ACADEMIC YEAR 2020-21 | JUNE 2021 | ISSUE 2

PANIMALAR ENGINEERING COLLEGE

(A Christian Minority Institution)
Jaisakthi Educational Trust
ACCREDITED BY NATIONAL BOARD OF ACCREDITATION
BANGALORE TRUNK ROAD, NASARATHPET ,
POONAMALLEE, CHENNAI - 600 123

TECH NEWS

The Half yearly newletter from Computer Science and Engineering

“ DEPARTMENT VISION

To provide an academically conducive environment for individuals to develop as technologically superior, socially conscious and nationally responsible citizens.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

- To impart and disseminate sound knowledge to the students on the fundamentals of mathematics and advanced fields of computer science and inter related disciplines to solve simple and complex engineering problems and train them to achieve sustainable growth in their professional career.
- To enhance the ability of students to evaluate the specific requirements of software industry and provide innovative engineering solutions and efficient product designs.
- To facilitate the students to make use of their technical competency to identify and develop appropriate product design, development, testing, maintenance, analysis of problems and provide corrective measures.
- To enable the students to develop strong leadership qualities with aggressive optimism, multidisciplinary skills, excellent communication skills and function as effective and reliable team members giving importance to professional and ethical principles.
- To inculcate in the students to associate in social networking, pursue continued learning of the latest developments in computer science and involve in higher research and contribute to the development of software industry and related engineering fields.

PROGRAM SPECIFIC OBJECTIVES (PSOs)

- PSO 1: To inculcate technical skills to analyze, design and implement software's related to algorithms, networking, web services, multimedia, big data analytics and recent topics of varying complexity.
- PSO 2: To develop the capability to comprehend and solve the interdisciplinary problems through appropriate technology with the understanding of contemporary business environment.
- PSO 3: To develop an ability to utilize the latest technology and platforms to become a triumphant professional, successful entrepreneur and an urge for pursuing higher studies.

“ DEPARTMENT MISSION

To develop our department as a center of excellence, imparting quality education, generating competent and skilled manpower. We prepare our students with high degree of credibility, integrity, ethical standards and social concern. We train our student to develop to devise and implement novel systems, based on Education and Research.

PROGRAM OUTCOMES(POs)

- Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems
- Problem analysis: Identify, formulate, research literature, and analyze complex engineering Problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
- Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the to provide valid conclusions.
- Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the Professional engineering practice.
- Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development
- Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings
- Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

APACE KAFKA – AN INTRODUCTION

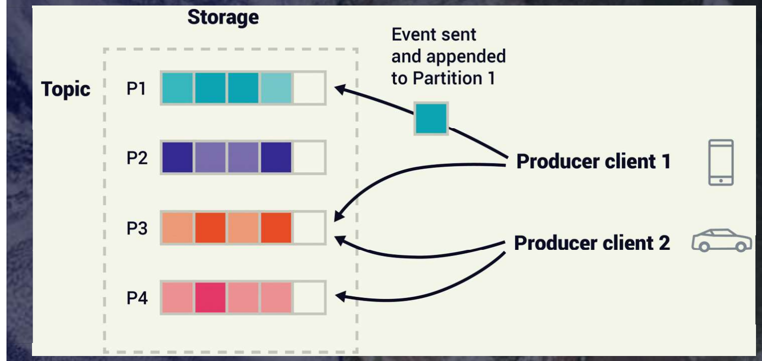
AN ARTICLE BY C JACULIN, ASSISTANT PROFESSOR, CSE

Apache Kafka is a distributed publish-subscribe messaging system and a robust queue that can handle a high volume of data and enables you to pass messages from one end-point to another. Kafka is suitable for both offline and online message consumption. Kafka messages are persisted on the disk and replicated within the cluster to prevent data loss. Kafka is built on top of the ZooKeeper synchronization service. It integrates very well with Apache Storm and Spark for real-time streaming data analysis.

Following are a few benefits of Kafka –

- Reliability – Kafka is distributed, partitioned, replicated and fault tolerance.
- Scalability – Kafka messaging system scales easily without down time..
- Durability – Kafka uses Distributed commit log which means messages persists on disk as fast as possible, hence it is durable..
- Performance – Kafka has high throughput for both publishing and subscribing messages. It maintains stable performance even many TB of messages are stored.

Kafka is very fast and guarantees zero downtime and zero data loss.



Kafka is a distributed system consisting of servers and clients that communicate via a high-performance TCP network protocol. It can be deployed on bare-metal hardware, virtual machines, and containers in on-premise as well as cloud environments.

An event records the fact that "something happened" in the world or in your business. It is also called record or message in the documentation. When you read or write data to Kafka, you do this in the form of events. Conceptually, an event has a key, value, timestamp, and optional metadata headers. Here's an example event:

- Event key: "Alice"
- Event value: "Made a payment of $200 to Bob"
- Event timestamp: "Jun. 25, 2020 at 2:06 p.m."

Producers are those client applications that publish (write) events to Kafka, and consumers are those that subscribe to (read and process) these events. In Kafka, producers and consumers are fully decoupled and agnostic of each other, which is a key design element to achieve the high scalability that Kafka is known for. For example, producers never need to wait for consumers. Kafka provides various guarantees such as the ability to process events exactly-once.

Events are organized and durably stored in topics. Very simplified, a topic is similar to a folder in a filesystem, and the events are the files in that folder. An example topic name could be "payments". Topics in Kafka are always multi-producer and multi-subscriber: a topic can have zero, one, or many producers that write events to it, as well as zero, one, or many consumers that subscribe to these events. Events in a topic can be read as often as needed—unlike traditional messaging systems, events are not deleted after consumption. Instead, you define for how long Kafka should retain your events through a per-topic configuration setting, after which old events will be discarded. Kafka's performance is effectively constant with respect to data size, so storing data for a long time is perfectly fine.

Topics are partitioned, meaning a topic is spread over a number of "buckets" located on different Kafka brokers. This distributed placement of your data is very important for scalability because it allows client applications to both read and write the data from/to many brokers at the same time. When a new event is published to a topic, it is actually appended to one of the topic's partitions. Events with the same event key (e.g., a customer or vehicle ID) are written to the same partition, and Kafka guarantees that any consumer of a given topic-partition will always read that partition's events in exactly the same order as they were written.

From the example in figure, topic has four partitions P1–P4. Two different producer clients are publishing, independently from each other, new events to the topic by writing events over the network to the topic's partitions. Events with the same key (denoted by their color in the figure) are written to the same partition. Note that both producers can write to the same partition if appropriate.

DOCKER

AN ARTICLE BY K SANGEETHA, ASSISTANT PROFESSOR, CSE

Docker is a container management service. The keywords of Docker are develop, ship and run anywhere. The whole idea of Docker is for developers to easily develop applications, ship them into containers which can then be deployed anywhere.

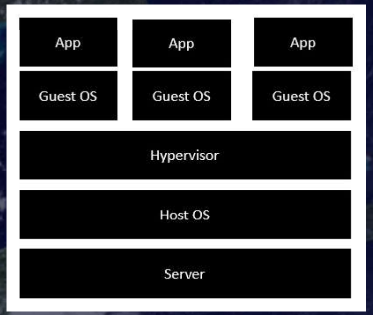
Features of Docker

- Docker has the ability to reduce the size of development by providing a smaller footprint of the operating system via containers.
- With containers, it becomes easier for teams across different units, such as development, QA and Operations to work seamlessly across applications.
- You can deploy Docker containers anywhere, on any physical and virtual machines and even on the cloud.
- Since Docker containers are pretty lightweight, they are very easily scalable.

Components of Docker
Docker has the following components

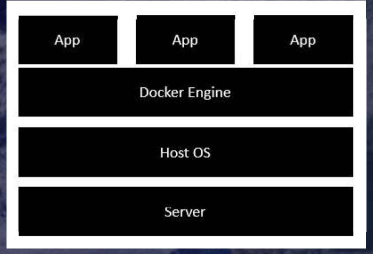
- Docker for Mac – It allows one to run Docker containers on the Mac OS.
- Docker for Linux – It allows one to run Docker containers on the Linux OS.
- Docker for Windows – It allows one to run Docker containers on the Windows OS.
- Docker Engine – It is used for building Docker images and creating Docker containers.
- Docker Hub – This is the registry which is used to host various Docker images.
- Docker Compose – This is used to define applications using multiple Docker containers.

The following image shows the standard and traditional architecture of virtualization.



- The server is the physical server that is used to host multiple virtual machines.
- The Host OS is the base machine such as Linux or Windows.
- The Hypervisor is either VMWare or Windows Hyper V that is used to host virtual machines.
- You would then install multiple operating systems as virtual machines on top of the existing hypervisor as Guest OS.
- You would then host your applications on top of each Guest OS.

The following image shows the new generation of virtualization that is enabled via Dockers. Let's have a look at the various layers.



- The server is the physical server that is used to host multiple virtual machines. So this layer remains the same.
- The Host OS is the base machine such as Linux or Windows. So this layer remains the same.
- Now comes the new generation which is the Docker engine. This is used to run the operating system which earlier used to be virtual machines as Docker containers.
- All of the Apps now run as Docker containers.

The clear advantage in this architecture is that you don't need to have extra hardware for Guest OS. Everything works as Docker containers.

NODE.JS - AN INTRODUCTION

AN ARTICLE BY SAKTHI SREE S, SECOND YEAR, CSE-B

Node.js is a very powerful JavaScript-based platform built on Google Chrome's JavaScript V8 Engine. It is used to develop I/O intensive web applications like video streaming sites, single-page applications, and other web applications. Node.js is open source, completely free, and used by thousands of developers around the world.

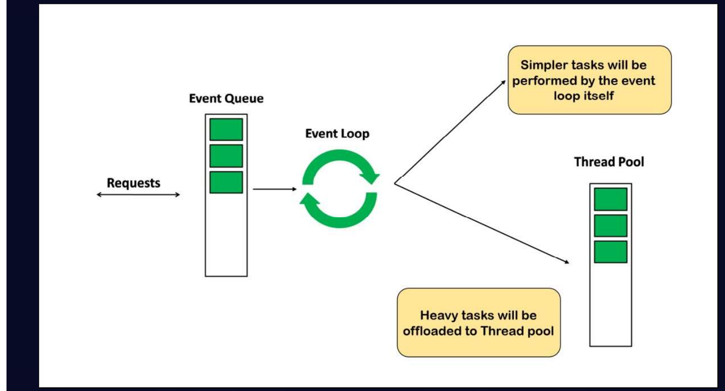
Node.js is an open source, cross-platform runtime environment for developing server-side and networking applications. Node.js applications are written in JavaScript, and can be run within the Node.js runtime on OS X, Microsoft Windows, and Linux.

Node.js also provides a rich library of various JavaScript modules which simplifies the development of web applications using Node.js to a great extent.

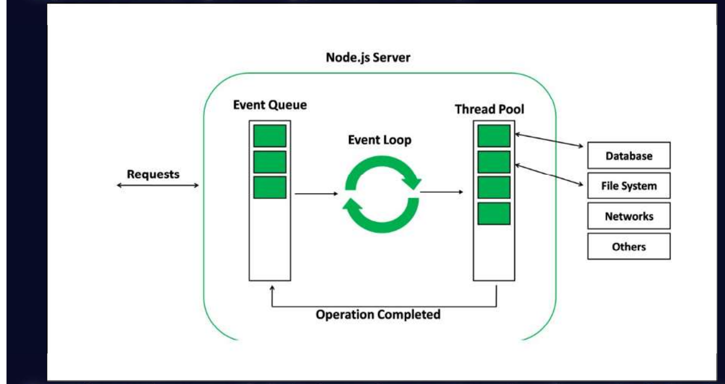
Node.js = Runtime Environment + JavaScript Library

Features of Node.js

- Asynchronous and Event Driven − All APIs of Node.js library are asynchronous, that is, non-blocking. It essentially means a Node.js based server never waits for an API to return data. The server moves to the next API after calling it and a notification mechanism of Events of Node.js helps the server to get a response from the previous API call.
- Very Fast − Being built on Google Chrome's V8 JavaScript Engine, Node.js library is very fast in code execution.
- Single Threaded but Highly Scalable − Node.js uses a single threaded model with event looping. Event mechanism helps the server to respond in a non-blocking way and makes the server highly scalable as opposed to traditional servers which create limited threads to handle requests. Node.js uses a single threaded program and the same program can provide service to a much larger number of requests than traditional servers like Apache HTTP Server.
- No Buffering − Node.js applications never buffer any data. These applications simply output the data in chunks.
- License − Node.js is released under the MIT license.


Event Queue
Event Loop
Thread Pool
Requests
Simpler tasks will be performed by the event loop itself
Heavy tasks will be offloaded to Thread pool

In languages like PHP, we have one new thread created for every task. So, again, if you run your Node application, it will run in just a single thread. No matter if you have 1 user or 100 users or maybe 100 million users accessing your application at the same time. The event loop is called the heart of the node.js. It executes all the callback functions(functions that are called as soon as some work is finished) in a single thread and it also offloads heavy or expensive tasks like compressing a file to a thread pool. Eventloop makes asynchronous programming possible in node.js.



Event loop takes care of all the incoming events and performs the balancing part by offloading heavier tasks into the thread pool, and doing the simpler tasks by itself.

REAL-TIME APPLICATION (RTA)

AN ARTICLE BY RAHUL S, SECOND YEAR, CSE-B

A real-time application (RTA) is an application program that functions within a time frame that the user senses as immediate or current. The latency must be less than a defined value, usually measured in seconds. Whether or not a given application qualifies as an RTA depends on the worst-case execution time (WCET), the maximum length of time a defined task or set of tasks requires on a given hardware platform. The use of RTAs is called real-time computing (RTC).

Increased Situational Awareness
Quickly and easily understand what is going on in and around your business at any given moment. Naturally Leverage IoT Devices & Sensors IoT needs real-time applications in order to efficiently process large amounts of streaming data and take immediate action.

Better Decision Making Capabilities
Put more information directly at your fingertips through the use of smart dashboards, real-time notifications, digital twins, human-machine collaboration, and more.

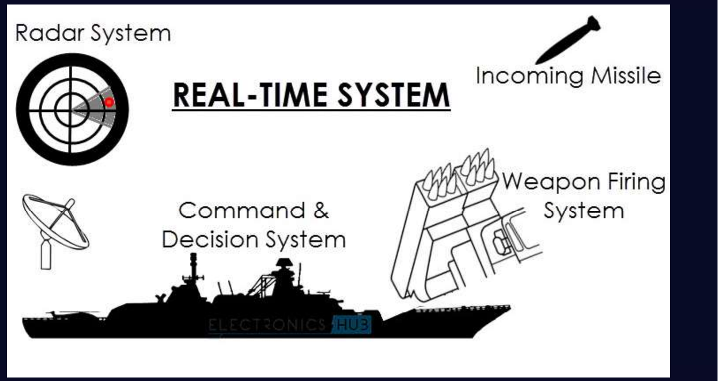
Greater Operational Responsiveness
Respond to mission-critical events as they happen instead of after the fact through real-time analysis and response to streaming data.

High Scalability:
Process data close to the source to scale without limits, and massively decrease bandwidth utilization.

Low Latency Edge Processing
Real-time applications enable very low latency data ingestion and analysis to fully leverage the benefits of edge computing.

Examples of RTAs include:

- Videoconference applications
- VoIP (voice over Internet Protocol)
- Online gaming
- Community storage solutions
- Some e-commerce transactions
- Chatting
- IM (instant messaging)



Consider a Weapons Defense System, whose job is to shoot down incoming missiles and protect the Naval Destroyer. This Weapons Defense System consists of three sub-systems: a Radar System, a Control System and a Weapons Firing System.
The Control System acts as a Command and Decision System, which is a controlling system and the Radar System and Weapons Firing System are the controlled systems. Now we will see how this Real Time System works in a real world scenario.

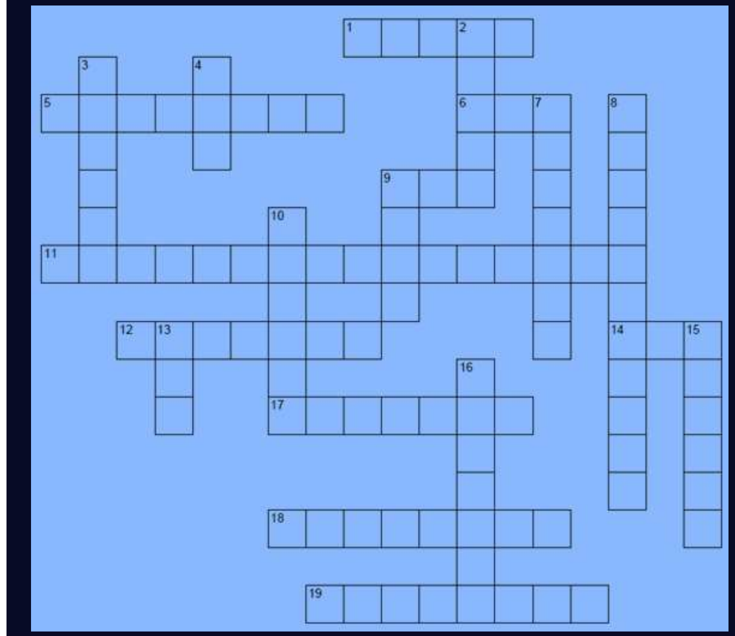
First, the Radar System continuously monitors for potential threats like incoming missiles and measures the coordinates of the targets. It then sends these coordinates to the Control System, which then determines the level of threat possessed by the target based on the information from the Radar System.

The Command and Decision System then calculates different parameters of the target like speed of the missile, flight path and possible point of impact.
Based on the above parameters, the Control System then activates the Weapons Firing System, which fires continuously until the target is destroyed.

In this example, the communication between the Command and Decision System and the Radar system happens in real time i.e. a potential threat can occur at any time and it is unpredictable. The second real time computing is the firing coordinates. Initially, they are determined by the flight path of the target but are updated in real time as per the actual location of the target.

CROSS WORD PUZZLE

BY MARIA SHERINE R, THIRD YEAR, CSE-B

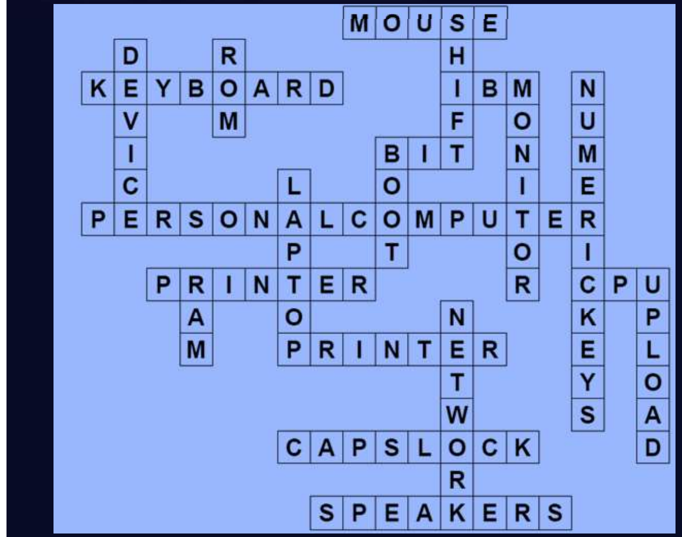


ACROSS

1 An input device
5 The ________ is a mean of sending information to the computer.
6 International Business Machines
9 The smallest piece of computer information, either the number 0 or 1.
11 A single-user computer containing a central processing unit (CPU) and one or more memory circuits.
12 An output device
14 The brain of the computer.
17 A mechanical device for printing a computer's output on paper.
18 change the case of the letter when pressed.
19 These are the devices that the sounds come out of which are sometimes built into the monitor.

DOWN

2 change the case of the letter when pressed.
3 Any hardware unit that is either part of, or is connected to, your PC
4 Read-Only Memory
7 A video display terminal.
8 _______are found on the right side of the keyboard and act much like a calculator pad,
9 To start up a computer
10 Small, lightweight, portable battery powered computers that can fit onto your lap.
13 Random Access Memory
15 The process of transferring information from a computer to a web site
16 A system of interconnected computers.



IT SECURITY AT HOME: HARD BUT NOT IMPOSSIBLE

AN ARTICLE BY DEEPIKA M, THIRD YEAR, CSE-A

When the COVID-19 quarantine hit in mid-March, it created an unprecedented situation in which the number of remote workers skyrocketed beyond anything anticipated.

The vast move to remote work is an exacerbation of the human element that "is often—frankly always—the most uncontrollable component of cybersecurity risk.

Common instructions came from every computing security specialist are starting with the need to equip your computer with a virtual private network (VPN) so that all of your activities are done on your company's network, not on your own, looser, more vulnerable one. This is just one difference between office security and remote security.

"In a workplace environment, you typically have a well-structured, highly controlled work environment where there are tight measures and controls on the type of traffic that can flow, what type of authentication is used, and what type of data can be stored.

At most enterprise or business locations, there are firewalls and the network is monitored by a networking team. Whereas at home, users basically just either doing Comcast or broadband or something like that, and there wont be any monitoring on the data traffic.

So antivirus and malware protection is not enough. need to do regular updates for the protection. And just like always, need to guard against phishing. User should always be at maximum awareness when dealing with messaging vulnerability on their remote connection and private computer, especially when the computer is shared with other family members or used for the personal business as well.

The two-factor authentication is best way to prevent phishing and other authentication attacks.

If user got the ability to use online services through a browser to do most of your work, that helps them segment the corporate data away from personal data. The big example would be to use the browser-based versions of the Microsoft Office programs (now Microsoft 365 apps) rather than the locally running programs.

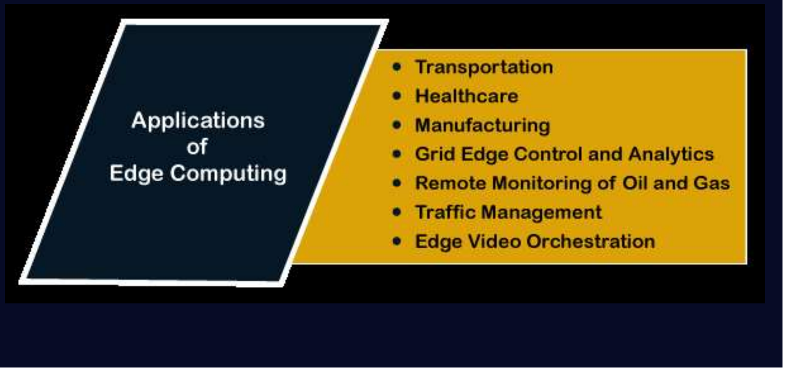
EDGE COMPUTING

AN ARTICLE BY KHADAR C, THIRD YEAR, CSE-C

Edge computing is a form of computing that is done on site or near a particular data source, minimizing the need for data to be processed in a remote data center.

dge Computing is a new type of technology that will not only save time but also save the cost of servicing and other charges like:

- Through edge computing, it allows smart applications and devices for responding to data very quickly as soon as it is created and thus removing the lag time.
- Edge Computing also enables data stream acceleration that includes real-time processing of data without latency use. Data Stream acceleration is, however, critical for self-driving cars type of technologies and provides equal and vital benefits for the businesses.
- Efficient processing of data at large scale by allowing processing close to the source, and it saves the use of internet bandwidth also. Thus, it reduces the cost and enables effective accessibility to the applications in remote locations.
- The ability of edge computing to provide services and processing data at the furthest distance makes a secured layer for the sensitive data without keeping it into a public cloud.



Applications of Edge Computing

- Transportation
- Healthcare
- Manufacturing
- Grid Edge Control and Analytics
- Remote Monitoring of Oil and Gas
- Traffic Management
- Edge Video Orchestration