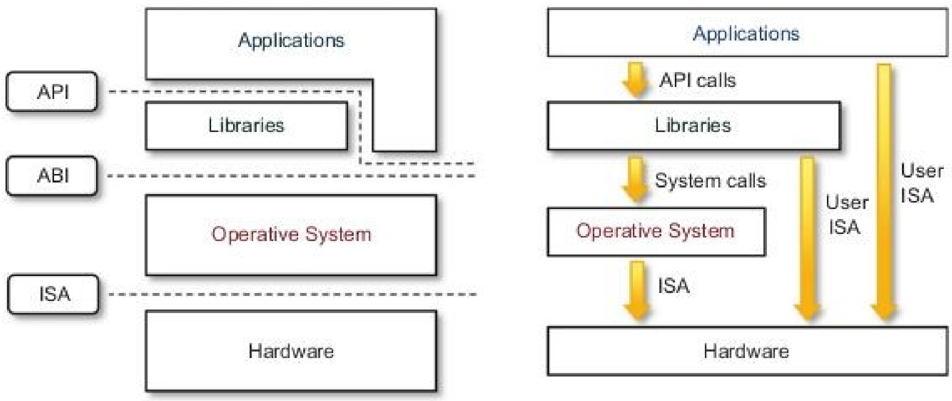
Seventh Semester BE Degree Examination DEC 2019/JAN 2020
15CS742 Cloud Computing and its Applications

VTU Exam Solutions

1(a) Describe the main characteristics of cloud computing [4 marks]

1. Standard instances. This class offers a set of configurations that are suitable for most applications. EC2 provides three different categories of increasing computing power, storage, and memory.
2. Micro instances. This class is suitable for those applications that consume a limited amount of computing power and memory and occasionally need bursts in CPU cycles to process surges in the workload. Micro instances can be used for small Web applications with limited traffic.
3. High-memory instances. This class targets applications that need to process huge workloads and require large amounts of memory. Three-tier Web applications characterized by high traffic are the target profile. Three categories of increasing memory and CPU are available, with memory proportionally larger than computing power.
4. High-CPU instances. This class targets compute-intensive applications. Two configurations are available where computing power proportionally increases more than memory.

1(b) With neat diagram explain cloud computing reference model []



Each of these layers specifies various types of entities that may exist in a cloud computing environment, such as compute systems, network devices, storage devices, virtualization software, security mechanisms, control software, orchestration software, management software, and so on. It also describes the relationships among these entities.The three cross-layer functions are business continuity, security, and service management. Business continuity and security functions specify various activities, tasks, and processes that are required to offer reliable and secure cloud services to the consumers. Service management function specifies various activities, tasks, and processes that enable the administrations of the cloud infrastructure and services to meet the provider's business requirements and consumer's expectations.

Physical Layer

•Foundation layer of the cloud infrastructure.

•Specifies entities that operate at this layer : Compute systems, network devices and storage devices. Operating environment, protocol, tools and processes.

•Functions of physical layer : Executes requests generated by the virtualization and control layer.

Virtual Layer

•Deployed on the physical layer.

•Specifies entities that operate at this layer : Virtualization software, resource pools, virtual resources.

•Functions of virtual layer : Abstracts physical resources and makes them appear as virtual resources (enables multitenant environment). Executes the requests generated by control layer.

Control Layer

•Deployed either on virtual layer or on physical layer

•Specifies entities that operate at this layer : control software

•Functions of control layer : Enables resource configuration, resource pool configuration and resource provisioning. Executes requests generated by service layer. Exposes resources to and supports the service layer. Collaborates with the virtualization software and enables resource pooling and creating virtual resources, dynamic allocation and optimizing utilization of resources.

Service Orchestration Layer

•Specifies the entites that operate at this layer : Orchestration software.

•Functions of orchestration layer : Provides workflows for executing automated tasks. Interacts with various entities to invoke provisionning tasks.

Service Layer

•Consumers interact and consume cloud resources via thos layer.

•Specifies the entities that operate at this layer : Service catalog and self-service portal.

•Functions of service layer : Store information about cloud services in service catalog and presents them to the consumers. Enables consumers to access and manage cloud services via a self-service portal.

Service operation management :

•Enables infrastructure configuration and resource provisioning

•Enable problem resolution

•Enables capacity and availability management

•Enables compliance conformance

•Enables monitoring cloud services and their constituent elements

1.(c) Explain the technologies on which cloud computing relies.

The main enabling technology for cloud computing is virtualization. Virtualization software separates a physical computing device into one or more "virtual" devices, each of which can be easily used and managed to perform computing tasks. With essentially creating a scalable system of multiple independent computing devices, idle computing resources can be allocated and used more efficiently. Virtualization provides the agility required to speed up IT operations and reduces cost by increasing infrastructure utilization. Autonomic computing automates the process through which the user can provision resources on-demand. By minimizing user involvement, automation speeds up the process, reduces labor costs and reduces the possibility of human errors.Cloud computing uses concepts from utility computing to provide metrics for the services used. Cloud computing attempts to address QoS (quality of service) and reliability problems of other grid models.

2.(a) Discuss classification or taxonomy of virtualization at different levels

There are five main domains in which virtualization technologies can be categorized. These are server, application, desktop, storage and network. In UML, the five domains are defined as objects that are called classes. In the second part of the analysis, two new types of virtualization technologies were introduced: management and security tools. Management and security tools are also added as classes in UML. They are comprised of a set of virtualization technologies that provide some form of management or security measures and must not be confused with other aspects of management and security, such as the making of policies and their execution. When we look at the relations of the management class, the security class and the five domain classes, both management and security classes are associated with the five domain classes. For example, management technologies can have a relation with all domains as each domain can be managed. The same goes for security, where virtualization security technologies can be used to provide security for each domain.

2(b) Explain the architecture of Hyper V and discuss its use in cloud computing.

Hyper-V supports isolation in terms of a partition. A partition is a logical unit of isolation, supported by the hypervisor, in which operating systems execute. The Microsoft hypervisor must have at least one parent, or root, partition, running Windows. The virtualization management stack runs in the parent partition and has direct access to hardware devices. The root partition then creates the child partitions which host the guest operating systems. A root partition creates child partitions using the hypercall application programming interface (API).Partitions do not have access to the physical processor, nor do they handle the processor interrupts. Instead, they have a virtual view of the processor and run in a virtual memory address region that is private to each guest partition. The hypervisor handles the interrupts to the processor, and redirects them to the respective partition. Hyper-V can also hardware accelerate the address translation between various guest virtual address spaces by using an Input Output Memory Management Unit (IOMMU) which operates independent of the memory management hardware used by the CPU. An IOMMU is used to remap physical memory addresses to the addresses that are used by the child partitions.

Child partitions also do not have direct access to other hardware resources and are presented a virtual view of the resources, as virtual devices (VDevs). Requests to the virtual devices are redirected either via the VMBus or the hypervisor to the devices in the parent partition, which handles the requests. The VMBus is a logical inter-partition communication channel. The parent partition hosts Virtualization Service Providers (VSPs) which communicate over the VMBus to handle device access requests from child partitions. Child partitions host Virtualization Service Consumers (VSCs) which redirect device requests to VSPs in the parent partition via the VMBus. This entire process is transparent to the guest operating system.Virtual Devices can also take advantage of a Windows Server Virtualization feature, named Enlightened I/O, for storage, networking, graphics, and input subsystems. Enlightened I/O is a specialized virtualization-aware implementation of high level communication protocols (such as SCSI) that utilize the VMBus directly, bypassing any device emulation layer. This makes the communication more efficient but requires an enlightened guest that is hypervisor and VMBus aware. Hyper-V enlightened I/O and a hypervisor aware kernel is provided via installation of Hyper-V integration services.

3(a).Explain various types of clouds

Cloud computing is the delivery of computing services like servers, storages and more over the Internet. The companies that offer these computing services are called cloud providers. They charge for cloud computing services based on usage.

Cloud computing is usually classified on the basis of location, or on the service that the cloud is offering. Based on a cloud location, we can classify cloud as:

- Public,

•Private,
•Hybrid
•Community Cloud

Based on a service that the cloud is offering, we classify as:

•IaaS(Infrastructure-as-a-Service)
•PaaS(Platform-as-a-Service)
•SaaS(Software-as-a-Service)
•or, Storage, Database, Information, Process, Application, Integration, Security, Management, Testing-as-a-service

3(b).Describe the fundamental features of economic and business model behind cloud computing.

Cloud computing is an appealing computing application providing affordable access to advanced technology and allowing end users to use and process their IT infrastructure, platforms and software on a host system over a communication network.

Business and Economic Service Models

Software as a Service, SaaS
In SaaS, multiple users are provided access to the application software hosted on the server by the service provider. Users can access and interact with the cloud applications via the Internet, using interfaces such as web browsers, without the need to install any applications on their own systems. In Saas, software is provided as a service via the Internet and the service is priced on a pay-per-use basis. In Saas model, users do not manage or monitor the infrastructure components such as network, platform, operating system and storage devices.

Platform as a Service, PaaS
The service provider delivers users a computing platform where they can develop and run their own applications using programming languages, software databases, services and tools provided by the service provider and also provides supplementary services. In PaaS model, users are not authorized to control or manage the servers, operating systems, storage spaces and other components that make up the platform infrastructure. Users' authority is limited to adjustments related to the software transferred to the cloud and configuration settings of the platform the software runs on.

Infrastructure as a Service, IaaS
In IaaS model, users can configure processing, storage, networks and other fundamental computing resources required for running applications and install the operating system and applications required. Users are not fully authorized to manage and control the physical infrastructure. However, users can control the system at the level of storage and operating system and manage specific network components. IaaS model is referred to as "Hardware as a service, HaaS" in some sources.

Analyze three inter-organizational economic models relevant to cloud networks. Main problems as non co-operative price and QoS games between multiple cloud providers existing in a cloud market. We prove that a unique pure strategy Nash equilibrium (NE) exists in two of the three models. Our analysis paves the path for each cloud provider to know what prices and QoS level to set for end-users of a given service type, such that the provider could exist in the cloud market.

4.(a) Explain the three types of services that are hosted inside the Aneka container.

Aneka is Manjrasoft Pty. Ltd.'s solution for developing, deploying, and managing cloud applications. Aneka consists of a scalable cloud middleware that can be deployed on top of heterogeneous computing resources. It offers an extensible collection of services coordinating the execution of applications, helping administrators monitor the status of the cloud, and providing integration with existing cloud technologies. One of Aneka's key advantages is its extensible set of application programming interfaces (APIs) associated with different types of programming models —such as Task, Thread, and MapReduce—used for developing distributed applications, integrating new capabilities into the cloud, and supporting different types of cloud deployment models: public, private, and hybrid (see Figure 5.1). These features differentiate Aneka from infrastructure management software and characterize it as a platform for developing, deploying, and managing execution of applications on various types of clouds.

4(b) Describe the features of Aneka management tools in terms of infrastructure, platform and applications.

Aneka is a pure PaaS implementation and requires virtual or physical hardware to be deployed. Hence, infrastructure management, together with facilities for installing logical clouds on such infrastructure, is a fundamental feature of Aneka's management layer. This layer also includes capabilities for managing services and applications running in the Aneka Cloud. Aneka leverages virtual and physical hardware in order to deploy Aneka Clouds. Virtual hardware is generally managed by means of the Resource Provisioning Service, which acquires resources on demand according to the need of applications, while physical hardware is directly managed by the Administrative Console by leveraging the Aneka management API of the PAL. The management features are mostly concerned with the provisioning of physical hardware and the remote installation of Aneka on the hardware. Platform management Infrastructure management provides the basic layer on top of which Aneka Clouds are deployed. The creation of Clouds is orchestrated by deploying a collection of services on the physical infrastructure that allows the installation and the management of containers. A collection of connected containers defines the platform on top of which applications are executed. The features available for platform management are mostly concerned with the logical organization and structure of Aneka Clouds. It is possible to partition the available hardware into several Clouds variably configured for different purposes. Services implement the core features of Aneka Clouds and the management layer exposes operations for some of them, such as Cloud monitoring, resource provisioning and reservation, user management, and application profiling. Application management Applications identify the user contribution to the Cloud.

5(a) Describe the two major techniques used for parallel computing with threads
parallel programming techniques

Simple Parallel Computations

Dynamic process creation in the master-slave approach:

This very likely represents your game project or server side threading example.

- the master thread creates the starts the slave threads (spawn)
- the master needs to communicate data to the slaves
- the master needs to collect back data
- the master proceeds after assured that data from slaves have been received.

Message passing is an alternative to solve the sync and communication issue.

Send (destination, message) method

- send data (message) to another thread
- can be synchronous-- the sending thread waits until the message is recieved and accepted by the other thread(s)
- asynchronous-- the sending thread sends the message which may go into a queue at the receiving end, but the sending thread can proceed.

Receive (threadId, message) method

- when reaching a recv() method, the thread usually waits (e.g. server connection to a client)
- but if there's a message in the queue, no wait
- responds with acceptance
- parameters are out type (call by ref): sending threadId and message
- can set up to timeout if no message, then thread proceeds anyway.

5(b) Explain the major difference between Aneka threads and local threads

The modern operating systems provide the abstractions of Process and Thread for defining the runtime profile of a software application. A Process is a software infrastructure that is used by the operating system to control the execution of an application. A Process generally contains one or more threads. A Thread is a sequence of instructions that can be executed in parallel with other instructions. When an application is running the operating system takes care of alternating their execution on the local machine. It is responsibility of the developer to create a consistent computation as a result of thread execution. The Aneka Thread Model uses the same abstraction of for defining a sequence of instructions that can be remotely executed in parallel with other instructions. Hence, within the Thread Model an application is a collection of remotely executable threads. The Thread Model allows developers to virtualize the execution of a local multi-threaded application (developed with the .NET threading APIs) in an almost complete transparent manner. This model represents the right solution when developers want to port the execution of a .NET multi-threaded application on Aneka and still use the same way of controlling the execution of application flow, which is based on synchronization between threads

6(a) List and explain various frameworks for task computing [8 marks].

Message Passing Interface (MPI) is a specification for developing parallel programs that communicate by exchanging messages. Compared to other models of task computing, MPI introduces the constraint of communication that involves MPI tasks that need to run at the same time. MPI has originated as an attempt to create common ground from the several distributed shared memory and message-passing infrastructures available for distributed computing. Nowadays, MPI has become a de facto standard for developing portable and efficient message passing HPC applications. Interface specifications have been defined and implemented for C/C11 and Fortran.

To create an MPI application it is necessary to define the code for the MPI process that will be executed in parallel. This program has, in general, the structure described in Figure below. The section of code that is executed in parallel is clearly identified by two operations that set up the MPI environment and shut it down, respectively. In the code section defined within these two operations, it is possible to use all the MPI functions to send or receive messages in either asynchronous or synchronous mode.

In SaaS, multiple users are provided access to the application software hosted on the server by the service provider. Users can access and interact with the cloud applications via the Internet, using interfaces such as web browsers, without the need to install any applications on their own systems. In Saas, software is provided as a service via the Internet and the service is priced on a pay-per-use basis. In Saas model, users do not manage or monitor the infrastructure components such as network, platform, operating system and storage devices.

Platform as a Service, PaaS

The service provider delivers users a computing platform where they can develop and run their own applications using programming languages, software databases, services and tools provided by the service provider and also provides supplementary services. In PaaS model, users are not authorized to control or manage the servers, operating systems, storage spaces and other components that make up the platform infrastructure. Users' authority is limited to adjustments related to the software transferred to the cloud and configuration settings of the platform the software runs on.

Infrastructure as a Service, IaaS

In IaaS model, users can configure processing, storage, networks and other fundamental computing resources required for running applications and install the operating system and applications required. Users are not fully authorized to manage and control the physical infrastructure. However, users can control the system at the level of storage and operating system and manage specific network components. IaaS model is referred to as "Hardware as a service, HaaS" in some sources.

6(b) Explain the feature provided by Aneka for parameter sweep applications [8 Marks]

Aneka integrates support for parameter-sweeping applications on top of the task model by means of a collection of client components that allow developers to quickly prototype applications through either programming APIs or graphical user interfaces (GUIs). The set of abstractions and tools supporting the development of parameter sweep applications constitutes the Parameter Sweep Model (PSM).

The PSM is organized into several namespaces under the common root Aneka.PSM. More precisely:

- Aneka.PSM.Core (Aneka.PSM.Core.dll) contains the base classes for defining a template task and the client components managing the generation of tasks, given the set of parameters.
- Aneka.PSM.Workbench (Aneka.PSM.Workbench.exe) and Aneka.PSM.Wizard (Aneka.PSM. Wizard.dll) contain the user interface support for designing and monitoring parameter sweep applications. Mostly they contain the classes and components required by the Design Explorer, which is the main GUI for developing parameter sweep applications.
- Aneka.PSM.Console (Aneka.PSM.Console.exe) contains the components and classes supporting the execution of parameter sweep applications in console mode. These namespaces define the support for developing and controlling parameter sweep applications on top of Aneka.

7(a) What is data intensive computing. Explain the open challenges in data intensive computing [8 marks]

Data-intensive computing is concerned with production, manipulation, and analysis of large-scale data in the range of hundreds of megabytes (MB) to petabytes (PB) and beyond. The term dataset is commonly used to identify a collection of information elements that is relevant to one or more applications. Datasets are often maintained in repositories, which are infrastructures supporting the storage, retrieval, and indexing of large amounts of information. To facilitate the classification and search, relevant bits of information, called metadata, are attached to datasets.

Data-intensive computations occur in many application domains. Computational science is one of the most popular ones. People conducting scientific simulations and experiments are often keen to produce, analyze, and process huge volumes of data. Hundreds of gigabytes of data are produced every second by telescopes mapping the sky; the collection of images of the sky easily reaches the scale of petabytes over a year. Bioinformatics applications mine databases that may end up containing terabytes of data. Earthquake simulators process a massive amount of data, which is produced as a result of recording the vibrations of the Earth across the entire globe.

Data-intensive applications not only deal with huge volumes of data but, very often, also exhibit compute-intensive properties. Data-intensive applications handle datasets on the scale of multiple terabytes and petabytes. Datasets are commonly persisted in several formats and distributed across different locations. Such applications process data in multistep analytical pipelines, including transformation and fusion stages. The processing requirements scale almost linearly with the data size, and they can be easily processed in parallel. They also need efficient mechanisms for data management, filtering and fusion, and efficient querying and distribution.

Challenges:

1. Scalable algorithms that can search and process massive datasets
2. New metadata management technologies that can scale to handle complex, heterogeneous, and distributed data sources
3. Advances in high-performance computing platforms aimed at providing a better support for accessing in-memory multiterabyte data structures
4. High-performance, highly reliable, petascale distributed file systems
5. Data signature-generation techniques for data reduction and rapid processing
6. New approaches to software mobility for delivering algorithms that are able to move the computation to where the data are located
7. Specialized hybrid interconnection architectures that provide better support for filtering multigigabyte datastreams coming from high-speed networks and scientific instruments
8. Flexible and high-performance software integration techniques that facilitate the combination of software modules running on different platforms to quickly form analytical pipelines

10(a) Describe how computing technology can be used for remote ECG monitoring [8 marks]

Healthcare is a domain in which computer technology has found several and diverse applications: from supporting the business functions to assisting scientists in developing solutions to cure diseases.

An important application is the use of cloud technologies to support doctors in providing more effective diagnostic processes. In particular, here we discuss electrocardiogram (ECG) data analysis on the cloud.

ECG is the electrical manifestation of the contractile activity of the heart's myocardium. This activity produces a specific waveform that is repeated over time and that represents the heartbeat. The analysis of the shape of the ECG waveform is used to identify arrhythmias and is the most common way to detect heart disease. Cloud computing technologies allow the remote monitoring of a patient's heartbeat data, data analysis in minimal time, and the notification of first-aid personnel and doctors should these data reveal potentially dangerous conditions. This way a patient at risk can be constantly monitored without going to a hospital for ECG analysis. At the same time, doctors and first-aid personnel can instantly be notified of cases that require their attention.

Wearable computing devices equipped with ECG sensors constantly monitor the patient's heartbeat. Such information is transmitted to the patient's mobile device, which will eventually forward it to the cloud-hosted Web service for analysis. The Web service forms the front-end of a platform that is entirely hosted in the cloud and that leverages the three layers of the cloud computing stack: SaaS, PaaS, and IaaS. The Web service constitute the SaaS application that will store ECG data in the Amazon S3 service and issue a processing request to the scalable cloud platform. The runtime platform is composed of a dynamically sizable number of instances running the workflow engine and Aneka. The number of workflow engine instances is controlled according to the number of requests in the queue of each instance, while Aneka controls the number of EC2 instances used to execute the single tasks defined by the workflow engine for a single ECG processing job. Each of these jobs consists of a set of operations involving the extraction of the waveform from the heartbeat data and the comparison of the waveform with a reference waveform to detect anomalies. If anomalies are found, doctors and first-aid personnel can be notified to act on a specific patient.

Even though remote ECG monitoring does not necessarily require cloud technologies, cloud computing introduces opportunities that would be otherwise hardly achievable. The first advantage is the elasticity of the cloud infrastructure that can grow and shrink according to the requests served. As a result, doctors and hospitals do not have to invest in large computing infrastructures

designed after capacity planning, thus making more effective use of budgets. The second advantage is ubiquity. Cloud computing technologies have now become easily accessible and promise to deliver systems with minimum or no downtime. Computing systems hosted in the cloud are accessible from any Internet device through simple interfaces (such as SOAP and REST-based Web ser-
vices). This makes these systems not only ubiquitous, but they can also be easily integrated with other systems maintained on the hospital's premises. Finally, cost savings constitute another reason for the use of cloud technology in healthcare. Cloud services are priced on a pay-per-use basis and with volume prices for large numbers of service requests. These two models provide a set of flexible options that can be used to price the service, thus actually charging costs based on effective use
rather than capital costs.10(b) Discuss three CRM and ERP technologies based on cloud computing [8 marks]
Social networking applications have grown considerably in the last few years to become the most active sites on the Web. To sustain their traffic and serve millions of users seamlessly, services such as Twitter and Facebook have leveraged cloud computing technologies. The possibility of continuously adding capacity while systems are running is the most attractive feature for social
networks, which constantly increase their user base.

Facebook

Facebook is probably the most evident and interesting environment in social networking. With more than 800 million users, it has become one of the largest Websites in the world. To sustain this incredible growth, it has been fundamental that Facebook be capable of continuously adding capacity and developing new scalable technologies and software systems while maintaining high performance to ensure a smooth user experience.

Currently, the social network is backed by two data centers that have been built and optimized to reduce costs and impact on the environment. On top of this highly efficient infrastructure, built and designed out of inexpensive hardware, a completely customized stack of opportunely modified and refined open-source technologies constitutes the back-end of the largest social network. Taken all together, these technologies constitute a powerful platform for developing cloud applications. This platform primarily supports Facebook itself and offers APIs to integrate third-party applications with Facebook's core infrastructure to deliver additional services such as social games and quizzes created by others.

The reference stack serving Facebook is based on LAMP (Linux, Apache , MySQL, and PHP). This collection of technologies is accompanied by a collection of other services developed in-house. These services are developed in a variety of languages and implement specific functionalities such as search, news feeds, notifications, and others. While serving page requests, the social graph of the user is composed. The social graph identifies a collection of interlinked information that is of relevance for a given user. Most of the user data are served by querying a distributed cluster of MySQL instances, which mostly contain key-value pairs. These data are then cached for faster retrieval. The rest of the relevant information is then composed together using the services mentioned before. These services are located closer to the data and developed in languages that provide better performance than PHP.

The development of services is facilitated by a set of internally developed tools. One of the core elements is Thrift. This is a collection of abstractions (and language bindings) that allow cross-language development. Thrift allows services developed in different languages to communicate and exchange data. Bindings for Thrift in different languages take care of data serialization
and deserialization, communication, and client and server boilerplate code. This simplifies the work of the developers, who can quickly prototype services and leverage existing ones. Other relevant services and tools are Scribe, which aggregates streaming log feeds, and applications for alerting and monitoring.

Besides the basic thread control operations, the most relevant properties have been implemented, such as name, unique identifier, and state. Whereas the name can be freely assigned, the identifier is generated by Aneka, and it represents a globally unique identifier (GUID) in its string form rather than an integer. Properties such as IsBackground, Priority, and IsThreadPoolThread have been provided for interface compatibility but actually do not have any effect on thread scheduling. Other properties concerning the state of the thread, such as IsAlive and IsRunning, exhibit the expected behavior, whereas a slightly different behavior has been implemented for the ThreadState property that is mapped to the State property. The remaining methods of the System.Threading.Thread class (.NET 2.0) are not supported.

Wearable computing devices equipped with ECG sensors constantly monitor the patient's heartbeat. Such information is transmitted to the patient's mobile device, which will eventually forward it to the cloud-hosted Web service for analysis. The Web service forms the front-end of a platform that is entirely hosted in the cloud and that leverages the three layers of the cloud computing stack: SaaS, PaaS, and IaaS. The Web service constitute the SaaS application that will store ECG data in the Amazon S3 service and issue a processing request to the scalable cloud platform. The runtime platform is composed of a dynamically sizable number of instances running the workflow engine and Aneka. The number of workflow engine instances is controlled according to the number of requests in the queue of each instance, while Aneka controls the number of EC2 instances used to execute the single tasks defined by the workflow engine for a single ECG processing job. Each of these jobs consists of a set of operations involving the extraction of the waveform from the heartbeat data and the comparison of the waveform with a reference waveform to detect anomalies. If anomalies are found, doctors and first-aid personnel can be notified to act on a specific patient.