NPTEL ONLINE CERTIFICATION COURSES

Cloud Computing

Prof. Soumya K Ghosh
Department of Computer Science and Engineering

Module 11: Cloud Computing Paradigms
Lecture 51: Dew Computing

CONCEPTS COVERED

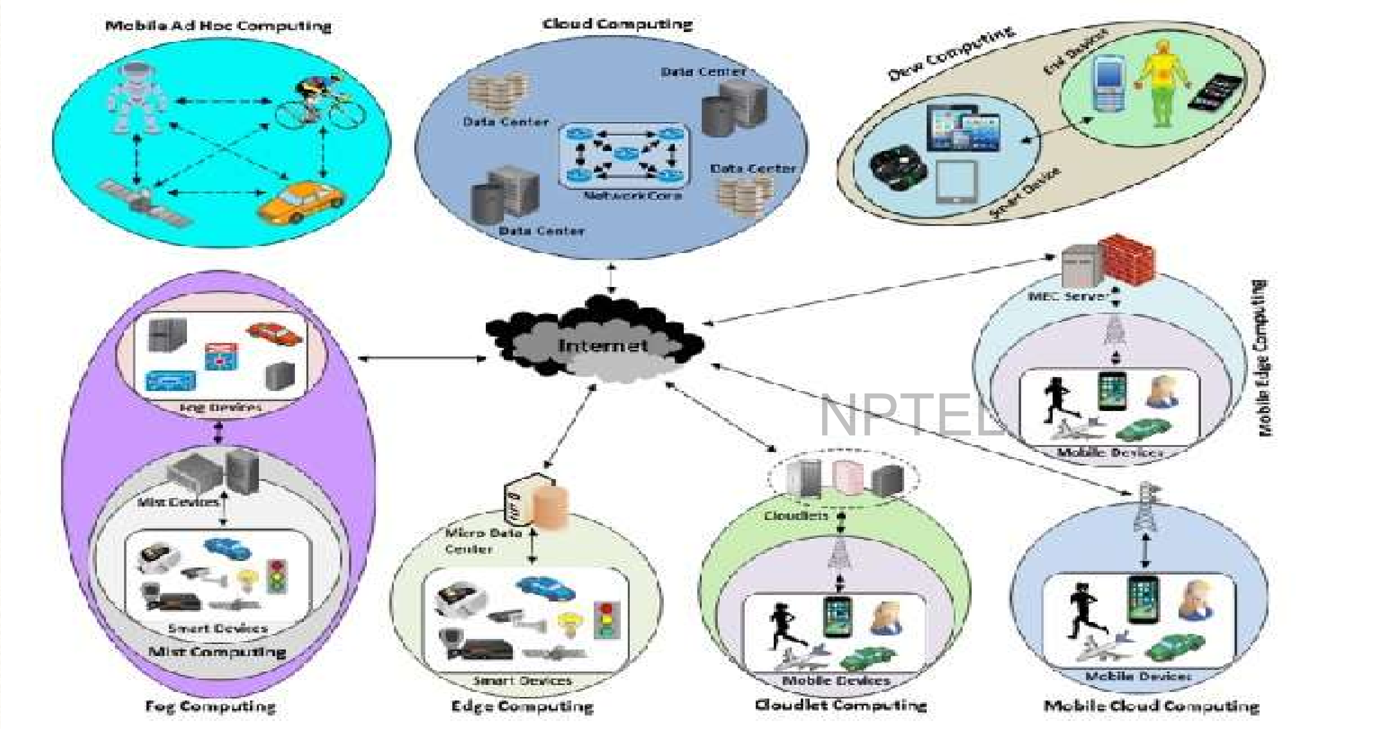
- Dew Computing – Overview
- Dew Computing – Features
- Dew Computing – Architecture
- Dew Computing – Applications

KEYWORDS

- Dew Computing

Dew Computing

Cloud Computing "Family"



Ref: Sunday Oyinlola Ogundoyin, Ismaila Adeniyi Kamil, Optimization techniques and applications in fog computing: An exhaustive survey, Swarm and Evolutionary Computation, Elsevier

Dew Computing (DC)

- Dew computing is a computing paradigm that combines the core concept of cloud computing with the capabilities of end devices (personal computers, mobile phones, etc.).
- It is used to enhance the experience for the end user in comparison to only using cloud computing.
- Dew computing attempts to solve one of the major problems related to cloud computing, such as reliance on internet access.

Dew Computing (DC)

- Dew Computing is a computing model for enabling ubiquitous, pervasive, and convenient ready-to-go, plug-in facility empowered personal network that includes Single-Super-Hybrid-Peer P2P communication link.
- Primary goal: To access a pool of raw data equipped with meta-data that can be rapidly created, edited, stored, and deleted with minimal internetwork management effort (i.e. offline mode).
- To utilise all functionalities of cloud computing , Network users are heavily dependent on *Internet Connectivity* all the time.

Dew Computing (DC)

- Dew computing (DC) is a new paradigm where user-centric, flexible, and personalized-supported applications are prioritized. It is located very close to the end devices and it is the first in the IoT-fog-cloud continuum.
- DC is a micro-service-based computing paradigm with vertically distributed hierarchy.
- DC comprises smart devices, such as smart-phones, smart-watches, tablets, etc., located at the edge of the network to connect with the end devices, collect and process the IoT sensed data, and offer other services.
- The services in DC are relatively available and it is not mandatory to have a permanent Internet connection.
- DC is micro-service based which means that it does not depend on any centralized server or cloud data center.
- DC does not rely on the centralized computing devices nor a permanent Internet connection.

DC – Typical Example

Dropbox is an example of the dew computing paradigm, as it provides access to the files and folders in the cloud in addition to keeping copies on local devices.

Allows the user to access files during times without an internet connection; when a connection is established again, files and folders are synchronized back to the cloud server.

Ref: https://www.dropbox.com

Dew Computing - Features

- Key features of dew computing are *independence* and *collaboration*.
- Independence means that the local device must be able to provide service without a continuous connection to the Internet.
- Collaboration means that the application must be able to connect to the cloud service and synchronize data when appropriate.

- The word "dew" reflects natural phenomena: clouds are far from the ground, fog is closer to the ground, and dew is on the ground. Analogically, cloud computing is a remote service, fog computing is beside the user, and dew computing is at the user end.

Dew Service Models and Typical Applications

Infrastructure-As-Dew

The local device is dynamically supported by cloud services.

Software-In-Dew

The configuration and ownership of software are saved in the cloud.

Platform-In-Dew

A software development suite must be installed on the local device with the settings and application data synchronized to the cloud service.

iCloud

Google Play App Store

GitHub

Dew Service Models and Typical Applications

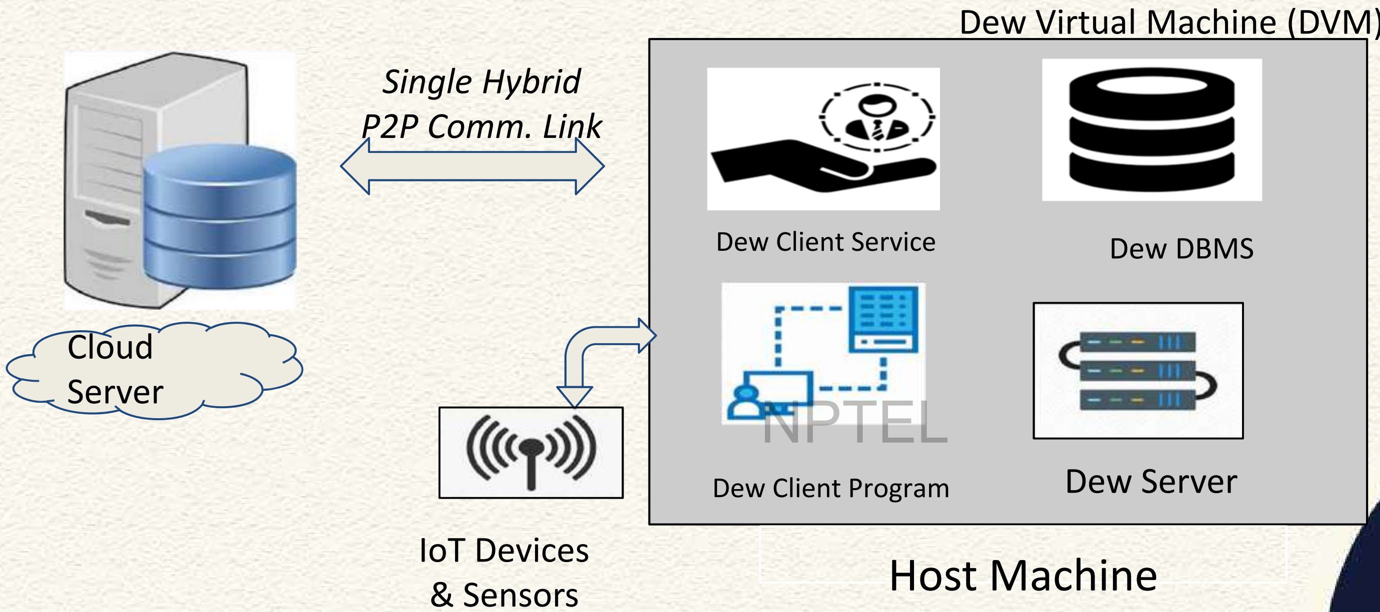
Storage-In-Dew

The storage of the local device is partially or fully copied into the cloud.

Web-in-Dew

The local device must possess a duplicated fraction of the World Wide Web (WWW).

Dew Computing Architecture



- *To establish a cloud-dew architecture on a local machine, a dew virtual machine (DVM) is needed. The DVM is an isolated environment for executing the dew server on the local system*

DC - Architecture

Dew Server functions:

- to serve user with requested services
- to perform synchronization and correlate between local data and remote data

Attempt to achieve three goals:

- Data Replication
- Data Distribution
- Synchronization

DC – Application Areas

- **Web in Dew** (WiD) - Possess a duplicated fraction of the World Wide Web (WWW) or a modified copy of that fraction to satisfy the independence feature. Because this fraction synchronizes with the web, it satisfies the collaboration feature of dew computing.
- **Storage in Dew** (SiD) The storage of the local device is partially or fully copied into the cloud. Since the user can access files at any time without the need for constant Internet access, this category meets the independence feature of dew computing. SiD also meets the collaboration feature because the folder and its contents automatically synchronize with the cloud service.
- **Database in Dew** (DBiD): The local device and the cloud both store copies of the same database. One of these two databases is considered the main version and can be defined as such by the database administrator. This service increases the reliability of a database, as one of the databases can act as the backup for the other.

DC – Application Areas

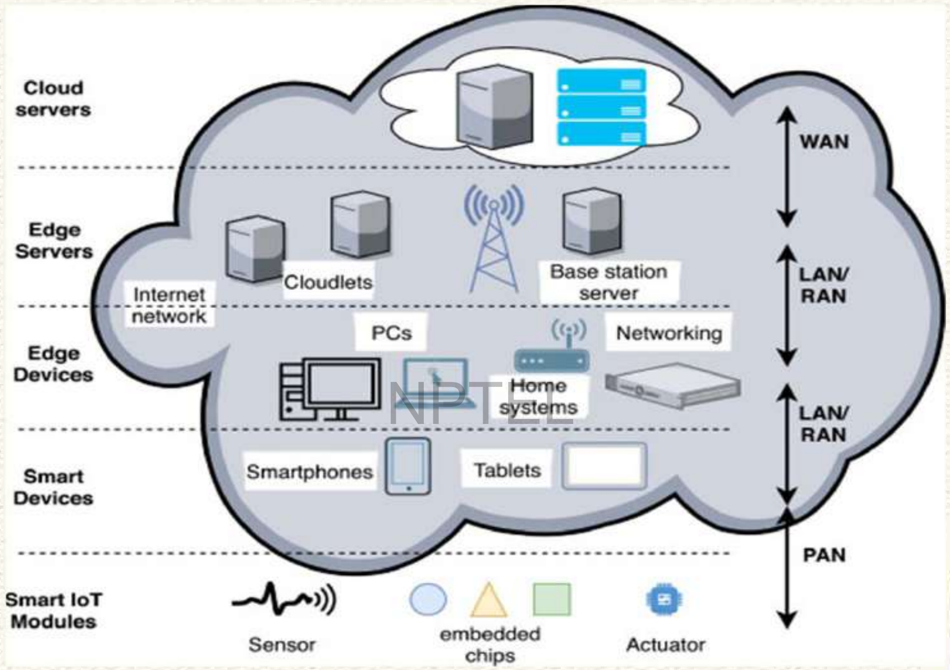
- **Software in Dew** (SiD): The configuration and ownership of software are saved in the cloud. Examples include the Apple App Store and Google Play, where the applications the user installs are saved to their account and can then be installed on any device linked to their account.
- **Platform in Dew** (PiD): A software development suite must be installed on the local device with the settings and application data synchronized to the cloud service. It must be able to synchronize development data, system deployment data, and online backups. Example: GitHub.
- **Infrastructure as Dew** (IaD): The local device is dynamically supported by cloud services. IaD can come in different forms, but the following two forms can be used: (1) the local device can have an exact duplicate DVM instance in the cloud, which is always kept in the same state as the local instance, or (2) the local device can have all its settings/data saved in the cloud, including system settings/data and data for each application

DC – Challenges

- Power Management
- Processor Utility
- Data Storage
- Viability of Operating System
- Programming Principles
- Database Security

Cloud Computing and Dew

Dew-enabled
Computing



REFERENCES

- https://en.wikipedia.org/wiki/Dew_computing
- Wang, Yingwei (2016). "Definition and Categorization of Dew Computing". Open Journal of Cloud Computing. 3 (1). ISSN 2199-1987.
- "Dew Computing and Transition of Internet Computing Paradigms" - ZTE Corporation
- Yingwei, Wang (2015). "The initial definition of dew computing". Dew Computing Research.
- Ray, Partha Pratim (2018). "An Introduction to Dew Computing: Definition, Concept and Implications - IEEE Journals & Magazine". IEEE Access. 6: 723–737. doi:10.1109/ACCESS.2017.2775042.
- Sunday Oyinlola Ogundoyin, Ismaila Adeniyi Kamil, Optimization techniques and applications in fog computing: An exhaustive survey, Swarm and Evolutionary Computation, Elsevier, Volume 66, 2021, https://doi.org/10.1016/j.swevo.2021.100937


Thank you

NPTEL ONLINE CERTIFICATION COURSES

Cloud Computing

Prof. Soumya K Ghosh
Department of Computer Science and Engineering

Module 11: Cloud Computing Paradigms
Lecture 52: Serverless Computing - I

CONCEPTS COVERED

- Serverless Computing
- Function-as-a-Service

KEYWORDS

- Serverless Computing
- Function-as-a-Service

Serverless Computing - I

NPTEL

Serverless Computing

- Serverless computing is a method of providing backend services on an as-used basis. A serverless provider allows users to write and deploy code without the hassle of worrying about the underlying infrastructure.
- Serverless architecture simplifies the code deployment and eliminates the need for system administration, allowing developers to focus on the core logic without creating additional overhead by instantiating resources, such as VMs or containers in the monitoring infrastructure.

Serverless Computing

- In this model, developers execute their logic in the form of *functions* and submit to the cloud provider to run the task in a shared runtime environment; cloud providers manage the scalability needs of the *function*, by running multiple functions in parallel.
- Following the wide scale application of the containerization approach, the cloud services have adapted to offer better-fitting containers that require less time to load (boot) and to provide increased automation in handling (orchestration) containers on behalf of the client.
- Serverless computing promises to achieve full automation in managing fine-grained containers.

Serverless Computing

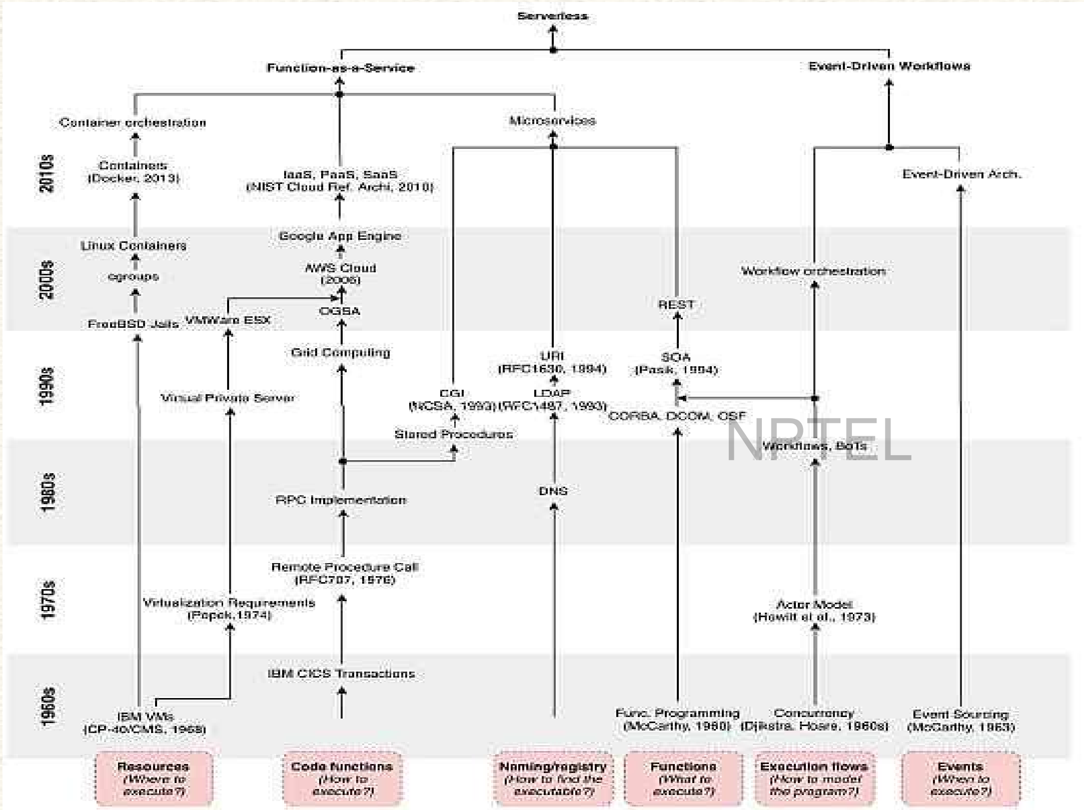
- *"Serverless computing is a form of cloud computing that allows users to run event-driven and granular applications, without having to address the operational logic"*
- Serverless as a computing abstraction: With serverless, developers focus on high-level abstractions (e.g., functions, queries, and events) and build applications that infrastructure operators map to concrete resources and supporting services.
- Developers focusing on the business logic and on ways to interconnect elements of business logic into complex workflows.
- Service providers ensure that the serverless applications are orchestrated—that is, containerized, deployed, provisioned, and available on demand—while billing the user for only the resources used.

Function-as-a-Service

- Clients of serverless computing can use the *function-as-a-service* (FaaS) model
- *Function as a service (FaaS) is a form of serverless computing in which the cloud provider manages the resources, lifecycle, and event-driven execution of user-provided functions.*
- With FaaS, users provide small, stateless functions to the cloud provider, which manages all the operational aspects to run these functions.
- For example, consider the *ExCamera* application, which uses cloud functions and workflows to edit, transform, and encode videos with low latency and cost.

  [Ref: S. Fouladi et al., “Encoding, fast and slow: Low-latency video processing using thousands of tiny threads,” Proceedings of the 14th USENIX Conference on Networked Systems Design and Implementation (NSDI 17), 2017, pp. 363–376]
- A majority of the tasks in these operations can be executed concurrently, allowing the application to improve its performance through parallelizing these tasks.

Evolution of Serverless



Serverless Computing

- In serverless, the cloud provider dynamically allocates and provisions servers.
- The code is executed in almost-stateless containers that are event-triggered, and ephemeral (may last for one invocation).
- Serverless covers a wide range of technologies, that can be grouped into two categories:
  – Backend-as-a-Service (BaaS)
  – Functions-as-a-Service (FaaS)

Backend-as-a-Service (BaaS)

- BaaS enables to replace server-side components with off-the-shelf services.
- BaaS enables developers to outsource all the aspects behind a scene of an application so that developers can choose to write and maintain all application logic in the frontend.
- Typical examples: remote authentication systems, database management, cloud storage, and hosting.
- Google Firebase, a fully managed database that can be directly used from an application.
- In this case, Firebase (the BaaS services) manage data components on the user's behalf.

Function-as-a-Service (FaaS)

- Serverless applications are event-driven cloud-based systems where application development relies solely on a combination of third-party services, client-side logic, and cloud-hosted remote procedure calls.
- FaaS allows developers to deploy code that, upon being triggered, is executed in an isolated environment.
- Each function typically describe a small part of an entire application. The execution time of functions is typically limited.
- Functions are not constantly active. Instead, the FaaS platforms listen for events that instantiate the functions.
- Thus, functions are triggered by events, such as client requests, events produced by any external systems, data streams, or others.
- FaaS provider is then responsible to horizontally scale function executions in response to the number of incoming events.

Serverless Computing - Challenges

- **Asynchronous calls**:
  - Asynchronous calls to and between Serverless Functions increase complexity of the system. Usually remote API calls follow request response model and are easier to implemented with synchronous calls.
- **Functions calling other functions**
  - Complex debugging, loose isolation of features. Extra costs if functions are called synchronously as we need to pay for two functions running at the same time.
- **Shared code between functions**
  - Might break existing Serverless Functions that depend on the shared code that is changed. Risk to hit the image size limit (50MB in AWS Lambda), warmup-time (the bigger the image, the longer it takes to start).

Serverless Computing - Challenges

- **Usage of too many libraries**
  - Increased space used by the libraries increase the risk to hit the image size limit and increase the warmup-time.
- **Adoption of too many technologies**
  - such as libraries, frameworks, languages.
  - Adds maintenance complexity and increases skill requirements for people working within the project.
- **Too many functions**
  - Creation of functions without reusing the existing one. Non-active Serverless Functions doesn't cost anything so there is temptation to create new functions instead of altering existing functionality to match changed requirements.
  - Decreased maintainability and lower system understandability.

Serverless Computing – Major Providers

Service Provider	Virtual Servers	Function	Database	Storage
aws	Instances	λ	amazon DynamoDB	S3
Google Cloud	VMs	Google Cloud Functions	Google Cloud Datastore	CLOUD STORAGE
Microsoft Azure	VM Instances	AzureFunctions	Azure Cosmos DB	S3 Azure Blob Storage

REFERENCES

- Sanghyun Hong and Abhinav Srivastava and William Shambrook and Tudor Dumitras, Go Serverless: Securing Cloud via Serverless Design Patterns, 10th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 18), 2018, https://www.usenix.org/conference/hotcloud18/presentation/hong
- E. van Eyk, L. Toader, S. Talluri, L. Versluis, A. Uță and A. Iosup, "Serverless is More: From PaaS to Present Cloud Computing," in IEEE Internet Computing, vol. 22, no. 5, pp. 8-17, Sep./Oct. 2018, doi: 10.1109/MIC.2018.053681358.
- J. Nupponen and D. Taibi, "Serverless: What it Is, What to Do and What Not to Do," 2020 IEEE International Conference on Software Architecture Companion (ICSA-C), 2020, pp. 49-50, doi: 10.1109/ICSA-C50368.2020.00016.


Thank you

NPTEL ONLINE CERTIFICATION COURSES

Cloud Computing

Prof. Soumya K Ghosh

Department of Computer Science and Engineering

Module 11: Cloud Computing Paradigms

Lecture 53: Serverless Computing - II

CONCEPTS COVERED

- Serverless Computing
- AWS Lambda
- Google Cloud Functions
- Azure Functions

KEYWORDS

- Serverless Computing
- AWS Lambda
- Google Cloud Functions
- Azure Functions

Serverless Computing - II

NPTEL

Serverless Computing

- Serverless computing hides the servers by providing programming abstractions for application builders that simplify cloud development, making cloud software easier to write.
- The focus/ target of Cloud Computing was system administrators and the Serverless is programmers. This change requires cloud providers to take over many of the operational responsibilities needed to run applications.

Serverless Computing

- To emphasize the change of focus from servers to applications, this new paradigm is known as serverless computing, although remote servers are still the invisible backend that powers it.
- This next phase of cloud computing will change the way programmers work as dramatically as the Cloud Computing has changed how operators work.
- Thus, Serverless applications are ones that don't need any server provision and do not require to manage servers.

Serverless Computing – Major Providers

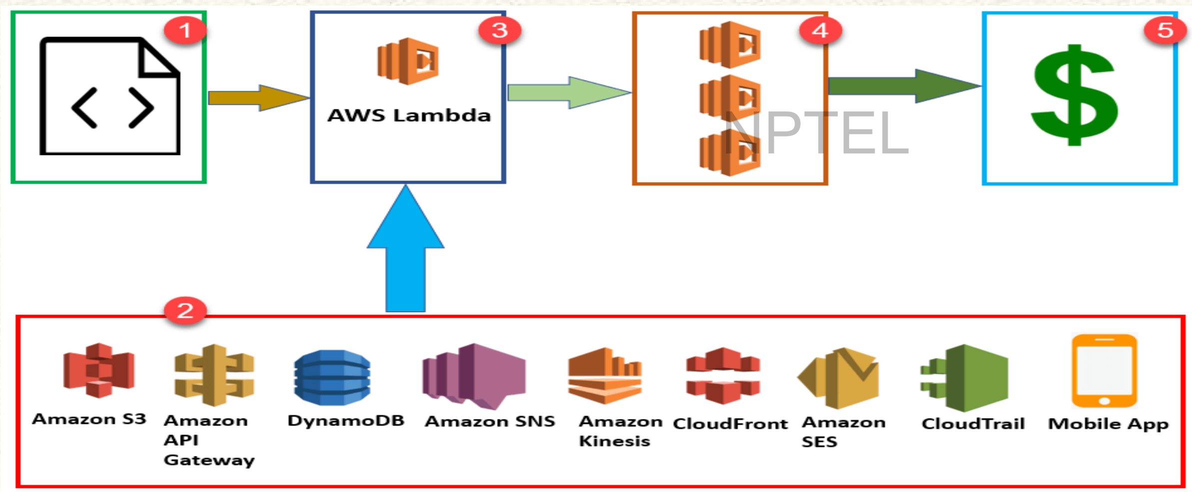
Service Provider	Virtual Servers	Function	Database	Storage
aws	**Instances**	λ	amazon DynamoDB	S3
Google Cloud	**VMs**	Google Cloud Functions	Google Cloud Datastore	CLOUD STORAGE
Microsoft Azure	**VM Instances**	AzureFunctions	Azure Cosmos DB	S3 Azure Blob Storage

AWS Lambda

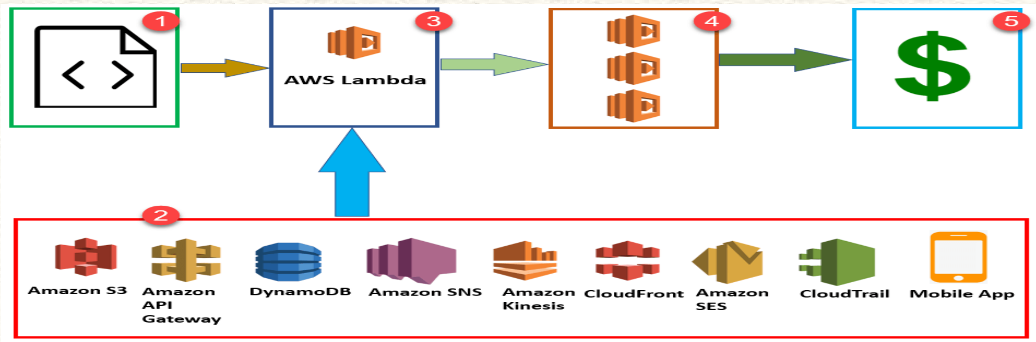
- AWS Lambda is an event-driven, serverless computing platform provided by Amazon as a part of Amazon Web Services.
- Thus one need to worry about which AWS resources to launch, or how to manage them. Instead, you need to put the code on Lambda, and it runs.
- In AWS Lambda the code is executed based on the response of events in AWS services such as add/delete files in S3 bucket, HTTP request from Amazon API gateway, etc.
- However, Amazon Lambda can only be used to execute background tasks.

AWS Lambda

- AWS Lambda function helps you to focus on your core product and business logic instead of managing operating system (OS) access control, OS patching, right-sizing, provisioning, scaling, etc.
- AWS Lambda Block Diagram:



AWS Lambda



1) First upload your AWS Lambda code in any language supported by AWS Lambda. Java, Python, Go, and C# are some of the languages that are supported by AWS Lambda function.
2) These are some AWS services which allow you to trigger AWS Lambda.
3) AWS Lambda helps you to upload code and the event details on which it should be triggered.
4) Executes AWS Lambda Code when it is triggered by AWS services
5) AWS charges only when the AWS lambda code executes, and not otherwise.

AWS Lambda Concepts

- **Function:** A function is a program or a script which runs in AWS Lambda. Lambda passes invocation events into your function, which processes an event and returns its response.
- **Runtimes:** Runtime allows functions in various languages which runs on the same base execution environment. This helps in configuring your function in runtime. It also matches your selected programming language.
- **Event source:** An event source is an AWS service, such as Amazon SNS (Simple Notification Service), or a custom service. This triggers function helps you to executes its logic.
- **Lambda Layers:** Lambda layers are an important distribution mechanism for libraries, custom runtimes, and other important function dependencies.
- **Log streams:** Log stream allows you to annotate your function code with custom logging statements which helps you to analyse the execution flow and performance of your AWS Lambda functions.

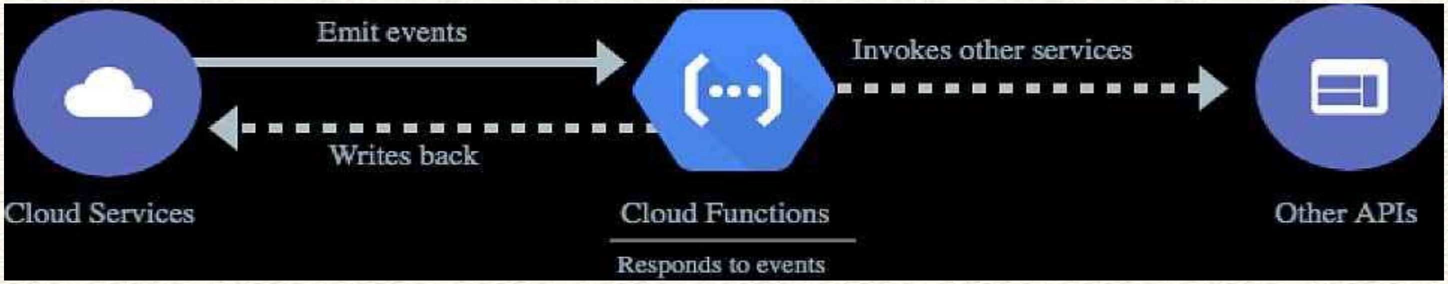
AWS Lambda – How to execute code

1) AWS Lambda URL: https://aws.amazon.com/lambda/
2) Create an Account or use Existing Account
   Edit the code & Click Run…
   1. Edit the code
   2. Click Run
3) Check output

Google Cloud Functions

- Google Cloud Functions is a **serverless** execution environment for building and connecting cloud services.
- With Cloud Functions you write simple, **single-purpose functions** that are **attached to events emitted from your cloud infrastructure and services**.
- Cloud Function is **triggered when an event being watched is fired**.
- The **code executes in a fully managed environment**. There is **no need to provision any infrastructure** or worry about managing any servers.

Google Cloud Functions - Working



Cloud Services. This is the Google Cloud Platform and its various services. Services like: Google Cloud Storage, Google Cloud Pub/Sub, Stackdriver, Cloud Datastore, etc. They all have events that happen inside of them. For e.g. if a bucket has got a new object uploaded into it, deleted from it or metadata has been updated.

Google Cloud Functions - Working

Cloud Functions:

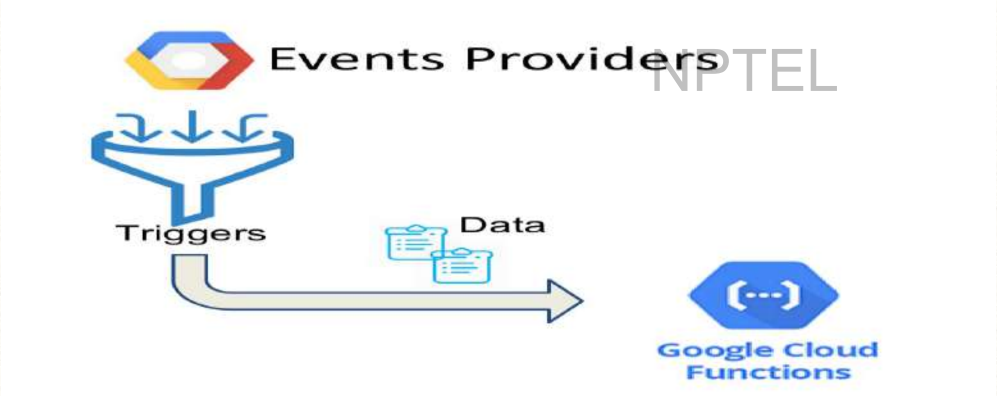
- Say an event (e.g. Object uploaded into a Bucket in Cloud Storage happens) is generated or fired or emitted. The Event data associated with that event has information on that event.
- If the Cloud Function is configured to be triggered by that event, then the Cloud Function is invoked or run or executed.
- As part of its execution, the event data is passed to it, so that it can decipher what has caused the event i.e. the event source, get meta information on the event and so on and do its processing.
- As part of the processing, it might also (maybe) invoke other APIs. (Google APIs or external APIs).
- It could even write back to the Cloud Services.

Google Cloud Functions - Working

- When it has finished executing its logic, the Cloud Function mentions or specifies that it is done.
- Multiple Event occurrences will result in multiple invocations of your Cloud Functions. This is all handled for you by the Cloud Functions infrastructure. You focus on your logic inside the function and be a good citizen by keeping your function single purpose, use minimal execution time and indicate early enough that you are done and don't end up in a timeout.
- This should also indicate to you that this model works best in a stateless fashion and hence you cannot depend on any state that was created as part of an earlier invocation of your function. You could maintain state outside of this framework

Google Cloud Functions - Events, Triggers

- Events : They occur in Google Cloud Platform Services E.g. File Uploaded to Storage, a Message published to a queue, Direct HTTP Invocation, etc.
- Triggers : You can chose to respond to events via a Trigger. A Trigger is the event + data associated the event.
- Event Data : This is the data that is passed to your Cloud Function when the event trigger results in your function execution.



Google Cloud Functions –Event Providers

- HTTP — invoke functions directly via HTTP requests
- Cloud Storage
- Cloud Pub/Sub
- Firebase (DB, Analytics, Auth)
- Stackdriver Logging
- Cloud Firestore
- Google Compute Engine
- BigQuery

Azure Functions

- Azure Functions is a serverless solution that allows you to write less code, maintain less infrastructure, and save on costs. Instead of worrying about deploying and maintaining servers, the cloud infrastructure provides all the up-to-date resources needed to keep your applications running.
- User focuses on the pieces of code, and Azure Functions handles the rest.
- A function is the primary concept in Azure Functions.
- A function contains two important pieces - your code, which can be written in a variety of languages, and some configurations, the function.json file.
- For compiled languages, this config file is generated automatically from annotations in your code. For scripting languages, you must provide the config file yourself.

Azure Functions – Build your Functions

Options and resources :

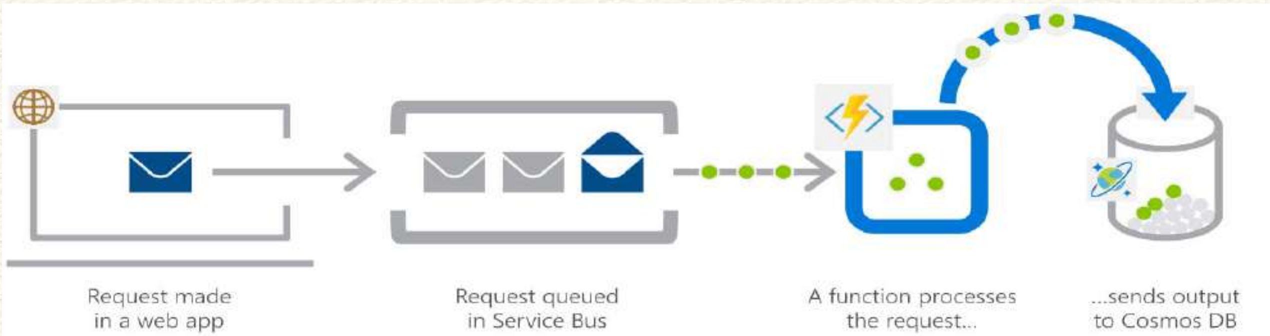
- **Use your preferred language**: Write functions in C#, Java, JavaScript, PowerShell, or Python, or use a custom handler to use virtually any other language.
- **Automate deployment**: From a tools-based approach to using external pipelines, there's a myriad of deployment options available.
- **Troubleshoot a function**: Use monitoring tools and testing strategies to gain insights into your apps.
- **Flexible pricing options**: With the Consumption plan, you only pay while your functions are running, while the Premium and App Service plans offer features for specialized needs.

Azure Functions

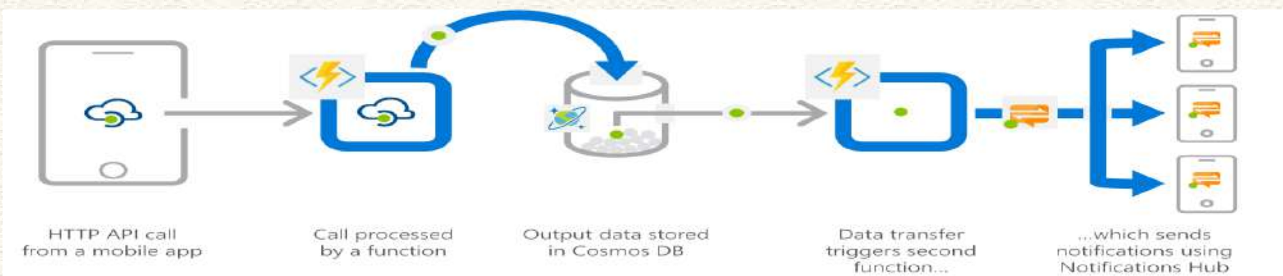
Common serverless architecture patterns include:

- Serverless APIs, mobile and web backends.
- Event and stream processing, Internet of Things (IoT) data processing, big data and machine learning pipelines.
- Integration and enterprise service bus to connect line-of-business systems, publish and subscribe (Pub/Sub) to business events.
- Automation and digital transformation and process automation.
- Middleware, software-as-a-Service (SaaS) like Dynamics, and big data projects.

Azure Functions - Scenarios



Web application backend: Retail scenario



Mobile application backend: Financial services scenario

REFERENCES

- Johann Schleier-Smith, Vikram Sreekanti, Anurag Khandelwal, Joao Carreira, Neeraja J. Yadwadkar, Raluca Ada Popa, Joseph E. Gonzalez, Ion Stoica, and David A. Patterson. 2021. What serverless computing is and should become: the next phase of cloud computing. Commun. ACM 64, 5 (May 2021), 76–84. DOI:https://doi.org/10.1145/3406011
- AWS Lambda: https://aws.amazon.com/lambda/
- AWS Lambda: https://www.guru99.com/aws-lambda-function.html
- Google Cloud Functions: https://cloud.google.com/functions
- Google Cloud Functions: https://iromin.medium.com/google-cloud-functions-tutorial-what-is-google-cloud-functions-8796fa07fc7a
- Azure Functions: https://docs.microsoft.com/en-us/azure/azure-functions/
- Azure Functions: https://azure.microsoft.com/en-in/services/functions/


Thank you

NPTEL ONLINE CERTIFICATION COURSES

Cloud Computing

Prof. Soumya K Ghosh

Department of Computer Science and Engineering

Module 11: Cloud Computing Paradigms

Lecture 54: Sustainable Cloud Computing - I

CONCEPTS COVERED

- Sustainable Computing
- Sustainable Cloud Computing

KEYWORDS

- Sustainable Cloud Computing
- Cloud Data Centre
- Energy Management
- Carbon Footprint

Sustainable Cloud Computing - I

NPTEL

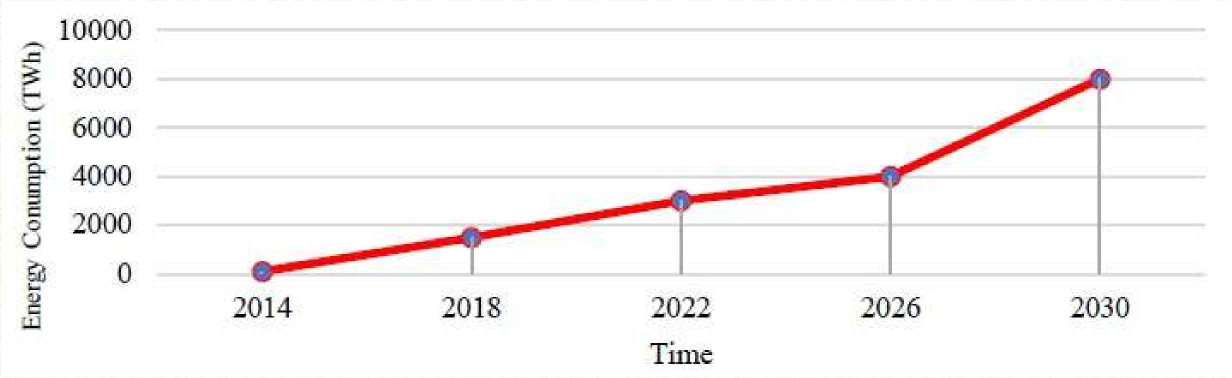
Sustainable Cloud Computing

- Cloud Service Providers (CSPs) rely heavily on the Cloud Data Centers (CDCs) to support the ever-increasing demand for their computational and application services.
- The financial and carbon footprint related costs of running such large infrastructure negatively impacts the sustainability of cloud services.
- Focus on minimizing the energy consumption and carbon footprints and ensuring reliability of the CDCs – goal of Sustainable Cloud Computing

Sustainable Cloud Computing

- Cloud computing paradigm offers on-demand, subscription-oriented services over the Internet to host applications and process user workloads.
- To ensure the availability and reliability of the services, the components of Cloud Data Centers (CDCs), such as network devices, storage devices and servers are to be made available round-the-clock.
- However, creating, processing, and storing each bit of data adds to the energy cost, increases carbon footprints, and further impacts the environment.

Sustainable Cloud Computing



Energy Consumption in Cloud Datacenters

- Amount of energy consumed by the CDCs is increasing regularly and it is expected to be 8000 Tera Watt hours (TWh) by 2030

Ref: (1) Rajkumar Buyya and Sukhpal Singh Gill. “Sustainable Cloud Computing: Foundations and Future Directions.” Business Technology & Digital Transformation Strategies, Cutter Consortium, Vol. 21, no. 6, Pages 1-9, 2018; (2) Anders SG Andrae, and Tomas Edler. “On global electricity usage of communication technology: trends to 2030.” Challenges, vol. 6, no. 1, pp. 117-157, 2015.

Sustainable Cloud Computing

- Components (networks, storage, memory and cooling systems) of CDCs are consuming huge amount of energy.
- To improve energy efficiency of CDC, there is a need for energy-aware resource management technique for management of all the resources (including servers, storage, memory, networks, and cooling systems) in a holistic manner.
- Due to the under-loading/ over-loading of infrastructure resources, the energy consumption in CDCs is not efficient; in fact, most of the energy is consumed while some resources (i.e., networks, storage, memory, processor) are in idle state, increasing the overall cost of cloud services.

Sustainable Cloud Computing

- CSPs are finding other alternative ways to reduce carbon footprints of their CDCs
- Major CSPs (like Google, Amazon, Microsoft and IBM) are planning to power their datacenters using renewable energy sources.
- Future CDCs are required to provide cloud services with minimum emissions of carbon footprints and heat release in the form of greenhouse gas emissions.

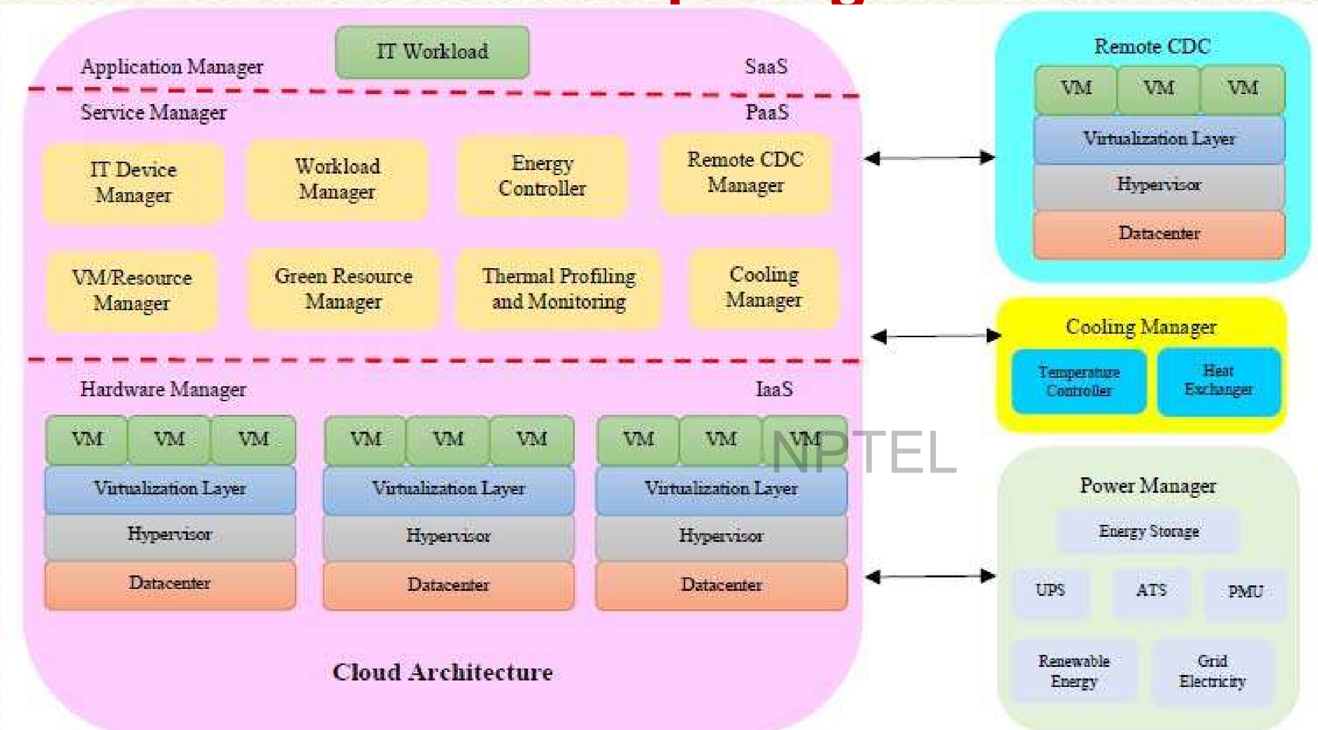
Sustainable Cloud Computing

To enable sustainable cloud computing, datacenters can be relocated based on:

- opportunities for waste heat recovery
- accessibility of green resource and
- proximity of free cooling resources

- To resolve these issues and substantially reduce energy consumption of CDCs, there is a need for cloud computing architectures that can provide sustainable cloud services through holistic management of resources.

Sustainable Cloud Computing



Sustainable Cloud Computing - A Conceptual Model

Ref: Rajkumar Buyya and Sukhpal Singh Gill. “Sustainable Cloud Computing: Foundations and Future Directions.” Business Technology & Digital Transformation Strategies, Cutter Consortium, Vol. 21, no. 6, Pages 1-9, 2018;

Sustainable Cloud Computing – Conceptual Model

Conceptual model for sustainable cloud computing in the form of layered architecture, which offers holistic management of cloud computing resources, to make cloud services more energy-efficient and sustainable.

- **Cloud Architecture**: This component is divided into three different sub-components: Software as a Service, Platform as a Service and Infrastructure as a Service.
- **Cooling Manager**: Thermal alerts will be generated if temperature is higher than the threshold value and heat controller will take an action to control the temperature with minimal impact on the performance of the CDC.

Ref: Rajkumar Buyya and Sukhpal Singh Gill. "Sustainable Cloud Computing: Foundations and Future Directions." Business Technology & Digital Transformation Strategies, Cutter Consortium, Vol. 21, no. 6, Pages 1-9, 2018;

Sustainable Cloud Computing

- **Power Manager**: It controls the power generated from renewable energy resources and fossil fuels (grid electricity). If there is execution of deadline oriented workloads, then grid energy can be used to maintain the reliability of cloud services. Automatic Transfer Switch (ATS) is used to manage the energy coming from both sources (renewable energy and grid electricity). Further, Power Distribution Unit is used to transfer the electricity to all the CDCs and cooling devices.

- **Remote CDC**: VMs and workloads can be migrated to a remote CDC to balance the load effectively.

Reliability and Sustainability - Issues

Energy

- To reduce energy consumption of cloud datacenter
- To reduce under loading and overloading of resources which improves load balancing
- To minimize heat concentration and dissipation in cloud datacenter
- To reduce carbon footprints to make environment more eco-friendly
- To improve bandwidth and computing capacity
- To improve storage management like disk-drives

Reliability

- To identify system failures and their reasons to manage the risks
- To reduce SLA violation and service delay
- To protect critical information from security attacks
- To make point to point communication using encryption and decryption
- To provide secure VM migration mechansim
- To improves capability of the system
- To reduce Turn of Investment (ToI)

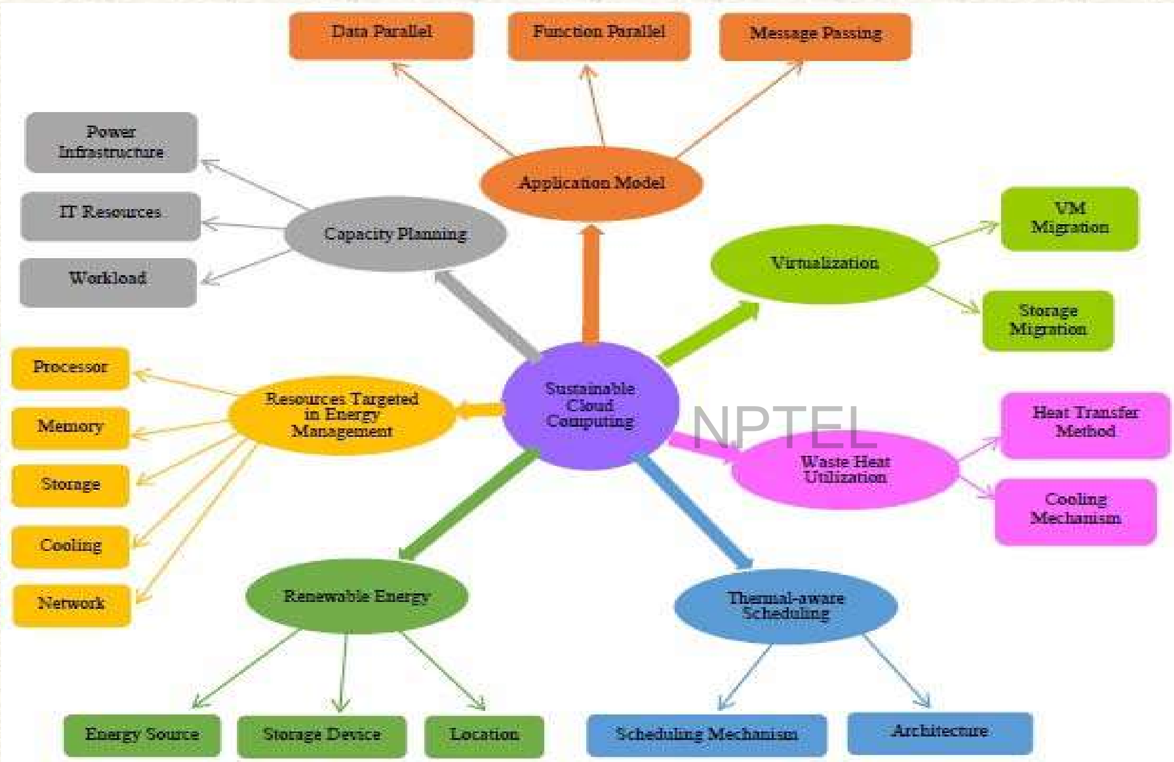
Implication of Reliability on Sustainability

- Improving energy utilization, which reduces electricity bills and operational costs to enables sustainable cloud computing.
- However, to provide reliable cloud services, the business operations of different cloud providers are replicating services, which needs additional resources and increases energy consumption.
- Thus, a trade-off between energy consumption and reliability is required to provide cost-efficient cloud services.
- Existing energy efficient resource management techniques consume a huge amount of energy while executing workloads, which decreases resources leased from cloud datacenters.
- Dynamic Voltage and Frequency Scaling (DVFS) based energy management techniques reduced energy consumption, but response time and service delay are increased due to the switching of resources between high scaling and low scaling modes.

Implication of Reliability on Sustainability

- Reliability of the system component is also affected by excessive turning on/off servers.
- Power modulation decreases the reliability of server components like storage devices, memory etc.
- By reducing energy consumption of CDCs, we can improve the resource utilization, reliability and performance of the server.
- There is a need of new energy-aware resource management techniques to reduce power consumption without affecting the reliability of cloud services.

Sustainable Cloud Computing – Components



Ref: Rajkumar Buyya and Sukhpal Singh Gill. "Sustainable Cloud Computing: Foundations and Future Directions." Business Technology & Digital Transformation Strategies, Cutter Consortium, Vol. 21, no. 6, Pages 1-9, 2018;

Sustainable Cloud Computing – Components

- **Application Model:**
  - In sustainable cloud computing, the application model plays a vital role and the efficient structure of an application can improve the energy efficiency of cloud datacenters.
  - Applications models can be data parallel, function parallel and message passing.
- **Resources Targeted in Energy Management**:
  - Energy consumption of processor, memory, storage, network and cooling of cloud datacenters is typically reported as 45%, 15%, 10%, 10% and 20% respectively
  - Power regulation approaches increase energy consumption during workload execution, which affects the resource utilization of CDCs.
  - DVFS attempts to solve the problem of resource utilization but switching of resources between high scaling and low scaling modes increases response time and service delay, which may violate the SLA.
  - Putting servers in sleeping mode or turning on/off servers may affects the availability/ reliability of the system components.
  - Thus improving energy efficiency of cloud datacenters affects the resource utilization, reliability and performance of the server.

Sustainable Cloud Computing – Components

- **Thermal-aware Scheduling**
  - Components of thermal-aware scheduling are architecture and scheduling mechanisms. Architecture can be single-core or multi-core while scheduling mechanism can be reactive or proactive.
  - Heating problem during execution of workloads reduces the efficiency of cloud datacenters. To solve the heating problem of CDCs, thermal-aware scheduling is designed to minimize the cooling set-point temperature, hotspots and thermal gradient
  - Existing thermal-aware techniques focused on reducing Power Usage Efficiency (PUE) can be found, but a reduction in PUE may not reduce the Total Cost of Ownership (TCO).
- **Virtualization**
  - During the execution of workloads, VM migration is performed to balance the load effectively to utilize renewable energy resources in decentralized CDCs.
  - Due to the lack of on-site renewable energy, the workloads to the other machines distributed geographically.
  - VM technology also offers migration of workloads from renewable energy based cloud datacenters to the cloud datacenters utilizing the waste heat at another site.
  - To balance the workload demand and renewable energy, VM based workload migration and consolidation techniques provide virtual resources using few physical servers.

Sustainable Cloud Computing – Components

- **Capacity Planning**
  - Cloud service providers must involve an effective and organized capacity planning to attain the expected return-on-investment (ROI). The capacity planning can be done for power infrastructure, IT resources and workloads.
  - There is a need to consider important utilization parameters per application to maximize the utilization of resources through virtualization by finding the applications, which can be merged. Merging of applications improves resource utilization and reduces capacity cost.
  - For efficient capacity planning, cloud workloads should be analysed before execution to finish its execution for deadline-oriented workloads.
  - There is also a need of effective capacity planning for data storage and their processing effectively at lower cost.

- **Renewable Energy**
  - Renewable energy source (e.g. solar or wind), the energy storage device and the location (off-site or on-site) are important factors, which can be optimized. Carbon Usage Efficiency (CUE) can be reduced by adding more renewable energy resources.
  - Major challenges of renewable energy are unpredictability and high capital.
  - Workload migration and energy-aware load balancing techniques attempt to address the issue of unpredictability in supply of renewable energy
  - Cloud datacenters are required to place nearer the renewable energy sources to make cost effective.

Sustainable Cloud Computing – Components

- **Waste Heat Utilization**
  - The cooling mechanism and heat transfer model plays an important role to utilize waste heat effectively.
  - Due to consumption of large amounts of energy, CDCs are acting as a heat generator. The vapor-absorption based cooling systems of CDCs can use waste heat then it utilizes the heat while evaporating.
  - Vapor-absorption based free cooling techniques may help in reducing the cooling expenses. The energy efficiency of CDCs can be improved by reducing the energy usage in cooling.

Sustainable Cloud Computing

- The ever-increasing demand for cloud computing services that are deployed across multiple cloud datacenters harnesses significant amount of power, resulting in not only high operational cost but also high carbon emissions
- The next generation of cloud computing must be energy efficient and sustainable to fulfill end-user requirements
- In sustainable cloud computing, the CDCs are powered by renewable energy resources by replacing the conventional fossil fuel-based grid electricity or brown energy to effectively reduce carbon emissions
- Sustainability with high performance and reliability is one of the primary goals

REFERENCES

- Rajkumar Buyya and Sukhpal Singh Gill. “Sustainable Cloud Computing: Foundations and Future Directions.” Business Technology & Digital Transformation Strategies, Cutter Consortium, Vol. 21, no. 6, Pages 1-9, 2018;
- Anders SG Andrae, and Tomas Edler. “On global electricity usage of communication technology: trends to 2030.” Challenges, vol. 6, no. 1, pp. 117-157, 2015.
- Zhenhua Liu, Yuan Chen, Cullen Bash, Adam Wierman, Daniel Gmach, Zhikui Wang, Manish Marwah, and Chris Hyser. “Renewable and cooling aware workload management for sustainable datacenters.” ACM SIGMETRICS Performance Evaluation Review, vol. 40, no. 1, pp. 175-186, 2012.
- Sukhpal Singh Gill and Rajkumar Buyya. 2018. A Taxonomy and Future Directions for Sustainable Cloud Computing: 360 Degree View. ACM Comput. Surv. 51, 5, Article 104 (December 2018), 33 pages.


Thank you

NPTEL ONLINE CERTIFICATION COURSES

Cloud Computing

Prof. Soumya K Ghosh
Department of Computer Science and Engineering

Module 11: Cloud Computing Paradigms
Lecture 55: Sustainable Cloud Computing - II

CONCEPTS COVERED

- Sustainable Computing
- Sustainable Cloud Computing

KEYWORDS

- Sustainable Cloud Computing
- Sustainable Cloud Computing - Taxonomy
- Energy Management
- Carbon Footprint

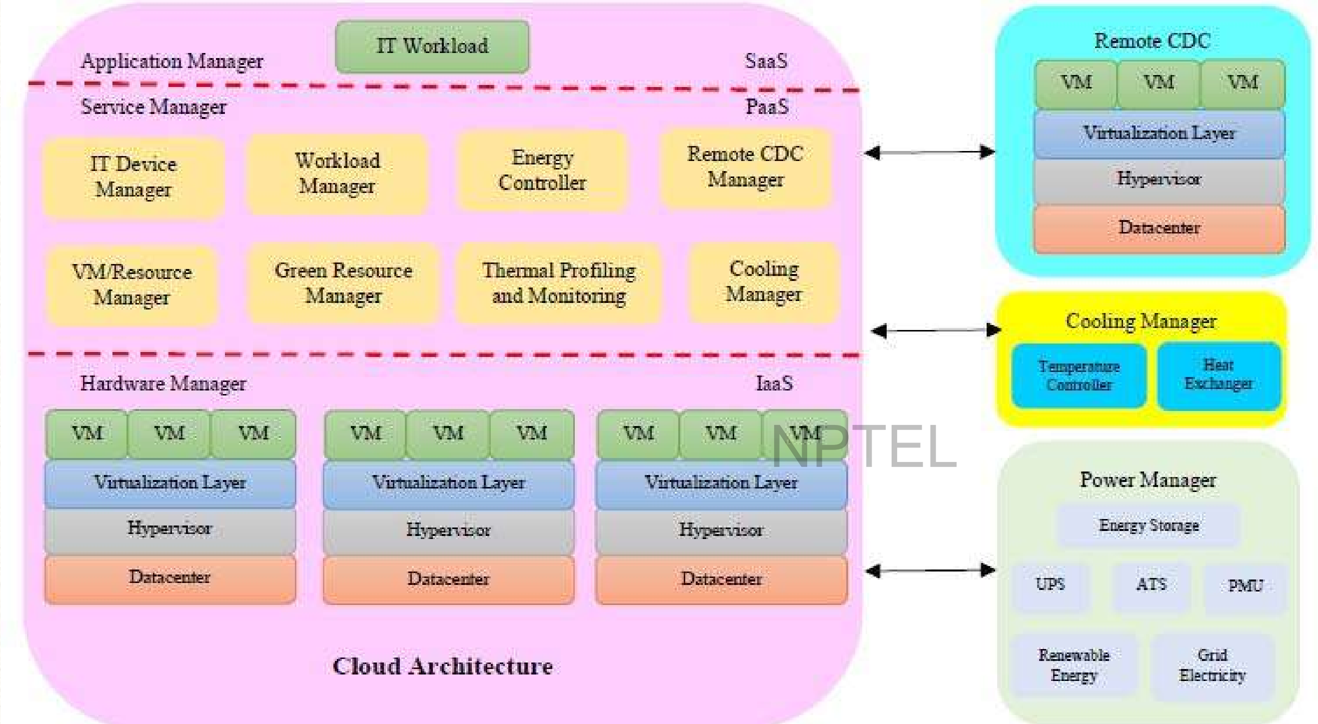
Sustainable Cloud Computing - II

NPTEL

Sustainable Cloud Computing

- Cloud Service Providers (CSPs) rely heavily on the Cloud Data Centers (CDCs) to support the ever-increasing demand for their computational and application services.
- The financial and carbon footprint related costs of running such large infrastructure negatively impacts the sustainability of cloud services.
- Focus on minimizing the energy consumption and carbon footprints and ensuring reliability of the CDCs – goal of Sustainable Cloud Computing

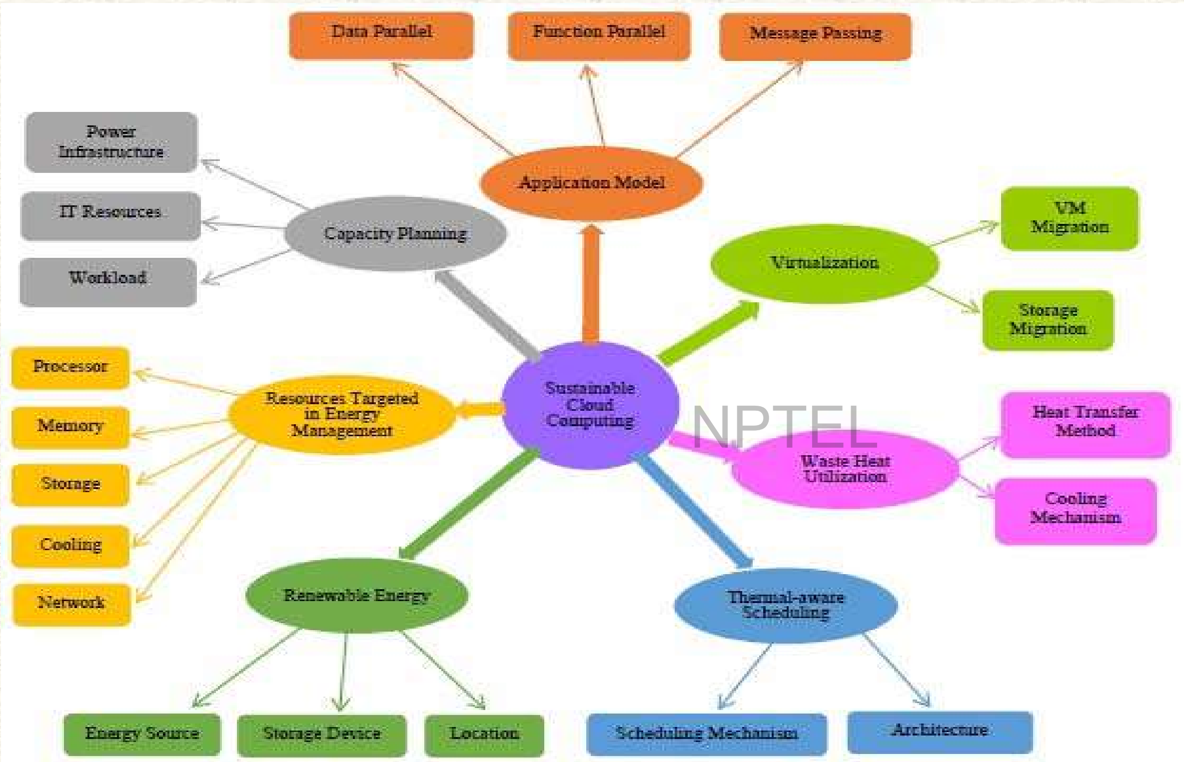
Sustainable Cloud Computing



Sustainable Cloud Computing - A Conceptual Model

Ref: Rajkumar Buyya and Sukhpal Singh Gill. "Sustainable Cloud Computing: Foundations and Future Directions." Business Technology & Digital Transformation Strategies, Cutter Consortium, Vol. 21, no. 6, Pages 1-9, 2018;

Sustainable Cloud Computing – Components



Ref: Rajkumar Buyya and Sukhpal Singh Gill. "Sustainable Cloud Computing: Foundations and Future Directions." Business Technology & Digital Transformation Strategies, Cutter Consortium, Vol. 21, no. 6, Pages 1-9, 2018;

Sustainable Cloud Computing – Components

- **Application Model:**
  - In sustainable cloud computing, the application model plays a vital role and the efficient structure of an application can improve the energy efficiency of cloud datacenters.
  - Applications models can be data parallel, function parallel and message passing.
- **Resources Targeted in Energy Management**:
  - Energy consumption of processor, memory, storage, network and cooling of cloud datacenters is typically reported as 45%, 15%, 10%, 10% and 20% respectively
  - Power regulation approaches increase energy consumption during workload execution, which affects the resource utilization of CDCs.
  - DVFS attempts to solve the problem of resource utilization but switching of resources between high scaling and low scaling modes increases response time and service delay, which may violate the SLA.
  - Putting servers in sleeping mode or turning on/off servers may affects the availability/ reliability of the system components.
  - Thus improving energy efficiency of cloud datacenters affects the resource utilization, reliability and performance of the server.

Sustainable Cloud Computing – Components

- **Thermal-aware Scheduling**
  - Components of thermal-aware scheduling are architecture and scheduling mechanisms. Architecture can be single-core or multi-core while scheduling mechanism can be reactive or proactive.
  - Heating problem during execution of workloads reduces the efficiency of cloud datacenters. To solve the heating problem of CDCs, thermal-aware scheduling is designed to minimize the cooling set-point temperature, hotspots and thermal gradient
  - Existing thermal-aware techniques focused on reducing Power Usage Efficiency (PUE) can be found, but a reduction in PUE may not reduce the Total Cost of Ownership (TCO).
- **Virtualization**
  - During the execution of workloads, VM migration is performed to balance the load effectively to utilize renewable energy resources in decentralized CDCs.
  - Due to the lack of on-site renewable energy, the workloads to the other machines distributed geographically.
  - VM technology also offers migration of workloads from renewable energy based cloud datacenters to the cloud datacenters utilizing the waste heat at another site.
  - To balance the workload demand and renewable energy, VM based workload migration and consolidation techniques provide virtual resources using few physical servers.

Sustainable Cloud Computing – Components

- **Capacity Planning**
  - Cloud service providers must involve an effective and organized capacity planning to attain the expected return-on-investment (ROI). The capacity planning can be done for power infrastructure, IT resources and workloads.
  - There is a need to consider important utilization parameters per application to maximize the utilization of resources through virtualization by finding the applications, which can be merged. Merging of applications improves resource utilization and reduces capacity cost.
  - For efficient capacity planning, cloud workloads should be analysed before execution to finish its execution for deadline-oriented workloads.
  - There is also a need of effective capacity planning for data storage and their processing effectively at lower cost.

- **Renewable Energy**
  - Renewable energy source (e.g. solar or wind), the energy storage device and the location (off-site or on-site) are important factors, which can be optimized. Carbon Usage Efficiency (CUE) can be reduced by adding more renewable energy resources.
  - Major challenges of renewable energy are unpredictability and high capital.
  - Workload migration and energy-aware load balancing techniques attempt to address the issue of unpredictability in supply of renewable energy
  - Cloud datacenters are required to place nearer the renewable energy sources to make cost effective.

Sustainable Cloud Computing – Components

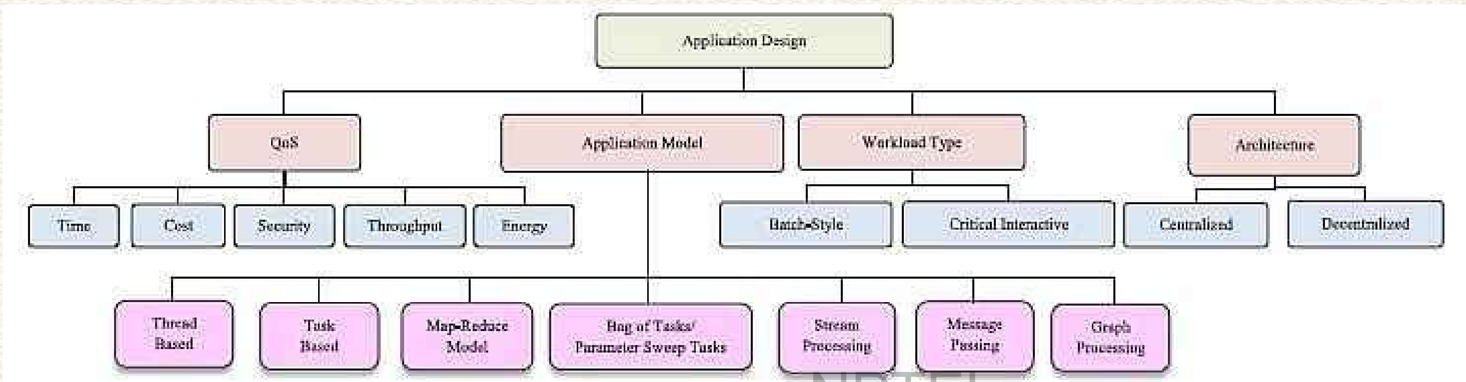
- **Waste Heat Utilization**
  - The cooling mechanism and heat transfer model plays an important role to utilize waste heat effectively.
  - Due to consumption of large amounts of energy, CDCs are acting as a heat generator. The vapor-absorption based cooling systems of CDCs can use waste heat then it utilizes the heat while evaporating.
  - Vapor-absorption based free cooling techniques may help in reducing the cooling expenses. The energy efficiency of CDCs can be improved by reducing the energy usage in cooling.

Sustainable Cloud Computing – Taxonomy

- With huge growth of Internet of Things (IoT)–based applications, the use of cloud services is increasing exponentially.
- Thus, cloud computing must be energy efficient and sustainable to fulfill the ever-increasing end-user needs.
- Research initiatives on sustainable cloud computing can be categorized as follows:
  - application design
  - sustainability metrics
  - capacity planning
  - energy management
  - Virtualization
  - thermal-aware scheduling
  - cooling management
  - renewable energy
  - waste heat utilization

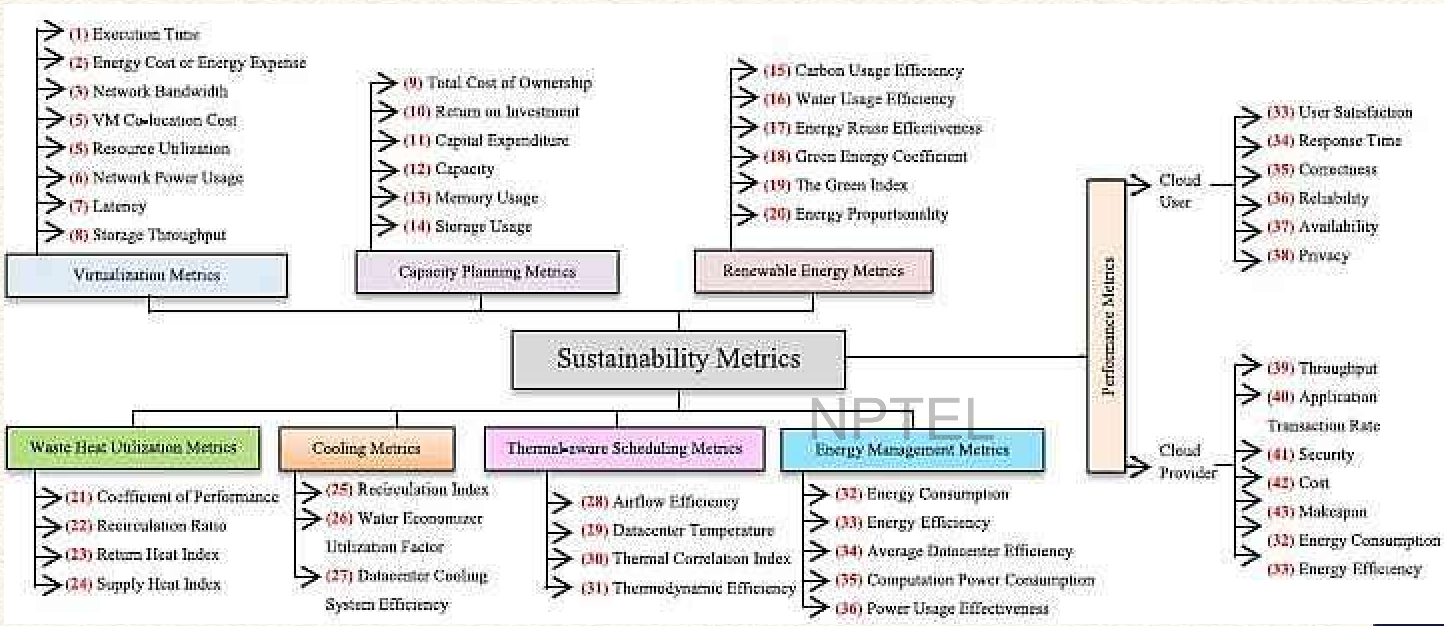
Ref: Sukhpal Singh Gill and Rajkumar Buyya. 2018. A Taxonomy and Future Directions for Sustainable Cloud Computing: 360 Degree View. ACM Comput. Surv. 51, 5, Article 104 (December 2018), 33 pages. https://doi.org/10.1145/3241038

Application Design

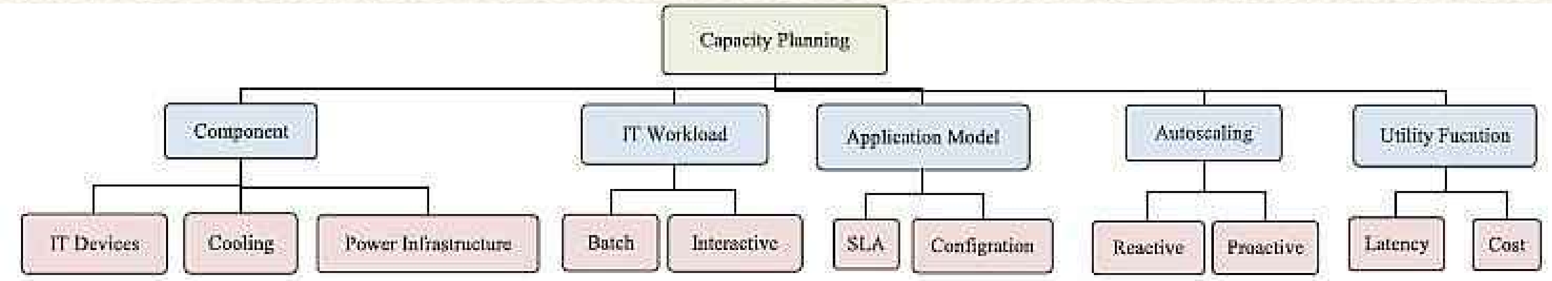


- Design of an application plays a vital role and the efficientvstructure of an application can improve energy efficiency of CDCs.
- The resource manager and scheduler follow different approaches for application modelling
- To make the infrastructure sustainable and environmentally eco-friendly, there is a need for green ICT-based innovative applications

Sustainability Metrics

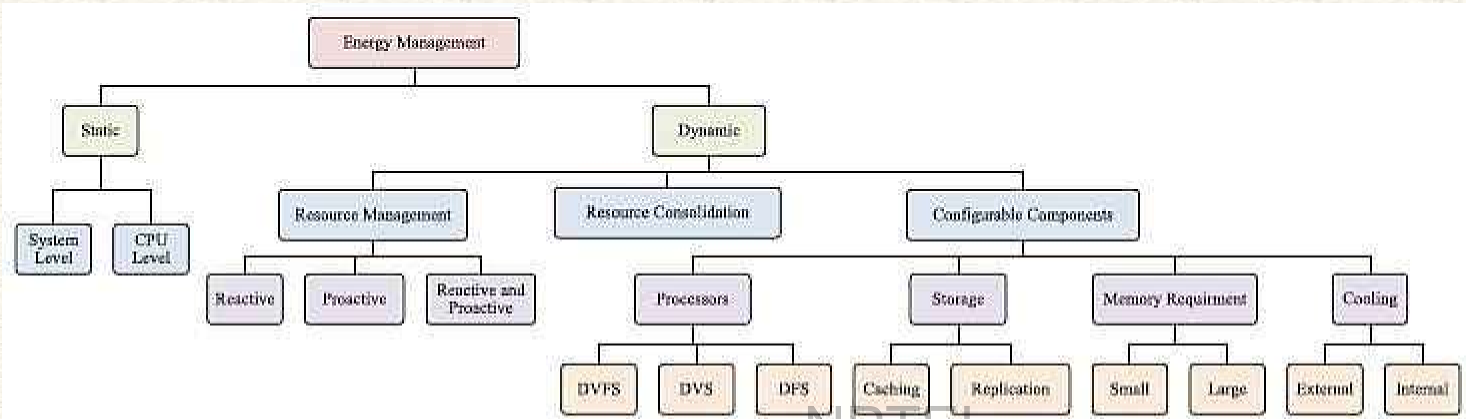


Capacity Planning



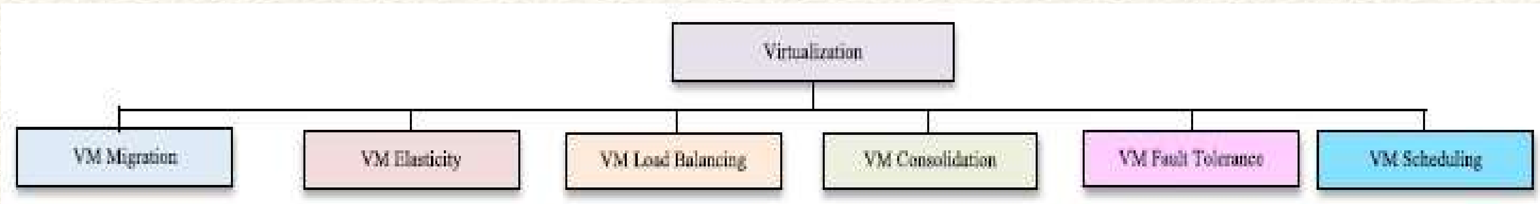
- CSPs must initiate effective and organized capacity planning to enable sustainable computing.
- Capacity planning can be done for power infrastructure, IT infrastructure, and cooling mechanism.

Energy Management



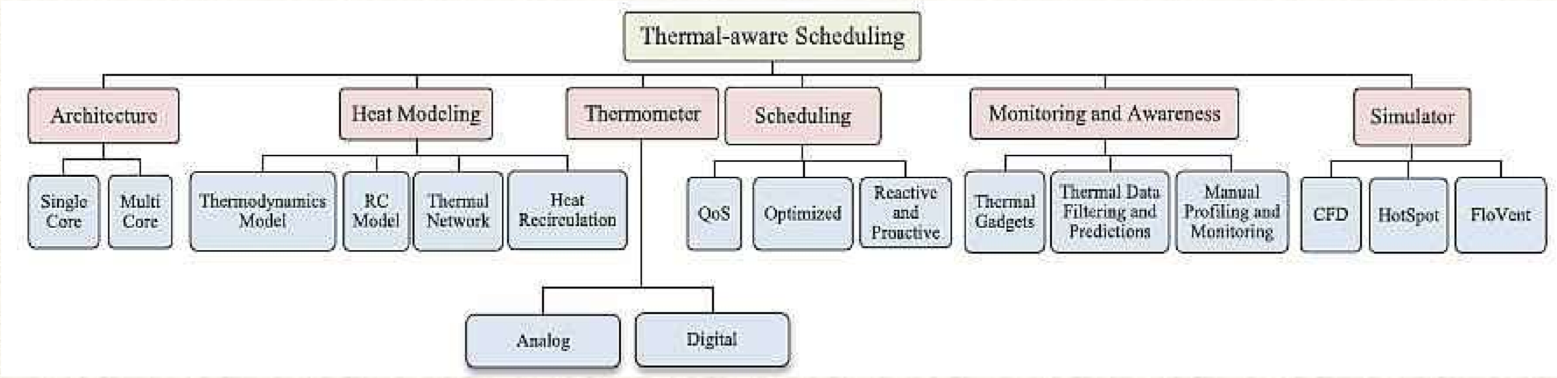
- Energy management in sustainable computing is an important factor for CSPs
- Improving energy use reduces electricity bills and operational costs to enable sustainable cloud computing.
- Essential requirements of sustainable CDCs are optimal software system design, optimized air ventilation, and installing temperature monitoring tools for adequate resource utilization, which improves energy efficiency

Virtualization



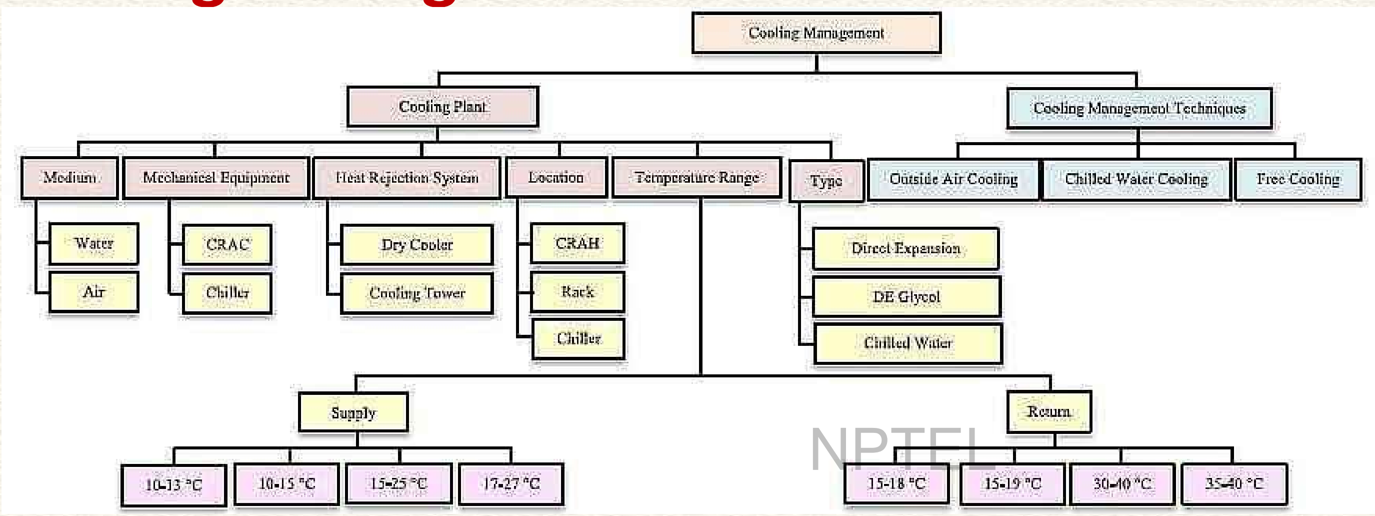
- Virtualization technology is an important part of sustainable CDCs to support energy-efficient VM migration, VM elasticity, VM load balancing, VM consolidation, VM fault tolerance, and VM scheduling
- Operational costs can be reduced by using VM scheduling to manage cloud resources using efficient dynamic provisioning of resources

Thermal-aware Scheduling



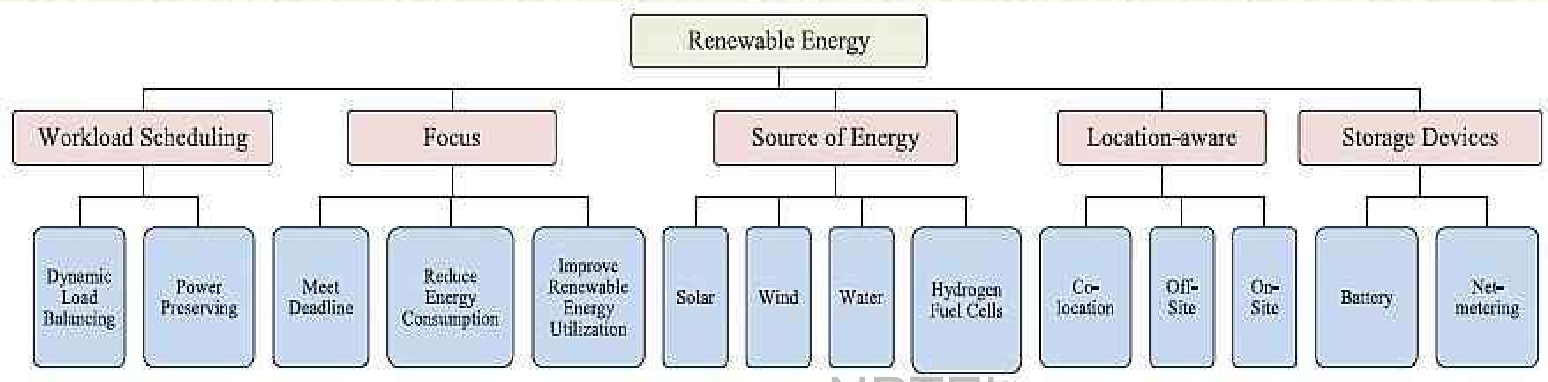
- CDCs consist of a chassis and racks to place the servers to process the IT workloads.
- To maintain the temperature of datacenters, cooling mechanisms are needed.
- Servers produce heat during execution of IT workload. The processor is an important component of a server and consumes the most electricity.
- Both cooling and computing mechanisms consume a huge amount of electricity. Proper management is needed.

Cooling Management



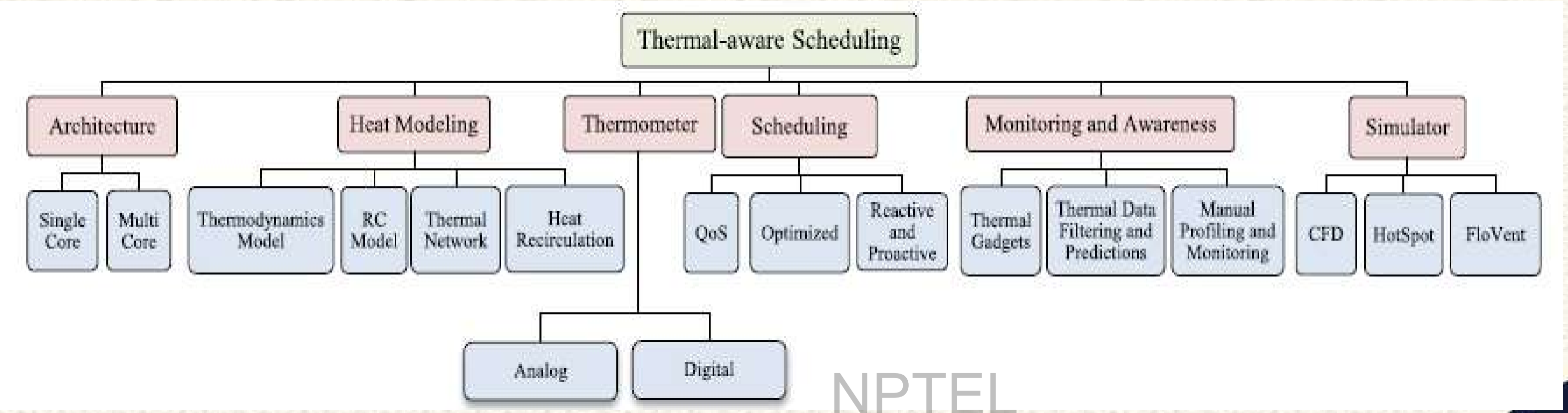
- The increasing demand for computation, networking, and storage expands the complexity, size, and energy density of CDCs exponentially, which consumes a large amount of energy and produces a huge amount of heat.
- To make CDCs more energy efficient and sustainable, we need an effective cooling management system, which can maintain the temperature of CDCs

Renewable Energy



- Sustainable computing needs energy-efficient workload execution by using renewable energy resources to reduce carbon emissions
- Green energy resources, such as solar, wind, and water generate energy with nearly zero carbon-dioxide emissions

Waste Heat Utilization



NPTEL

- Reuse of waste heat is becoming a solution for fulfilling energy demand in energy conservation systems
- The vapor-absorption-based cooling systems can use waste heat, and remove the heat while evaporating.
- Vapor-absorption-based free cooling mechanisms can make the value of PUE (Power Usage Effectiveness) ideal by neutralizing cooling expenses.

Sustainable Cloud Computing

- The ever-increasing demand for cloud computing services that are deployed across multiple cloud datacenters harnesses significant amount of power, resulting in not only high operational cost but also high carbon emissions
- The next generation of cloud computing must be energy efficient and sustainable to fulfill end-user requirements
- In sustainable cloud computing, the CDCs are powered by renewable energy resources by replacing the conventional fossil fuel-based grid electricity or brown energy to effectively reduce carbon emissions
- Sustainability with high performance and reliability is one of the primary goals

REFERENCES

- Rajkumar Buyya and Sukhpal Singh Gill. “Sustainable Cloud Computing: Foundations and Future Directions.” Business Technology & Digital Transformation Strategies, Cutter Consortium, Vol. 21, no. 6, Pages 1-9, 2018
- Sukhpal Singh Gill and Rajkumar Buyya. 2018. A Taxonomy and Future Directions for Sustainable Cloud Computing: 360 Degree View. ACM Comput. Surv. 51, 5, Article 104 (December 2018), 33 pages.
- Anders SG Andrae, and Tomas Edler. “On global electricity usage of communication technology: trends to 2030.” Challenges, vol. 6, no. 1, pp. 117-157, 2015.
- Zhenhua Liu, Yuan Chen, Cullen Bash, Adam Wierman, Daniel Gmach, Zhikui Wang, Manish Marwah, and Chris Hyser. “Renewable and cooling aware workload management for sustainable datacenters.” ACM SIGMETRICS Performance Evaluation Review, vol. 40, no. 1, pp. 175-186, 2012.
- Sukhpal Singh Gill, Inderveer Chana, Maninder Singh and Rajkumar Buyya. 2018. RADAR: Self-Configuring and Self-Healing in Resource Management for Enhancing Quality of Cloud Services, Concurrency and Computation: Practice and Experience (CCPE), 2018.


Thank you