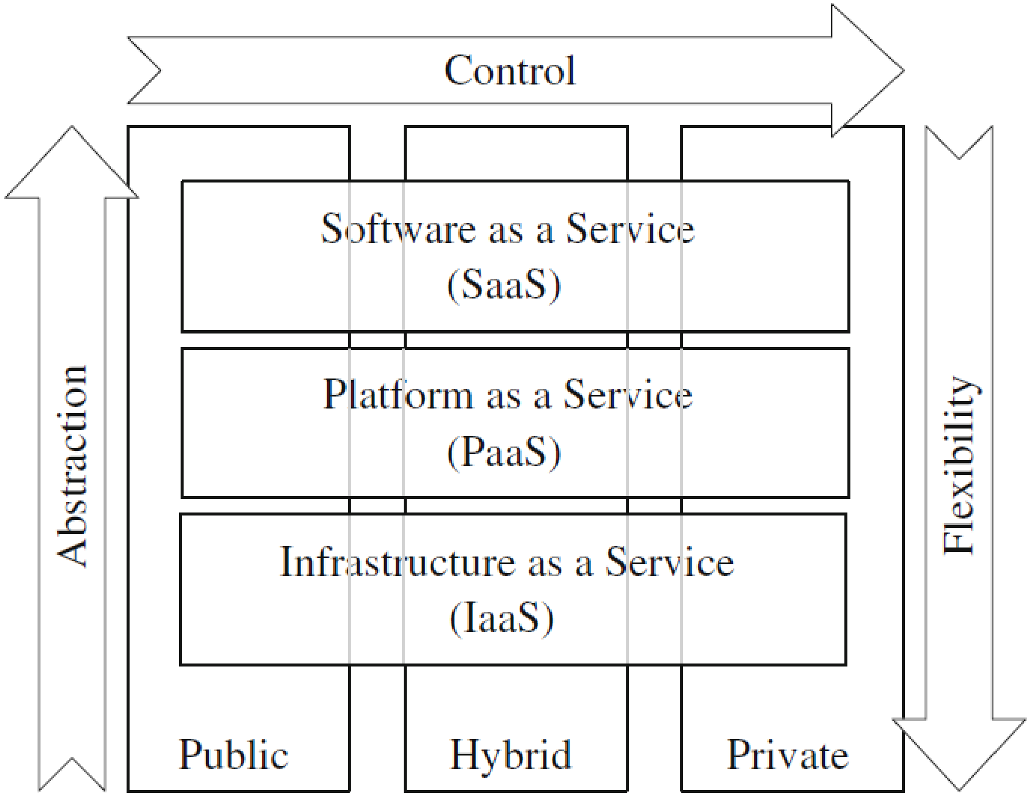


Fig 18.2: An illustration of cloud service models

Cloud Storage Example: Amazon S3

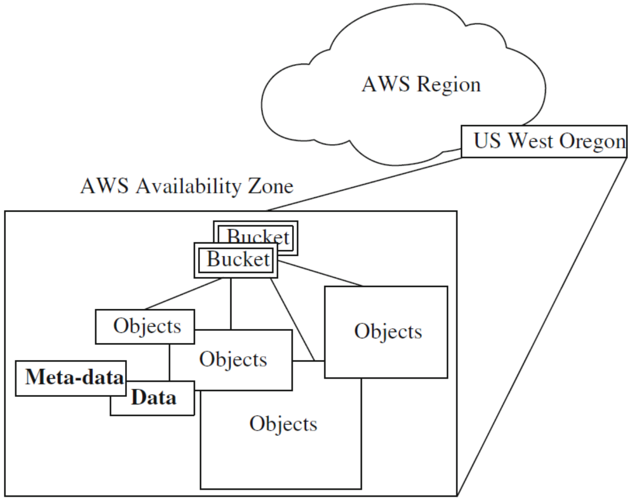
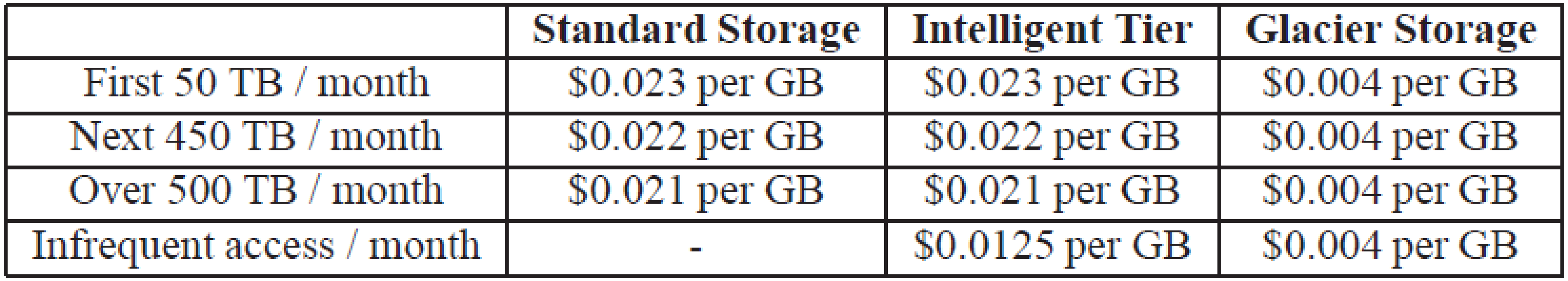


Fig 18.3: An example of a data object stored in an Amazon AWS region (US West)

- S3 provides a web interface to store and retrieve practically any amount of data.
- Intentionally built with a minimal feature set, to provide maximum performance and flexibility for end users and developers.
- Features include encryption, access control as well a durability for stored data. With these features developers can create robust applications.

Cloud Storage Example: Amazon S3

Table 18.1: Sample storage pricing of US East Region (Year 2020)

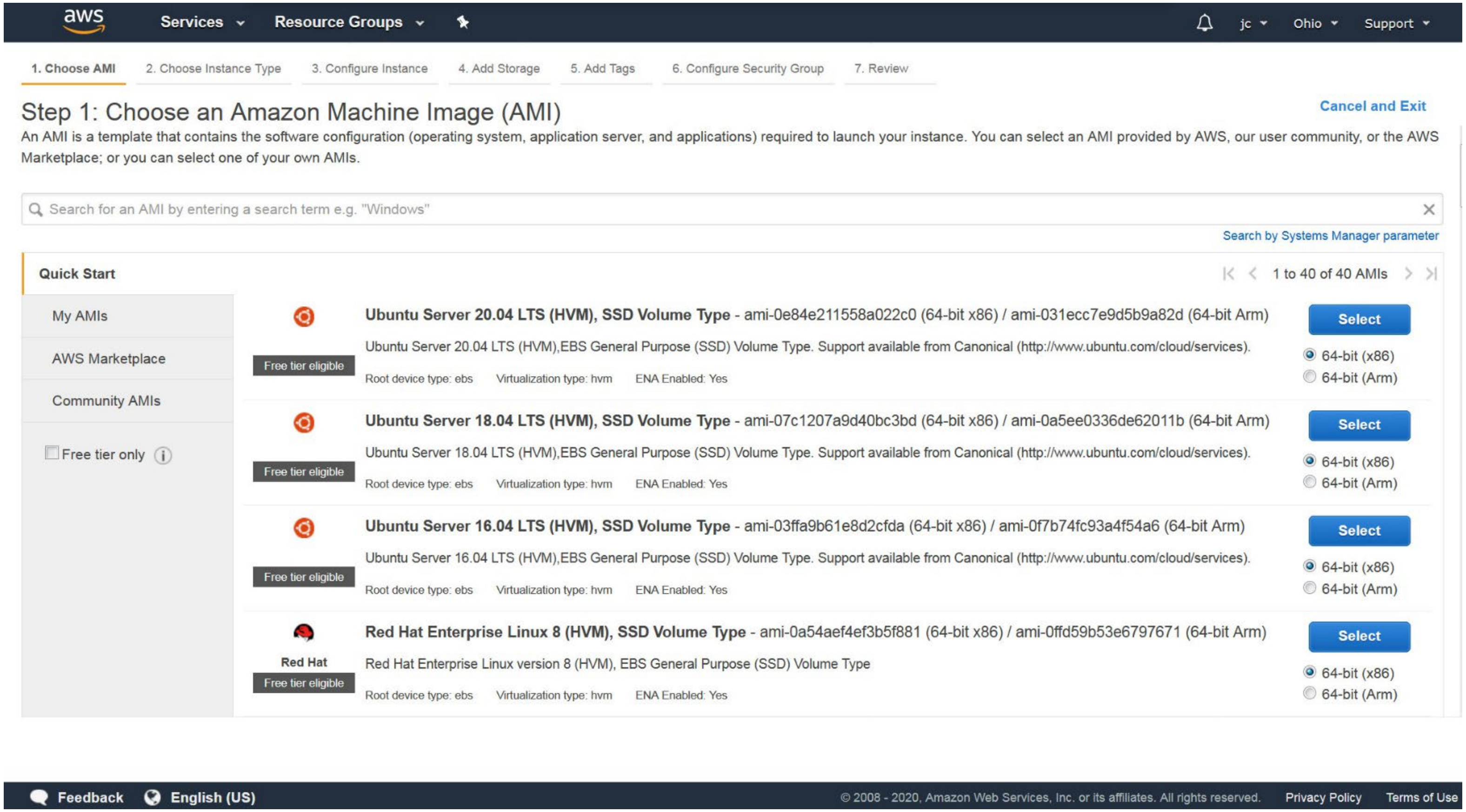


	Standard Storage	**Intelligent Tier**	**Glacier Storage**
First 50 TB / month	\$0.023 per GB	\$0.023 per GB	\$0.004 per GB
Next 450 TB / month	\$0.022 per GB	\$0.022 per GB	\$0.004 per GB
Over 500 TB / month	\$0.021 per GB	\$0.021 per GB	\$0.004 per GB
Infrequent access / month	-	\$0.0125 per GB	\$0.004 per GB

Cloud Computation Example: Amazon EC2

- Amazon's Elastic Compute Cloud (EC2) is a public computational cloud that provides resizable compute instances.
  - Implemented using Virtual Machines (VM) provided by the Xen virtualization system.
  - VMs can be resized dynamically by the users, ex. To increase CPU resources, available ram, and disk space.
- Different instance provisioning strategies exist, namely "on-demand" and "spot" instances.
  - **On-Demand:** Allows users to purchase computation capacity by the hour at a set price.
  - **Spot:** Users bid on unused EC2 capacity and their instances run while their bid price exceeds the current spot price.
- EC2 user has the choice of highly configurable instance types, including memory, CPU, storage and operating system.

Amazon EC2: Create an Instance



Step 1: Choose an Amazon Machine Image (AMI)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Search for an AMI by entering a search term e.g. "Windows"

□ Free tier only

1 to 40 of 40 AMIs

Ubuntu Server 20.04 LTS (HVM), SSD Volume Type - ami-0e84e211558a022c0 (64-bit x86) / ami-031ecc7e9d5b9a82d (64-bit Arm)

Free tier eligible

Ubuntu Server 20.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-07c1207a9d40bc3bd (64-bit x86) / ami-0a5ee0336de62011b (64-bit Arm)

Free tier eligible

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)

Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-03ffa9b61e8d2cfda (64-bit x86) / ami-0f7b74fc93a4f54a6 (64-bit Arm)

Free tier eligible

Ubuntu Server 16.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)

Red Hat Enterprise Linux 8 (HVM), SSD Volume Type - ami-0a54aef4ef3b5f881 (64-bit x86) / ami-0ffd59b53e6797671 (64-bit Arm)

Red Hat
Free tier eligible

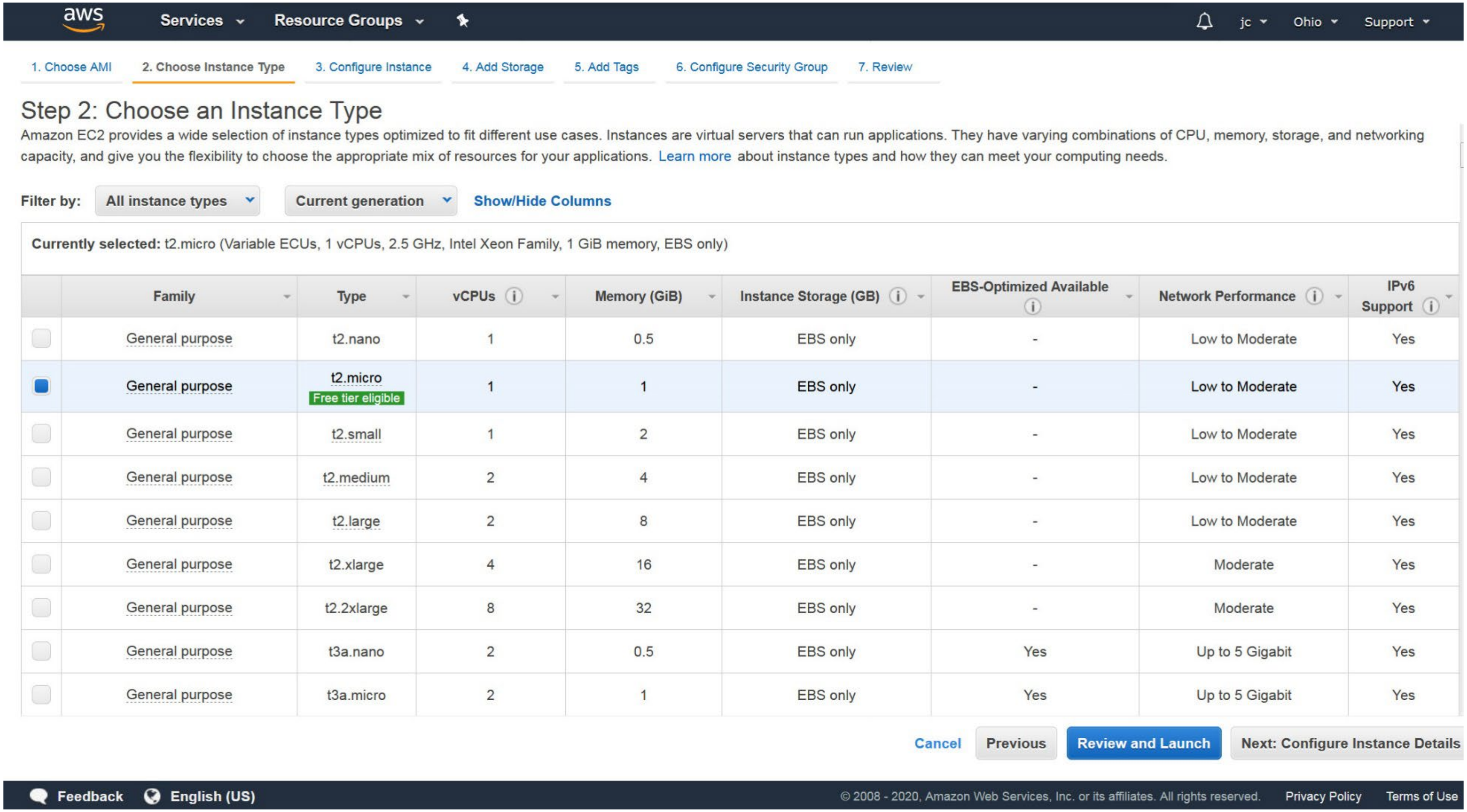
Red Hat Enterprise Linux version 8 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

● 64-bit (x86)
○ 64-bit (Arm)

© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon EC2: Choose Instance Type



Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. Learn more about instance types and how they can meet your computing needs.

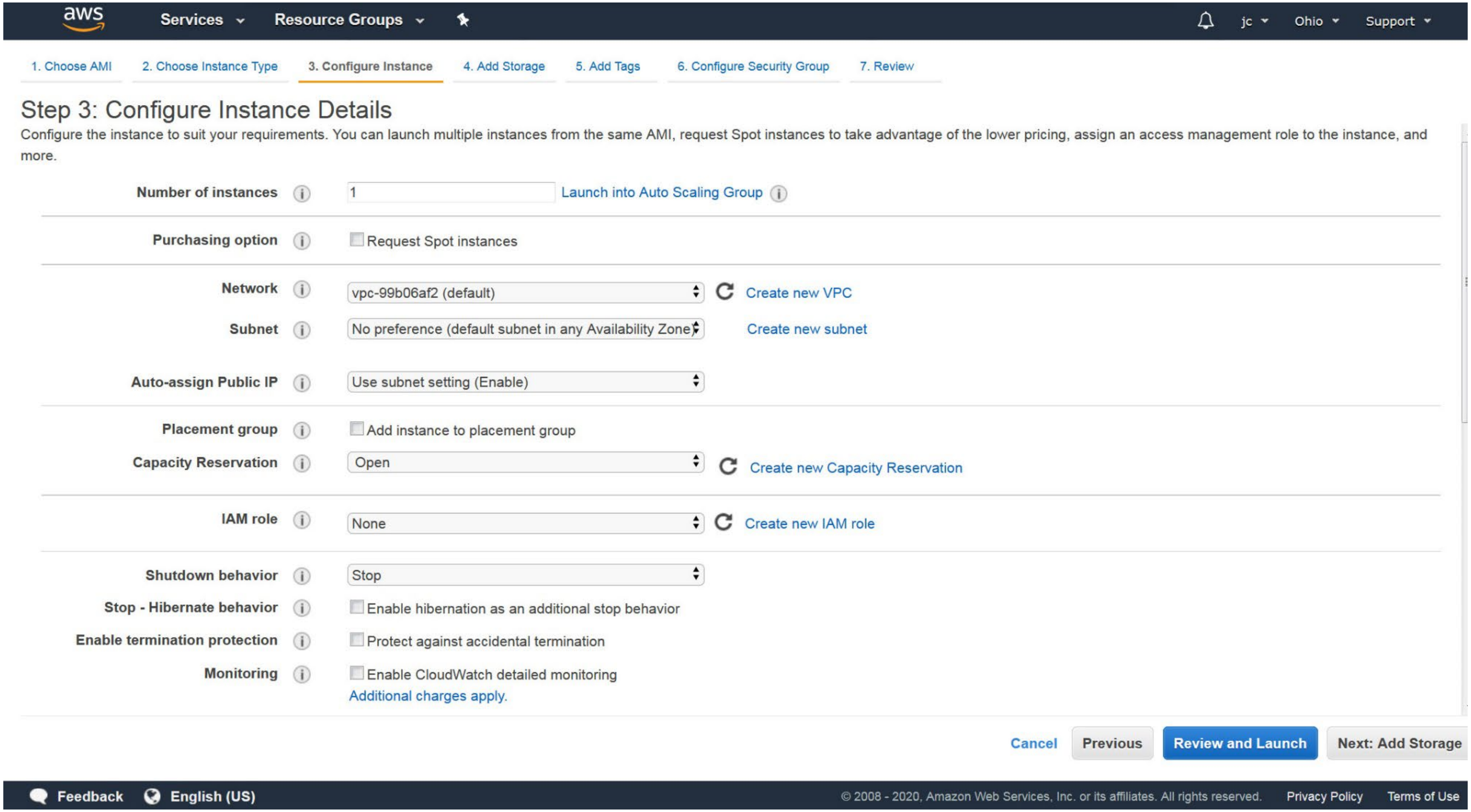
Filter by: All instance types | Current generation | Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
□	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
☑	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
□	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
□	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes
□	General purpose	t3a.nano	2	0.5	EBS only	Yes	Up to 5 Gigabit	Yes
□	General purpose	t3a.micro	2	1	EBS only	Yes	Up to 5 Gigabit	Yes

Cancel
Previous
Review and Launch
Next: Configure Instance Details
Feedback
English (US)
© 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

Amazon EC2: Configure Instance



Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 — Launch into Auto Scaling Group

Purchasing option: □ Request Spot instances

Network: vpc-99b06af2 (default) — Create new VPC

Subnet: No preference (default subnet in any Availability Zone) — Create new subnet

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: □ Add instance to placement group

Capacity Reservation: Open — Create new Capacity Reservation

IAM role: None — Create new IAM role

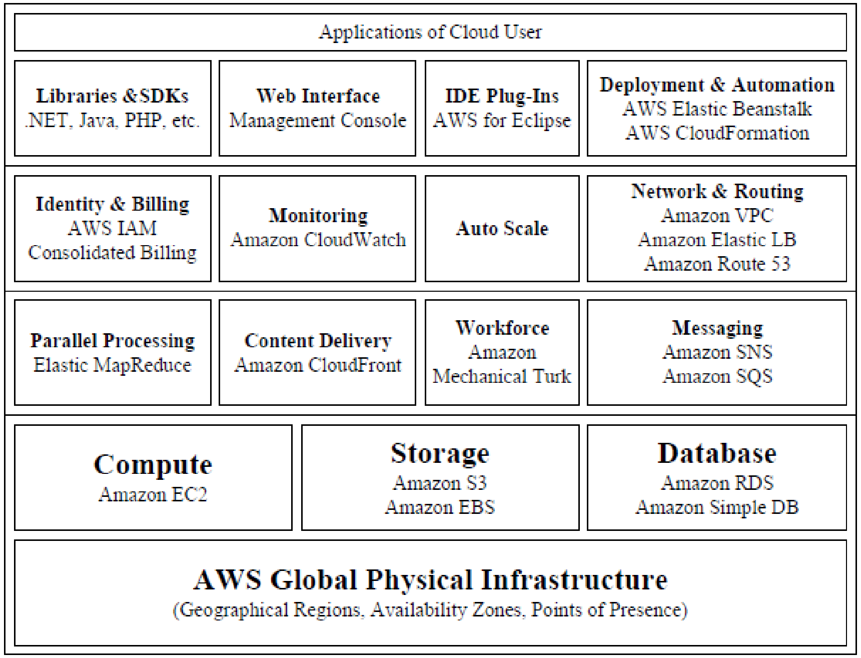
Shutdown behavior: Stop

Stop - Hibernate behavior: □ Enable hibernation as an additional stop behavior

Enable termination protection: □ Protect against accidental termination

Monitoring: □ Enable CloudWatch detailed monitoring
Additional charges apply.

Amazon Web Service (AWS)



Applications of Cloud User

Libraries &SDKs
.NET, Java, PHP, etc.

Web Interface
Management Console

IDE Plug-Ins
AWS for Eclipse

Deployment & Automation
AWS Elastic Beanstalk
AWS CloudFormation

Identity & Billing
AWS IAM
Consolidated Billing

Monitoring
Amazon CloudWatch

Auto Scale

Network & Routing
Amazon VPC
Amazon Elastic LB
Amazon Route 53

Parallel Processing
Elastic MapReduce

Content Delivery
Amazon CloudFront

Workforce
Amazon
Mechanical Turk

Messaging
Amazon SNS
Amazon SQS

Compute
Amazon EC2

Storage
Amazon S3
Amazon EBS

Database
Amazon RDS
Amazon Simple DB

AWS Global Physical Infrastructure
(Geographical Regions, Availability Zones, Points of Presence)

Fig. 18.5: Relations among the different components in Amazon Web Service (AWS).

Designed for use with other AWS modules, EC2 works seamlessly in conjunction with Amazon S3, and such other Amazon services as Relational Database Service (RDS), SimpleDB and Simple Queue Service (SQS) to provide a complete solution for computing, query processing and storage across a wide range of applications.

18.2 Multimedia Cloud Computing

- Distribution of multimedia content has always been a fundamental issue for media service and content providers.

- The large computational, storage and bandwidth capacity of the Cloud make it a very attractive platform for multimedia applications. Further, large public cloud often have many strategically geographically located data-centers allowing for more efficient distribution of content.

- Multimedia Cloud Computing and general purpose Cloud Computing share many commonalties. However, multimedia services are highly diverse, examples include voice over IP, video conferencing, and video streaming.

Modules and their relation in Multimedia Cloud Computing

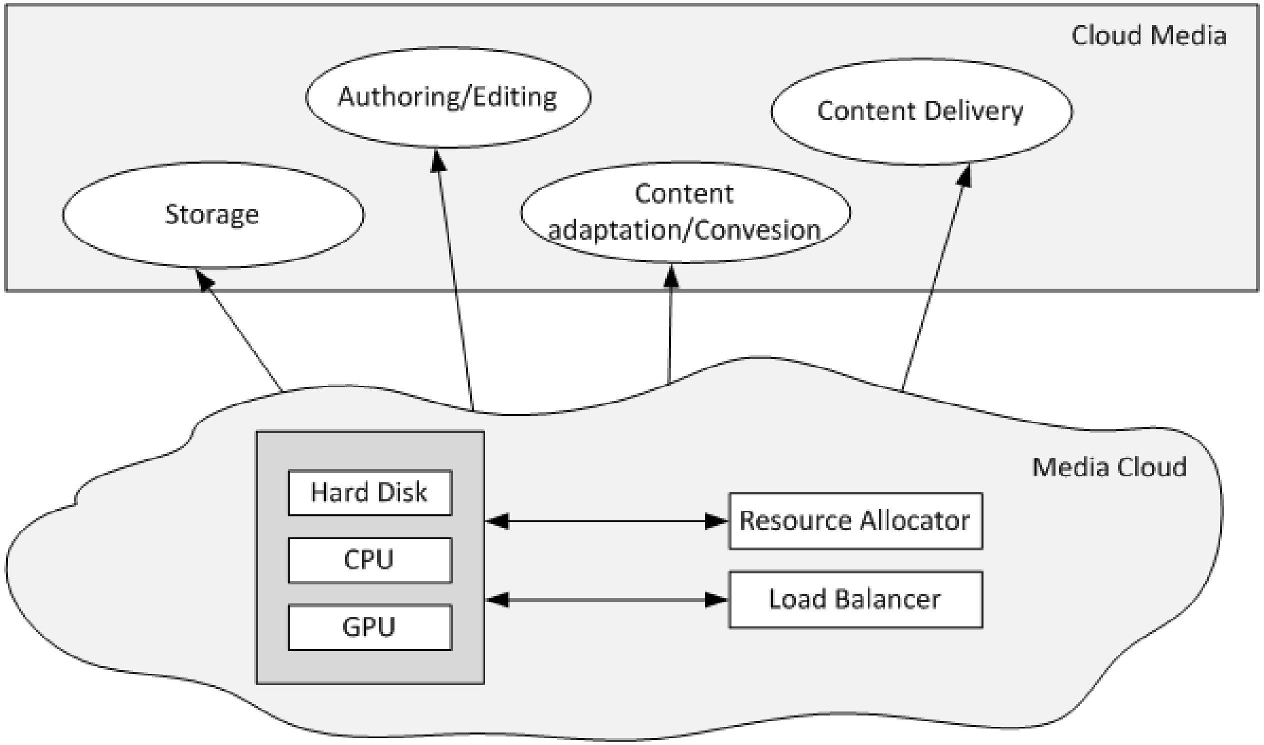


Fig 18.7: Modules and their relations in multimedia cloud computing

18.3 Multimedia Content Sharing over Cloud

- Media sharing and distribution services are evolving extremely quickly. New services and applications are being developed constantly, some become hugely successful like Youtube, many others fade into obscurity.

- Developers face a trade off in the early stage of any multimedia service.
  - How to provision enough resources to provide a high quality experience to their current users, as well as allowing room for the user base to grow, while not overextending themselves on costly infrastructure projects and overhead.

- The Cloud's "pay-as-you-go" service allows a service to start small and scale large without huge initial capital investments.

Generic Framework Media Streaming

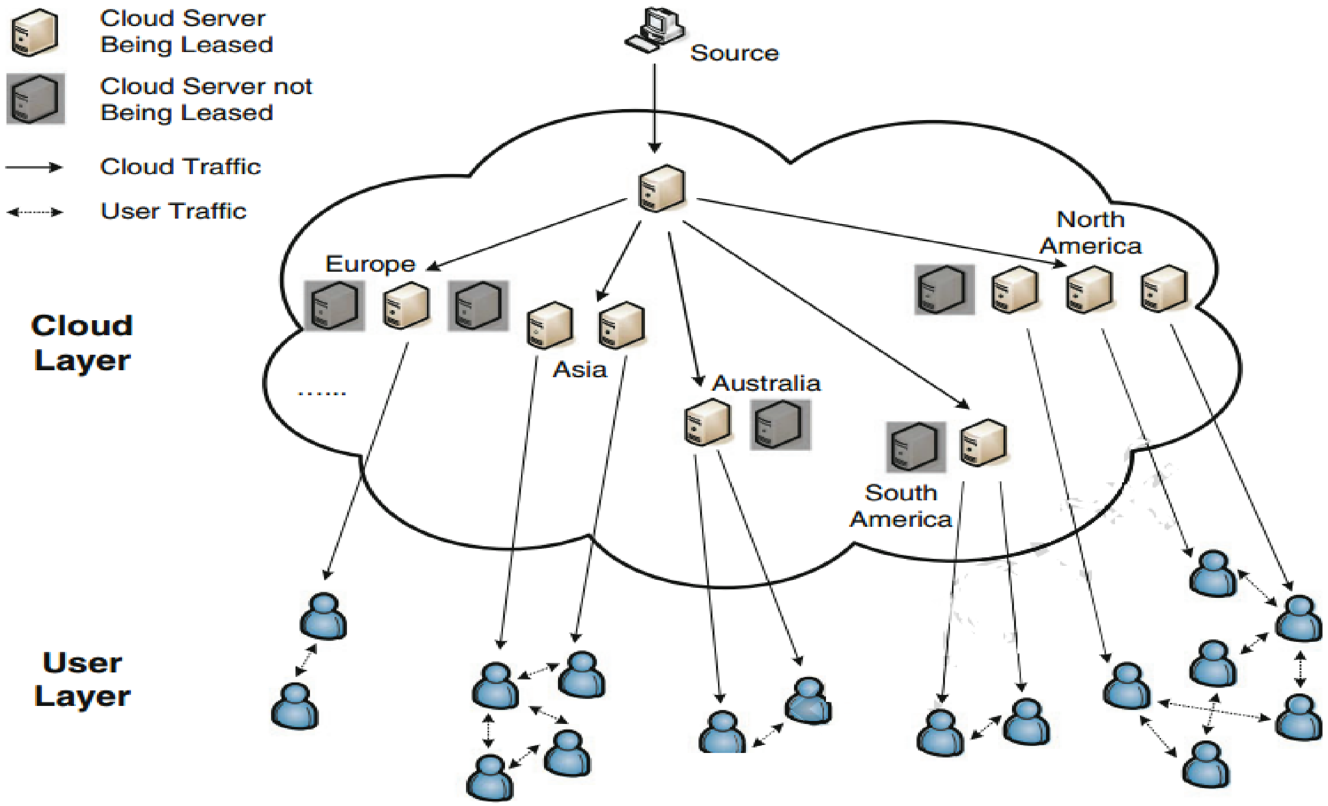


Fig 18.8: A generic framework for migrating live media streaming service to the cloud

Case Study: NetFlix

- Combing huge network traffic requirements and dynamic unpredictable demand bursts Netflix has become a prominent example of a business greatly enhanced by multimedia Cloud Computing.
- As of 2018, all of Netflix's media infrastructure has been moved to Amazon public cloud.
- Netflix leverages Amazon's cloud services to provide the following features:
  - **Content Conversion:** Using EC2 to convert master copies to over 50 different versions, with varying levels of video resolution, video quality and audio quality.
  - **Content Storage:** Utilizing S3 to store the many converted copies.
  - **Content Distribution:** Data is distributed from S3 to OpenConnect, the company's proprietary CDN, where the media is ultimately streamed to the end user.

Case Study: NetFlix

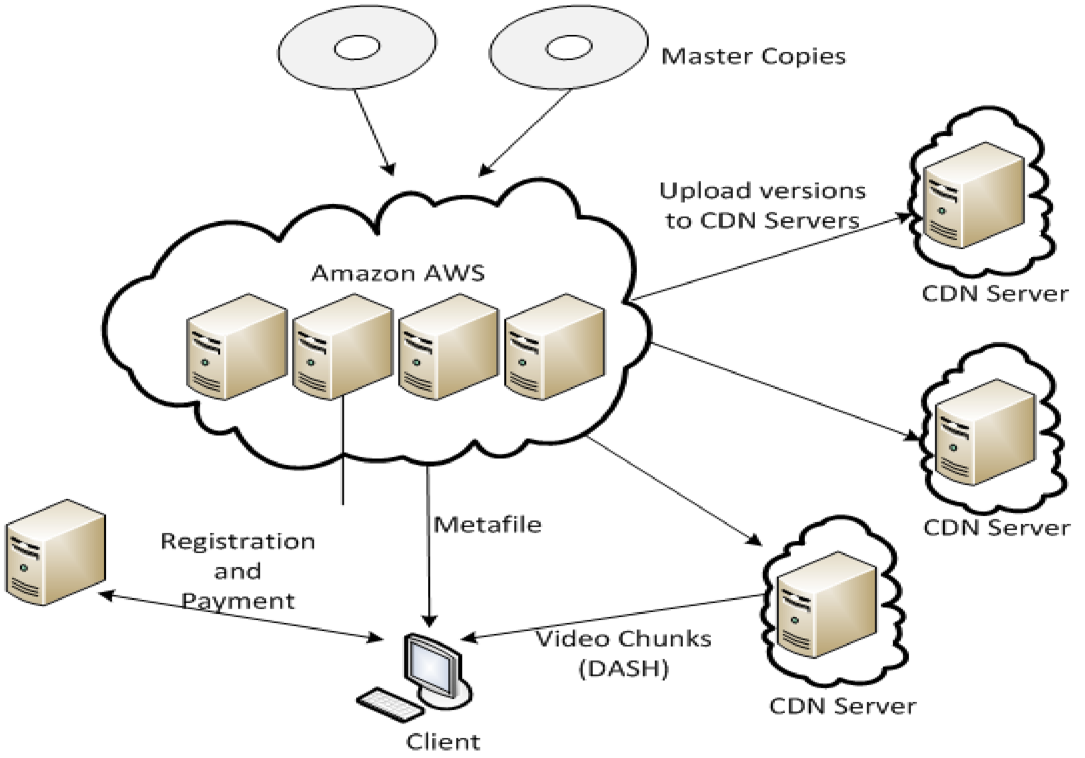


Fig 18.11: The cloud-based Netflix architecture during migration. Its own servers were finally eliminated by 2018.

Case Study: NetFlix

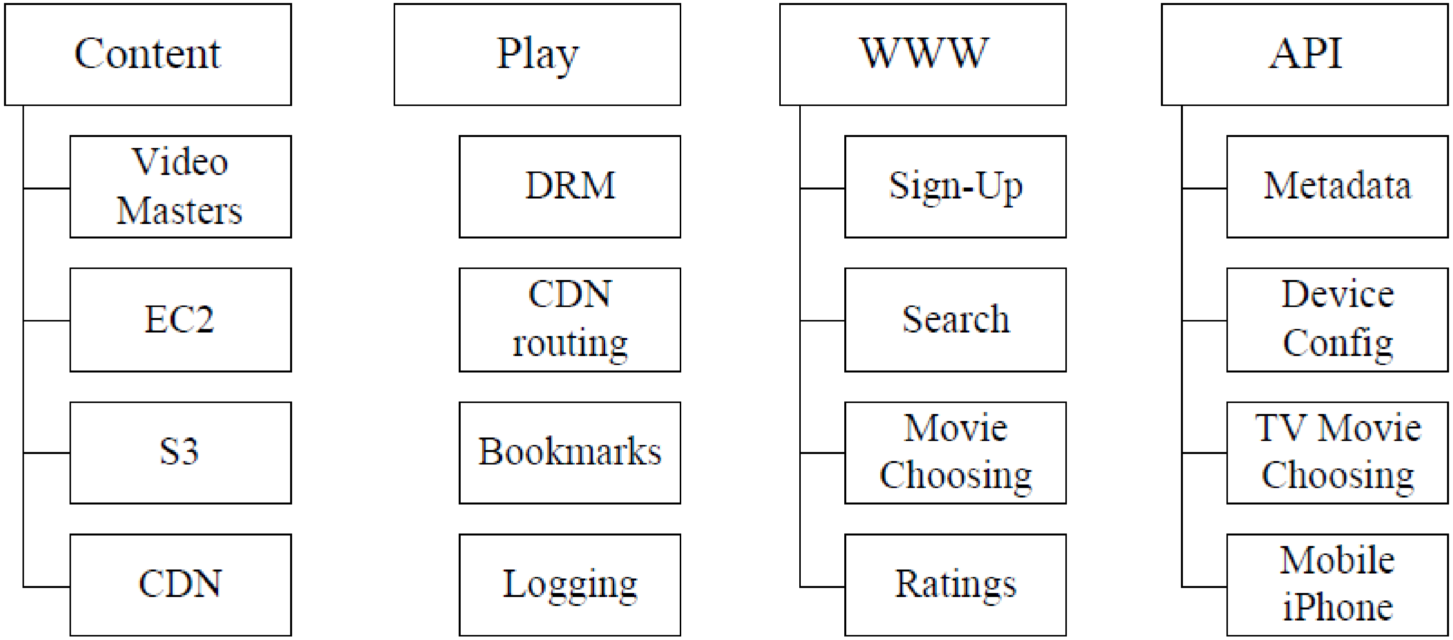


Fig. 18.12: The services integrated in Netflix and the cloud modules for them.

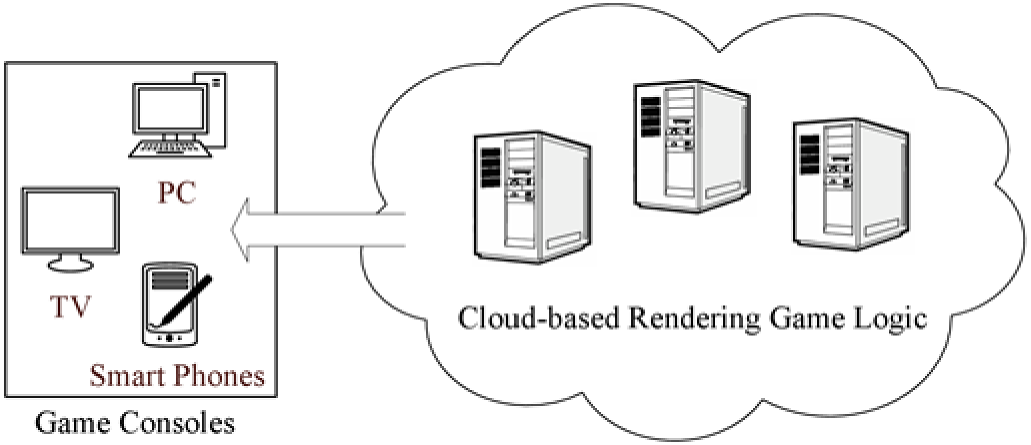
18.4 Multimedia Computation Offloading

- Many computation-intensive applications are now being migrated to the cloud.
  - Users can access high-performance servers and clusters as needed meaning they do not have to maintain expensive infrastructure.
  - Further, users on relatively weak computational platforms such as cell phones and tablets can offload some of their computational tasks to save battery and improve performance.
    - For example, Apple's Siri can offload some tasks to a remote cloud server if the computation would be too expensive to perform locally.

Requirements for Offloading

- Motivation for offloading.
  - to save energy locally, to improve computation performance, or both.
- Gain of offloading.
  - A profiling or breakdown analysis of the application is needed.
- Decision of offloading.
  - Offloading decision can be made statically or dynamically.

18.5 Interactive Cloud Gaming



- Cloud Gaming at its core uses thin-clients to interact with powerful cloud-based servers to render interactive games remotely and stream the images back.
- Many of today’s top interactive games require powerful desktop class components to provide an enjoyable gaming experience. Many relatively weaker devices such as tablets and smart phones do not meet the minimum requirements.

Cloud Gaming Challenges

- From ultra low latency live video streaming to high performance 3D processing, cloud gaming has overcome considerable challenges.

- In the time frame of only 100-200 milliseconds a cloud gaming system must collect a player's actions, transmit them to the cloud server, process the action, render the results, encode/compress the resulting changes to the game-world, and stream the video (game scenes) back to the player.
  - Very little time to employ video streaming buffers often found in other media streaming applications.

- Many optimizations and technologies must be used to lower the players interaction delay (delay between a player's action and resultant game scene).
  - GPU manufactures such as NVidia are developing cloud enable GPUs specifically targeted at cloud gaming.

Cloud Gaming Architecture Overview

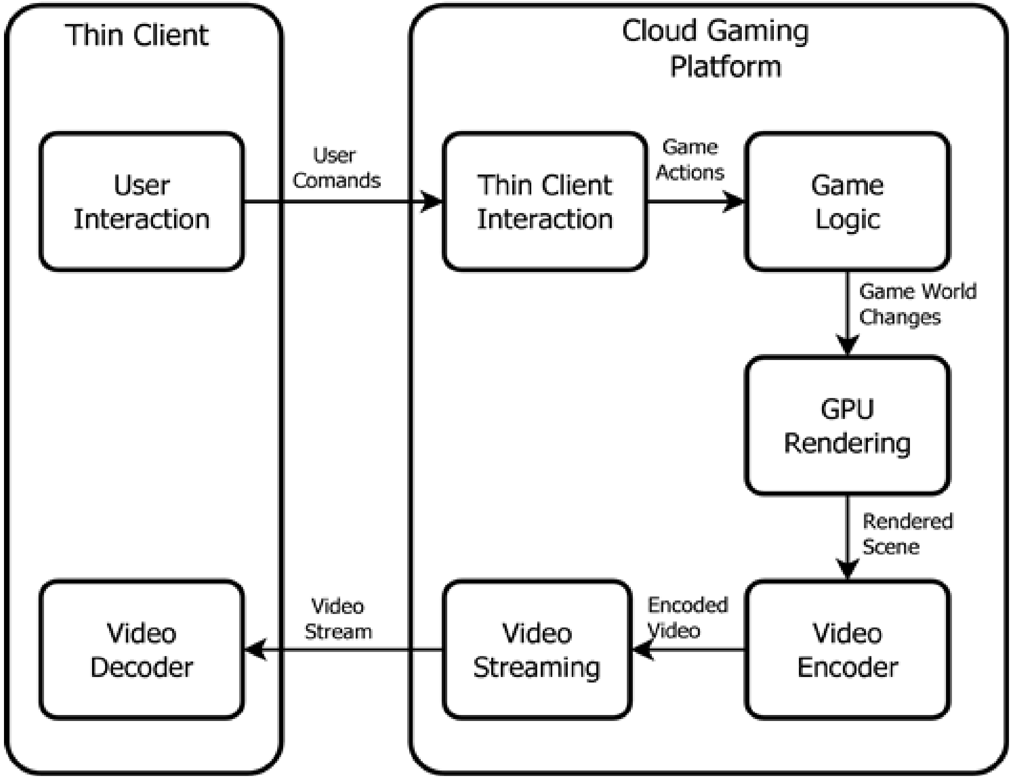


Fig 18.14: A generic framework of cloud gaming

Case Study: Gaikai

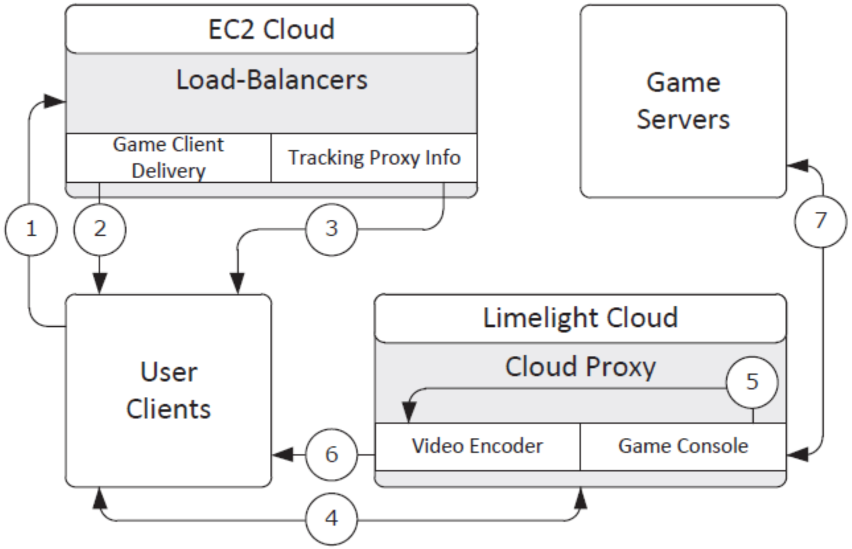


Fig. 18.15: The workflow of the Gaikai cloud gaming platform.

Gaikai is implemented using two public clouds, namely Amazon EC2 and Limelight.

When a user selects a game on Gaikai (*Step1* in Figure 18.15), an EC2 virtual machine will first deliver the Gaikai game client to the user (*Step2*).

After that, it forwards the IP addresses of game proxies that are ready to run the selected games to the user (*Step3*). The user will then select one game proxy to run the game (*Step4*). The game proxy starts to run the game and the game screens will be streamed to the user via UDP (*Step5* and *Step6*).

For multi-player games, these proxies also forward user operations to game servers (mostly deployed by the game companies) and send the related information/reactions back to the users (*Step7*).

Case Study: Onlive

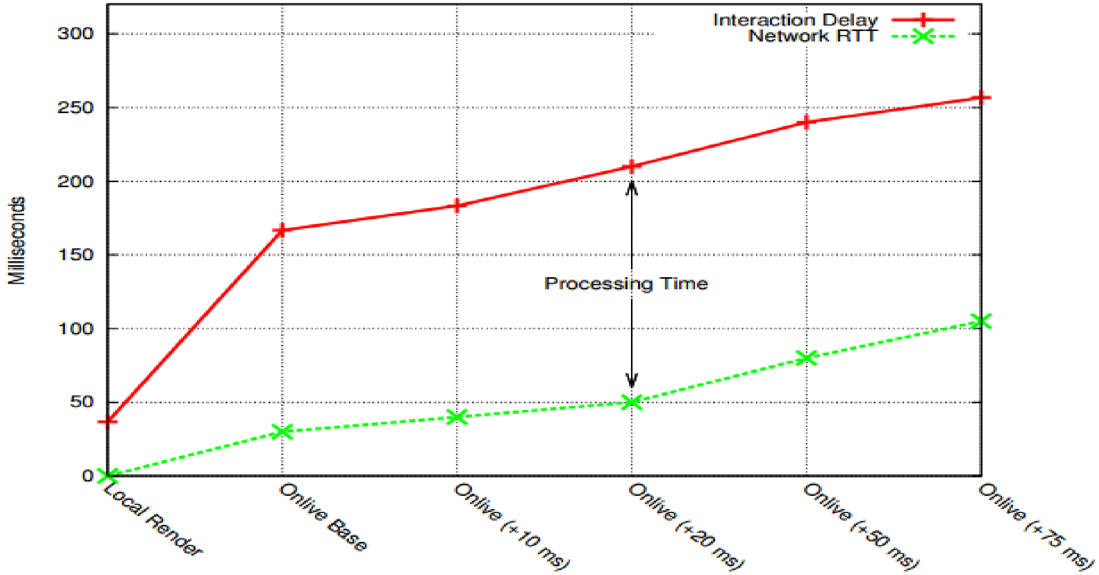
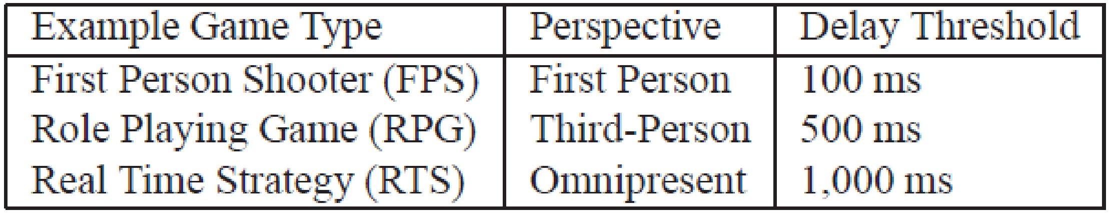


Fig 18.16: Interaction delay in Onlive

- Measurements of the Onlive cloud gaming service indicate that given good network conditions total interaction delay can be just over 150 milliseconds. A local rendering of the same game takes approximate 37 milliseconds to render the same action.

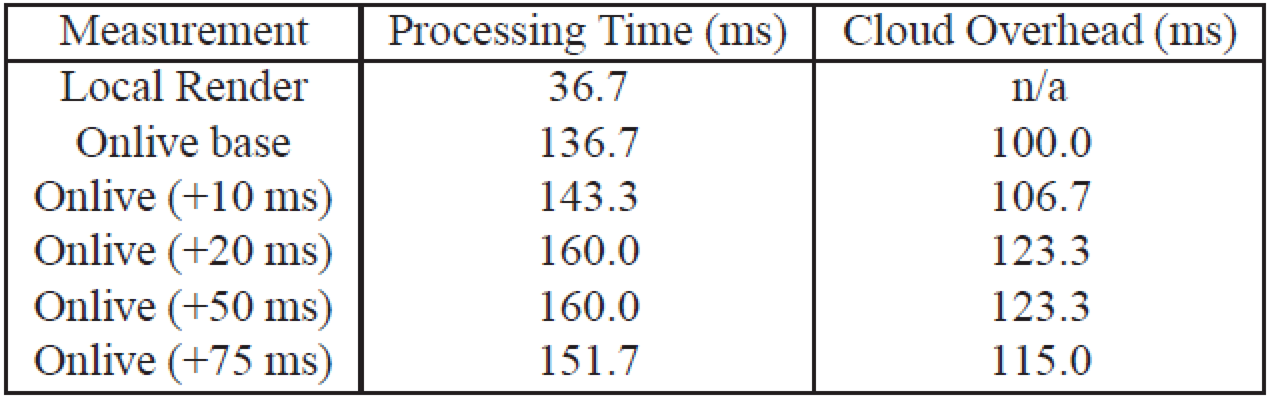
Case Study: Onlive

Table 18.2: Delay tolerance in traditional gaming.



Example Game Type	Perspective	Delay Threshold
First Person Shooter (FPS)	First Person	100 ms
Role Playing Game (RPG)	Third-Person	500 ms
Real Time Strategy (RTS)	Omnipresent	1,000 ms

Table 18.3: Processing time and cloud overhead



Measurement	Processing Time (ms)	Cloud Overhead (ms)
Local Render	36.7	n/a
Onlive base	136.7	100.0
Onlive (+10 ms)	143.3	106.7
Onlive (+20 ms)	160.0	123.3
Onlive (+50 ms)	160.0	123.3
Onlive (+75 ms)	151.7	115.0

18.6 Edge Computing and Serverless Computing for Multimedia

- For delay-sensitive multimedia applications, aggregating all the resources in a centralized data center is not ideal.
- Fine-grained resource partition and distribution is needed.
- Two directions:
  - Edge computing (on geo-distribution)
  - Serverless computing (on service abstraction).

18.6.1 Edge Computing

- Edge computing
  - A distributed computing paradigm that expands the cloud computing paradigm by bringing computation and data storage closer to the location where it is needed, i.e., network edge, so as to improve response times and save bandwidth.
  - Examples: AWS IoT Greengrass, CloudFront, Azure IoT Edge, Akamai Intelligent Edge.

- Key Modules
  - **Service module**, which includes containers that run native or third-party services locally at the edge node;
  - **Runtime**, which runs and manages the service modules deployed to the node;
  - **Cloud interface**, which offers the connection, interaction and collaboration with the cloud data center.

Mobile Edge Computing

- Mobile Edge Computing
  - Or Multi-access Edge Computing (MEC)
  - Initially presented by European Telecommunications Standards Institute (ETSI) in 2015.
  - Bring storage and computing capability into cellular base stations.

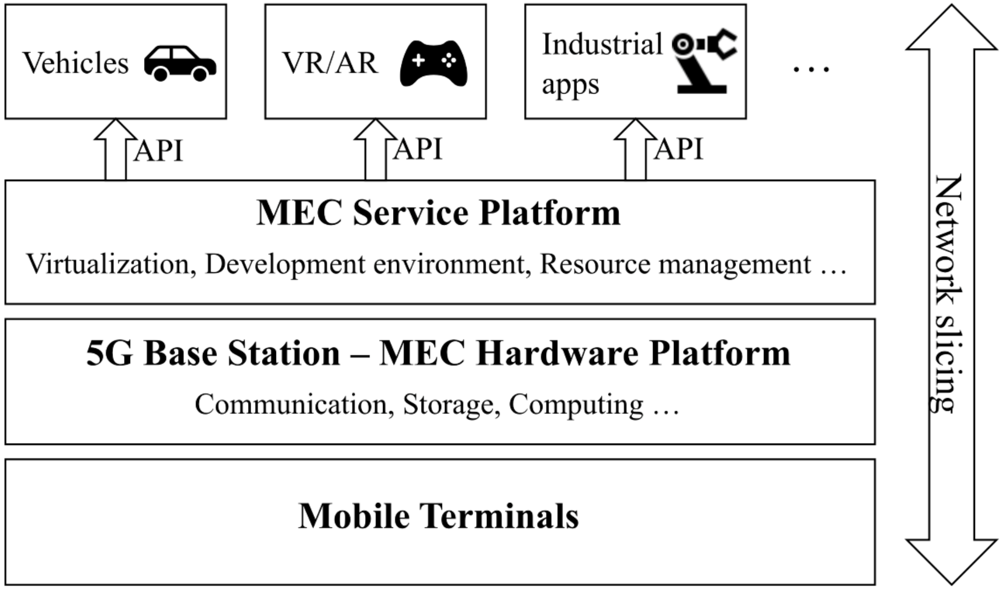

Vehicles
VR/AR
Industrial apps
…
API
API
API
MEC Service Platform
Virtualization, Development environment, Resource management …
5G Base Station – MEC Hardware Platform
Communication, Storage, Computing …
Mobile Terminals
Network slicing

Fig. 18.17: An architectural view of mobile edge computing (MEC)

Mobile Edge Computing

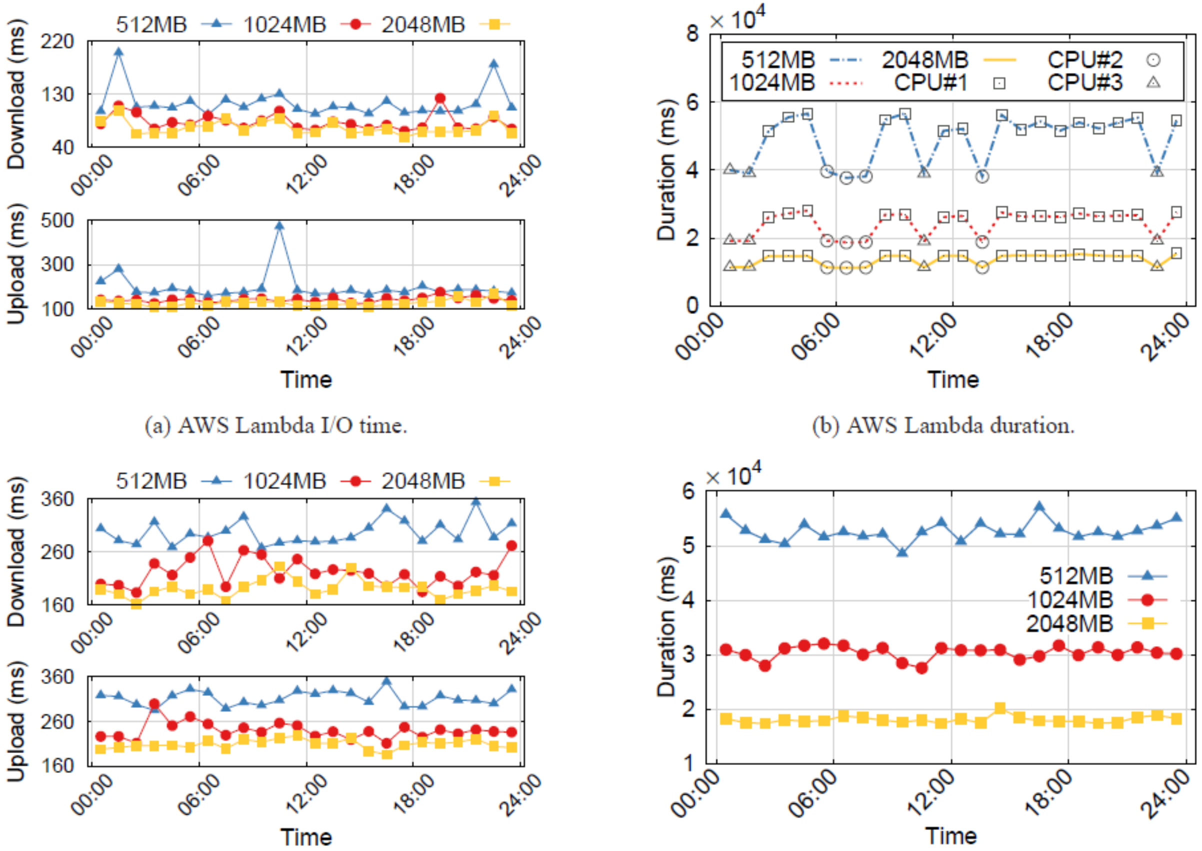
- Enabling technology of 5G
- URLLC (Ultra-reliable and low-latency communications) cases
  - With the assistance of an edge server, Augmented Reality (AR) will be accelerated and timely feedback will help with modify a live representation of the world.
  - Video streaming services will operate much more efficiently given the awareness of local network conditions.
  - For smart transportation, connected and automated cars can efficiently and reliably coordinate with other cars nearby in realtime, in particular, within the coverage of the same base station.
- mMTC (massive Machine Type Communications) cases
  - The tiny connected devices in smart home and smart city can more easily access the computing resources at the edge than from the remote cloud data center, in terms of speed, cost, and reliability.

18.6.2 Serverless Computing

- Offloads the task of server provisioning and management from developers to platforms.
- Developers only need to break up application codes into a collection of stateless functions and set events to trigger their executions.
- Platforms are responsible for handling every trigger and scaling precisely with the size of workloads.

- Examples: AWS Lambda, Google Cloud Functions (GCF), and Apache OpenWhisk

Transcoding with Serverless Computing



(a) AWS Lambda I/O time.

(b) AWS Lambda duration.

(c) GCF I/O time.

(d) GCF duration.

Fig. 18.18: Changes in execution time for the transcoding function in one day.

Transcoding with Serverless Computing

- Increasing the memory size from 1,024MB to 2,048MB does not improve the download and upload latencies much
  - For larger memory sizes, CPU power is not the bottleneck for I/O tasks.
- As the memory size decreases, the execution duration changes more and more dramatically
  - The longest execution duration of 512MB (56,640ms) is 1.5 times as much as the shortest execution duration (37,735ms), and the increment is even larger than the average execution duration of 2,048MB (13,754ms)
- The influence of the heterogeneous underlying infrastructures on performance cannot be ignored

Transcoding Function Execution Duration

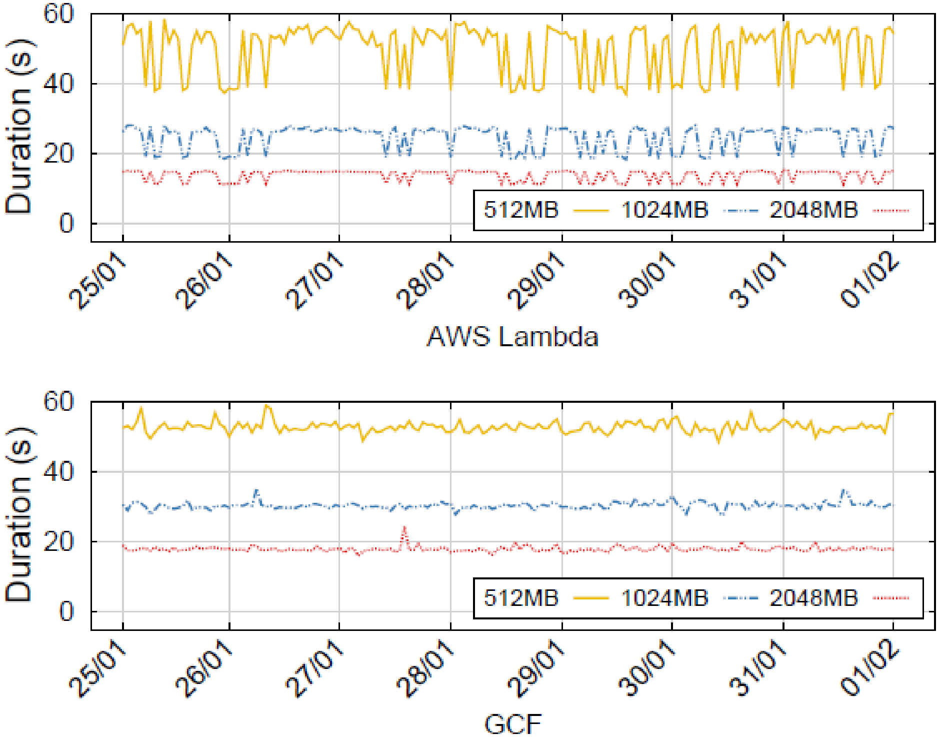


Fig. 18.19: Transcoding function execution duration changes within one week on different serverless platforms.

Transcoding Function Execution Duration

- For AWS Lambda, the performance shows more drastic changes as the memory size decreases.
  - The changes do not show a daily periodic pattern but are closely related to the heterogeneous underlying infrastructures.
  - This implies that the scheduling and allocation of VMs are random and independent of time.
- In contrast, GCF exhibits a more stable execution duration regardless of the memory size.

Further Exploration

- Cloud and edge computing remains a new field for both industry and research community.

- Many of the related materials can be found as white papers from such major cloud and edge computing providers as Amazon, Google, and Microsoft.