BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

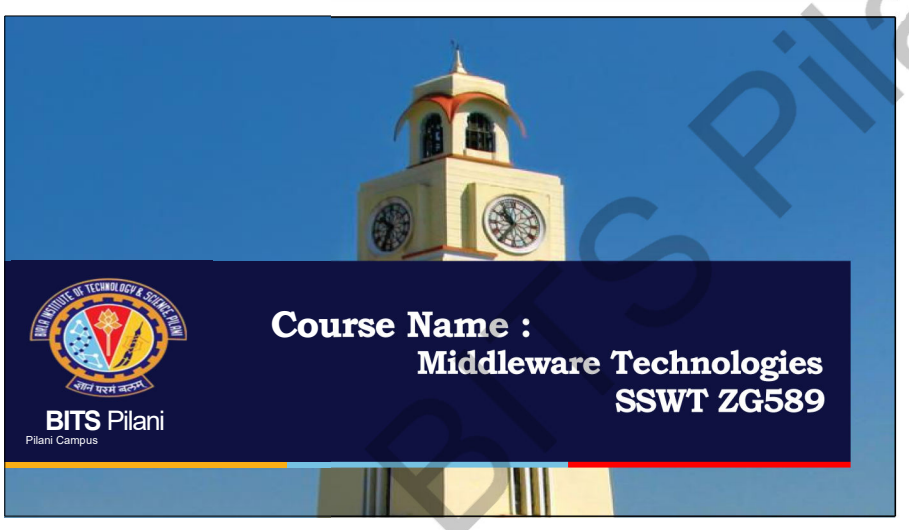
innovate achieve lead

Start Recording

innovate achieve lead

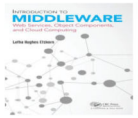
IMP Note to Students

- It is important to know that just login to the session does not guarantee the attendance.
- Once you join the session, continue till the end to consider you as present in the class.
- IMPORTANTLY, you need to make the class more interactive by responding to Professors queries in the session.
- **Whenever Professor calls your number / name ,you need to respond, otherwise it will be considered as ABSENT**



innovate achieve lead

Textbooks



T1: Letha Hughes Etzkorn - Introduction to middleware _ web services, object components, and cloud computing- Chapman and Hall_CRC (2017).

T2: William Grosso - Java RMI (Designing & Building Distributed Applications)

R1: Gregor Hohpe, Bobby Woolf - Enterprise Integration Patterns_ Designing, Building, and Deploying Messaging Solutions -Addison-Wesley Professional (2003)

R2: MongoDB in Action

Note: In order to broaden understanding of concepts as applied to Indian IT industry, students are advised to refer books of their choice and case-studies in their own organizations

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

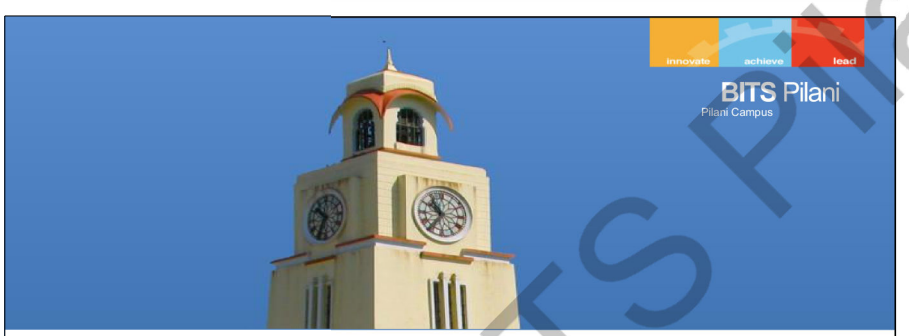
Evaluation Components

Evaluation Component	Name	Type	Weight	Duration	Schedule
EC – 1	Quiz I & II	Individual / Take-home	10%		Pre/Post Mid-Sem
EC – 2	Assignment/ Laboratory Exercises	Practical	10%		TBA
EC – 3	Mid-Semester Examination	Closed Book	30%	2 Hrs.	TBA
EC – 4	End-Semester Examination	Open Book	50%	3 Hrs.	TBA

Modular Structure

innovate achieve lead

No	Title of the Module
M1	Introduction and Evolution
M2	Enterprise Middleware
M3	Middleware Design and Patterns
M4	Middleware for Web-based Application and Cloud-based Applications
M5	Specialized Middleware



CS 13: Comparison of Services provided by Various Cloud providers

Agenda

innovate achieve lead

- Creating S3 bucket in AWS and hosting static website in S3
- Key Advantage of Cloud
- Various cloud provider
- Case study

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Creating and Hosting Static website with AWS S3

S3 is one of the oldest and most popular service provide by AWS with high availability, durability, security and scalability.

S3 can be used to store backup of the database, Big Data Analytics, media and much more.

It provides an object storage mechanism with an abstraction of Buckets, folders and files. The abstraction is what makes S3 easy to use.

With S3 (a storage mechanism), there is no need to perform the capacity planning and to specify the initial capacity.

As we put more data and delete data, S3 will shrink and expand automatically.

S3 provides different storage classes to store different types of data (old/new, frequently/infrequently accessed) and the data can be moved from one storage class to another using S3 Object Life Cycle Management.

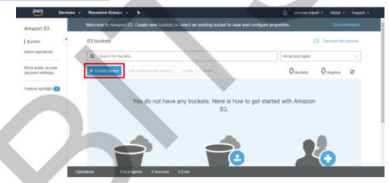
Or

else use the AWS S3 Intelligent Tiering to let AWS decide when to move the data from storage class to another. AWS CloudFront which is a CDN (Content Distribution Network) can optionally be used to make the website load faster to the end user.

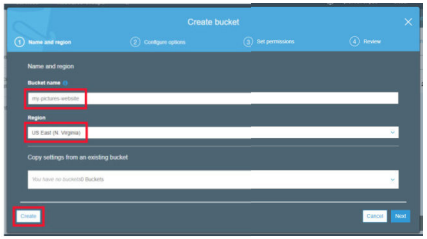
S3 with the free tier provides 5GB of storage, 20,000 Get Requests, 2,000 Put Requests for free every month for the first year and beyond at on a pay-on-usage model.

Step 1: Creating Bucket in S3

Step 1.1: Go to the S3 Management Console and click on “Create Bucket”.



Step 1.2: Enter the Bucket name. Note that the Bucket name should be unique. Add something at the end to get a unique Bucket name. Select the Region where the data must be stored. Click on Create and the Bucket should be created as shown below. A Bucket is a container for storing folders and files.

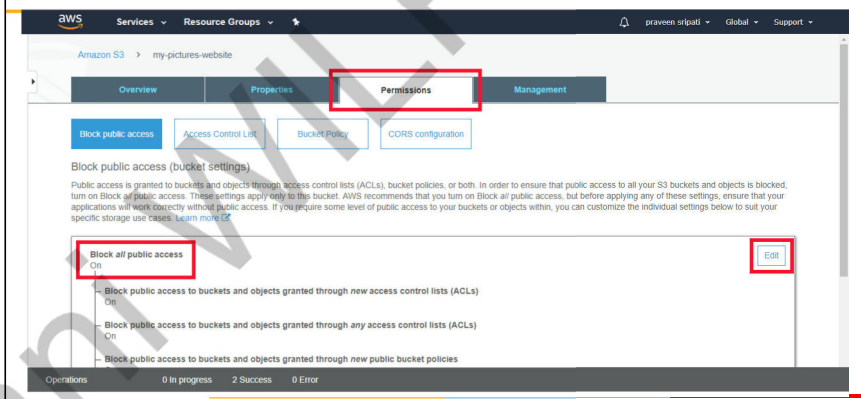


innovate achieve lead

Step 2: Giving the public permissions to the S3 Bucket

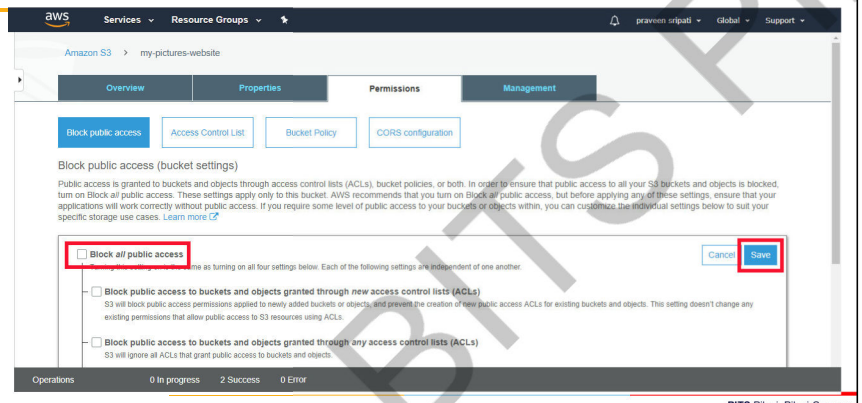
Step 2.1: Any folder/file in the Bucket would only be visible to the owner who created it. For a website, the Bucket should be given public access for the rest of the world to access it as a webpage. Click on the Properties tab, make sure the "Block public access" is selected and click on the Edit button. Unselect "Block all public access" and click on Save. Type the word confirm and click on the "Confirm" button.

This step doesn't give the public permission to the S3 Bucket, but will allow us to make the Bucket and its content public later in the next step. AWS has introduced these additional step and hoops, as there had been a good number of incidents where sensitive data has been put in the S3 Bucket and without the proper settings it was made public for everyone to access the sensitive data.



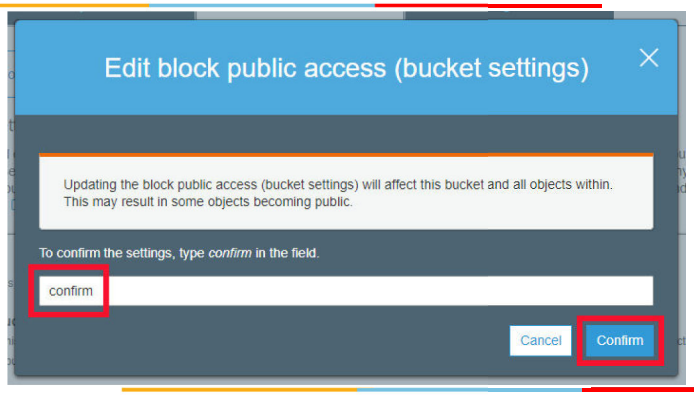
innovate achieve lead

BITS Pilani, Pilani Campus



innovate achieve lead

BITS Pilani, Pilani Campus



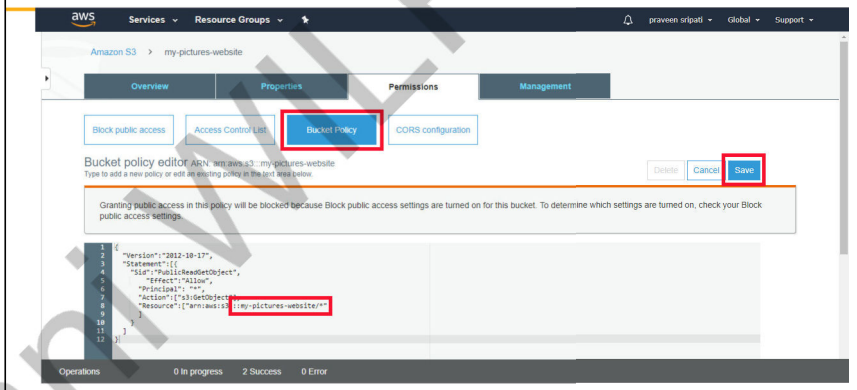
innovate achieve lead

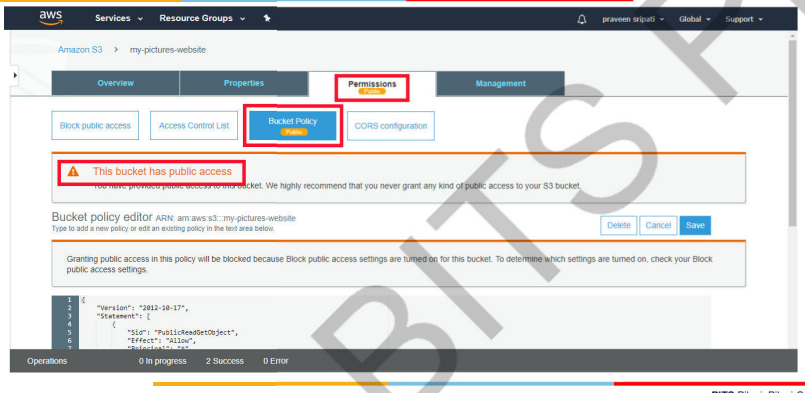
BITS Pilani, Pilani Campus

innovate achieve lead

Step 2.2: Now is the time to make the Bucket public. Click on the “Bucket Policy” and enter the below policy, make sure to change the Bucket name to what was created in Step 1. Click on Save to make the Bucket public. Note that AWS will let us know three times that the Bucket has been made public, just to make sure that we don’t make in public accidentally.

```
{
"Version":"2012-10-17",
"Statement":[{
"Sid":"PublicReadGetObject",
"Effect":"Allow",
"Principal": "*",
"Action":["s3:GetObject"],
"Resource":["arn:aws:s3:::my-pictures-website/*"]
}]
}
```

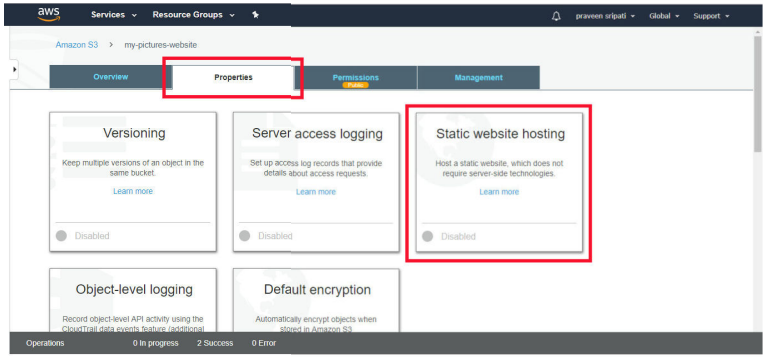



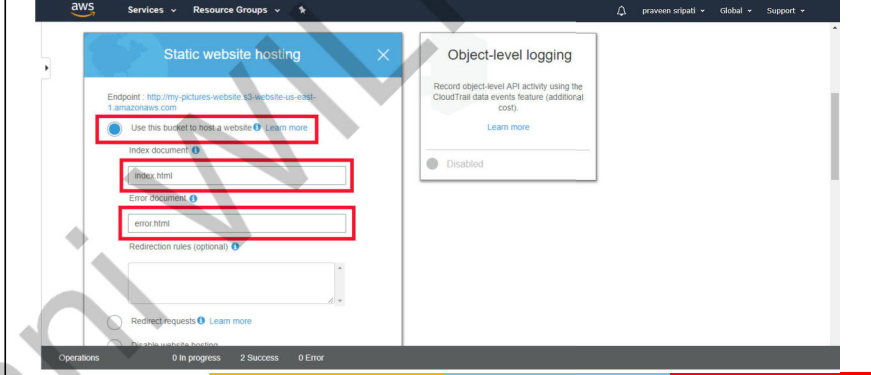


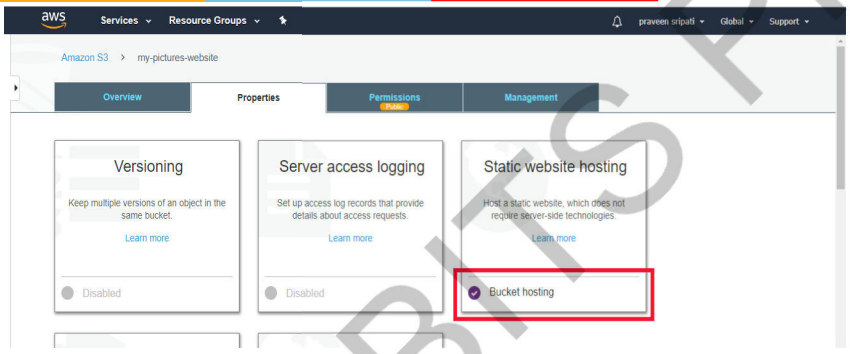
Step 3: Enabling Static website hosting and uploading the website to S3

Step 3.1: Now is the time to enable “Static website hosting” for S3. Notice that by default, it is disabled. Click on the card and select “Use this Bucket to host a website”. Enter the Index document as index.html and the Error document as error.html. Make sure to note down the Endpoint, this is the URL used to access the S3 website. Click on Save.

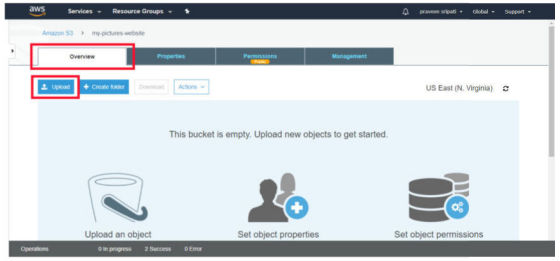
Note that the “Static website hosting” would be enabled by now. The Index document is the default HTML to be displayed and the Error document is the HTML to be displayed when the HTML page which we are trying to access is not there in S3.



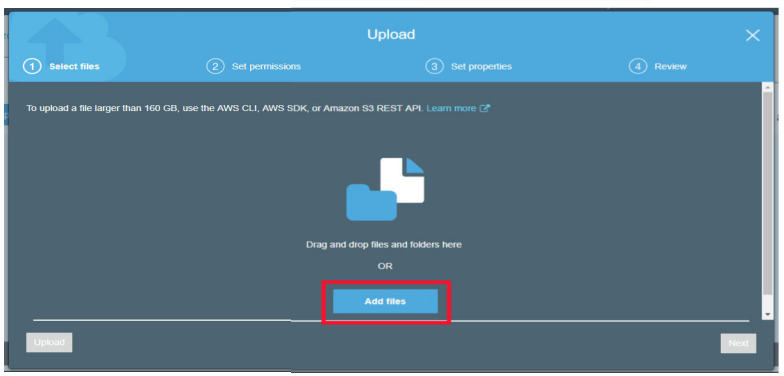


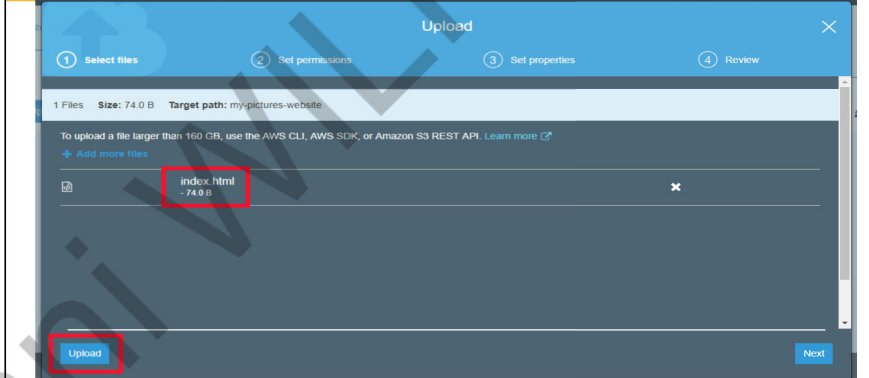


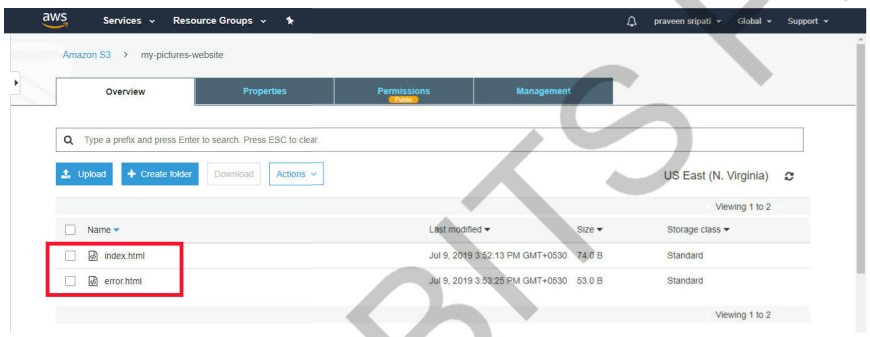
Step 3.2: Now is the time to upload the index.html and error.html. Go to the Overview tab and click on Upload. Click on “Add Files” and then Upload. The same steps must be repeated for index.html and the error.html pages.



innovate achieve lead







Here is the content of index.html and error.html. Here the webpage is a simple one, but it can be as complicated as it gets as long as we stick to HTML and JavaScript. The JavaScript gets executed in the browser. There is no Server-side code execution with S3 website hosting.

index.html

```
<html>
<body>
Welcome to AWS
</body>
</html>
```

error.html

```
<html>
<body>
Oops! Not here.
</body>

</html>
```


innovate achieve lead

Step 4: Access the webpage hosted in S3

Step 4.1: Open the URL got from "Step 5" to get to get the index.html displayed. To the end of the URL anything beside the index.html and the error page would be displayed. The URL is not user friendly, a user-friendly URL can be created using AWS Route53.

Key Advantage of Cloud

- Whether it's infrastructure, software, applications, services, products, or even an operating system, everything is making its way to the cloud.
  - Cloud computing allows a business to cut their operational and fixed monthly costs of hardware, databases, servers, software licenses. Eventually, it will reduce the need for IT resources, including people. All hardware, database servers, web servers, software, products, and services are hosted in the cloud and added to an account as needed.
  - Cloud computing offers 24/7 uptime (99.99% uptime). Cloud servers and data centers are managed by the cloud service provided. Therefore, there is no need for employee management.
  - Cloud computing is scalable and reliable. There is no limit to the number of users or resources. Furthermore, the cloud increases processing and resources as needed. If you do not need resources, you can always scale down.
  - Cloud computing provides maintainability and automatic updates of new software, OS, databases, and third-party software. It also reduces IT labor cost for a business.
  - Cloud service providers have data centers in various locations, which makes them faster and more reliable. Larger companies such as Microsoft and AWS even have data centers around the world.

Categories of Cloud Computing

- Software as a Service (SaaS)
  - SaaS is a software developed and hosted by someone else. Businesses or individuals are able to use them as needed.
- Platform as a Service (PaaS)
  - The PaaS component of cloud computing offers a full development and deployment environment in the cloud, including dev, test, QA, debugging, and deployment tools and services.
- Infrastructure as a Service (IaaS)
  - IaaS offers entire IT computing infrastructure, provisioned and managed over the internet.
  - The key components of IaaS are used to replace existing development and test environments, virtual machines, website hosting, storage, backup, networking, servers, operating systems, middleware, data, and applications, and high-performance computing (HPC).

Top Cloud Service Provider

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud
- Alibaba Cloud
- IBM Cloud
- Oracle
- Hp
- AT&T
- Salesforce
- Rackspace

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Amazon Web Services (AWS)

- Amazon Web Services (AWS) is an Amazon company that was launched in the year 2002. AWS is the most popular cloud service provider in the world.
- Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform, offering over 165 fully-featured services from data centers globally. This service is used by millions of customers.
- AWS's revenue in the year 2018 was $25.6 billion with a profit of $7.2 billion. The revenue is expected to grow to $33 billion in 2019.
- AWS offers hundreds of services. Some of these include Virtual Private Cloud, EC2, AWS Data Transfer, Simple Storage Service, DynamoDB, Elastic Compute Cloud, AWS Key Management Service, AmazonCloudWatch, Simple Notification Service, Relational Database Service, Route 53, Simple Queue Service, CloudTrail, and Simple Email Service.

- AWS Security
- Cloud security is the highest priority for AWS. As a customer, you will benefit from a data center and network architecture built to meet the requirements of the most security-sensitive organizations.
- AWS security offers services such as infrastructure security, DDoS mitigation, data encryption, inventory and configuration, monitoring and logging, identity and access control, and penetration testing.

Microsoft Azure

- Microsoft Azure is one of the fastest-growing clouds among them all. Azure was launched years after the release of AWS and Google Cloud but is still knocking on the door to become the top cloud services provider. Microsoft Azure recently won a $10 billion US government contract.
- While Microsoft Azure revenue is difficult to predict, Microsoft broke down its revenue of the last quarter into three categories, Productivity and Business Processes, Intelligent Cloud, and Personal Computing. The respective revenue was $11.0 billion, $11.4 billion, and $11.3 billion.
- Microsoft's Azure revenue is expected to grow between $33 billion to $35 billion. This makes Azure one of the most profitable cloud services in the world.
- Azure offers hundreds of services within various categories including AI + Machine Learning, Analytics, Blockchain, Compute, Containers, Databases, Developer Tools, DevOps, Identity, Integration, Internet of Things, Management, Media, Microsoft Azure Stack, Migration, Mixed Reality, Mobile, Networking, Security, Storage, Web, and Windows Virtual Desktop.

IBM Cloud

- IBM Cloud developed by IBM is a set of cloud computing services for businesses. Similar to other cloud service providers, the IBM cloud includes IaaS, SaaS, and PaaS services via public, private, and hybrid cloud models.

- Compute, Network, Storage, Cloud Packs, Management, Security, Database, Analytics, AI, IoT, Mobile, Dev Tools, Blockchain, Integration, Migration, Private Cloud, and VMware.

Google Cloud

- Google cloud platform is Google's cloud. Similar to AWS and Azure, Google Cloud also offers similar services in various categories, including compute, storage, identity, security, database, AI and machine learning, virtualization, DevOps and more.
- Here is a list of complete products and services categories Google Cloud Platform services:
- AI and Machine Learning, API Management, Compute, Containers, Data Analytics, Databases, Developer Tools, Healthcare and Life Sciences, Hybrid and Multi-cloud, Internet of Things, Management Tools, Media and Gaming, Migration, Networking, Security and Identity, Serverless Computing, and Storage.
- Google products are also offered in the cloud, including G Suite, Google Maps Platform, Google Hardware, Google Identity, Chrome Enterprise, Android Enterprise, Apigee, Firebase, and Orbitera.
- Google Cloud Services are available in 20 regions, 61 zones, and 200+ countries.
- Google Cloud's annual revenue is close to $8 billion.

innovate achieve lead

Oracle Cloud

- Oracle cloud platform is the cloud offering of Oracle corporation. Oracle cloud offers IaaS, PaaS, SaaS, and Data as a Service (DaaS).
- Oracle offerings include the following:
- Oracle IaaS offerings are Compute, Storage, Networking, Governance, Database, Load Balancing, DNS Monitoring, Ravello, and FastConnect.
- Oracle PaaS offerings are Data Management, Application Development, Integration, Business Analytics, Security, Management, and Content and Enterprise.
- Oracle SaaS offerings are CX, HCM, ERP, SCM, EPM, IoT, Analytics, Data, and Blockchain Applications.
- Oracle DaaS is the Oracle Data Cloud.

Alibaba Cloud

- Alibaba Cloud, founded in 2009, is registered and headquartered in Singapore. It was initially built to serve Alibaba's own e-commerce ecosystem and is now offered to the public. Alibaba Cloud is the largest cloud provider in China.
- Alibaba offers various products and services in various categories, including Elastic Computing, Storage and CDN, Networking, Database Services, Security, Monitoring and Management, Domains and Websites, Analytics and Data Technology, Application Services, Media Services, Middleware, Cloud Communication, Apsara Stack, and Internet of Things.
- Alibaba Cloud is available in 19 regions and 56 availability zones around the globe.
- Alibaba Cloud's revenue is $4.5 million annually.

HP Cloud

- HP [10] is one of famous hardware company in the world with excellent market share in servers and data centers. In last years, HP started to offer cloud services such as:
  - HP Cloud Compute: scalable processing power that customers can control and pay as you use.
  - HP Cloud Storage: offers range of storage options for individuals and business sectors.
  - HP Cloud CDN: refers to Content Delivery Networks and it is a web service that delivers data from HP
- Cloud Storage to customers around the world at high speed using global network of servers from HP and Akamai
- HP Cloud Relational Database: offers an environment for database developers to configure and process relational databases.
- HP Cloud Application platform: offers a platform that enables an enterprise to develop, deploy and scale application in the cloud.
- HP Cloud DNS: a user can manage his/her DNS zones securely and efficiently.
- HP Cloud Identity Service: provides a single method for managing HP cloud users' identities and authentication.

AT & T

- AT&T [12] is the leader American international communication and information technologies company. It starts providing cloud computing service with four major services:
- Cloud Compute: provides computing resources for business and individual sectors.
- Cloud Storage: provides Storage as service.
- Network Enablement: provides networking services in cloud by providing VPN to customers.
- Platform as Service provides platform for developers to develop an application on the cloud

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Salesforce

- Salesforces mainly focuses on specific cloud applications related to sales and customer relationship management. It provides some cloud products such as:
- Sales Cloud: a platform dedicated for Sales application on the cloud.
- Service Cloud: a platform dedicated for customer service management system on the cloud.
- Platform: as a majority of cloud providers they also provide platform as a service.

Rackspace

- Unlike previous providers, Rackspace Company [9] focuses on cloud computing as main core business. They have about 11 different cloud services as follows:
- Cloud Servers: on-demand servers featuring local, powerful Intel® Xeon® processors, and 40Gbps of highly available throughput to every host.
- Block storage: fast, reliable storage for I/O-intensive apps. Use standard or SSD volumes, connected to Cloud Servers via fast network.
- Cloud Files: easy-to-use online storage for files and media. Deliver content globally to users over Akamai's content delivery network (CDN).
- Cloud backup: file-level backup for Cloud Servers. Cloud Backup lets users quickly protect and restore important files.
- Cloud Networks: fully isolated, single-tenant networks which provides control over network topology, IP addressing (IPv4 or IPv6), and architecture.

Rackspace

- Cloud Queues: a scalable cloud message queue service that allow users to easily connect distributed applications without installing complex software.
- Cloud Monitoring: allows users to stay on top of their infrastructure by receiving alerts anytime their URLs, ports, or protocols need attention.
- Cloud Databases: high-performance MySQL databases in the cloud, with built-in redundancy and automated configuration to save time.
- Big Data Platform: production-ready, performance-tested big data clusters on OpenStack-powered cloud, supported by a broad ecosystem of partners.
- Load Balancers: easy-to-configure, reliable failover for high-traffic site or applications hosted on Cloud Servers or Cloud Databases.
- Cloud DNS: allows user to add, modify, and remove domains, subdomains, and records, as well as import and export domains and records.

Case Study – Cloud Computing in a Chosen Public Sector Subject

- In order to illustrate the utilisation of cloud computing in the public sector, an in-depth interview was conducted with a secondary school's IT administrator.
- Profile of the School
- The chosen subject is a secondary school. Its official name is Střední škola a mateřska škola, o.p.s. Litoměřice. It consists of a kindergarten for 40 pre-school children and a secondary school, where students can study at a grammar school (an academically oriented secondary school) or several vocational education courses including those for beauticians, hairdressers, cabinetmakers, shopfitters, graphical designers. There is also an academy of the third age and a university of the third age. The school uses three buildings and employs more than 30 people. The information for the case study was provided by the school's (organization's) IT administrator of the school, Ing. Karel Klatovský.

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Reasons for Implementing Cloud Technology

- The main motivations were cost and human resources savings in relation to streamlined administration.
- As the organization uses three separate buildings several kilometers from one another, it is necessary to connect them in a simple way.
- Previous IT solution did not solve difficult data synchronization among the three buildings.
- The organization considered whether to implement a stand-alone cloud solution to be able to implement such a solution only for data storage and data backup. Nevertheless, it was also necessary to address e-mail accounts, document sharing and register system.
- The organization used all these services before, but they did not want to administer them on their own anymore.

Implementation

- Cloud service implementation had impact on all employees as all of them use e-mail addresses and record information in the register system (attendance, etc.). Students have e-mail accounts, too.
- The IT administrator provided five afternoon training courses. Further training was planned for any occasion when Microsoft would update any service or issue a new one within Microsoft Office 365.
- The employees welcomed cloud services as they had had problems with data synchronization among the three buildings. Currently, they can record anything and have access to their e-mails from anywhere (including their homes), which saves time. So far, nobody has had any problems with the new services.

Benefits

- The employees singled out as the main benefit the fact that cloud services are time-saving.
- The teachers spare much time as they can work from anywhere and IT administrators can do other things as the system administration is easier and it is not necessary to take care of backups and configurations. A lot of routine activities are no longer needed.
- The number of IT employees was reduced from four to only two
- The lower number of employees brought about financial savings. The organization saved money on the register system, too. The iŠkola cloud service is cheaper than the previous system Bakalaři. As the organization uses the Microsoft Office 365 service free of charge and the e-mail communication was done from the employees' private accounts, users benefit from document sharing and the possibility to backup school materials.
- Therefore, it was not necessary to acquire a backup server and pay various licences related to the infrastructure.Cloud services brought a lot of new opportunities for users and solved the problem with data synchronization among the three buildings.

Case Study – Airbnb

- Airbnb is a community marketplace that allows property owners and travelers to connect with each other for the purpose of renting unique vacation spaces around the world. The Airbnb community users' activities are conducted on the company's Website and through its iPhone and Android applications. The San Francisco-based Airbnb began operation in 2008 and currently has hundreds of employees across the globe supporting property rentals in nearly 25,000 cities in 192 countries.
- According to Nathan Blecharczyk, Co-founder & CTO of Airbnb, due to service administration challenges experience with Airbnb original provider, Airbnb migrated to cloud computing functions of AWS (Amazon Web Services) after a year of its creation to gain the ease of managing and customizing its stack. From this, the company has continued to grow upon the reliance on AWS cloud. Thus, AWS is the easy answer for any internet business that wants to scale to the next level.

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Challenges experienced by Airbnb before migrating to AWS cloud.

- Small 5-person operations team
- Infrastructure scalability problem
- Huge traffic load during peak period like; festivals, public holidays etc.

Benefits of AWS to Airbnb

- Reduction in operation expense
- Automation of scaling
- Faster responsiveness
- Flexibility
- Running MySQL in the Cloud

Case Study – in Business: Cloud Strategies

- **Problem / Motivation**
- Identify special causes that relate to bad outcomes for the quality-related parameters of the products and visually inspected defects
- Complex upstream process conditions and dependencies making the problem difficult to solve using traditional statistical / analytical methods
- Determine the optimal process settings that can increase the yield and reduce defects through predictive quality assurancePotential savings huge as the cost of rework and rejects are very high
- **Solution**
- Use ontology to model the complex manufacturing processes and utilize semantic technologies to provide key insights into how outcomes and causes are related
- Develop a rich internet application that allows the user to evaluate process outcomes and conditions at a high level and drill down to specific areas of interest to address performance issues

Why Cloud Computing for this Project

- Well-suited for incubation of new technologies
  1) Semantic technologies still evolving
  2) Use of Prototyping and Extreme Programming
  3) Server and Storage requirements not completely known
- Technologies used (TopBraid, Tomcat) not part of emerging or core technologies supported by corporate IT
- Scalability on demand
- Development and implementation on a private cloud

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Public vs Private Cloud

- Rationale for Private Cloud:
- Security and privacy of business data was a big concern
- Potential for vendor lock-in
- SLA's required for real-time performance and reliability
- Cost savings of the shared model achieved because of the multiple projects involving semantic technologies that the company is actively developing

Cloud Computing for the Enterprise What should IT Do

- Revise cost model to utility-based computing: CPU/hour, GB/day etc.
- Include hidden costs for management, training
- Different cloud models for different applications - evaluate
- Use for prototyping applications and learn
- Link it to current strategic plans for Services-Oriented Architecture, Disaster Recovery, etc.

Thank You

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

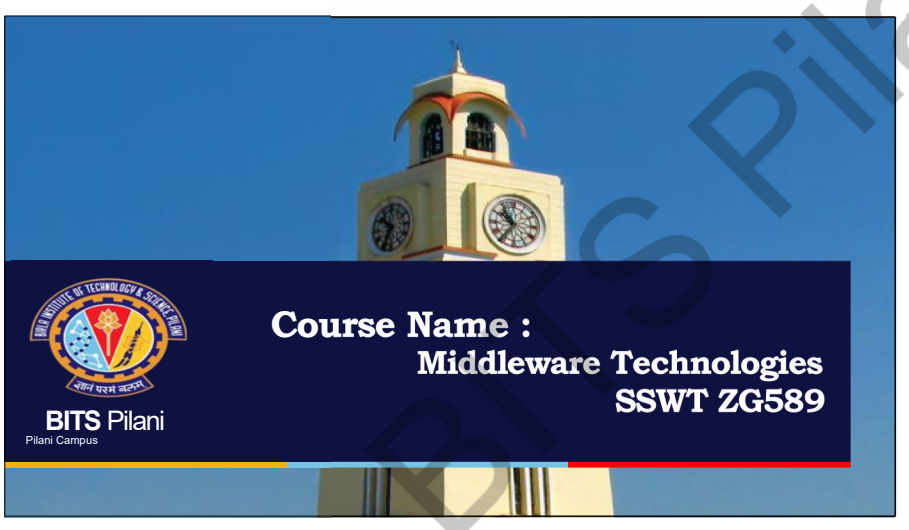
innovate achieve lead

Start Recording

innovate achieve lead

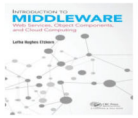
IMP Note to Students

- It is important to know that just login to the session does not guarantee the attendance.
- Once you join the session, continue till the end to consider you as present in the class.
- IMPORTANTLY, you need to make the class more interactive by responding to Professors queries in the session.
- **Whenever Professor calls your number / name ,you need to respond, otherwise it will be considered as ABSENT**



innovate achieve lead

Textbooks



T1: Letha Hughes Etzkorn - Introduction to middleware _ web services, object components, and cloud computing- Chapman and Hall_CRC (2017).

T2: William Grosso - Java RMI (Designing & Building Distributed Applications)

R1: Gregor Hohpe, Bobby Woolf - Enterprise Integration Patterns_ Designing, Building, and Deploying Messaging Solutions -Addison-Wesley Professional (2003)

R2: MongoDB in Action

Note: In order to broaden understanding of concepts as applied to Indian IT industry, students are advised to refer books of their choice and case-studies in their own organizations

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

CS 14: P2P Overlay Middleware

Agenda

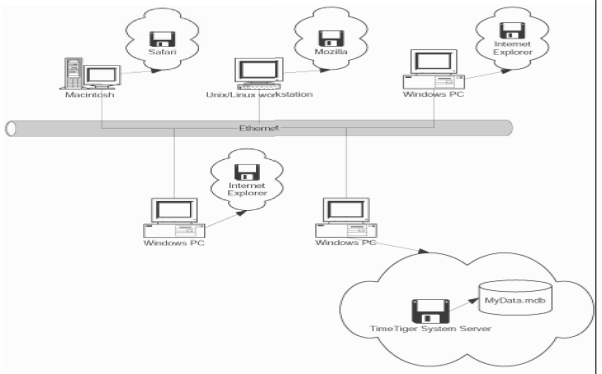
- Introduction
- Napster and its legacy
- Peer-to-peer middleware
- Routing overlays
- Overlay case studies
- Application case studies

Introduction - Traditional client/server systems

- Manage and provide access to resource on a single server or cluster of tightly-couples servers.
- Centralized design.
- Limitations
  - Server hardware resource
  - Network connectivity
  - Requires administration
- **What is P2P system ?**
- P2P systems aim to support useful distributed services and applications using data and computing resources available in the personal computers and workstations that are present on the Internet and other networks in ever-increasing numbers.
- The main goal:
- To allow users to share data without putting them on central servers.

Architecture and Characteristics of P2P System

- Their design to ensure that each user contributes resource to the system.
- All nodes have the same functional capabilities and responsibilities.
- Nodes are autonomous.
- Nodes collaborate directly with each other.

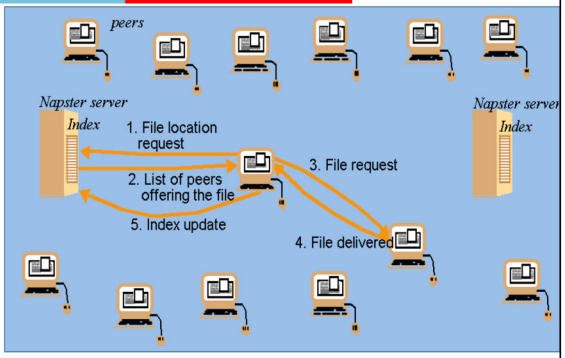


innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Napster and its legacy - Introduction

- A very popular for music exchange soon after its launch in 1999.
- At its peak:
  - Several million users were registered
  - Thousands users were swapping music files simultaneously.



Architecture and method of operation

Napster and its legacy - Shutdown

- On March 5, 2001, an injunction was issued ordering Napster to prevent the trading of copyrighted music on its network.
- The developer of Napster argued:
- They were not participant in the copying process.
- The copying process was performed entirely between user's machines.
- Argument failed:
- The index server were deemed an essential part of the process.
  - In July 2001, Napster shut down its entire network in order to comply with the injunction.

Napster and its legacy – Lesson Learnt

- The feasibility of building a useful large-scale service which depends almost wholly on data and computers owned by ordinary internet users.
- The service to scale to meet the needs of large numbers of users.
- Limitations
  - Object discovery and addressing is likely to become a bottle-neck.
  - No guarantees are required concerning the availability of individual files.

P2P Middleware

- Design to meet the need for automatic placement and subsequent location of the distributed objects managed by peer-to-peer system and application.
- Functional Requirements
  - Simplify the construction of services that are implemented across many hosts in a widely distributed network.
  - Add new resources and remove them.
  - Add hosts to service and remove them.
  - Offer a simple programming interface that is independent of the type of distributed resource that application manipulates.

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

P2P Middleware

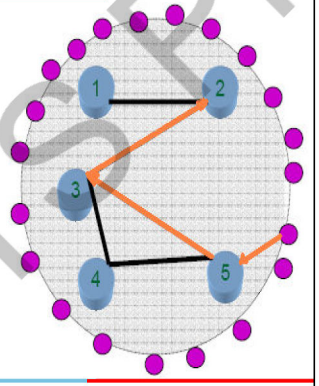
- Non-Functional Requirements
  - Global scalability: support application access millions of object on very large numbers of hosts.
  - Load balancing: Exploit a large number of computers depends upon the balanced distributed of workload across them
  - Optimization for local interactions between neighboring peers: The middleware should aim to place resources close to the nodes that access them the most.
  - Accommodating to highly dynamic host availability: hosts are free to join or leave the system at any time.
  - Security of data in an environment with heterogeneous trust: using authentication and encryption mechanisms to ensure the integrity and privacy of information.

Routing Overlay

- Routing overlay is a distributed algorithm takes responsibility for locating nodes and objects.
- The middleware takes the form of a layer that is responsibility for routing request from any client to the host that the object to which the request is addressed.
- The routing overlay ensures that any node can access any object by routing request through a sequence of node.
- Every nodes and objects has a identifier call GUID.
- The GUID is generated by a hash function using the object's value.
- Overlay routing systems are sometimes described as distributed hash tables (DHT) and this is reflected by the simplest form of API used to access them.

Routing Overlay

- **Main Task - 1**
- Client submit a request including the object's GUID to the routing overlay.
- Overlay must "search" for "closest" node that has a resource.
- Query is "Routed" through the overlay until object is located
- **Main Task - 2**
- When client request the removal of objects from the service the routing overlay must make them unavailable.
- Node may join and leave the service. When the node joins the service, the routing overlay arranges for it to assume some of the responsibility of other nodes. When the node leave , its responsibility are distributed amongst the other nodes.



Routing Overlay

- **API used to access DHT (1)**
- Basic programming interface for DHT
  - put(GUID, data)
  - remove(GUID)
  - value=get(GUID)
- The DHT layer takes responsibility for choosing a location for it, storing it and providing access to it via the get() operation.
- **API used to access DHT (2)**
- Basic programming interface for distributed object location and routing (DOLR)
  - publish(GUIG)
  - unpublidh(GUID)
  - sendToObj(msg, GUID,[n])
- Object can be stored anywhere and the DOLR layer is responsible for maintaining a mapping between GUIDs and the addresses of the nodes at which replicas of the objects are located.

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

innovate achieve lead

Overlay Case Study - Pastry

- Pastry is a routing overlay
- All the nodes and objects are assigned 128-bit GUIDs.
- In a network with N nodes, the Pastry routing algorithm will correctly route a message addressed to any GUID in O(logN) steps.
  - If a node is currently active, the message is delivered to that node
  - Otherwise, the message is delivered to the active node whose GUID is numerically closest to it.
- Active nodes take responsibility for processing requests addressed to all objects in their numerical neighborhood.
- Routing steps involve the use of an underlying transport protocol (normally UDP) to transfer the message to a Pastry node that is 'closer' to its destination.
- The real transport of a message across the Internet between two Pastry nodes may require a substantial number of IP hops.

innovate achieve lead

Overlay Case Study - Pastry

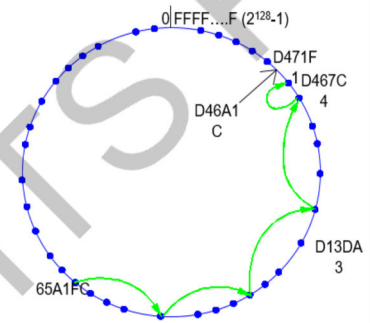
- To minimize the risk of unnecessarily extended transport paths, Pastry selects appropriate neighbours when setting up the routing tables used at each node.
- New nodes join the Pastry overlay:
  - They obtain the data needed to construct a routing table and other required state from existing members in O(logN) messages (N: number of hosts).
- A node failure or departure, the remaining nodes can detect its absence and reconfigure to reflect the required changes in the routing structure.
- **Routing Algorithm – Stage 1(1)**
- Each active node stores a leaf set – a vector L (of size 2l) containing the GUIDs and IP addresses of the nodes whose GUIDs are numerically closest.
- Leaf sets are maintained by Pastry as nodes join and leave. Even after a node failure they will be corrected within a short time.
- The leaf sets reflect a recent state of the system

innovate achieve lead

Overlay Case Study - Pastry

- **Routing Algorithm – Stage 1(2)**
- The GUID space is treated as circular:
  - GUID 0's neighbour is 2128-1.
  - Leaf set includes the GUIDs and IP addresses of the current node's immediate neighbours
- A Pastry system with correct leaf sets of size at least 2 can route messages to any GUID trivially as follows:
  - any node A that receives a message M with destination address D routes the message by comparing D with its own GUID A and with each of the GUIDs in leaf set → forwarding M to the node that is numerically closest to D.
  - at each step M is forwarded to a node that is closer to D than the current node and that this process will eventually deliver M to the active node closest to D.
- Requiring ~ N/2l hops to deliver a message in a network with N nodes → very inefficient

- **Routing Algorithm – Stage 1(3)**



innovate achieve lead

Overlay Case Study - Pastry

- **Routing Algorithm – Stage 2(1)**
- Each Pastry node maintains a tree-structured routing table giving GUIDs and IP addresses
- The routing table is structured as follows:
  - GUIDs are viewed as hexadecimal prefixes.
  - The table has as many rows as there are hexadecimal digits in a GUID
  - Any row n contains 15 entries – one for each possible value of the nth hexadecimal digit excluding the value in the local node's GUID.
  - Each entry points to all nodes whose GUIDs have the relevant prefix.
- **Stage 2(2)**

p=	GUID prefixes and corresponding nodehandles n															
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	n	n	n	n	n	n		n	n	n	n	n	n	n	n	n
1	60	61	62	63	64	65	66	67	68	69	6A	6B	6C	6D	6E	6F
	n	n	n	n	n		n	n	n	n	n	n	n	n	n	n
2	650	651	652	653	654	655	656	657	658	659	65A	65B	65C	65D	65E	65F
	n	n	n	n	n	n	n	n	n	n		n	n	n	n	n
3	65A 0	65A 1	65A 2	65A 3	65A 4	65A 5	65A 6	65A 7	65A 8	65A 9	65A A	65A B	65A C	65A D	65A E	65A F
	n		n	n	n	n	n	n	n	n	n	n	n	n	n	n

Structure of the routing table for specific node (65A1FC)

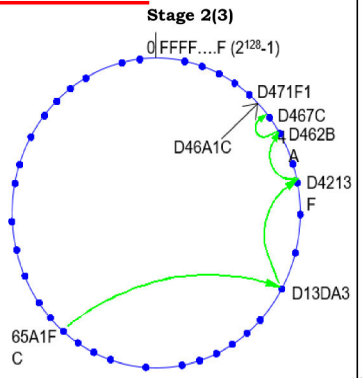
innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Overlay Case Study - Pastry

- **Routing Algorithm:** To handle a message M addressed to a node D (where R[p,i] is the element at column i, row p of the routing table):

1. If (L-l<D<Ll)
// the destination is within the leaf set or is the current node {
2. Forward M to the element Li of the leaf set with GUID closest to D or the current node A }
3. else
// use the routing table to send M to a node with a closer GUID {
4. find p, the length of the longest common prefix of D and A, and i, the (p+1)th hexadecimal digit of D.
5. If (R[p,i]≠null) forward M to R[p,i]
// route M to a node with a longer common prefix.
6.else
// there is no entry in the routing table {
7. Forward M to any node in L or R with a common prefix of length i, but a GUID that is numerically closer.
} }



Overlay Case Study - Pastry

- **Routing Algorithm: Host Integration (1)**
- New nodes use a joining protocol in order to acquire their routing table and leaf set contents and notify other nodes of changes they must make to their tables.
- The new node computes a suitable GUID (applying the SHA-1 hash function to the node's public key), then it makes contact with a nearby Pastry node
- **Host failure or Departure**
- Node in the Partry infrastructure may fail or depart without warning.
- A Pastry node is failed when its immediate neighbours can no longer communicate with it → repair the leaf sets:
  - The node that discovers the failure looks for a live node close to the failed node in L and requests a copy of that node's leaf set, L'.
  - L' overlap L, including an appropriate value to replace the failed node.
- Other neighbor nodes are informed of the failure

Overlay Case Study - Pastry

- **Locality**
- Reduce actual message transmission times by exploiting the locality properites of nodes
- Each row in a routing table contains sixteen entries. The entries in the ith row give the addresses of sixteen nodes with GUIDs with i-1 initial hexadecimal digits that match the current node's GUID.
- A locality metric (number of IP hops or measured latency) is used to compare candidates and the closest available.
- **Fault tolerance**
- All nodes send 'heartbeat' messages to neighbouring nodes in their leaf sets
- Information about failed nodes may not be disseminated sufficiently rapidly to eliminate routing errors.
- Clients employ an 'at-least-once' delivery mechanism and repeat several times in the absence of a response → detect and repair node failures.

Overlay Case Study - Pastry

- **Dependability:** Include the use of acknowledgements at each hop in the routing algorithm. If the sending host does not receive an acknowledgement after a specified timeout, it selects an alternative route and retransmits the message.
- The node that failed to send an acknowledgement is noted as a suspected failure. To detect failed nodes, Pastry node periodically sends a heartbeat message to its neighbour on the left (with a lower GUID). Each node also records the time of the last heartbeat message received from its immediate neighbour on the right (with a higher GUID).
- If the interval since the last heartbeat exceeds a timeout threshold, the detecting node starts a repair procedure that involves contacting the remaining nodes in the leaf set with a notification about the failed node and a request for suggested replacements. If multiple simultaneous failures, this procedure terminates with all nodes on the side of the failed node having leaf sets that contain the i live nodes with the closest GUIDs.

innovate achieve lead

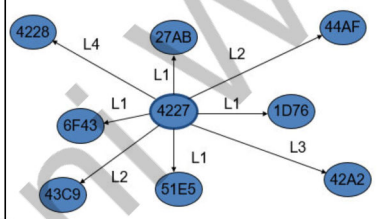
BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Overlay Case Study - Tapestry

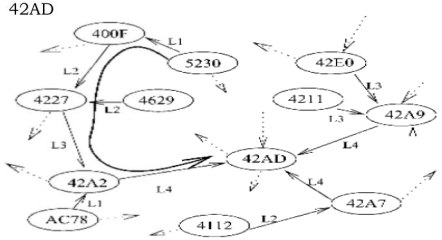
- A Tapestry is a P2P overlay network provides:
  - high-performance
  - scalable
  - fault-tolerant
  - location-independent routing
- Tapestry use adaptive algorithms with soft state to maintain fault tolerance.
- Tapestry nodes are assigned nodeIDs uniformly at random from a large identifier space
- More than one node may be hosted by one physical host.
- Application-specific endpoints are assigned globally unique identifiers (GUIDs) selected from the same identifier space
- Tapestry currently uses an identifier space of 160-bit values.

Overlay Case Study - Tapestry

- Node's neighbors at level "i"
  - Match prefix for -1 digits
  - All possible variations for the ith digit
- For example, node 4227:

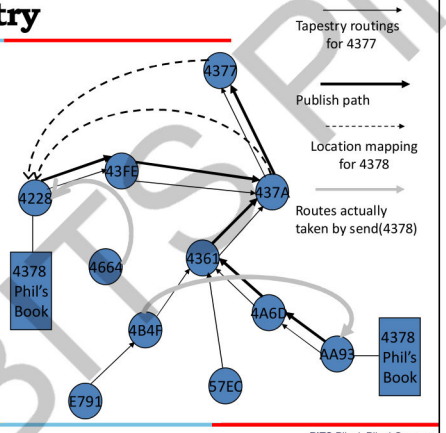


- Path of a message: messages are forwarded progressively closer to the destination node in the ID space
- Example: routing a message from node 5230 to node 42AD



Overlay Case Study - Tapestry

- Guarantees that any existing node can be reached in at most logb N logical hops
- Surrogate routing: when a digit cannot be matched, Tapestry looks for a "close" digit in the routing table.
- Object location and publication:
  - A server S publishes an object O (with GUID, OG, and root, OR) by routing a publish message toward OR
  - Each node along the publication path stores a pointer mapping instead of a copy of the object
  - Nodes store location mappings for object replicas in sorted order of network latency
  - A client locate object O by routing a message to OR, check if whether it has a location mapping for O redirect the message to S. Otherwise, it forwards the message onwards to OR (guaranteed to have a location mapping)



Overlay Case Study - Tapestry

- When a new node joins the network:
  - Need-to-know nodes are notified of N
  - N might become the new root for existing objects, so references to those objects must be moved to N to main object availability
  - Construct a near optimal routing table for N
  - Nodes near N may use N in their routing tables as an optimization
- Voluntary Node Deletion
  - Notified nodes need to have replacement nodes for each routing level
  - Notified nodes send republish traffic to both N and its replacement
- Involuntary Node Deletion
  - Build redundancy into routing tables to improve object availability

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Application Case Study – Squirrel Web Cache

- Web browsers generate HTTP GET requests for Internet objects like HTML pages, images, …
  - a browser cache on the client machine,
  - a proxy web cache: a service running on another computer in the same local network or on a nearby node in the Internet
  - the origin web server : the server whose domain name is included in the parameters of the GET request.
- Some objects are uncacheable
- When a browser cache or proxy web cache receives a GET request:
  - the requested object is uncacheable
  - there is a cache miss
  - the object is found in the cache.

Application Case Study – Squirrel Web Cache

- Web objects are stored in web servers and cache servers with some additional metadata values:
  - a timestamp giving a date of last modification T
  - and possibly a time-to-live t or an eTag.
- For objects that have an associated time-to-live t → fresh if T+t later than the current real time.
- For objects without a time-to-live, an estimated value for t is used. If the result of this freshness evaluation is positive, the cached object is returned to the client without contacting the origin web server. Otherwise a conditional GET(cGET) request is issued to the next level for validation
- Two basic types of cGET requests:
  - an If-Modified-Since request containing the timestamp of the last known modification.
  - an If-None-Match request containing an eTag representing the object contents.
- Whenever a newly-modified cacheable object is received from the origin server, it is added to the set of objects in the local cache together with a timestamp, a time-to-live and eTag if available.

Application Case Study – Squirrel Web Cache

- Performs the same functions using a small part of the resources of each client computer on a local network.
- The SHA-1 secure hash function is applied to the URL of each cached object to produce a 128-bit Pastry GUID.
- The authenticity of a web page may be compromised at many points in its journey from the host to the client; authentication of cached pages only adds little to any overall guarantee of authenticity – the HTTPS protocol should be used to achieve a much better guarantee for those interactions that require it.
- In the simplest implementation of Squirrel – which proved to be the most effective one – the node whose GUID is numerically closest to the GUID of an object becomes that object's home node, responsible for holding any cached copy of the object.

Application Case Study – Squirrel Web Cache

- Client nodes are configured to include a local Squirrel proxy process which takes responsibility for both local and remote caching of web objects.
- If a fresh copy of a required object is not in the local cache Squirrel routes a Get request or a cGet request via Pastry to the home node.
- If the home node has a fresh copy it directly responds to the client with a not-modified message or a fresh copy, as appropriate.
- If the home node has a stale copy or no copy of the object it issues a cGet or a Get to the origin server, respectively.
- If the origin server may respond with a not-modified or a copy of the object. The home node re-validates its cache entry and forwards a copy of the object to the client. In the latter case, it forwards a copy of the new value to the client and places a copy in its local cache if the object is cacheable

innovate achieve lead

innovate achieve lead

Thank You

innovate achieve lead

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

innovate achieve lead

Start Recording

innovate achieve lead

IMP Note to Students

- It is important to know that just login to the session does not guarantee the attendance.
- Once you join the session, continue till the end to consider you as present in the class.
- IMPORTANTLY, you need to make the class more interactive by responding to Professors queries in the session.
- **Whenever Professor calls your number / name ,you need to respond, otherwise it will be considered as ABSENT**

Disclaimer

- *The slides presented here are obtained from the authors of the books and from various other contributors.*
- *I hereby acknowledge all the contributors for their material and inputs.*
- *I have added and modified a few slides to suit the requirements of the course.*

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Agenda

- NoSQL Databases – Introduction
- RDBMS Vs NoSQL DB
- MongoDB – Overview Introduction

Specialized Middleware - NoSQL Databases

- Deals with data that is modeled in forms other than tabular relations (as defined in relational data bases)
- Large volumes of rapidly changing structured, semi-structured, and unstructured data
- Geographically distributed scale-out architecture instead of expensive, monolithic architecture
- Data types such as documents, images, videos, machine learning models etc.
- NoSQL Data base types
  - Document databases
  - Graph stores
  - Key-value stores
  - Wide-column stores

What is NoSQL?

NoSQL databases are a special kind of database system made to handle lots of unstructured and semi-structured data well. Unlike regular relational databases with fixed table structures, NoSQL databases use flexible models that can adjust to changes in data. They're also built to grow horizontally, which means they can handle more and more data as needed.

Originally, "NoSQL" stood for "non-SQL" or "non-relational" databases. But now, it's grown to mean "not only SQL." This change shows how NoSQL databases have expanded to include all sorts of different ways of organizing data, not just the traditional SQL methods.

NoSQL databases find extensive use in applications requiring real-time processing and analysis of large data volumes, such as social media analytics, e-commerce, and gaming. They also serve in other areas like content and document management systems, as well as customer relationship management.

However, it's essential to note that NoSQL databases may not suit all applications due to potential limitations in data consistency and transactional guarantees compared to traditional relational databases. Hence, careful assessment of the application's requirements is crucial in selecting the appropriate database management system.

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

Specialized Middleware - NoSQL Databases

- RDBMS vs NoSQL

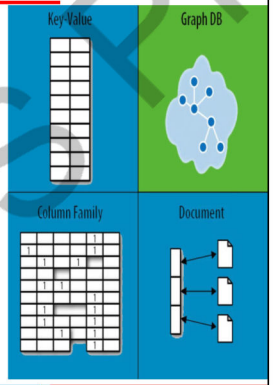
	SQL	NoSQL
Type	Relational	Non-Relational
Data	Structured Data stored in Tables	Un-structured stored in JSON files but the graph database does supports relationship
Schema	Static	Dynamic
Scalability	Vertical	Horizantal
Language	Structured Query Language	Un-structured Query Language
Joins	Helpful to design complex queries	No joins, Don't have the powerful interface to prepare complex query
OLTP	Recommended and best suited for OLTP systems	Less likely to be considered for OLTP system
Support	Great support	community depedent, they are expanding the support model
Integrated Caching	Supports In-line memory(SQL2014 and SQL 2016)	Supports integrated caching
flexible	rigid schema bound to relationship	Non-rigid schema and flexible
Transaction	ACID	CAP theorem
Auto elasticity	Requires downtime in most cases	Automatic, No outage required

NoSQL - Distinguishing Characteristics

- Large data volumes
  - Google's "big data"
- Scalable replication and distribution
  - Potentially thousands of machines
  - Potentially distributed around the world
- Queries need to return answers quickly
- Mostly query, few updates
- Asynchronous Inserts & Updates
- Schema-less
- ACID transaction properties are not needed – BASE
- CAP Theorem
- Open source development

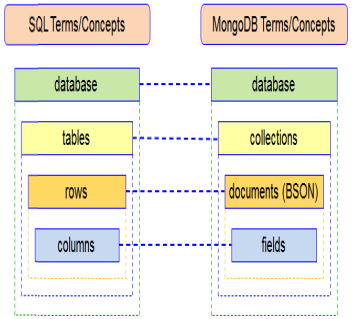
NoSQL – Database types

- Discussing NoSQL databases is complicated because there are a variety of types:
- **Sorted ordered Column Store**
  - Optimized for queries over large datasets, and store columns of data together, instead of rows
- **Document databases:**
  - pair each key with a complex data structure known as a document.
- **Key-Value Store :**
  - are the simplest NoSQL databases. Every single item in the database is stored as an attribute name (or 'key'), together with its value.
- **Graph Databases :**
  - are used to store information about networks of data, such as social connections.



NoSQL – Mongo DB

- MongoDB is a document oriented DB following NoSQL paradigm
- Released in 2009, latest version 4.0
- Written in C++, Go, Python and Javascript
- Offers a JS shell for all operations
- Available as a standalone server and Cloud offering
- Features
  - Adhoc Querying
  - Indexes
  - Replication
  - Speed and durability
  - Scaling



innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

NoSQL – Mongo DB

- MongoDB – Document Data Model
- Basic elements in MongoDB are: documents and fields

```
{
   _id: ObjectID('4bd9e8e17cefd64108961bb'),
   title: 'Adventures in Databases',
   url: 'http://example.com/databases.txt',
   author: 'msmith',
   vote_count: 20,
   tags: ['databases', 'mongodb', 'indexing'],
   image: {
      url: 'http://example.com/db.jpg',
      caption: 'A database.',
      type: 'jpg',
      size: 75381,
      data: 'Binary'
   },
   comments: [
      {
         user: 'bjones',
         text: 'Interesting article.'
      },
      {
         user: 'sverch',
         text: 'Color me skeptical!'
      }
   ]
}
```

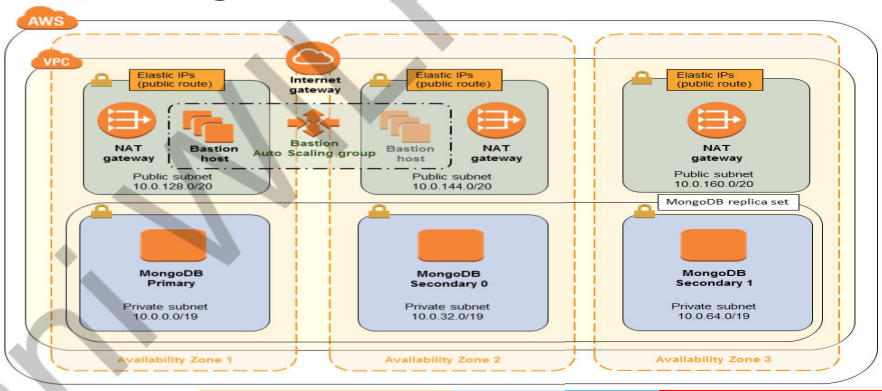
_id field, primary key

Tags stored as array of strings

1 Attribute pointing to another document

Comments stored as array of comment objects

NoSQL : Mongo DB – AWS



NoSQL – Mongo DB

- Data types: bool, int, double, string, object(bson), oid, array, null, date.
- Database and collections are created automatically.
- Lots of Language Drivers.
- Capped collections are fixed size collections, buffers, very fast, FIFO, good for logs. No indexes.
- Object id are generated by client, 12 bytes packed data. 4 byte time, 3 byte machine, 2 byte pid, 3 byte counter.
- Possible to refer other documents in different collections but more efficient to embed documents.
- Replication is very easy to setup. You can read from slaves.
- Connection pooling is done for you. Sweet.
- Supports aggregation.
  - Map Reduce with JavaScript.
- You have indexes, B-Trees. Ids are always indexed.

NoSQL – Mongo DB

- Updates are atomic. Low contention locks.
- Querying mongo done with a document:
  - Lazy, returns a cursor.
  - Reduceable to SQL, select, insert, update limit, sort etc.
    - There is more: upsert (either inserts of updates)
  - Several operators:
    - $ne, $and, $or, $lt, $gt, $incr,$decr and so on.
- Repository Pattern makes development very easy.

```
> // map function
> m = function(){
...    this.tags.forEach(
...        function(z){
...            emit( z , { count : 1 } );
...        }
...    );
...};
```

```
> // reduce function
> r = function( key , values ){
...    var total = 0;
...    for ( var i=0; i <values. length; i++
)
...        total += values[i].count;
...    return { count : total };
...};
```

```
> // execute
> res = db.things.mapReduce(m, r, { out : "myoutput" } );
```

MongoDB Example

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

NoSQL – Mongo DB

- **Performance:**
- There is no perfect NoSQL database
- Every database has its advantages and disadvantages
- Depending on the type of tasks (and preferences) to accomplish
- NoSQL is a set of concepts, ideas, technologies, and software dealing with
- Big data
- Sparse un/semi-structured data
- High horizontal scalability
- Massive parallel processing
- Different applications, goals, targets, approaches need different NoSQL solutions

Key benefits

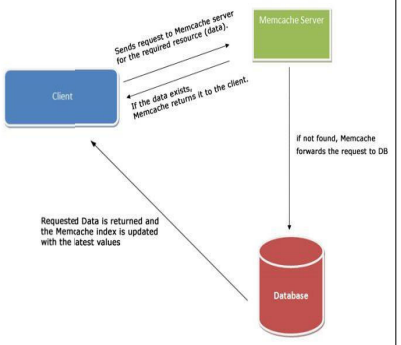
- **Scalability:** NoSQL databases utilize sharding for horizontal scaling, where data is partitioned and distributed across multiple machines while maintaining data order. Unlike vertical scaling, which involves adding resources to existing machines, horizontal scaling, exemplified by MongoDB and Cassandra, is easier to implement and allows for efficient handling of increasing data volumes.
- **Flexibility:** NoSQL databases excel in handling unstructured or semi-structured data, adapting seamlessly to dynamic changes in data models. This adaptability makes them well-suited for applications with evolving data requirements.
- **High Availability:** NoSQL databases ensure high availability through automatic replication, wherein data replicates itself to maintain consistency in case of failure, thus enhancing reliability.
- **Performance:** Designed to handle large data volumes and high traffic, NoSQL databases offer superior performance compared to traditional relational databases, ensuring efficient data processing.
- **Cost-effectiveness:** NoSQL databases often prove more cost-effective than traditional relational databases due to their simpler architecture and reduced hardware and software requirements.
- **Agility:** NoSQL databases are conducive to agile development practices, facilitating rapid adaptation and iteration in software development processes.

Specialized Middleware - Caching

- Cache is a high-speed data storage layer which stores a subset of data, typically transient in nature, so that future requests for that data are served up faster
- Distributed Cache an extension of the traditional concept of cache, spanning multiple servers
- Examples – Redis, Memcached, Hazelcast, Spark, Couchbase, Apache Ignite
- Use cases
  - Database Caching
  - Content Delivery Networks
  - Session Management
  - APIs
  - Web caching

Specialized Middleware - Caching

- **Caching for Web Services - Memcached**
- Stores data in key value pair
- Supports distributed caching using client – server
- The servers store and fetch items and manage when to evict or reuse memory
- It gives near-deterministic query speeds for all use cases O(1)
- Uses Least Recently Used cache, which discards items after specific amount of time

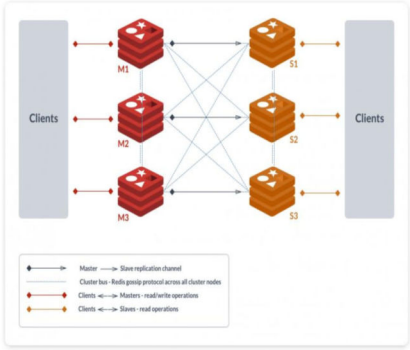


innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

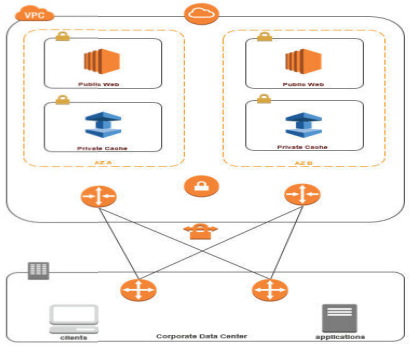
Specialized Middleware - Caching

- **Caching for Web Services - Redis**
- Redis is an open source (BSD licensed), in-memory data structure store
- Supports data structures such as strings, hashes, lists, sets, sorted sets with range queries, bitmaps, hyperlog logs, geospatial indexes with radius queries and streams
- Redis works with an in-memory dataset
- Redis also supports trivial-to-setup master-slave asynchronous replication, with very fast non-blocking first synchronization



Specialized Middleware - Caching

- **Caching for Elastic Cache - AWS**
- ElastiCache is caching-as-a-service offering from AWS
- Provides encapsulation on caching implementation, which seamlessly integrates with other AWS offerings
- Offers two caching engines
  - Memcached
  - Redis
- Comes in two flavors – Standalone Node and Cluster



Use cases of No SQL

Choosing the right type of NoSQL database depends on how your organization intends to use it. Let's break down some specific uses for different types of NoSQL databases:

- **Managing Data Relationships**: When dealing with complex data relationships, like those in recommendation engines, knowledge graphs, fraud detection, and social networks, a graph-based NoSQL database is typically used. These databases excel at connecting various data points efficiently.
- **Low-Latency Performance**: For applications requiring high throughput and real-time data management, such as gaming, home fitness apps, and ad technology, a NoSQL database with low-latency performance is essential. These databases ensure quick responses, crucial for tasks like market bidding updates and delivering relevant ads. Web applications often utilize in-memory NoSQL databases to swiftly handle usage spikes without delays from disk storage.
- **Scaling and Handling Large Data Volumes**: E-commerce platforms must handle massive spikes in usage, especially during events like one-day sales or holiday shopping seasons. Key-value databases are commonly employed here due to their simple structure, which allows easy scaling during high-traffic periods. This flexibility is also beneficial for gaming, ad tech, and Internet of Things (IoT) applications.

How Does NoSQL Work and Why is it Faster Than Relational Databases?

- Unlike relational databases, NoSQL operates quicker because it doesn't need to sift through multiple tables for answers. Instead of the traditional rows and columns setup, NoSQL arranges data in a tabular format, often using JSON documents.
- For instance, let's consider a major retail chain. Instead of accessing various tables for shoe size, brand, and color, all details about the shoes are stored in a single document. This document can easily incorporate new parameters like shoe width or material as needed.
- NoSQL databases excel in handling large, intricate datasets or situations where data structures are frequently changing to adapt to new business needs.

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad | Mumbai
WORK INTEGRATED LEARNING PROGRAMMES

When to Avoid Using a NoSQL Database?

innovate achieve lead

- NoSQL databases are designed for applications that require simplified data structures with fewer tables or containers. They work best for systems where data relationships are represented through embedded records or documents rather than traditional references. If your application heavily relies on highly normalized data, like in finance, accounting, or enterprise resource planning, NoSQL may not be the ideal choice. These systems typically need the strict structure and data integrity provided by relational databases to avoid anomalies and duplication.
- Another factor to consider is query complexity. While NoSQL databases excel with simple queries against a single table, they struggle with more complex queries. Relational databases are better suited for handling intricate joins, sub-queries, and nested queries in a WHERE clause.
- However, it's worth noting that there's not always a clear-cut decision between relational and nonrelational databases. Many companies opt for a hybrid approach, using databases that offer a mix of both models. This hybrid model provides the flexibility to handle different data types while maintaining read and write consistency without sacrificing performance.

Disadvantages of NoSQL Databases

innovate achieve lead

- Each NoSQL database has its unique way of querying and managing data, unlike SQL, which is universally understood across relational and SQL database systems.
- NoSQL databases lack a strict database schema and constraints, which means they don't have the same data integrity safeguards found in relational and SQL database systems.
- Developers must create some form of structure for the schema to utilize the data in NoSQL databases, whereas in SQL databases, this is typically handled by the database administrator.
- Most NoSQL databases employ the eventual consistency model, leading to lower levels of data consistency compared to SQL databases. This inconsistency makes them unsuitable for transactions requiring immediate integrity, like banking or ATM transactions.

Examples of NoSQL databases

innovate achieve lead

Many companies have entered the NoSQL landscape

- Apache CouchDB, an open source, JSON document-based database that uses JavaScript as its query language.
- Elasticsearch, a document-based database that includes a full-text search engine.
- Couchbase, a key-value and document database that empowers developers to build responsive and flexible applications for cloud, mobile, and edge computing.

innovate achieve lead

Thank You

innovate achieve lead