**UNIT-V**

1. **Data Collection, Storage and Computing using a Cloud Platform for IoT/M2M Applications/service.**
2. **Data collection, storage and computing using a cloud platform everything as a service and Cloud Service Models.**
3. **IoT Cloud based Services using the Xively (Pachube/COSM).**
4. **Nimbits and other platforms Sensors.**
5. **Participatory sensing.**
6. **Actuator**
7. **Radio frequency Identification and Wireless Sensor Network Technology.**
8. **Sensors Technology, Sensing the World.**

**INTRODUCTION**

**A few conventional methods for data collection and storage are as follows:**

**● Saving devices’ data at a local server for the device nodes**

**● Communicating and saving the devices’ data in the files**

**locally on removable media, such as micro SD cards and**

**computer hard disks**

**● Communicating and saving the data and results of computations in a dedicated data**

**store or coordinating node locally**

**● Communicating and saving data at a local node, which is a part of a distributed DBMS**

**● Communicating and saving at a remote node in the distributed DBMS**

**● Communicating on the Internet and saving at a data store in a web or enterprise server**

**● Communicating on the Internet and saving at data centre for an enterprise**

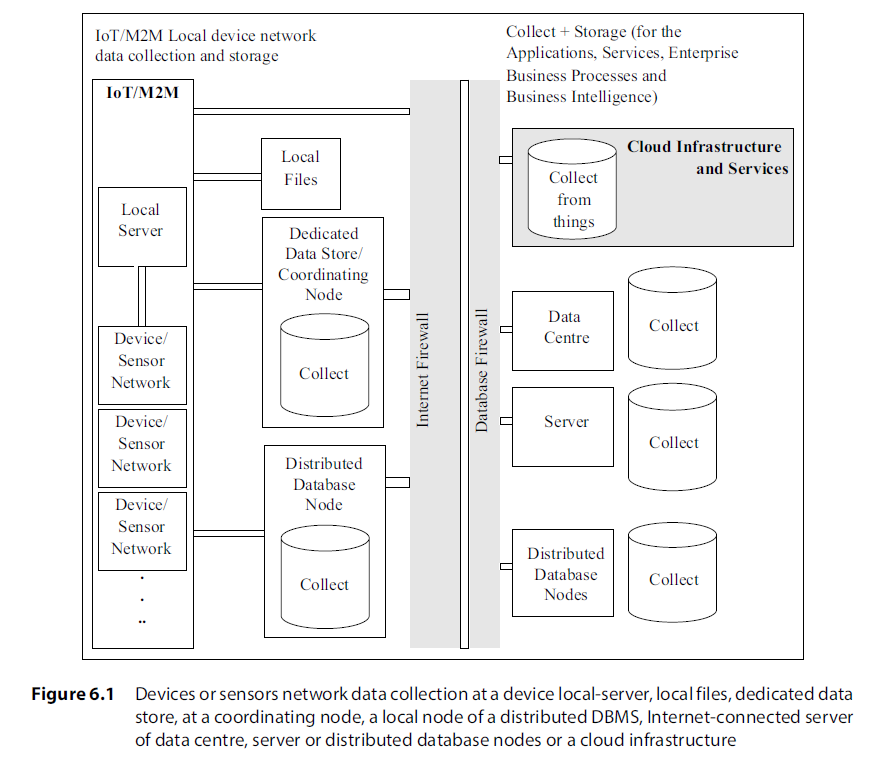
**Cloud is a new generation method for data collection, storage and computing.**

1. **Data collection, storage and computing using a cloud platform everything as a service and Cloud Service Models.**
2. **IoT Cloud based Services using the Xively (Pachube/COSM).**
3. **Nimbits and other platforms Sensors.**
4. **Participatory sensing.**
5. **Actuator**
6. **Radio frequency Identification and Wireless Sensor Network Technology.**
7. **Sensors Technology, Sensing the World.**
8. **Data collection, storage and computing using a cloud platform everything as a service and Cloud Service Models.**

Different methods of data collection, storage and computing are shown in Figure 6.1. The figure shows (i) Devices or sensor networks data collection at the device web server, (ii) Local files, (iii) Dedicated data store at coordinating node, (iii) Local node in a distributed DBMS, (iv) Internet-connected data centre, (v) Internet-connected server, (vi) Internet-connected distributed DBMS nodes,and (vii) Cloud infrastructure and services.

Cloud computing paradigm is a great evolution in Information and Communications Technology (ICT). The new paradigm uses XAAS at the Internet connected clouds for collection, storage and computing. Following are the key terms and their meanings, which need to be understood before learning about the cloud computing platform.

Resource refers to one that can be read (used), written (created of changed) or executed (processed). A path specification is also a resource. The resource is atomic (not-further divisible) information, which is usable during computations. A resource may have multiple instances or just a single instance. The data point, pointer, data, object, data store or method can also be a resource.



System resource refers to an operating system (OS), memory, network, server, software or application.

Environment refers to an environment for programming, program execution or both. For example, cloud9 online provides an open programming environment for BeagleBone board for the development of IoT devices; Windows environment for programming and execution of applications; Google App Engine environment for creation and execution of web applications in Python or Java.

Platform denotes the basic hardware, operating system and network, and is used for software applications or services over which programs can be run or developed.1

Edge computing is a type of computing that pushes the frontier of computing applications, data and services away from centralised nodes to IoT data generating nodes, that means at logical extremes of the network

Distributed computing refers to computing and usage of resources which are distributed at multiple computing environments over the Internet.

Service is a software which provides the capabilities and logically grouped and encapsulated functionalities. A service is called by an application for utilising the capabilities.

Web Service, according to the W3C definition, is an application identified by a URI,

described and discovered using the XML based Web-Service Description Language (WSDL)

Service-oriented architecture consists of components which are implemented as independent services which can be dynamically bonded and orchestrated, and which possess loosely coupled configurations, while the communication between them uses messages.

Web computing refers to computing using resources at computing environment of web

server(s) or web services over the Internet.

Grid computing refers to computing using the pooled interconnected grid of computing resources and environments in place of web servers.

Utility computing refers to computing using focus on service levels with optimum

amount of resources allotted when required and takes the help of pooled resources and

environments for hosting applications.

Cloud computing refers to computing using a collection of services available over the Internet that delivers computational functionality on the infrastructure of a service provider for connected systems and enables distributed grid and utility computing.

Key Performance Indicator (KPI) refers to a set of values, usually consisting of one or more raw monitored values including minimum, average and maximum values specifying the scale. A service is expected to be fast, reliable and secure.

Localisation means cloud computing content usage is monitored by determining localisation of the QoS level and KPIs.

Seamless cloud computing means during computing the content usages and computations continue without any break when the service usage moves to a location with similar QoS level and KPIs. For example, continue using same cloud platform when developer of software shifts.

Elasticity denotes that an application can deploy local as well as remote applications or services and release them after the application usage. The user incurs the costs as per the usages and KPIs.

Measurability (of a resource or service) is something which can be measured for controlling or monitoring and enables report of the delivery of resource or service.

Homogeneity of different computing nodes in a cluster or clusters refers to integration with the kernel providing the automatic migration of the processes from one to otherhomogeneous nodes.

Scalability in cloud services refers to the ability using which an application can deploy smaller local resources as well as remotely distributed servers and resources, and then increase or decrease the usage, while incurring the cost as per the usage on increasing scales.

Maintainability in cloud services refers to the storage, applications, computing infrastructure, services, data centres and server’s maintenances which are responsibilitiesof the remotely connected cloud services with no costs to the user.

XAAS is a software architectural concept that enables deployment and development ofapplications, and offers services using web and SOA. A computing paradigm is to integratecomplex applications and services (Section 5.3.5) and use XAAS concept for deploying a cloud platform.

**Cloud Computing Paradigm**

Cloud computing means a collection of services available over the Internet. Cloud delivers the computational functionality. Cloud computing deploys infrastructure of acloud-service provider.

The infrastructure deploys on a utility or grid computing or webservices environment that includes network, system, grid of computers or servers or data centres.

Cloud platform offers the following:

● Infrastructure for large data storage of devices, RFIDs, industrial plant machines, automobiles and device networks

● Computing capabilities, such as analytics, IDE (Integrated Development Environment)

● Collaborative computing and data store sharing

**Cloud Platform Usages**

Cloud platform usages are for connecting devices, data, APIs, applications and services, persons, enterprises, businesses and XAAS.

The following Equation 6.1 describes a simple conceptual framework of the Internet

Cloud:

**Internet Cloud + Clients = User applications and services with ‘no boundaries and no walls’ …**

An application or service executes on a platform which includes the operating system

(OS), hardware and network. Multiple applications may initially be designed to run on

diversified platforms (OSs, hardware and networks). Applications and services need to

integrate them on a common platform and running environment.

Cloud storage and computing environment offers a virtualised environment, which refers

to a running environment made to appear as one to all applications and services, but in

fact physically two or more running environments and platforms may be present.

**EVERYTHING AS A SERVICE AND CLOUD SERVICE MODELS (XAAS)**

Cloud connects the devices, data, applications, services, persons and business. Cloud services can be considered as distribution service—a service for linking the resources (computing functions, data store, processing functions, networks, servers and applications) and for provision of coordinating between the resources. Figure 6.2 shows four cloud service models and examples.

Cloud computing can be considered by a simple equation:

**Cloud Computing = SaaS + Paas + IaaS + DaaS … 6.2**

**SaaS means** Software as a Service. The software is made available to an application or service on demand. SaaS is a service model where the applications or services deploy and host at the cloud, and are made available through the Internet on demand by the service user.

The software control, maintenance, updation to new version and infrastructure, platform and resource requirements are the responsibilities of the cloud service provider.

**PaaS means** Platform as a Service. The platform is made available to a developer of an application on demand. PaaS is a service model where the applications and services develop and execute using the platform (for computing, data store and distribution services) which is made available through the Internet on demand for the developer of the applications. The platform, network, resources, maintenance, updation and security as per the developers’ requirements are the responsibilities of the cloud service provider.

**IaaS means** Infrastructure as a Service. The infrastructure (data stores, servers, data centres and network) is made available to a user or developer of application on demand. Developer installs the OS image, data store and application and controls them at the infrastructure.

**DaaS means** Data as a Service. Data at a data centre is made available to a user or developer of application on demand. DaaS is a service model where the data store or data warehouse is made available through the Internet on demand on rent (pay as per use in multi tenancy model) to an enterprise.

**Cloud Computing Features and Advantages**

Essential features of cloud storage and computing are:

● On demand self-service to users for the provision of storage, computing servers,

software delivery and server time

● Resource pooling in multi-tenant model

● Broad network accessibility in virtualised environment to heterogeneous users, clients,

systems and devices

● Elasticity

● Massive scale availability

● Scalability

● Maintainability

● Homogeneity

● Virtualisation

● Interconnectivity platform with virtualised environment for enterprises and

provisioning of in-between Service Level Agreements (SLAs)

● Resilient computing

● Advanced security

● Low cost

1. **IoT Cloud-based Data Collection, Storage, Computing using Xively (Pachube/COSM).**

Pachube is a platform for data capture in real-time over the Internet. Cosm is a changed

domain name, where using a concept of console, one can monitor the feeds. Xively is the

latest domain name. Xively is an open source platform for Arduino which is an opensource

prototyping platform that provides connectivity with web deploying Internet.

Xively is a commercial PaaS for the IoT/M2M.6 It is used as a data aggregator and data

mining website often integrated into the Web of Things. Xively is an IoT PaaS for services

and business services. The platform supports the REST, WebSockets and MQTT protocols

and connects the devices to Xively Cloud Services. There are native SDKs for Android,

Arduino, ARM mbed, Java, PHP, Ruby and Python languages. Developers can use the

workflow of prototyping, deployment and management through the tools provided by

Xively.3

* **Xively PaaS services offers the following features**:

It enables services, business services platform which connects the products, including

collaboration products, Rescue, Boldchat, join.me, and operations to the Internet.

● Data collection in real-time over the Internet.

● Data visualisation for data of connected sensors to IoT devices.

● Graphical plots of collected data.

● It generates alerts.

● Access to historical data.

● It supports Java, Python and Ruby, and Android platform.

● It generates feeds which can be real-world objects of own or others.

● It supports the ARM mBedTM-based, Arduino-based and other hardware-platformbased

IoT devices, and HTTP-based APIs which are easy to implement on device

hardware acting as clients to Xively web services, and connect to the web service and

send data.

● It supports REST.

A user creates an account with Xively when deploying Xively APIs for the data collectionand other functions. An API key from my settings is necessarily copied. Xively APIs enable interface with Python, HTML5, HTML5 server, tornado, webSocket,

webSocket Server and WebSockets and interface with an RPC (Remote Procedure Call).Devices get an online presence. For example, an Arduino climate logging client that can be accessed via a browser or mobile using Xively. Arduino is an open-source prototyping platform which is based on ATmega microcontroller for embedded applications and IoT/M2M. Another platform is mBedTM. It is based on ARM Cortex-microprocessors. ThemBedTM is also for embedded applications and IoT/M2M (Chapter 8).

Xively is an opensource platform that enables IoT devices or sensors network to connect the sensor data to the web.7,8 Xively provides services for logging, sharing and displaying sensor data of all kinds using an HTTP based API.

Xively is based on the concept of users, feeds, data streams, data points and triggers. A feed is typically a single location (e.g. a house), and data streams are of individual sensors associated with that location (for example, ambient lights, temperatures, power consumption).

**Pull or Push (Automatic or Manual Feed) Methods for IoT Devices Data**

Xively provides two modes for data capture, viz. a pull method (automatic feed type) where data is collected from an http server, and a push method (manual feed type) where data is written to Xively using an http client.

**Data Formats and Structures**

Number of data formats and structures enable interaction, data collection and services with Xively. The support exists for JSON (Section 3.2.3), XML (Section 3.3.3) and CSV (a tabular, spreadsheet, excel, data numbers and text with a comma-separated values in file).

**Private and Public Data Access**

A free account supports up to 10 sensor feeds updated in near real time and the data is stored for up to 3 months. Interactive graphs provided by the service can be embedded onto the mobile giving the application access to data anywhere. The application can even use other user’s data feeds as inputs.

**Data streams, Data points and Triggers**

Xively enables data streams, data points and triggers. Data stream means continuous sensed data flow over the Internet. Data points mean data values, whereas trigger means action on a state change. For example, ambient light sensing below a threshold value near a group of streetlights.

1. **IoT Cloud-based Data Collection, Storage and Computing Services**

**Using Nimbits**

Nimbits enables IoT on an open-source distributed cloud. Nimbits cloud PaaS deploys aninstance of Nimbits Server at the device nodes. Nimbits functions as an M2M system data store, data collector and logger with access to historical data.

Nimbits architecture is a cloud-based Google App Engine. Nimbits server is a class hierarchy com.nimbits.server. system.ServerInfo of java.lang.Object.

**Nimbits PaaS services offer the following features:**

● Edge computing locally on embedded systems, built up of local applications. It runs

the rules and pushes important data up to the cloud running when connected over

Internet and an instance of Nimbits Server hosts at the device nodes which is then

enabled.

● It supports multiple programming languages, including Arduino, new Arduino

library, push functions from Arduino cloud, JavaScript, HTML or the Nimbits.io Java

library.

● Nimbits server functions as a backend platform. Nimbits data point can relay data

between the software systems, or hardware devices such as Arduino, using the cloud

as a backend.

● An open-source Java library called nimbits.io enables easy development of JAVA, web

and Android solutions (Nimbits data, alerts, messages on mobile).

● It provides a rule engine for connecting sensors, persons and software to the cloud

and one another. Rules can be for calculations, statistics, email alerts, xmpp messages

push notifications and more.

It provides a data logging service and access, and stores the historical data points and

data objects.

● Storage in any format that can be serialised into a string, such as JSON or XML.

● It filters the noise and important changes sent to another larger central instance.

● It processes a specific type of data and can store it.

● Time- or geo-stamping of the data.

● Nimbits clients provide over Internet, data collection in real time, charts, chart and

graphical plots of collected data and data entry.

● Data visualisation for data of connected sensors to IoT devices.

● Supports the alerts subscription, generation and sending in real time over the Internet.

● It creates streams of data objects and stores them in a data point series.

● Data accessibility and monitoring from anywhere, and is used to shape the behaviour

of connected devices and software.

● It supports the mBedTM, Arduino, Raspberry Pi based and other hardware platform

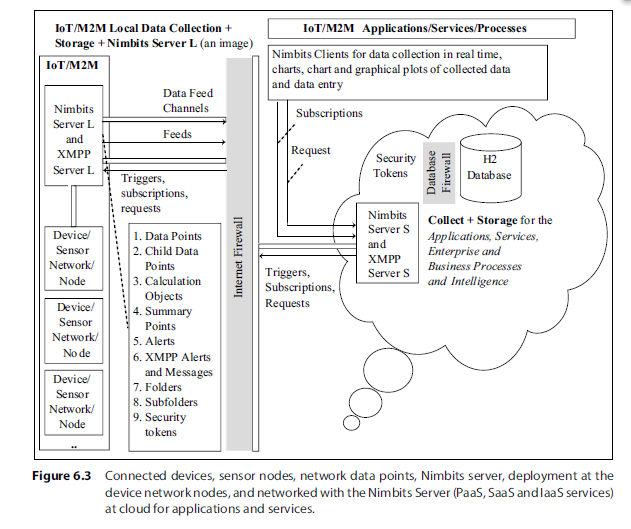
based IoT devices.

● Web service APIs are easy to implement on device hardware acting as clients to

Nimbits web services, and connect to the web service and send data.

● It deploys software on Google App Engine, any J2EE server on Amazon EC2 or on a

Raspberry Pi.



**Using Advanced Features**

An application can create a connection to another Nimbits application or service.10 The

application sends an invitation and if the invitee approves, then the application can see

10 http://bsautner.github.io/com.nimbits/

**Data Collection, Storage and Computing Using a Cloud Platform 221**

invitee’s points and data in their tree (if they set the objects permission level to public or

connection viewable).

Nimbits 3.6.6 introduced H2 database engine. Nimbits 3.8.10 includes H2 database

engine. H2 is Java SQL database. APIs are in pure Java, The main features of H2 are:

● Very fast, open source, JDBC API

● Embedded and server modes; in-memory databases

● Encrypted database

● ODBC driver

● Full text search

● Multi-version concurrency

● Browser-based console application

● Small footprint (around 1.5 MB jar file size)

MySQL is not in pure Java and have no provisions for in-memory or encrypted

databases. Footprint (DLL) is nearly 4 MB. (JAR means Java archive while DLL stands for

dynamically linked library.)

**Security Tokens**

Nimbits 3.9.6 provides security tokens in a new way.

Breakthrough Performance and Data Integrity

Nimbits server 3.9.10 version launched in June 2015 provisions for the breakthrough

performance and data integrity.

**Alerts**

A filter means applying some rule to get new data for a data point. The filter item in the

tree called “ah” is for XMPP alerts (Section 3.3.3). A user application can now have many

JIDs (Jabber IDs) for a single point—alerts and messages can be sent over XMPP using the

custom JID of points.

**Jabbing**

Jabbing means pushing the alerts or messages down quickly or pushing repeatedly. Each type of alert or message is assigned a Jabber ID, called JID. Each JID consists of three main parts, viz. the node identifier (optional), domain identifier (required) and resource identifier (optional). A JID is written in notations as <JID>:= [<node>”@”]<domain>[“/”<resource>.

**Subscriptions**

A user can create many subscriptions for a single point. It may subscribe to one of the points, other user, or anyone’s public point to get the alerts. The user gets alerts when the point goes into an alarm state, or receives new data. Alarm state means reaching preset value. For example, a pressure boiler reaching critical pressure necessitating action or water reaching boiling point necessitating action.

1. **Participatory sensing.**

Information collected from sensors of multiple heterogeneous sources can lead to knowledge discovery after analytics and data visualisation. A web source defines Participatory Sensing (PS) as “sensing by the individuals and groups of people contributing sensory information to form a body of knowledge”.

(OR)

Deborah Estrin, University of California, Los Angeles now at Cornell University, defines participatory sensing as, “Participatory sensing is the process whereby individuals and communities use evermore-capable mobile phones and cloud services to collect and analyse systematic data for use in discovery.”

A participant of a PS process can be sensors used in mobile phones. Mobile phones have camera, temperature and humidity sensors, an accelerometer, a gyroscope, a compass, infrared sensors, NFC sensors, bar or QR code readers, microphone and GPS. Mobiles communicate on the Internet the sensed information with time, date and location stamps.

Applications of PS include retrieving information about weather, environment information, pollution, waste management, road faults, health of individuals and group of people, traffic congestion, urban mobility, or disaster management, such as flood, fire, etc. Participatory sensing has many challenges such as—security, privacy, reputation and ineffective incentives to participating entities.

Figure 7.9 (a) shows the sources of data in the PS process for IoT applications. Figure 7.9 (b) shows the phases of a PS process. Phase 1 is coordination, in which the participants of a PS process organise after identifying the sources. Next two phases, i.e. phases 2 and 3 involve data capture, communication and storage on servers or cloud. Next two phases, i.e. phases 4 and 5 involve PS data processing and analytics, visualisation and knowledge discovery. Last phase, i.e. phase 6 is for initiating appropriate actions.

