Gayatri Vidya Parishad College of Engineering For Women

tech niyati

TECH NIYATI

Quarterly Technical Magazine Volume No. 3 Issue No. 2 2018

Institute Vision

To emerge as an acclaimed centre for learning that provides value based technical education for the holistic development of students.

Institute Mission

Undertake activities that provide value based knowledge in science, engineering & Technology.

Provide opportunities for learning through industry-institute interaction on the state-of-the-art technologies.

Create collaborative environment for research, innovation and entrepreneurship to flourish.

Promote activities that bring in a sense of social responsibility.

Editorial Message

We hereby convey our immense happiness with the release of yet another successful issue of the technical magazine "*TechNiyati*". This time, the cover story focuses on the buzz word 'big data'. One regular article is well placed that inherently talks about reinforcement learning. Trending news and trending articles are suitably placed in regular manner to enlighten one's insight into latest technology scenario. Workbench section featuring the control of PC using NodeMCU will definitely be liked by the readers. Other contents of the magazine like '*mysteries of India*', '*riddles*', and '*fun facts*' will be worth enjoyable by the readers.

Thank you for showing your interest in "*TechNiyati*". We always value your genuine feedback and suggestion towards continuous growth of the outlook of the magazine.

Editorial team

editor@gvpcew.ac.in

Cover Story:

History of Big Data

Kalli Bala Parameswaran

Senior Software Architect, Equinix

First time the term "big data" was used in a context in July, 1997 [1]. Michael Cox and David Ellsworth, scientists/ researchers from NASA, described the problem they faced when processing humongous amounts of data with the traditional computers of that time. In the early 2000s, Lexis Nexis has designed a proprietary system, which later went on to become the High-Performance Computing Cluster (HPCC), to address the growing needs of processing data on a cluster. It was later open sourced in 2011.

In the era of dot coms, Google was challenging the limits of the internet by crawling and indexing the entire internet. With the rate at which the internet was expanding, Google knew it would be difficult; if not impossible to scale vertically to process data of that size. Distributed computing, though still in its infancy, caught Google's attention. They not only developed a distributed fault tolerant file system, Google File System (GFS), but also a distributed processing engine/system called Map Reduce. It was then in 2003-2004 that Google released the white paper titled "*The Google File System*" [2] by Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, and shortly thereafter they released another white paper titled Map Reduce: Simplified Data Processing on Large Clusters by Jeffrey Dean and Sanjay Ghemawat [3].

Doug Cutting, an open source contributor, around the same time was looking for ways to make an open source search engine and like Google; was failing to process the data at the internet scale. By 1999, Doug Cutting had developed *Lucene*, a Java library with the capability of text/web searching among other things. Nutch, an open source web crawler and data indexer built by Doug Cutting along with Mike Cafarella, was also not scaling well. Above said Google's white paper has caught Doug Cutting's attention. He began working on similar concepts calling them Nutch Distributed File System (NDFS) and Nutch Map Reduce. By 2005, he was able to scale Nutch, which could index 100 million pages to multi-billion pages using the distributed platform.

However, it wasn't just Doug Cutting but Yahoo! also started showing interest in the development of the Map Reduce computing framework to serve its processing capabilities. It is here that Doug Cutting re-factored the distributed computing framework of Nutch and named it after his kid's elephant toy, Hadoop. By 2008, Yahoo! was using Hadoop in its production cluster to build its search index and metadata called web map. Despite being a direct competitor to Google, one distinct strategic difference that Yahoo! took while co-developing Hadoop was the nature in which the project was to be developed: they open sourced it. And the rest, as we know is history! A timeline representing the evolution of Big Data is shown in Fig. 1.

Dimensioning the Big Data

Big data can be best described by using its dimensions. Those dimensions are called the Vs of big data. To categorize a problem as a big

data problem, it should lie in one or more of these dimensions.

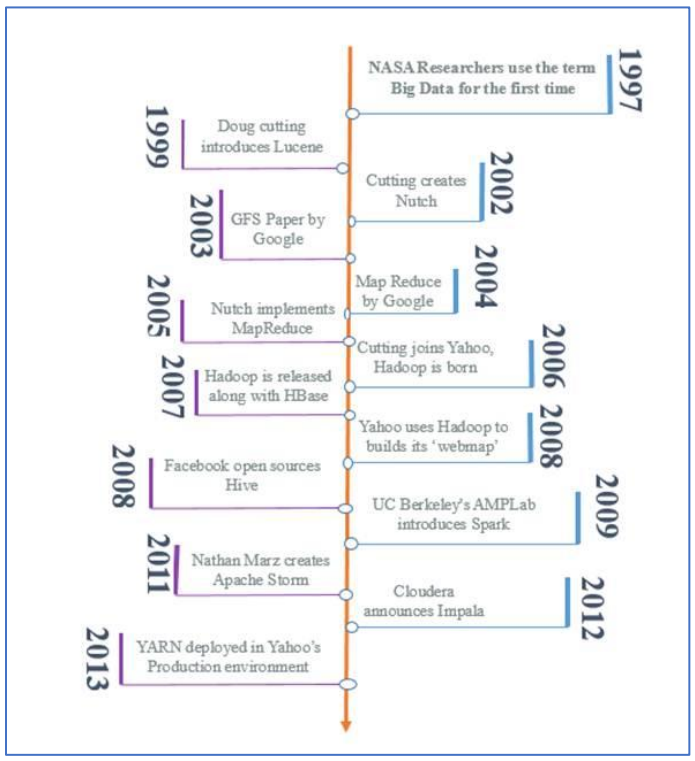


Fig. 1. Representing Evolution in a Timeline.

The big data world started with three dimensions or 3Vs of big data, which are as follows:

- Volume
- Variety
- Velocity

Let us now take a look at each one in detail:

- **Volume**: The amount of data being generated in the world is increasing at an exponential rate. So, to analyze the amount of data being generated, they need to find a solution out of the existing RDBMS world. Performing analytics on such a humongous amount of data is a big data problem.
- **Variety**: There was a time when only structured data was meant to be processed. In real world you often need to analyze every sort of data that can be structured or unstructured which can increase value of the business.
- **Velocity**: Data is not only increasing in size but the rate at which it is arriving is also increasing rapidly. Take the example of Twitter: billions of users are tweeting at a time. Twitter has to handle such a high velocity of data in almost real time. Also, you can think of YouTube. A lot of videos are being uploaded or streamed from YouTube every minute. So, this dimension of big data deals with a high velocity of data and helps to provide persistence or analyze the data in near real time so as to generate real value.
- **Veracity**: The truthfulness and completeness of the data is really important. Take an example of a machine learning algorithm that involves automated decision making based on the data it analyzes. If the data is not accurate, this system can be disastrous. If the data that is fed to such a system is inaccurate or incomplete, analytics will not be meaningful or beneficial for the system. So, as per this dimension, before processing/analyzing, data should be validated.
- **Variability:** This dimension of big data mainly deals with the quality or degree of being subject to variation i.e., natural language processing or sentiment analytics. In language, one word can have multiple usages based on the sentiments of the user. So, to find sentiments, you should be able to comprehend the exact meaning. To analyze the sentiments, the system should be fed with lot of other information such as the statistics of the events, and so on. Semantic analytics or natural language processing can only be

accurate if you can understand sentiments behind the data.

- **Value:** There is lot of cost involved in performing big data analytics: the cost of getting the data, the cost for arranging hardware on which this data is saved and be analyzed, the cost of employees and time that goes into these analytics. All these costs are justified if the analytics provide value to the organization.
- **Visualization:** Visualization is another important aspect of the analytics. No work can be useful until it is visualized in a proper manner. Visualization is a really important aspect of big data analytics because things can only be highlighted if they are visible.

Components of Hadoop

In a classical sense, Hadoop comprises of two components: a storage layer called HDFS and a processing layer called Map Reduce. Resource management task prior to Hadoop 2.X was done using the Map Reduce framework of Hadoop itself. However, that changed with the introduction of YARN. In Hadoop 2.0, YARN was introduced as the third component of Hadoop to manage the resources of the Hadoop cluster and make it more Map Reduce agnostic. The components are shown in Fig. 2.

Defining HDFS

The Hadoop Distributed File System **(**HDFS**),** as the name suggests, is a distributed file system based on the lines of the Google File System written in Java. In practice, HDFS resembles closely any other UNIX file system with support for common file operations such as ls, cp, rm, du, cat, and so on.

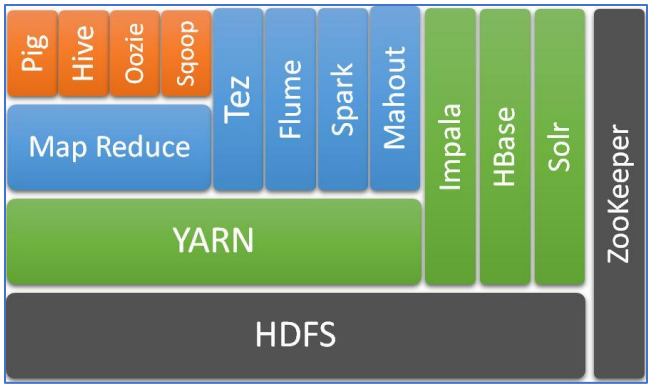


Fig. 2. Components of Hadoop.

However what makes HDFS stand out, despite its simplicity, is its mechanism to handle node failure in the Hadoop cluster without effectively changing the search time for accessing stored files. The HDFS cluster consists of two major components: Data Nodes and Name Node.
HDFS has a unique way of storing data on HDFS clusters (cheap commodity networked computers). It splits the regular file in smaller chunks called blocks and then makes an exact number of copies of such chunks depending on the replication factor for that file. After that, it copies such chunks to different Data Nodes of the cluster.

Name Node: Name Node is the centerpiece of HDFS and if only stores the metadata of HDFS – the directory tree of all files in the file system, and tracks the files across the cluster.

Data Node: Data Node is responsible for storing the actual data in HDFS and Name Node and Data Node are in constant communication. Data Node is usually configured with a lot of hard disk space. This is because the actual data is stored in the Data Node.

HDFS I/O:
An HDFS read operation from a client involves the following:

1. The client requests Name Node to determine where the actual data blocks are stored for a given file.
2. Name Node obliges by providing the block IDs and locations of the hosts (Data Node) where the data can be found.
3. The client contacts Data Node with the respective block IDs to fetch the data from Data Node while preserving the order of the block files.

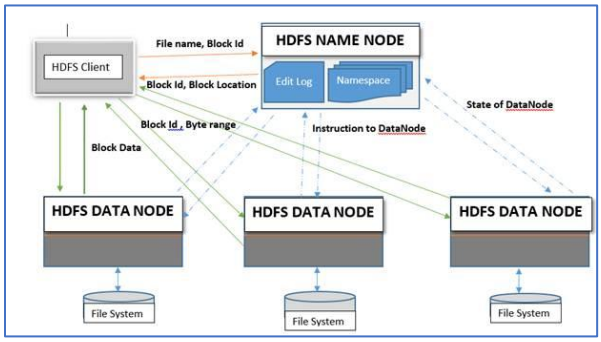


Fig. 3. The HDFS architecture.

An HDFS (Fig. 3) write operation from a client involves the following:

1. The client contacts Name Node to update the namespace with the filename and verify the necessary permissions.
2. If the file exists, then Name Node throws an error; otherwise, it returns the client FSDataOutputStream which points to the data queue.
3. The data queue negotiates with the Name Node to allocate new blocks on suitable Data Nodes.
4. The data is then copied to that Data Node, and, as per the replication strategy, the data is further copied from that Data Node to the rest of the Data Nodes.
5. It's important to note that the data is never moved through the Name Node as it would cause a performance bottleneck.

YARN

The simplest way to understand YARN (Yet-Another-Resource-Negotiator) is to think of it as an operating system i.e., Data Operating System on a cluster; provisioning resources, scheduling jobs and node maintenance. With Hadoop 2.x, the Map Reduce model of processing the data and managing the cluster (Job Tracker/Task Tracker) was divided. While data processing was still left to Map Reduce, the cluster's resource allocation (or rather, scheduling) task was assigned to a new component called YARN. Another objective that YARN met was that it made Map Reduce one of the techniques to process the data rather than being the only technology to process data on HDFS, as was the case in Hadoop 1.x systems. This paradigm shift opened the floodgates for the development of interesting applications around Hadoop and a new ecosystem other than the classical Map Reduce processing system evolved. It didn't take much time after that for Apache Spark to break the hegemony of classical Map Reduce and become arguably the most popular processing framework for parallel computing as far as active development and adoption is concerned.

Components of YARN

In order to serve multi-tenancy, fault tolerance, and resource isolation in YARN, it developed the following components as shown in Fig. 4 to manage the cluster seamlessly:

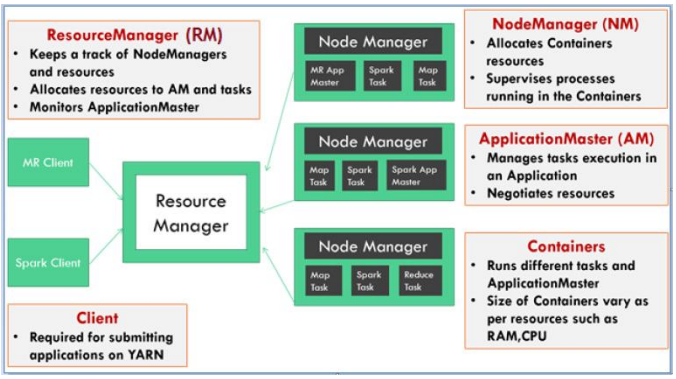


Fig. 4. Components of YARN.

Overview of Map Reduce

Map Reduce is a framework for processing large datasets in parallel across the Hadoop clusters. It comprises of two mutually exclusive but dependent phases, each capable of running on two different machines or nodes:

Map: In the Map phase, the transformation of the data takes place. It splits data into key value pairs by splitting it on a keyword. Process Flow of application submission in YARN is shown in Fig. 5.

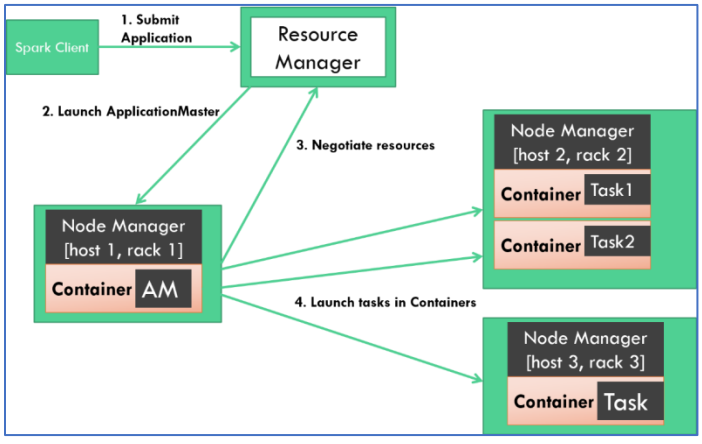


Fig. 5. Process flow of application submission in YARN.

Suppose we have a text file and we would want to do an analysis such as counting the total number of words or even the frequency with which the word has occurred in the text file. This is the classical word count problem of Map Reduce. To address this problem, first we will have to identify the splitting keyword so that the data can be spilt and be converted into a key value pair.

Let's begin with *John Lennon's* song, *Imagine*.

Sample text:

> Imagine there's no heaven
> It's easy if you try
> No hell below us
> Above us only sky
> Imagine all the people living for today

After running the Map phase on the sampled text and splitting it over <space>, it will get converted to a key value pair as shown here:

> <imagine, 1><there's, 1><no, 1><heaven, 1><it's, 1><easy, 1><if, 1><you, 1><try, 1><no, 1><hell, 1><below, 1><us, 1><above, 1><us, 1><only, 1><sky, 1><imagine, 1><all, 1><the, 1><people, 1><living, 1><for, 1><today, 1>]

The key here represents the word and the value represents the count. Also it should be noted that we have converted all the keys to lowercase to reduce any further complexity arising out of matching case sensitive keys.

Reduce: The Reduce phase deals with aggregation of the Map phase results and hence all the key value pairs are aggregated over the key.

So the Map output of the text would get aggregated as follows:

> [<imagine, 2><there's, 1><no, 2><heaven, 1><it's, 1><easy, 1><if, 1><you, 1><try, 1><hell, 1><below, 1><us, 2><above, 1><only, 1><sky, 1><all, 1><the, 1><people, 1><living, 1><for, 1><today, 1>]

As we can see, both the Map and Reduce phases can be run exclusively and hence can use independent nodes in the cluster to process the data. This approach of separation of tasks into smaller units called Map and Reduce has revolutionized general purpose distributed/parallel computing, which we now know as Map Reduce.

Apache Hadoop's Map Reduce has been implemented pretty much the same way as discussed, except for adding extra features into how the data from the Map phase of each node gets transferred to their designated Reduce phase node (Fig. 6).

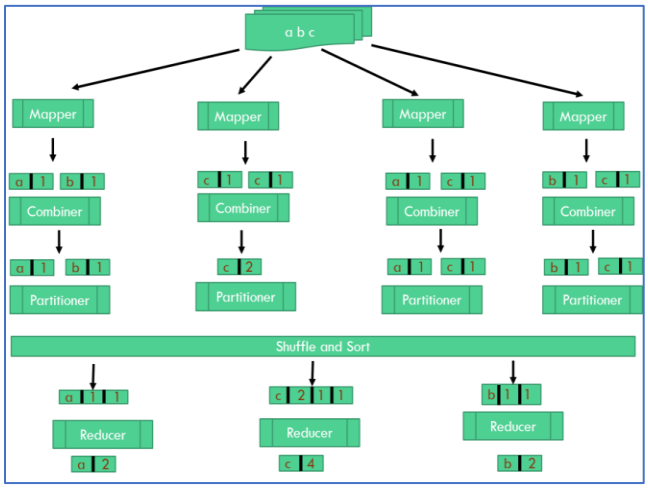


Fig. 6. A snapshot of the implementation of reduce strategy.

Hadoop's implementation of Map Reduce enriches the Map and Reduce phases by adding a few more concrete steps in between to make it fault tolerant and truly distributed. We can describe MR jobs on YARN in five stages (Fig. 7):

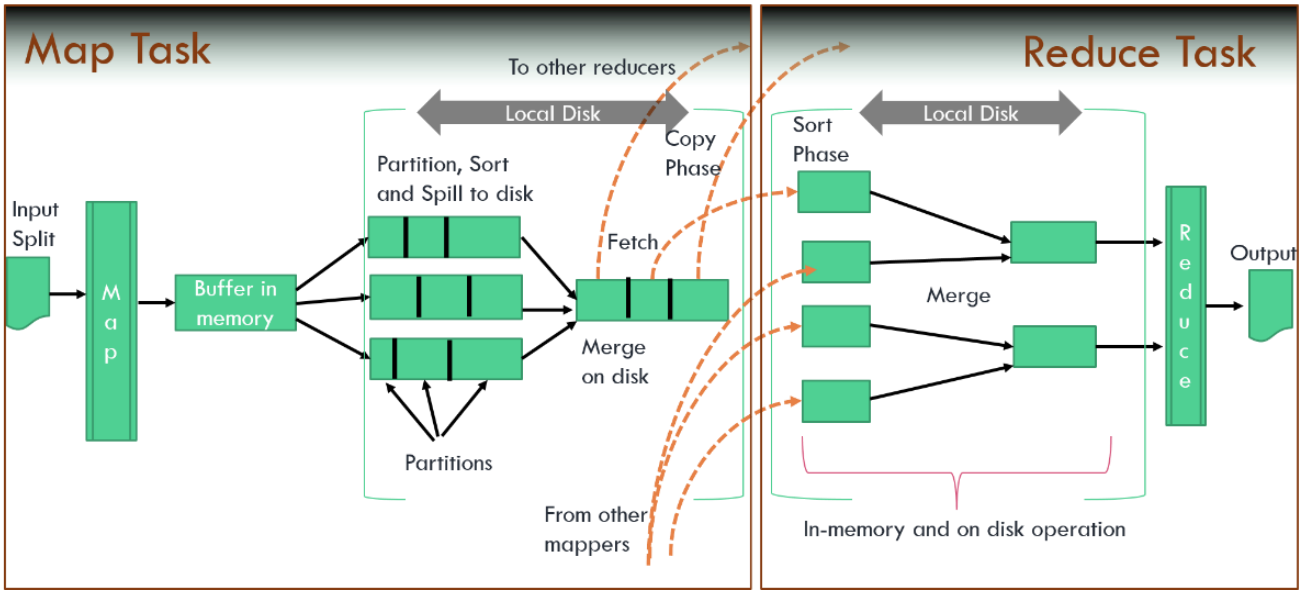


Fig. 7. Stages in Map Reduce.

1. Job Submission Stage: When a client submits an MR job, the following things happen:
   a. The RM is requested for an application ID
   b. The input data location is checked and if present then the file split size is computed
   c. The job's output location needs to exist as well

   If all the three conditions are met, then the MR job jar along with its configuration details of input split are copied to HDFS in a directory named the application ID provided by RM. Then the job is submitted to RM to launch a job-specific Application Master, MRAppMaster.

2. MAP Stage: Once RM receives the client's request for launching MRAppMaster, a call is made to the YARN scheduler for assigning a container. As per the resource availability, the container is granted and hence the MRAppMaster is launched at the designated node with provisioned resources. After this, MRAppMaster fetches input split information from the HDFS path that was submitted by the client and computes the number of mapper tasks that will be launched based on the splits. Depending on the number of

mappers, it also calculates the required number of reducers as per the configuration, If MRAppMaster now finds the number of mapper, reducer and size of input files to be small enough to be run in the same JVM, then it goes ahead in doing so. Such tasks are called Uber tasks. However, in other scenarios, MRAppMaster negotiates container resources from RM for running these tasks, albeit mapper tasks have a higher order and priority. This is why Mapper tasks must finish before the sorting phase can start.

Data locality is another concern for containers hosting mappers, as local data nodes are preferred over rack locals, with the least preference being given to remote node hosted data. But when it comes to the Reduce phase no such preference of data locality exists for containers. Containers hosting function mappers first copy map Reduce JAR and configuration files locally and then launch a class called Yarn Child in the JVM. The mapper then starts reading the input file, processes them by making key value pairs, and writes them in a circular buffer.

3. Shuffle and Sort Phase: Considering that circular buffers have a size constraint, after a certain percentage, the default being 80, a thread gets spawned which spills the data from the buffer. But, before copying the spilled data to disk, it is first partitioned with respect to its reducer and then the background thread also sorts the partitioned data on a key and if the combiner is mentioned it then combines the data too. This process optimizes the data once it is copied to its respective partitioned folder. This process is continued until all the data from circular buffer gets written to disk. A background thread again checks if the number of spilled files in each partition is within the range of the configurable parameter or else the files are merged and the combiner is run over them until it falls within the limit of the parameter.

   A Map task keeps updating the status to ApplicationMaster for its entire life cycle. It is only when 5 percent of a Map task has been completed that the Reduce task starts. An auxiliary service in the NodeManager serving the Reduce task starts a Netty web server that makes a request to MRAppMaster for Mapper hosts having specific Mapper partitioned files. All the partitioned files that pertain to the Reducer are copied to their respective nodes in a similar fashion. Since multiple files get copied as data from various nodes representing that Reduce nods gets collected, a background thread merges the sorted map file and again sorts them and if the combiner is configured, then combines the result too.
4. Reduce Stage: It is important to note here that at this stage every input file of each reducer should have been sorted by key. This is the presumption with which the reducer starts processing these records and converts the key value pair into an aggregated list. Once the reducer has processed the data, it writes them to the output folder as was mentioned during the job submission.
5. Clean-up: Each reducer sends a periodic update to MRAppMaster about the task completion. Once the Reduce task is over, the ApplicationMaster starts the clean-up activity. The submitted job status is changed from running to successful, and all the temporary and intermediate files and folders are deleted. The application statistics are archived to a job history server.

References

[1] "*What is Data Science? Fundamental Concepts and a Heuristic Example*", In Hayashi Chikio, Yajima Keiji, Bock Hans-Hermann, Ohsumi Noboru, Tanaka Yutaka and BabaYasumasa, Data Science, Classification, and Related Methods Studies in Classification, Data Analysis, and Knowledge Organization. Springer Japan, pp. 40–51, 1998.

[2] "*The Google File System*", Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, Google SOSP'03, October 19–22, 2003, Bolton Landing, New York, USA.

[3] "*MapReduce: Simplified Data Processing on Large Clusters*", Jeffrey Dean and Sanjay Ghemawat, 2004, OSDI'04: Sixth Symposium on Operating System Design and Implementation, 137–150, San Francisco, CA.

. . . Mysteries of India[2] (*Excerpt from factslegend.org*)

Starting July 25, 2001, Kerala experienced an unusual phenomenon. It was monsoon and it rained. While that falls pretty much into the natural cycle as we know but what was really unusual was that this time, the color of the rain was red. This red rain continued till September 23, 2001. Prior to that, red rain in Kerala was reported in 1896. Most recently, the phenomenon was reported in July 2012. The blood rain followed a specific pattern. It always rained over a much localized area of no more than a few square kilometers. Why was the water red? This question led scientists to test it. They found that about 9 million red particles were present in every millimeter of the blood rain. They further stated that in every one liter of the water from blood rain contain approximately 100 mg of solid. Based on these calculations, the scientists told that for the total amount of red rain that poured down on Kerala, a total of 50,000 kilograms of the red particles came down.

Upon further examination, the scientists found that the solid particles they separated from the water were brownish-red in color with 90% of the solid made up of round particles and remain were debris. But, the boldest of all explanations and this theory in particular stirred up the case of red rain globally. This theory was proposed by Kottayam's Mahatma Gandhi University's Santhosh Kumar and Godfrey Louis. Both these physicists proposed that the red cells were extraterrestrial life form released in the atmosphere by the meteor blast.

Article:

Reinforcement Learning

M/s. V. Gowtami Annapurna

Assistant Professor, CSE

Reinforcement Learning is a type of Machine Learning, and thereby also a branch of Artificial Intelligence. It allows machines and software agents to automatically determine the ideal behavior within a specific context, in order to maximize its performance. Simple reward feedback is required for the agent to learn its behavior; this is known as the reinforcement signal. Reinforcement learning, in the context of artificial intelligence, is a type of dynamic programming that trains algorithms using a system of reward and punishment.

Reinforcement learning is an approach to machine learning that is inspired by behaviorist psychology. It is similar to how a child learns to perform a new task. Reinforcement learning contrasts with other machine learning approaches in that the algorithm is not explicitly told how to perform a task, but works through the problem on its own.

As an agent, which could be a self-driving car or a program playing chess, interacts with its environment, receives a reward state depending on how it performs, such as driving to destination safely or winning a game. Conversely, the agent receives a penalty for performing incorrectly, such as going off the road or being checkmated. The agent over time makes decisions to maximize its reward and minimize its penalty using dynamic programming. The advantage of this approach to artificial intelligence is that it allows an AI program to learn without a programmer spelling out how an agent should perform the task.

You may start with a blank state, and then strive to reach the goal, under the right conditions. Just like a whiz kid tries out different ways to achieve his goals, through trial and error, learning from his mistakes, Reinforcement Learning is attaining success through a series of steps. These steps may vary widely from problem to problem. But the result is the same; making the right decisions and getting rewarded when you make the right ones – this is reinforcement (Fig. 1).

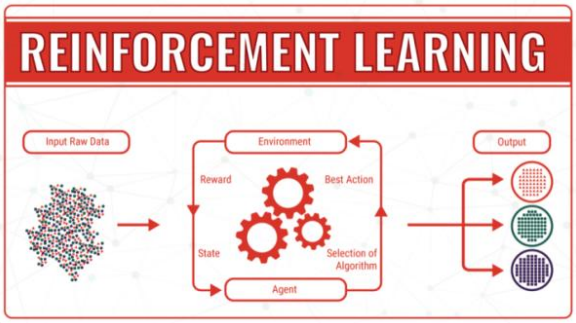


Fig. 1. Reinforcement learning process.

Why Reinforcement Learning?

Reinforcement Learning allows the machine or software agent to learn its behavior based on feedback from the environment. This behavior can be learnt once and for all, or keep on adapting as time goes by. If the problem is modeled with care, some Reinforcement Learning algorithms can converge to the global optimum; this is the ideal behavior that maximizes the reward.

When does Reinforcement Learning fail?

There are many challenges in current Reinforcement Learning research. Firstly, it is often too memory expensive to store values of each state, since the problems can be pretty complex. Solving this involves looking into value approximation techniques, such as Decision Trees or Neural Networks. There are many consequences of introducing these imperfect value estimations, and research tries to minimize their impact on the quality of the solution.
Moreover, problems are also generally very modular; similar behaviors reappear often, and modularity can be introduced to avoid learning everything all over again. Hierarchical approaches are common-place for this, but doing this automatically is proving a challenge. Finally, due to limited perception, it is often impossible to fully determine the current state. This also affects the performance of the algorithm, and much work has been done to compensate this Perceptual Aliasing.

Reinforcement learning components (Fig. 2):
Agent: Our robot.
Environment: The game, or where the agent lives.
A set of states, $s \in S$.
Policy: Map between states to actions.
Reward Function $R(s,a,s')$: Gives immediate reward for each state.

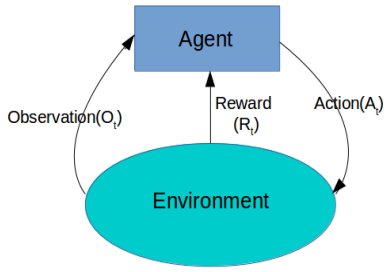


Fig. 2. Components of reinforcement learning.

Approaches to Reinforcement Theory of Learning:

Value Based Learning Approach:

Value-based Learning estimates the optimal value function, which is the maximum value achievable under any policy. Storing the value function (or) policy might not be possible especially if the state-action pairs are high dimensional. Thus, function approximates like linear regression and neural networks are used. In value-based RL, the goal is to optimize the value function V(s). The value function is a function that tells us the maximum expected future reward the agent will get in each state.
The value of each state is the total amount of the reward an agent can expect to accumulate over the future, starting in that state. Then the agent uses this value function to select which state to choose at each step. The agent decides to take up the state with the biggest value.

Policy-Based Learning Approach:

Policy-based Learning searches directly for the optimal policy which achieves the maximum future reward. In policy-based approach, we want to directly optimize the policy function $\pi(s)$ without using a value function. The policy is what defines the agent behavior at a given time. We learn a policy function. This lets us map each state to the best corresponding action.
This approach has two types of policy:
Deterministic: a policy at a given state will always return the same action.
Stochastic: output a distribution probability over actions.

Model-Based Learning Approach:

In Model-based RL, the environment is treated as a model for learning. This means a model of the environmental behavior is created. This is

a great approach until you discover that each environment will need a different model representation.

Reinforcement Learning Applications:

(i) *Robotics and Industrial Automation:*

Reinforcement Learning (RL) enables a robot to autonomously discover an optimal behavior through trial-and-error interactions with its environment. In Reinforcement Learning, the agent (i.e., the designer of a control task) provides constructive feedback in terms of a scalar objective function that measures the one-step performance of the robot. This serves as a guideline for deciding the next action.
Industrial automation is another major area where Reinforcement Learning has contributed significantly. A classic example would be of Google, which has reduced energy consumption (HVAC) in its own data centres through RL technologies from DeepMind. Start-ups like Bonsai use RL for industrial applications.

(ii) *Data Science and Machine Learning:*

With machine learning libraries becoming more accessible, deep learning techniques are widely being used by data scientists and machine learning engineers, to help people identify and tune neural network architectures are active areas of research. Several research groups have used RL to make the process of designing neural network architectures easier. AutoML from Google, for example, uses RL to produce state-of-the-art machine-generated neural network architectures for computer vision and language modeling.

(iii) *Education and Training:*

Reinforcement Learning is already showing ripples in online tutorials and virtual classrooms. Deep Learning researchers are looking for new ways use RL and other machine learning methods in online tutoring systems and personalized learning. Reinforcement learning tutorials will be instrumental in providing custom instruction and materials to serve the needs of individual students. RL algorithms and statistical methods may also be developed in such a way that requires less data for use in future tutoring systems.

(iv) *Healthcare:*

Healthcare is another area where Reinforcement Learning is fast creating impressions. The RL setup of an agent may interact with an environment receiving feedback based on actions taken. Several RL applications in health care mostly pertain to finding optimal treatment policies. Deep learning scientists are researching on RL applications that serve the purpose of medical equipment, medication dosing, and two-stage clinical trials.
Some of the other applications of Reinforcement Learning include cross-channel marketing optimization and real-time bidding systems for online display advertising.

References:

[1] Abhijit Gosavi, "T Reinforcement Learning: A Tutorial Survey and Recent Advances", Journal of Informs Journal on Computing, vol. 21, no. 2, pp. 178-192, 2009.

[2] Richard S. Sutton and Andrew G. Barto, "Reinforcement Learning: An Introduction", Bradford Book, The MIT Press, Cambridge, 2015.

[3] Richard S. Sutton and Andrew G. Barto, "Reinforcement Learning: An Introduction", Second Edition, MIT Press, Cambridge, MA, 2018.

. . . Fun Facts (*Excerpt from globallead.co.za, Feb-2018*)

- ✓ PCs were initially named as “Electronic Brains” in the 1950s.
- ✓ Email is older than the World Wide Web.
- ✓ HP, Google, Microsoft and Apple have just one thing in common, other than the fact that they are IT companies. They were all started in garages.
- ✓ Bill Gates’ house was designed by using a Mac computer.
- ✓ There are approximately 6000 new viruses released every month.
- ✓ Computer programming is currently one of the fastest growing occupations related to technology.
- ✓ 28% of IT professionals hide their career from friends and family to get out of giving free tech support.
- ✓ The 30th of November is known as “Computer Security Day”.
- ✓ Technophobia is the fear of technology, Nomophobia is the fear of being without a mobile phone, Cyberphobia is the fear of computers.
- ✓ The original name of Windows was Interface Manager.
- ✓ The QWERTY keyboard was designed to slow you down. If you want to type faster, try the Dvorak Keyboard.
- ✓ Currently, the world’s largest hard drive is a 60TB SSD.
- ✓ The Apple II had a hard drive of only 5 megabytes when it was launched.
- ✓ 51% of internet traffic is “non-human”. 31% is made up from hacking programs, spammers and malicious phishing.
- ✓ The first computer was almost 2.5 meters high and weighed nearly 30,000kg.
- ✓ The name Google was created accidentally. A spelling error was made by the original founders who were under the impression they were going for Googol.
- ✓ The average computer user blinks 7 times a minute, less than half the normal rate of 20.
- ✓ The first computer mouse was invented by Doug Engelbart and it was carved from wood.

Trending Article:

Advanced Driver-Assistance Systems: A path toward autonomous vehicles

V. K. Kukkala, J. Tunnell,
S. Pasricha and T. Bradley

Excerpt from IEEE Consumer Electronics Magazine, vol. 7, no. 5, pp. 18-25, Sept. 2018

Advanced driver-assistance systems (ADASs) have become a salient feature for safety in modern vehicles. They are also a key underlying technology in emerging autonomous vehicles. State-of-the-art ADASs are primarily vision based, but light detection and ranging (lidar), radio detection and ranging (radar), and other advanced-sensing technologies are also becoming popular (Fig. 1 & 2). Several original equipment manufacturers (OEMs) have attempted to address this issue by developing various safety systems to protect occupants within a vehicle as well as prevent injuries to people outside the vehicle. These systems are mainly classified into two types:

1) Passive (or reactive), and
2) Active (or proactive).

Passive safety systems protect vehicle occupants from injuries after a crash, e.g., seat belts, air bags, and padded dashboards. Due to a consistent consumer demand for safer vehicles, passive safety systems that have been under continuous development for many decades have been augmented by active safety systems, which seek to prevent a crash from happening altogether. Active systems are one of the main areas of interest and have seen major growth in today's vehicles. Examples of such systems include lane keeping, automatic braking, and adaptive cruise control. These systems are commonly known as ADASs and are becoming increasingly popular as a way for automotive manufacturers to differentiate their offerings while promoting safety.

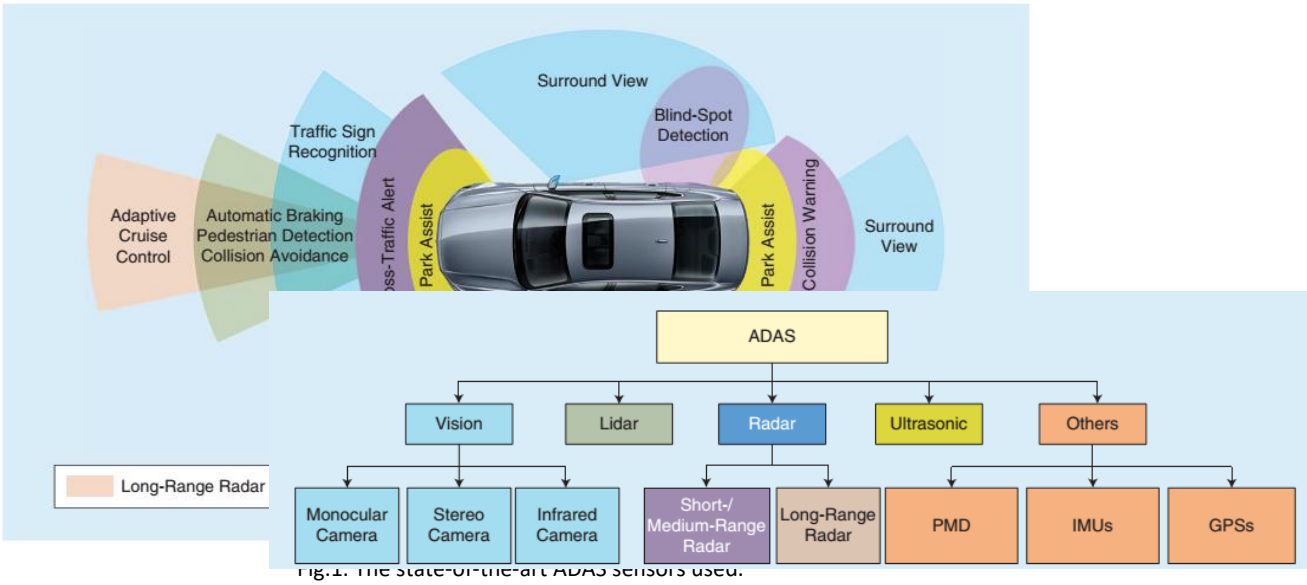


Fig.1. The state-of-the-art ADAS sensors used.

Fig.2. The taxonomy of an ADAS.

Sensor Fusion:

Sensor fusion refers to combining information from multiple homogenous or heterogeneous sensors to find a single best estimation of the state of the environment. Fusion helps sensors complement each other's limitations and offers greater leverage to the system compared to a system with individual sensors. Sensor fusion offers high precision, reliability, robustness to uncertainty, extended spatial and temporal coverage, and improved resolution, which are crucial in safety critical systems, such as vehicles. Although this comes at a higher computation cost, the computation power available in modern-day cars and the reducing cost of the sensors are facilitating the widespread integration of these systems. The classification of different levels of sensor fusion along with the most commonly used techniques for fusing data are discussed in. The growing interest in deep learning and other ML methods in recent years has driven researchers toward exploring more efficient and intelligent techniques that enhance ADASs with sensor fusion capabilities.

V2X Communication:

V2X communication represents a class of communication systems that provide the vehicle with an ability to exchange information with other systems in the environment. Examples include vehicle-to-vehicle (V2V) for collision avoidance, vehicle-to-infrastructure (V2I) for traffic signal timing, vehicle-to-network for real-time traffic updates, and vehicle-to-pedestrian for pedestrian signaling. State-of-the-art V2X communication is based on either dedicated short-range communications (DSRC) or cellular networks. The IEEE 1609 family of standards for Wireless Access in Vehicular Environment (WAVE), which is developed based on the IEEE 802.11p standard, defines architecture and a set of services and interfaces to enable DSRC-based secure V2V and V2I communication.

Autonomous Vehicles:

Next-generation ADASs using sensor fusion and V2X communication are paving the way for autonomous driving. The Society of Automotive Engineers (SAE) J3016 standard defines six different levels of driving automation for on-road vehicles. A vehicle is categorized as level zero if there are no ADASs assisting the driver in handling steering and acceleration/deceleration and everything is handled manually by the driver. Level-one vehicles consist of ADAS's assisting the driver in handling the steering, acceleration, or deceleration under certain cases with human driver input. ADASs in level-two vehicles handle all the steering, acceleration, and deceleration under certain environments with human driver input. In general, in lower-level vehicles (levels zero to two), the driver monitors the driving environment. In contrast, ADAS monitors the driving environment in higher-level (levels three to five) vehicles. Modern vehicles with the top-of-the-line ADASs, such as the 2016 Tesla models are level three,

where multiple safety systems are handled by them, but the driver intervenes when it is needed. Level four vehicles handle multiple safety systems and operate in a wider range of environments. Level-five automation is the end goal of autonomous driving, where all of the systems in the car are operated by the ADAS, under all driving conditions (such as snow-covered roads and unmarked dirt roads) and would not require any human intervention. This, however, still requires significant advances in multiple areas, such as sensor d

Riddles (Excerpted from *boredpanda.com*)

* A murderer is condemned to death; he has to choose between three rooms:

  (1) The first is full of raging fires;

  (2) In the second one, lions who have not eaten anything for years;

  (3) Third one is full of assassins with loaded guns.

  Which room he should choose as safest?

* What can you hold in your right hand, but not in your left?

* 16, 06, 68, 88, ? , 98 (What will come in '?')

* What was the largest island in the world before Australia was discovered?

* I am the *first* on earth, the *second* in heaven. I appear *twice* in a week, though you can see me *once* in a year. What I am?

[For solutions: Browse some other pages of this magazine.]

Trending news:

Excerpt from *scitechdaily*

November, December, 2018.

1. **IMAGING TECHNIQUES WILL HELP BRING FUSION ENERGY CLOSER TO REALITY**

(*Excerpt from scitechdaily, Dec-2018*)

Harnessing nuclear fusion, which powers the sun and stars, to help meet earth's energy needs, is a step closer after researchers showed that using two types of imaging can help them assess the safety and reliability of parts used in a fusion energy device. Scientists from Swansea University, Culham Centre for Fusion Energy, ITER in France, and the Max-Planck Institute of Plasma Physics in Germany paired x-ray and neutron imaging to test the robustness of parts.

They found that both methods yield valuable data which can be used in developing components. The sun is a shining example of fusion in action. In the extremes of pressure and temperature at the centre of the sun atoms travel fast enough to fuse together, releasing vast amounts of energy. For decades, scientists have been looking at how to harness this safe, carbon-free and virtually limitless source of energy. Various techniques have been examined for automated design however the most effective techniques have proven to be the machine learning and search algorithms themselves. Evolutionary algorithms, specifically genetic programming and variations thereof have played pivotal role in inducing new constructs such as construction heuristics, operators and software. Evolutionary algorithms have also been used for designing architectures E.g. Neural networks architecture and parameter control and tuning.

2. STUDY SHOWS CONNECTION BETWEEN BRAIN SIZE AND COGNITIVE PERFORMANCE

(*Excerpt from scitechdaily, Dec-2018*)

A new study, the largest of its kind, led by Gideon Nave of the Wharton School and Philipp Koellinger of Vrije Universiteit Amsterdam, has clarified the connection. Using MRI-derived information about brain size in connection with cognitive performance test results and educational-attainment measures obtained from more than 13,600 people, the researchers found that, as previous studies have suggested, a positive relationship does exist between brain volume and performance on cognitive tests. But that finding comes with important caveats.

"The effect is there," says Nave, an assistant professor of marketing at Wharton. "On average, a person with a larger brain will tend to perform better on tests of cognition than one with a smaller brain. But size is only a small part of the picture, explaining about 2 percent of the variability in test performance. For educational attainment the effect was even smaller: an additional 'cup' (100 square centimetres) of brain would increase an average person's years of schooling by less than five months." Koellinger says "this implies that factors other than this one single factor that has received so much attention across the years account for 98 percent of the other variation in cognitive test performance."

"Yet, the effect is strong enough that all future studies that will try to unravel the relationships between more fine-grained measures of brain anatomy and cognitive health should control for total brain volume. Thus, we see our study as a small, but important, contribution to better understanding differences in cognitive health."

3. NEW BATTERY DESIGN EXTENDS THE LIFE OF LOW-COST, LIGHTWEIGHT BATTERIES

(*Excerpt from scitechdaily, Nov-2018*)

The MIT design overcomes the problem of corrosion in aluminium-air batteries by introducing an oil barrier between the aluminium electrode and the electrolyte — the fluid between the two battery electrodes that eats away at the aluminium when the battery is on standby. The oil is rapidly pumped away and replaced with electrolyte as soon as the battery is used. As a result, the energy loss is cut to just 0.02 percent a month — more than a thousand fold improvement. The findings are

reported today in the journal Science by former MIT graduate student Brandon J. Hopkins ’18, W.M. Keck Professor of Energy Yang Shao-Horn, and professor of mechanical engineering Douglas P. Hart. A key to the new system is a thin membrane placed between the battery electrodes. When the battery is in use, both sides of the membrane are filled with a liquid electrolyte, but when the battery is put on standby, oil is pumped into the side closest to the aluminium electrode, which protects the aluminium surface from the electrolyte on the other side of the membrane.

The new battery system also takes advantage of a property of aluminium called “underwater oleophobicity” — that is, when aluminium is immersed in water, it repels oil from its surface. As a result, when the battery is reactivated and electrolyte is pumped back in, the electrolyte easily displaces the oil from the aluminium surface, which restores the power capabilities of the battery. Ironically, the MIT method of corrosion suppression exploits the same property of aluminium that promotes corrosion in conventional systems. The result is an aluminium-air prototype with a much longer shelf life than that of conventional aluminium-air batteries. The researchers showed that when the battery was repeatedly used and then put on standby for one to two days, the MIT design lasted 24 days, while the conventional design lasted for only three. Even when oil and a pumping system are included in scaled-up primary aluminium-air battery packs, they are still five times lighter than and twice as compact as rechargeable lithium-ion battery packs for electric vehicles, the researchers report.

4. EXISTING LASER TECHNOLOGY COULD BE USED TO ATTRACT ALIENS

(*Excerpt from scitechdaily, Nov-2018*)

If extraterrestrial intelligence exists somewhere in our galaxy, a new MIT study proposes that laser technology on Earth could, in principle, be fashioned into something of a planetary porch light — a beacon strong enough to attract attention from as far as 20,000 light years away. The research, which author James Clark calls a “feasibility study,” appears today in The Astrophysical Journal. The findings suggest that if a high-powered 1- to 2-megawatt laser were focused through a massive 30- to 45-meter telescope and aimed out into space, the combination would produce a beam of infrared radiation strong enough to stand out from the sun’s energy. Such a signal could be detectable by alien astronomers performing a cursory survey of our section of the Milky Way — especially if those astronomers live in nearby systems, such as around Proxima Centauri, the nearest star to Earth, or TRAPPIST-1, a star about 40 light-years away that hosts seven exo-planets, three of which are potentially habitable. If the signal is spotted from either of these nearby systems, the study finds, the same megawatt laser could be used to send a brief message in the form of pulses similar to Morse code.

Workbench:
Python based Wireless Control of PC using NodeMCU

Mr. Satyanarayana V V Vedula

Assistant Professor Department of Physics

PowerPoint presentations always need a wireless control of PC for independent operation by the presenter. Bluetooth based remote controls are available in the market. But they will provide only limited controls like slide up/down; Quit presentation etc. This is a Python project (Fig. 1) with customized control of PC/Laptop through Wireless communication, preferably Wi-Fi. The same project can be implemented using Bluetooth, Xbee or RF communication. The total budget of the project is just worth a NodeMCU board as it does not involve any other electronics.

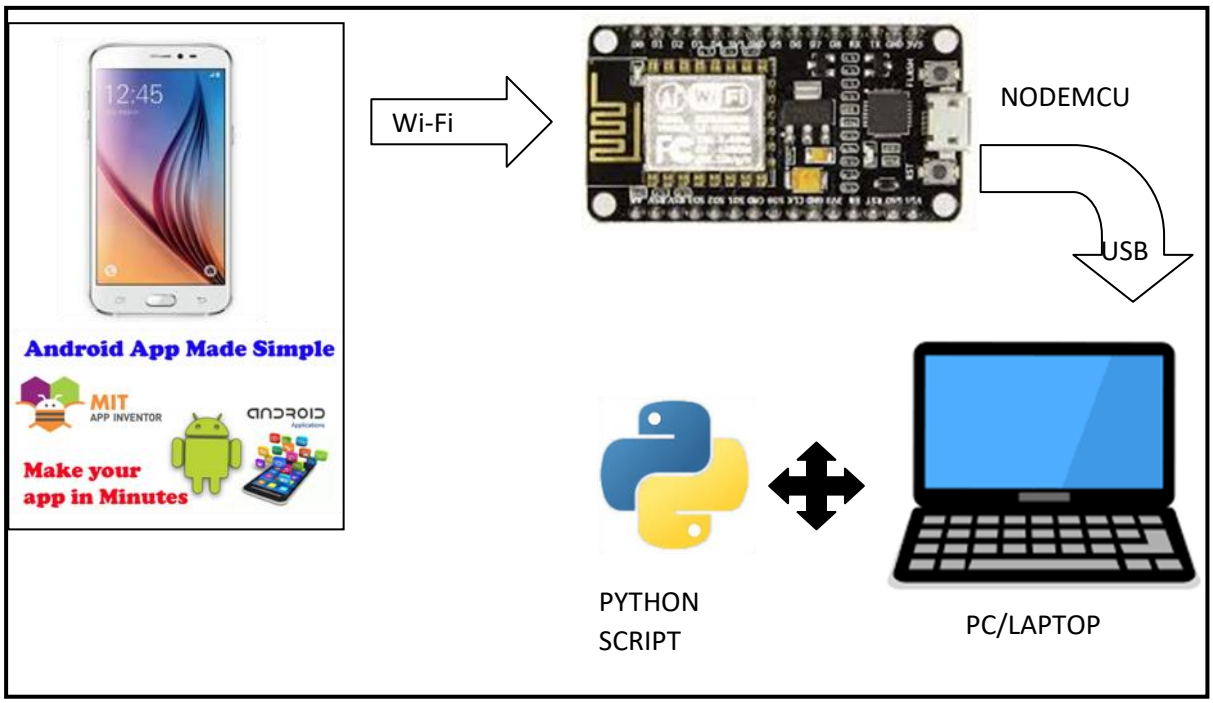
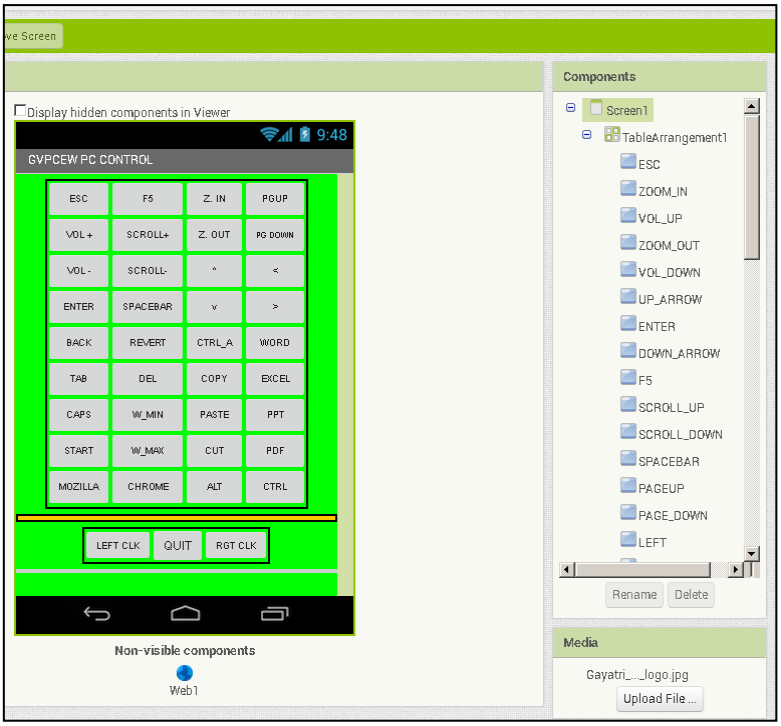


Fig. 1. Block diagram of the system.

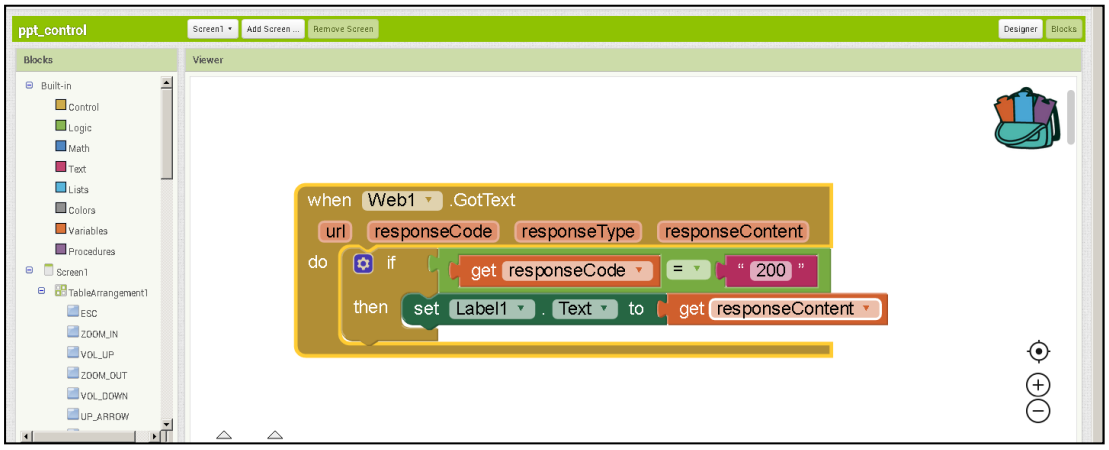
The Android mobile Wi-Fi client gets connected to NODEMCU Wi-Fi web server, working also as Wi-Fi access point. The MIT app running on the Android communicates to NODEMCU through Wi-Fi, using http protocol. The NODEMCU acting as Wi-Fi access point cum web server processes each client request made by the Android mobile. The desired operation on PC, say slide up, can be sent from the Android to the NODEMCU using http request. This request is processed by NODEMCU and the corresponding instruction, say "slide up" is transmitted to PC using its USB port. A python script always scans for the instructions received at USB serial port of the PC. Python controls the PC using Pyautogui library which can control the keyboard and mouse functions of the PC.

Steps to build the project:

1. MIT app:



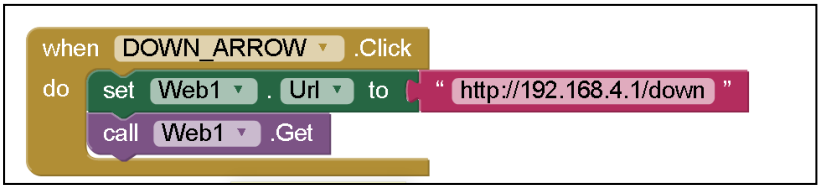
1. In the default “Designer” window, Choose and drag the “Web” from “connectivity” palette to the app screen area.
2. Choose a Table arrangement from the “Layout” palette with desired number of rows and columns. In this case the author chose 9 rows with 4 columns.
3. Choose another table arrangement with only one row and three columns.
4. In the first table add buttons from the “user interface” palette and name them accordingly as shown in the above picture.
5. Follow the next steps in the “blocks” part of the App inventor.



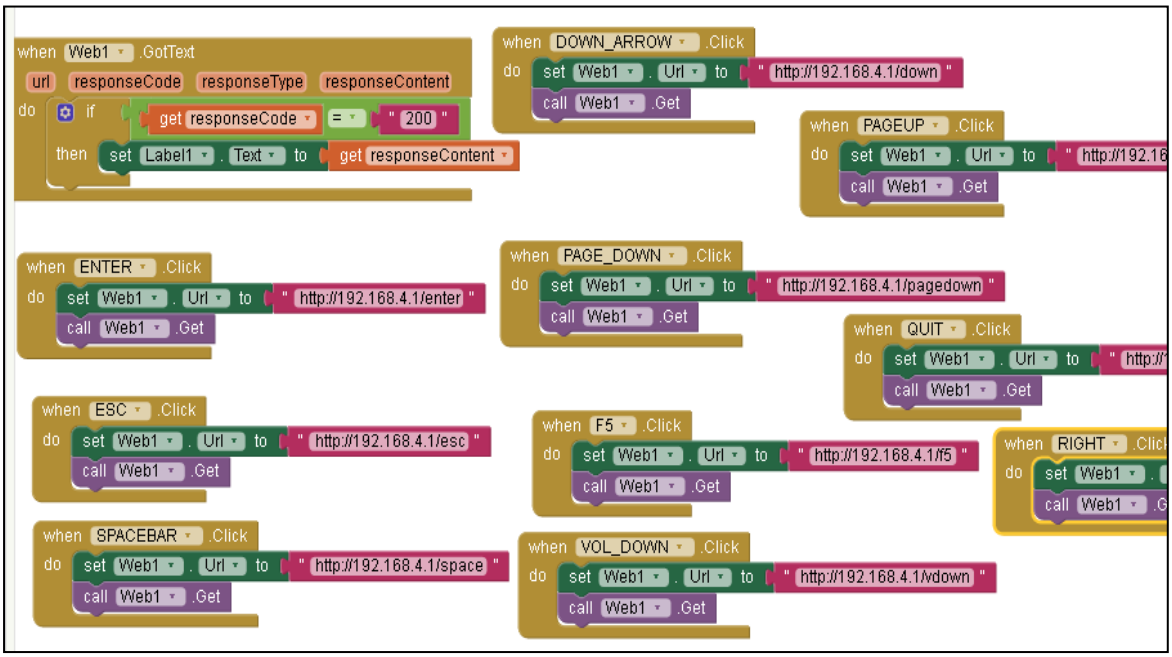
6. Choose the “web1 got text” brick and attach a “if then” conditional block.
7. Check the “get response code” with decimal value of 200. (Explanation: When an http request is made by the client device, server processes it and bounces back to the client with an http header which contains a response code, response type and response content. If the request made by client is successfully processed by the server, the “response code” will be 200. If the code is 404 then the result not found by the server. One can refer to types of response codes of http on the web. The “response type” part

describes the type of content sent by the server. It may be a plain text file, an html code, mp3 file or any other file type. The "response content" is the string which contains the actual data sent by the server to client.)

8. If satisfied, then print the response content of http request on a label on the home screen. Now add the following block to the existing block.



9. "When button got click" is chosen from the palette, one can find all the buttons added to the screen in "designer" window. The above screen shot describes a button named "DOWN_ARROW".
10. Select the "set web1 url to" brick from the "web" palette and add the text "http://192.168.4.1/down" to it. The IP address 192.168.4.1 is default for the NODEMCU Wi-Fi web server. /down is the request made by client. Hence for the action of pressing of down arrow on the PC one sends the command "/down" to the Wi-Fi web server.
11. Add the brick "call web1 get" from the "web" palette. This will send the http request to the server.
12. Add similar blocks like UP_ARROW, ZOOM_IN, ZOOM_OUT, VOLUME_UP, VOLUME_DOWN, MSWORD, MSPPT, MSEXCEL, F5 (for screen refresh, start of slide show), QUIT (ALT+F4) etc, so many desired functions to "blocks" part of App inventor.



Finally, build the apk file using "build" menu. This will save the apk file to the computer which can be transferred to an Android mobile and then installed.

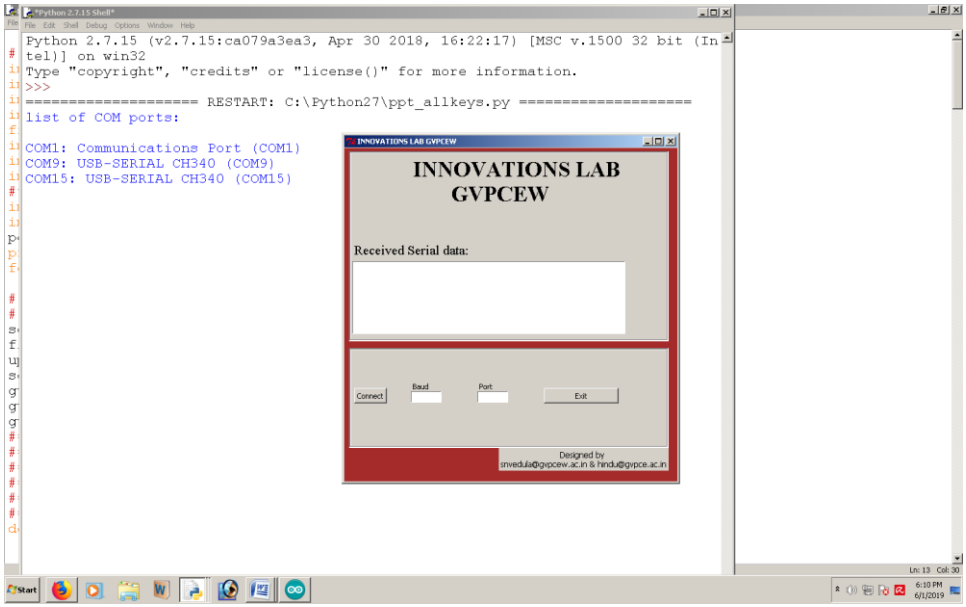
2. NODEMCU CODE:

Type the following code in the Arduino IDE.

/********* Readers are requested to contact the author for the complete source code *********/

Explanation of Python code:

The import directive import the libraries installed earlier. It is similar to #include in case of C language. The "import time" enables the usage of time delays in the code. "import Tkinter" enables the GUI functions. Buttons, window controls, scroll bars, Text boxes, Labels, radio buttons etc functions are created using this library. Import pyautogui enables the control of keyboard and mouse functions. Import os and import serial.port.tools, import pywin32 libraries enables the usage of Serial ports and listing of existing ports on the pc. Threading enables the division of works into threads. For example the user interface created using Tkinter is refreshed every 5second in the code. But one does not know when the data comes from the serial port. Polling for serial port data blocks the execution of other routines and hence threading is used. In the background this thread continuously monitors the serial port data and holds the data in some buffer. When refreshed the GUI shows the data received in the back end. Rest of the code has comments in each line which makes it easier to understand. The following is the GUI when the Python script is executed.



In this case the PC has COM9 and COM15 Ports which are the NodeMCUs connected to the PC. Enter the com port number in the Port tab and the press connect to run the python script. Simultaneously the NodeMCU provides an access point with name "innovationslab" with password 12345678. Connect the Android to this access point and then run the app. Enjoy controlling the pc using the Android.

Solutions to the Riddles given on Page. 15

1. The second one; the poor lions died of starvation.
2. Your left hand.
3. 78; all of the numbers, when read upside down, are the numbers 86 - 91.
4. Australia; it was still there, just not discovered.
5. The letter 'E'.

College Events:

1. **Engineers day celebration (September-2018)**

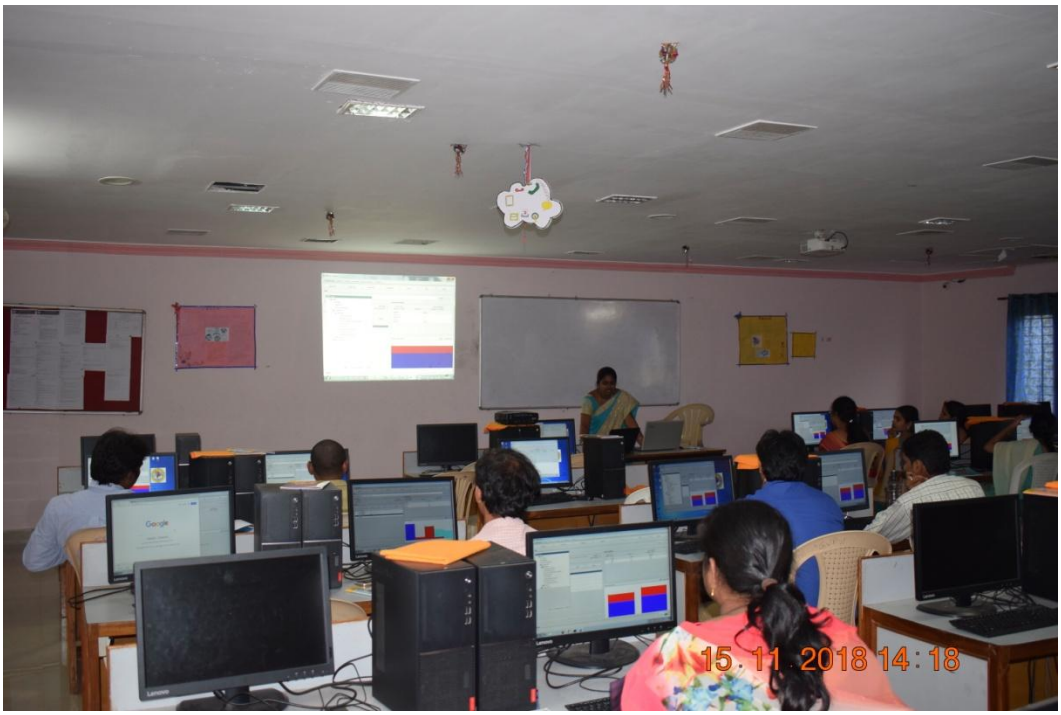
GVP COLLEGE OF ENGINEERING FOR WOMEN (GVPCEW) has taken initiative to bring out the hard efforts of engineers and to bring their ideas from mind to market. GVPCEW organized "Engineers Day on 15 -09-2018 where hard efforts of future engineers got appreciated and awarded.

Important Attendees:

1. Dr. DVLN Somayajulu, professor(HAG), Dept. CSE, NIT Warangal (chief guest),
2. Prof. (Dr.) KVS Rao, Principal,
3. Prof. (Dr.) G. Sudheer ,Vice-Principal ,
4. Prof (Dr.)PVSL Jagadamba, HoD, Dept. CSE,
5. Mr. C. Srinivas, HoD, Dept. of IT.

2. **Workshop by ML certified faculty members on Weka tools, Dept. CSE (November-2018)**

GVPCEW conducted a two day faculty development program (FDP) on WEKA Tool for the faculty belonging to Dept. of CSE & IT. The FDP is conducted on 15th and 16th, Nov-2018. Three distinguished resource persons are actively involved in conducting this event. They are M/s. B. Jaya Lakshmi, (Asst. Prof., Dept. of IT, GVPCE (A), Prof. (Dr.) N. B. Venkateswarulu, Dept. of CSE, and M/s. P. Sridevi , Asst. Prof., Dept. of IT ,GVPCEW.



3. **<u>Helping hands for Kerala flood relief (October-2018)</u>**

4. Talk on "Quest for Happiness" by Sushila ji (November-2018)

5. Workshop on Cyber Security (December-2018)

GVPCEW in collaboration with Andhra Pradesh Information Technology Academy (APITA) has conducted a three days workshop on Cyber Security and its application for the fourth year B.Tech students from the Dept. Of IT who have registered with APITA. The workshop is conducted on 6th – 8th, Dec-2018. Eminent resource person from APITA, Mr. Phani carried out the workshop that included both theoretical and practical sessions.

GAYATRI VIDYA PARISHAD

College of Engineering for Women

Approved by AICTE New Delhi, Affiliated to JNTUK Kakinada
Madhurawada, Kommadi, Visakhapatnam - 530 048.

Other Institutions

GVP College for Degree & PG Courses

GVP College of Engineering (Autonomous)

GVP Centre for Policy Studies

GVP Junior College

Indo German Institute of Advanced Technology (IGIAT)

GVP College for Degree & PG Courses School of Engineering

GVP Institute of Health Care & Medical Technology

GVP MLBT School