

Research Article

Eucalyptus Cloud to Remotely Provision e-Governance Applications

Sreerama Prabhu Chivukula,¹ Rajasekhar Krovvidi,¹ and Aneesh Sreevallabh Chivukula²

¹ Ministry of Communications and Information Technology, Department of Information and Communication Technology, National Informatics Centre, New Delhi 110003, India

² Center For Data Engineering, International Institute of Information Technology, Hyderabad 500032, India

Correspondence should be addressed to Sreerama Prabhu Chivukula, ddg@ap.nic.in

Received 10 November 2010; Revised 6 January 2011; Accepted 16 February 2011

Academic Editor: Daniele Tarchi

Copyright © 2011 Sreerama Prabhu Chivukula et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Remote rural areas are constrained by lack of reliable power supply, essential for setting up advanced IT infrastructure as servers or storage; therefore, cloud computing comprising an Infrastructure-as-a-Service (IaaS) is well suited to provide such IT infrastructure in remote rural areas. Additional cloud layers of Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) can be added above IaaS. Cluster-based IaaS cloud can be set up by using open-source middleware Eucalyptus in data centres of NIC. Data centres of the central and state governments can be integrated with State Wide Area Networks and NICNET together to form the e-governance grid of India. Web service repositories at centre, state, and district level can be built over the national e-governance grid of India. Using Globus Toolkit, we can achieve stateful web services with speed and security. Adding the cloud layer over the e-governance grid will make a grid-cloud environment possible through Globus Nimbus. Service delivery can be in terms of web services delivery through heterogeneous client devices. Data mining using Weka4WS and DataMiningGrid can produce meaningful knowledge discovery from data. In this paper, a plan of action is provided for the implementation of the above proposed architecture.

1. Introduction

Cloud computing can be defined as on-demand, scalable and elastic web services on public or private fabric consisting any of grid, cluster, virtual machines, and physical machines. Ensuring high reliability, scalability, high availability of citizen-centric e-governance services is very important. Cloud computing makes it possible to accomplish this task cost effectively.

The open source Infrastructure-as-a-Service (IaaS) cloud based on operating system virtualization (Xen, KVM, VMWare, HyperV) allows leasing computing as a utility [1–4].

IaaS cloud allocation is

- (1) set of virtual machines,
- (2) set of storage resources,
- (3) private network to minimize security vulnerabilities,

- (4) application virtualization.

IaaS benefits:

- (1) share underutilized software, network, and storage resources;
- (2) efficient server provisioning;
- (3) effective data persistence.

The Government of India, Department of Information Technology, has initiated national e-governance plan (NeGP) for the execution of e-governance projects in the country, both at central and state levels. It has identified “Mission Mode” projects at both the levels. The NeGP proposes citizen service delivery up to the village level through common service delivery outlets and ensure efficiency, transparency, and reliability of such services at affordable costs to realize the basic needs of the common man [5]. The citizen services to be delivered

could be based on the service-oriented architecture paradigm (as against the present web-enabled services). These services expect adequate networking and computing resources for effective and efficient service delivery. In this paper, we present how grid/cloud/web services technologies could be applied to e-governance application architecture.

The remainder of the paper is organized as follows.

In Section 2, we review the current status of the prototype. An empirical high-performance architecture for e-governance services is presented in Section 3. Technologies that enable the required infrastructure are given in Section 4. Section 5 refers related work. Some of the business cases are listed in Section 6. Sections 7 and 8 propose a future course of action. Sections 9 and 10 detail the implementation progress. The appendix, the acknowledgment, and the references conclude the paper.

2. Current Status of Prototype

National Informatics Centre (NIC) of the Department of Information Technology is providing network backbone and e-governance support to central government, state governments, UT administrations, districts and other government bodies. It offers a wide range of ICT services including nationwide communication network for decentralized planning, improvement in government services and wider transparency of national and local governments. SAN (storage area network) data centers and SWANs (state wide area network) are being established in all 35 states/UTs through NIC as a part of NICNET. Figure 1 shows a typical network diagram with national data centers.

Presently, SAN and SWANs are individually connected and are independently operating without any resource sharing or even without any replica or mirroring storage elsewhere. By connecting all these data centers (SAN) into a cloud, all the computational resources, such as the CPUs, disk storage systems, specialized software systems, and so forth, can be provisioned to all the users connecting to the cloud, including sophisticated users needing advanced capabilities like remote application hosting space, data storage on cloud, persistent transaction states, and distributed data mining.

Also, NIC is having various applications which are running under different platforms on different operating systems (Linux, Windows) without any resource sharing. These applications often need to interact with each other and may also need additional resources temporarily, for a small duration of time. There are many critical mission mode applications where computer services are continuously required for any kind of citizen services. Under these circumstances, breakdown of any machine or operating system or database server or application server brings the services to the citizens to a standstill.

Hence, it is required to plan a business continuity model for such applications. Using the enabling technologies enumerated below, the applications can be deployed as web services in the container to make them interoperable and a solution for business continuity plan (BCP), and disaster management and recovery (DR) can be provided.

3. High-Performance Architecture for a Typical e-Governance Service

To ensure high reliability, availability, and business continuity, following empirical architecture is suggested for e-governance applications.

As shown in Figure 2, the architecture has been devised based on the experience gained in launching several e-governance applications by NIC. The client applications access the remote content through a layered architecture. The operations and integration layers span across application and system architectures. The architecture comprises the following layers:

- (1) governance content management layer,
- (2) application frameworks layer/service provisioning layer,
- (3) service mediation layer,
- (4) process service layer,
- (5) interface integration layer,
- (6) client layer,
- (7) management and monitoring layer.

3.1. Governance Content Management Layer

3.1.1. Content. This layer consists of all data, information, customized analytical reports, and associated applications. Knowledge management systems, enterprise databases, content management systems, and data warehouses will hold both the application and service data, structured or unstructured.

3.1.2. Business Intelligence. Distributed data mining workflows executing distributed algorithms on inherently distributed data can be used to capture rich data mining service semantics. Some of the tools [6] available on the grid for this purpose are Weka4ws [7], Vega [8], DataMiningGrid [9]. Such products can be deployed atop grid-based cloud containers for heterogeneous, large-scale, and distributed data mining applications consisting of common data mining algorithms and workflow components.

3.1.3. Performance. Given a common web service registry on the cloud, the performance benefits accrued by applications include hardware reuse, remote maintenance of hosted servers, storage provisioning and network provisioning with interoperability, scalability, and security. The application's transaction semantics can be reliably represented using some of the cloud infrastructure capabilities like storage snapshots, elastic addresses, and autoscaling. Also, we should mention in this context that e-governance applications can be replicated easily on a cloud resulting in economies of scale.

3.2. Application Frameworks Layer/Service Provisioning Layer. Application services encompass reusable business logic that

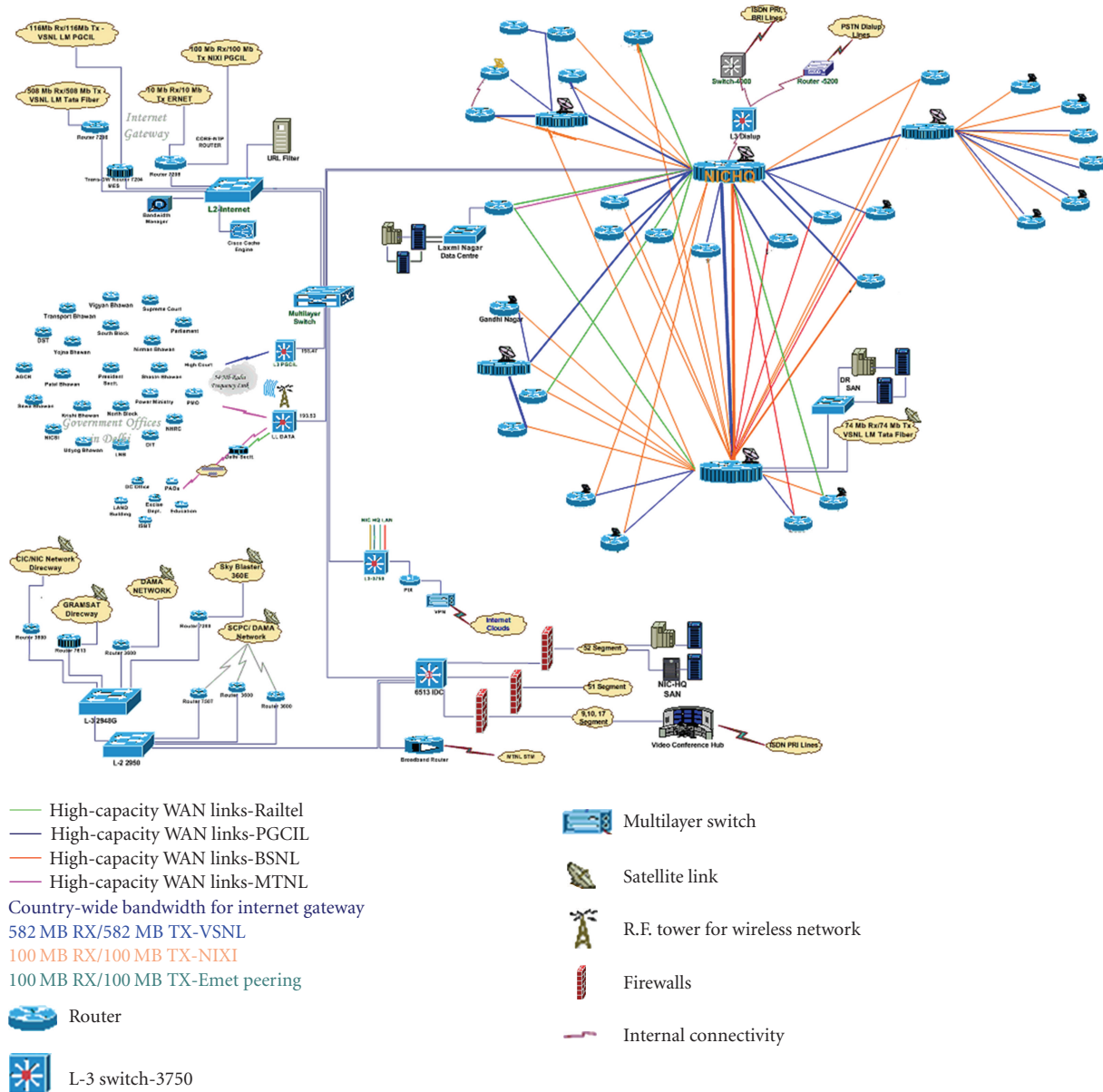


FIGURE 1: Typical network diagram.

is derived from determining the redundancies within application portfolio in a bottom-up approach. Framework services encompass management of structured, semistructured and unstructured data such as information services, portal services, interaction services, infrastructure services, security services, and so forth. The following are such services.

3.2.1. Enterprise Information Integration (EII). This layer exposes data directly using SOAP web services, based on XML, XSLT, XPATH, XQuery, and so forth. XML-based messaging also facilitates legacy system integration.

3.2.2. Access Services. These include common services to validate, enrich, transform, route, and operate. Common

services consist of data collection services—ability to capture electronic data and integrate into backend systems.

3.2.3. Interaction Services. Interaction management involves information aggregation, categorization, classification, and retrieval for farmers, citizens, private sector, and government agencies. This layer handles the conversion, workflow, staging, versioning, and deployment of content across portals and channels.

3.2.4. Localisation and Customisation Services. In India, not only the laws but also languages vary from region to region. So, a layered approach to application design facilitates easy customization according to end-user needs. Language

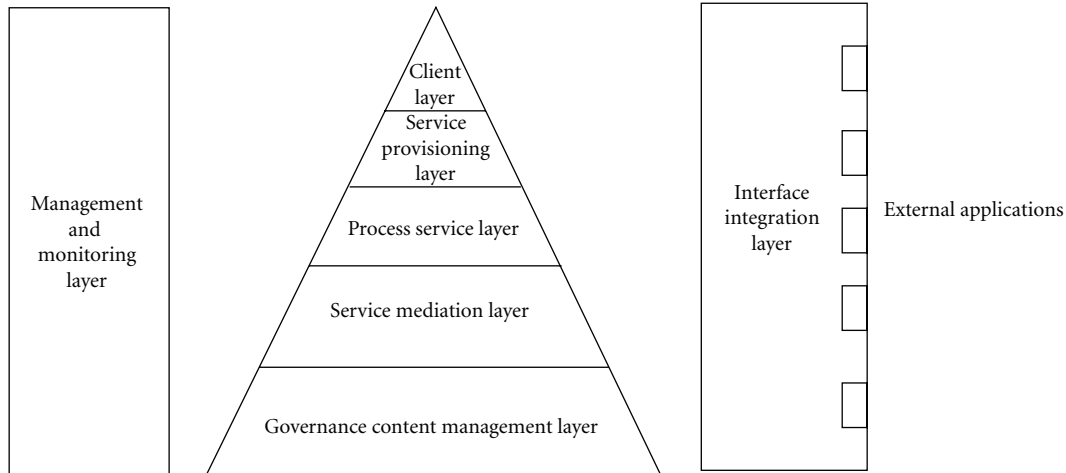


FIGURE 2: Typical e-governance application architecture.

scripts, fonts, transliteration, and local bylaws dictate transaction semantics of this layer. However, for federal applications such as income tax, passports and immigration, and so forth, such customization may be an unnecessary overhead due to uniformity of applicable laws.

3.2.5. Security Services. These are essential for all SOA layers. Important things required are authentication, authorization, single-sign-on (SSO), and central security services for access control. Authentication based on an active directory system allows end users to enter e-governance services delivery channel. The parameters for this authentication would be user ID and password. Authorization is a single repository of access control for all users at state level. This component would be responsible for defining access to various functionalities based on the access control defined in the system.

Some of the security issues implemented in this layer include authentication, access control, logs, damage mitigation, SQL. Injection, cross-site scripting, biometrics, HTTPS, digital certification, and so forth. In a cloud, the security semantics can be addressed at various levels of application isolation using public key infrastructure certificates, overlay networks/virtual local area networks and WS-security specifications [10].

3.2.6. Portal Services. They are information retrieval services on top of ECM services, personalization services, presentation framework, portlet communication framework, and so forth.

3.2.7. Operational/Infrastructure Services. This encompasses generic services across all applications such as logging, auditing, notification, exception handling, and so forth. This would also include device-specific services. These are very specific to accessing some devices in general through application or service interfaces like printer, fax machine, scanner, plotter, and so forth. Session management enables user session management at the server. Audit and logging

enables logging of the events/transactions at the server. These events would then be used to generate security alerts and notifications. Input validation enables validation of the raw data uploaded by the end user before saving into the database. Caching management would cache frequently used resources for faster response. For example, caching of frequently visited web pages for faster delivery at local machine is proposed.

3.2.8. Delivery Services. The overall objective of these services is message cohesion and improved service delivery. For village level citizens, such as farmers, without access to the internet, mobile platform is one of the service delivery channels in the recommended architecture shown in Figure 3.

3.2.9. Channel Services. These consist of various devices like IVRS, mass media, mobile phones, private kiosks, and so forth, to access e-governance services, resources are limited and infrastructure creation involves resources. The objective is to reach/serve maximum number of clientele with minimum resources.

Commercially, all the delivery channels should merge into a single, one-stop and affordable window. Technically, the delivery channels and associated protocols are different, these channels are Internet, mobile, land line phone, and fax. The deliverable could be web pages, SMS, MMS, voice call, a fax message, or an email. A layered approach is adopted to facilitate easy customization and localization. The local language and script issues are handled in the channel layer.

3.3. Service Mediation Layer. This consists of service infrastructure components like service bus, service gateway for external integration, service registry, service repository, and BPEL processor. Service bus will carry the service invocation payloads/messages between consumers and providers. The other important functions expected out of it are itinerary-based routing, distributed caching of routing information,

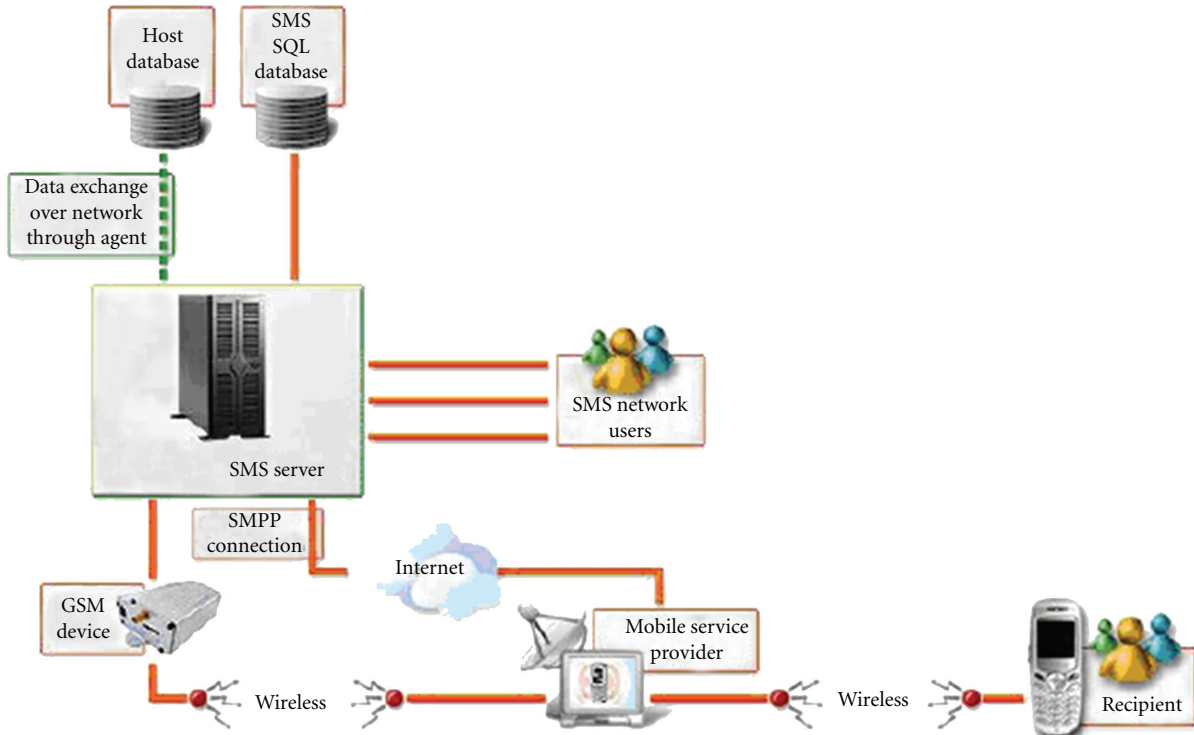


FIGURE 3: Delivery channel.

transformations, and all qualities of service for messaging like reliability, scalability and availability, and so forth. Service registry will hold all contracts (WSDL) of services and help developers to locate or discover service during design time or runtime.

3.3.1. Service-Oriented Architecture (SOA). Web service provides an abstraction layer on top of an enterprise application system. The centre and state services will be exposed through the web services for information dissemination across the states. This paradigm is also known as SOA (service-oriented architecture).

Comprehensive SOA can be taken up only for new information system development, mostly the legacy systems have to be provided with web service wrappers to enable interoperability.

3.3.2. Business Process Execution Language (BPEL). e-governance activities primarily comprise a variety of workflows and processes. Every government department has many processes which can be modeled and implemented as web services. The interoperability requirements of web services so deployed necessitate BPEL-based workflows. In a workflow involving several web services, we require mechanisms like BPEL. Proper workflow of web services ensures consistency and integrity in e-government transactions and activities.

BPEL processor is used for orchestrating the services to compose a complex business scenario or process. Workflow and business rules management supports manual triggering of certain activities within business process based on the rules

setup and also the state information. Thus, BPEL processor can be used for the following:

- (1) interorganizational and intraorganizational process modeling, execution, and analysis;
- (2) defining collaborations between interacting parties in terms of G2G and G2B choreographies;
- (3) avoiding typical anomalies like deadlocks and live-locks in G2G and G2B collaborations;
- (4) conformance testing by process mining the event logs;
- (5) process behavioral mediation in case of multiple service orchestration within a process.

e-governance services delivery is through web services. The major delivery requirements are interoperability, security, scalability, and remote provisioning. To satisfy these requirements, the web services repositories are being created in all states and centre. Within a states interoperability of web services is essential for consistent and effective service delivery. Workflow of web services can be defined using industry-standard BPEL [11]. Thus, orchestrated workflows consist of interactions that are not only internal to the organization but also across organizational boundaries (see, e.g., web service repositories in the appendix). Such workflow among different organizations cannot be aggregated to be rendered as a single RIA without using BPEL. The business logic processing can be done within an n -tier application architecture on a cloud. Therefore, this paradigm necessitates the use of server-side infrastructure resources like data center hardware, rendering client-side processing to a minimum.

3.3.3. Management. Management and configuration component facilitates interoperability between client layer and the various businesses and enterprise services using XML standards.

3.4. Process Service Layer. The e-governance application aimed at rendering services to end users includes e-commerce services comprising domain specific services. These are typically the intermediate services layer and represent shared business process services.

3.4.1. Enterprise Application. The enterprise application represents all the enterprise-related applications and related components such as payment gateway, workflow management, and so forth.

3.4.2. Business Data Services. Business data services encapsulate business data, typically providing a create, read, update and delete (CRUD) type of functionality. They access data from data layer using data access components.

3.4.3. Data Access Components. They are responsible for exposing the data stored in the database to the business data components. Data access object (DAO) pattern is used by these components to encapsulate all access to the data sources, which manages the connection with data sources.

3.5. Interface Integration Layer. This layer consists of composite applications and portals (for interface integration) to integrate to external systems. It consists of integration components, adaptors, and service communication infrastructure. In case of many point to many point communication, the messaging infrastructure is generally modeled after an enterprise service bus (ESB). It may also consist of custom middleware protocols.

3.6. Client Layer. This designates the different types of users accessing the e-governance services. Type of user typically would be an important factor to determine the depth of access to applications.

3.6.1. Thin Client. Thin Client refers to the web browser component installed on the end users' machines. All services would be accessed through the browser, and the proposed system would be capable of delivering the services through all the standard browsers like Internet Explorer, Firefox, Opera, and so forth.

3.6.2. Kiosk. Kiosk refers to the specially installed PC booths for disseminating information to the citizens at village level. These kiosks would be connected to server through SWAN/Internet for information dissemination.

3.6.3. Thick Client. Thick client refers to service delivery component installed on end users' machine. These components would be responsible for interacting with server-side

components. For example, clients for uploading any knowledge base in to the e-governance knowledge repository.

3.6.4. RIA and Mobile Phones. RIA (rich Internet applications) on mobile phone-based e-governance services are to be given importance and throughst so as to reach the unreached and overcome infrastructure constraints in the last mile. In India, some of the villages do not even have unstable power lines. So, to reach the unreached, mobile technology is used as delivery channel with both pull and push approach. To power the mobile in the absence of power grid, proprietary but least cost methodologies are being adopted.

Availability of good signal is another problem. To solve the signal strength problem, also low-cost proprietary methods have been adopted successfully. These last mile solutions are outside the purview of this paper.

3.7. Management and Monitoring Services. Management and monitoring involves all aspects of application architecture like services, SLAs and other QOS, and life cycle processes for both applications and services surrounding application life-cycle. Security is distributed across all layers. Vertical pieces like management, monitoring, security, and development cut across all horizontal layers of applications to manage/maintain the quality of service requirements of e-governance service suite.

We have the option of harnessing either grid technology or cloud computing technology for online mirroring of the key services to ensure business continuity. The SLA guarantees apply to each user who applies for service registration [12]. Further, SLAs are usually application specific.

4. Enabling Technologies

Given below are the cloud technology platform, grid and grid-based cloud platform, and data mining platform over grid and cloud.

4.1. Introduction to Eucalyptus. Elastic utility computing architecture for linking your programs to useful systems (EUCALYPTUS) [4]—is an open-source software infrastructure for implementing elastic/utility/cloud computing using computing clusters and/or workstation farms. Eucalyptus is a distributed computing system implemented using commonly available Linux tools and basic web service technologies. Eucalyptus implements private/hybrid cloud. An Eucalyptus cloud setup consists of five types of components. The cloud controller (CLC) and “Walrus” are top-level components, with one of each in a cloud installation. The cloud controller is a Java program that offers EC2-compatible SOAP and “Query” interfaces, as well as a web interface to the outside world. In addition to handling incoming requests, the cloud controller performs high-level resource scheduling and system accounting. Walrus, also written in Java, implements bucket-based storage, which is available outside and inside a cloud through S3-compatible SOAP and

REST interfaces. Figure 4 shows the component architecture of Eucalyptus.

Top-level components can aggregate resources from multiple clusters (i.e., collections of nodes sharing a LAN segment, possibly residing behind a firewall). Each cluster needs a cluster controller (CC) for cluster-level scheduling and network control and a “storage controller” (SC) for EBS-style block-based storage. The two cluster-level components would typically be deployed on the head-node of a cluster (in fact, this is required if the cluster is behind a firewall). Finally, every node with a hypervisor will need a node controller (NC) for controlling the hypervisor. CC and NC are written in C and deployed as web services inside Apache; the SC is written in Java. Communication among these components takes place over SOAP with WS-security.

Euca2ools are command-line tools for interacting with Web services that export a REST/query-based API compatible with Amazon EC2 and S3 services. The tools were inspired by command-line tools distributed by Amazon (api-tools and ami-tools) and largely accept the same options and environment variables. Euca2ools use cryptographic credentials for authentication. Two types of credentials are issued by EC2- and S3-compatible services: x509 certificates and keys. Euca2ools are used to learn about installed images, start VM instances using those images, describe the running instances, and terminate them. Eucalyptus versions 1.5 and higher include a highly configurable VM networking subsystem that can be adapted to a variety of network environments. There are four high-level networking “modes,” each with its own set of configuration parameters, features, benefits, and in some cases restrictions placed on local network setup.

Features of Eucalyptus 1.6.1 include the following:

- (1) deployment on multiple clusters deployment of components (cloud controller, walrus, storage controller, and cluster controller) on different machines;
- (2) enhanced maintenance support: components are now “crash consistent,” maintaining state across process restart or machine crash;
- (3) enhanced concurrency management: cloud requests are serviced asynchronously with minimal locking using eventual consistency for scale;
- (4) networking improvements, including multicloud support;
- (5) building and installation improvements.

4.2. Introduction to Globus Toolkit. Globus Toolkit [13] is a middleware container that evolved out of various experimental grids in the world. Figure 5 shows the toolkits component architecture. Globus Toolkit includes web services to build grid applications. These services, meet most of the abstract requirements set forth in OGSA. The goal of open grid services architecture (OGSA) is to standardize the services one commonly finds in a grid application (job management services, resource management services, security services, etc.) by specifying a set of standard interfaces for these services. Web services resource framework (WSRF) specifies stateful services. The Globus Toolkit 4 includes a complete

implementation of the WSRF specification. The grid security infrastructure (GSI) enforces security by using a public key infrastructure (PKI) implemented in X.509-compliant certificates for authorization. For communication with a grid system built on the Globus Toolkit, so-called “proxy certificates” are used.

These are only valid for fixed periods of time, and are created for a user using the Globus client-side security API. The data management components from Globus Toolkit 4 are GridFTP, the Reliable File Transfer (RFT), and the data services provided by OGSA-DAI. GridFTP is a basic platform on which a variety of higher-level functions can be built. The RFT facilitates reliable management of multiple GridFTP transfers. The Globus Toolkit 4 data access and integration tools (OGSA-DAI component) provide grid-enabled access to files, relational databases, and XML databases. Other grid services exposed by the tool kit include execution, information, security, and application services.

4.3. Introduction to Globus Nimbus. Globus Nimbus [14] is an open source toolkit from Globus Alliance. It allows you to turn your cluster into an Infrastructure-as-a-Service (IaaS) cloud with an Amazon EC2-like interface. Mission is to evolve the infrastructure with emphasis on the needs of research in science, but many nonscientific use cases are supported as well. Nimbus allows a client to lease remote resources by deploying virtual machines (VMs) on those resources and configuring them to represent an environment desired by the user. Virtualization implementation is based on Xen and KVM. Nimbus 2.4 also provides an implementation of the Amazon elastic compute cloud (EC2) web services description (WSDL) that allows integration of clients developed for the real EC2 system against Nimbus-based clouds.

Figure 6 shows the components architecture of Globus Nimbus. Workspace service is a standalone site virtual machine manager that different remote protocol frontends can invoke. The workspace-control agent implements virtual machine monitor and network-specific tasks on each hypervisor. Context Broker is a service that allows clients to coordinate large virtual cluster launches automatically and repeatedly. It also provides a facility to “personalize” virtual machines. This requires that the virtual machines run a lightweight script at boot time called the Context Agent. Context Agent is a lightweight agent on each virtual machine—its only dependencies are Python and the curl program—securely contacts the Context Broker using a secret key. The cloud client aims to get users up and running in minutes with instance launches and one-click clusters. The EC2 backend allows the service to turn around and secure remote resources off site.

4.4. Introduction to Weka4WS, the Distributed Data Mining Platform. Figure 7 shows the components architecture of Weka4WS. Weka4WS [7] is a framework developed that extends the widely used Weka toolkit for supporting distributed data mining on Grid environments. Weka provides a large collection of machine learning algorithms written in

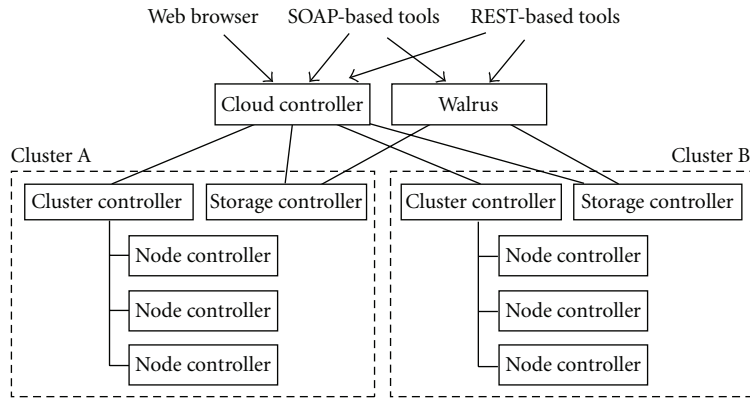
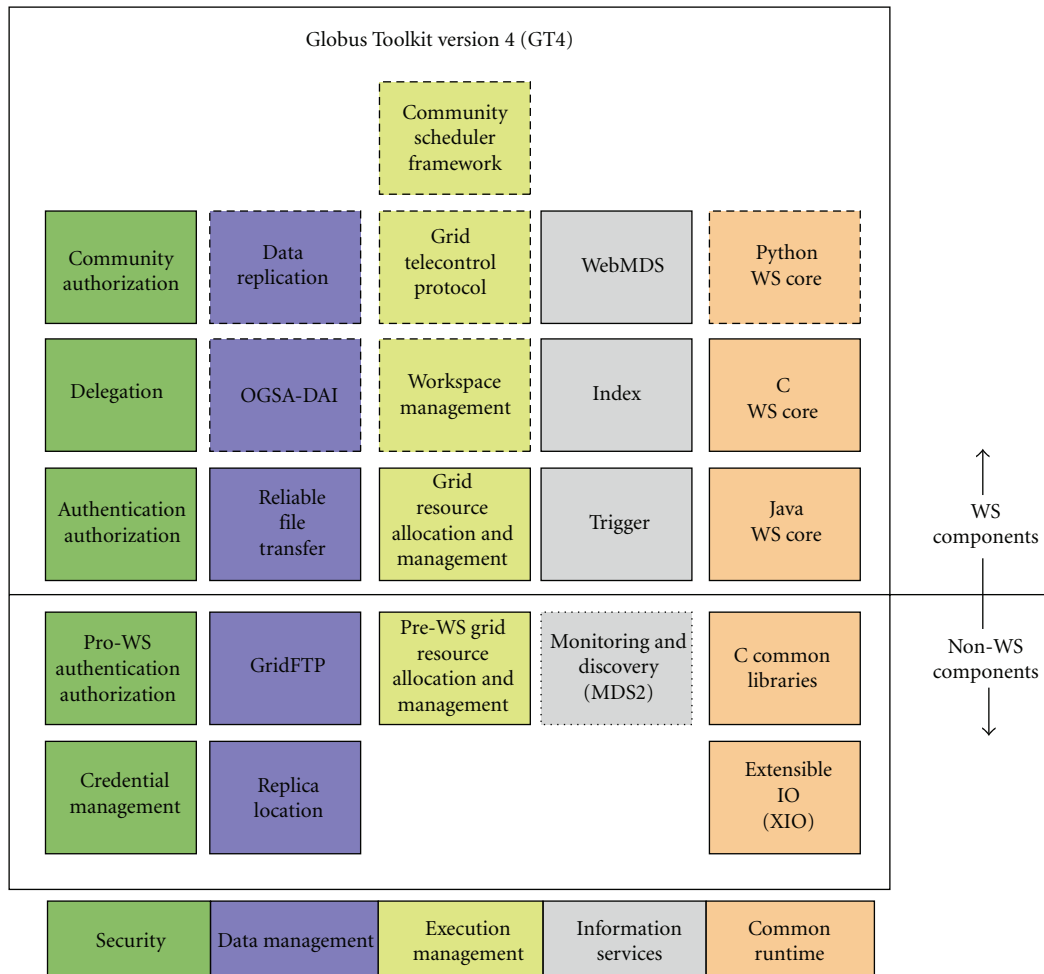


FIGURE 4: Eucalyptus architecture.



- Core GT component: public interfaces frozen between incremental releases; best effort support
- Contribution/tech preview: public interfaces may change between incremental releases
- Deprecated component: not supported; will be dropped in a future release

FIGURE 5: GT4 component architecture.

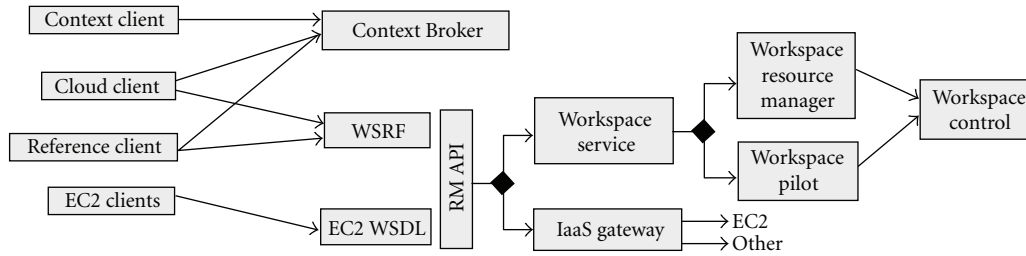


FIGURE 6: Nimbus components.

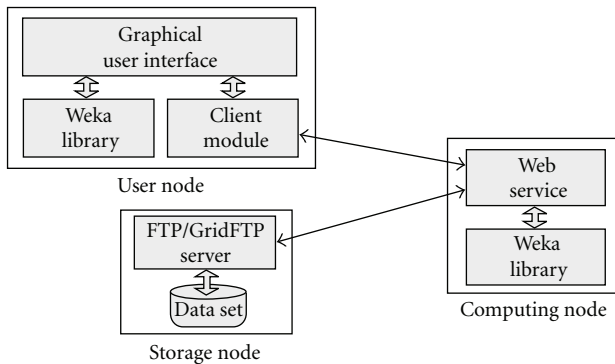


FIGURE 7: Weka4WS architecture.

Java for data preprocessing, classification, clustering, association rules, and visualization, which can be invoked through a common graphical user interface. In Weka, the overall data mining process takes place on a single machine, since the algorithms can be executed only locally. The goal of Weka4WS is to extend Weka to support remote execution of the data mining algorithms through WSRF Web Services.

In such a way, distributed data mining tasks can be concurrently executed on decentralized grid nodes by exploiting data distribution and improving application performance. To achieve integration and interoperability with standard grid environments, Weka4WS has been designed by using the web services resource framework (WSRF) as enabling technology. In particular, Weka4WS has been developed by using the WSRF Java library provided by Globus Toolkit 4.0.x (GT4). Weka4WS is based on the Weka version 3.4.12. In the Weka4WS framework, all nodes use the GT4 services for standard grid functionalities, such as security and data management.

4.5. DataMiningGrid. Based on the Globus Toolkit and other open technology and standards, the DataMiningGrid [9] system provides a set of generic tools and services for deploying data mining applications on grid service infrastructures without any intervention on the application side. The system has been developed and evaluated on the basis of a diverse set of use cases from different sectors in science and technology. DataMiningGrid has been developed based on existing open technology such as Globus Toolkit 4, OGSA-DAI [15], Triana [16], and GridBus [17]. Major features of the DataMiningGrid technology include high

performance, scalability, flexibility, ease of use, conceptual simplicity, compliance with emerging grid (e.g., WSRF) and data mining standards (e.g., CRISP-DM [18]), and use of mainstream grid and open technology.

5. Related Work

Since late 90s, most countries have released their e-government strategies defining their milestones and action plans and have thereafter made significant progress on e-government at all levels of public administration [19]. However, it soon became apparent that absence of common technologies standards and interoperability guidelines yielded considerable leeway to government authorities and let them be focused on their own requirements and define inflexible information systems according to their own assumptions and interpretations [20]. Interoperability has become the key issue in the agenda of the government and public sector [21] since providing one-stop services calls for collaboration within and across public authorities essential for ICT-enabled public services. Lack of interoperability can produce undesirable results in e-governance applications.

National efforts aimed at setting up an interoperability framework have usually devoted efforts to produce standards and guidelines addressing the three levels of interoperability: organizational, semantic, and technical levels [22]. Common principles, such as scalability, reusability, flexibility, security, concurrency, reliability, availability, open standards, and market support, have been adopted over G2G, G2B, G2C national, and cross-country transactions.

In this context, articulating organizational and semantic interoperability issues deserves more priority and effort than the technical interoperability layer that already has mechanisms and standards in place. Significant effort has to be devoted to the development of registries incorporating service descriptions, data definitions, standard codelists, certification schemes, and application metrics in a common repository.

All the above frameworks meant for interoperability proposes are providing very closely or tightly coupled interoperability procedures, methodologies, and mechanisms. In the Indian context, such efforts towards implementation of schema-level interoperability result in significant time delays. The e-governance grid of India (e-ggI) framework for implementation proposed in this paper will be comparatively easier due to loose coupling nature of web services used

for implementing SOA on an IaaS cloud. Further, for legacy systems, it will be an infeasible proposition to redevelop the entire application with SOA. Instead, wrapping of the legacy systems with web services should be sufficient for achieving interoperability. Further, web services used for interoperability are normally stateless (as in the above-described frameworks), but many situations of e-governance applications will require stateful web services in addition to stateless web services. In the generic layered e-governance service architecture described in previous sections, the governance content management layer and service provisioning layer together recommend a maximum of 12 kinds of services to be rendered. Also, grid/cloud technologies can be harnessed for deploying e-governance applications. Private grids/clouds are to be used for the government to create security and privacy of ownership within the government.

6. Issues

Some of the business cases include the following.

6.1. Remote Provisioning of Virtual Servers for Application Development and Hosting. Till now, the physical servers were highly under-utilized, and huge investments have been made towards server procurement and associated power and other infrastructure such as floor space resulting in large wastage. Such physical servers cannot be installed in rural areas such as districts, blocks, and villages where satisfactory infrastructure such as power and floor space is unavailable.

The virtualization technologies [1] in Eucalyptus and Nimbus allow efficient resource usage of the servers by decoupling an operating system and the services and applications supported by that system from a specific physical hardware platform. Given specifications, suitable virtual machine can be created and maintained at the national and state data centers where required hardware and network exist. These virtual machines are remotely accessible by the users from interior areas of the states without the need to have the same facilities as those in state capitals. The provisioned virtual machines can then be used for application development and prototyping, hosting production environments consisting of operating systems, application servers and database servers, middleware systems.

6.2. Web Services. Web services technology allows encapsulation of legacy systems, web-enabled services operationalised in a heterogeneous environment into loosely coupled, self-describing software entities which can be helpful to facilitate interoperability, BPM (business process modeling), and so forth. While REST-based interfaces may be available in some cases, web services developed for various purposes in many states and centre are in .NET and Java/PHP. For example, a web service providing birth certificate can be used as a common web service by a large number of e-governance applications by invocation. Similarly, driving license details should be available as a web service to be consumed by other web services. Many e-governance applications require a workflow of multiple web services belonging to different

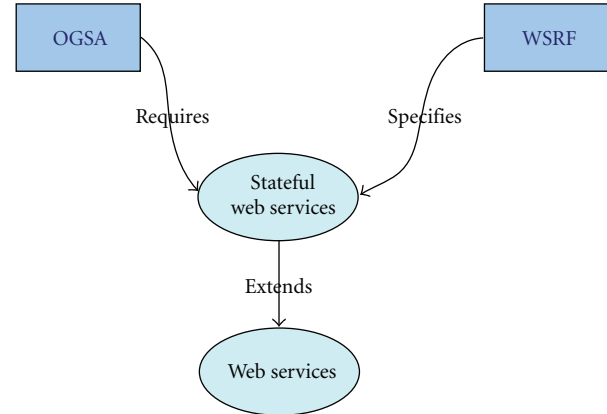


FIGURE 8: Relationship between OGSA, WSRF, and web services.

departments. For interoperability, such workflows can be described using BPEL (business process execution language). Legacy e-governance applications without web services are in silos. Without interoperability, they can produce spurious and unreliable results. Interoperability is possible by using web services. Web service repositories are proposed to be created at centre, state, and district levels. Proposed central web service repository (CWSR) has web services of all applications of all sectors/departments of the central government applicable throughout the country across all the states (example given in Table 1). The state-level web service repository (SWSR) is proposed to be created for all states separately. For each state, there will be a repository of 40–50 web services pertaining to various sectors/departments of the respective state governments. They need to operate within purview of local laws, procedures, and languages. In this context, web service wrappers need to be created around the existing legacy applications.

6.3. Persistent Transaction States. Most of the web services in use do not have the need to store data state across transactions. However, due to complexity of the systems and processes involved in government information systems if there occurs a need to manage application state for interoperability, business continuity, risk management, and so forth, the WSRF stateful grid services techniques of the Globus Toolkit can come in handy.

OGSA specifies a set of standard interfaces for grid services one commonly finds in a grid application, namely, job management services, resource management services, security services, and common runtime, as shown in Figure 8.

Web services resource framework (WSRF), specifies stateful services and the Globus Toolkit 4 includes a complete implementation of the WSRF specification.

Alternatively, the SOAP/REST interfaces for Amazon EC2, S3, and EBS available in Eucalyptus can also be used for persisting data across multiple web service invocations. Further enhancements such as process orchestration with BPEL (business process execution language) and other enterprise architecture styles can be implemented. Example use-cases include integration of land registration and land

TABLE 1: Cloud web services functional description.

Web service	Functionality description
Application Details	All services provide data through web service to respective department
PendingServiceSelect	Selected service-wise pending application list
UpdateServiceStatus	Update service status
DistrictSelect	All Gujarat State District in Gujarati and Eng. name
TalukaSelect	All Taluka in Gujarati and Eng. name
VillageSelect	All Village in Gujarati and Eng. name
All citizen information and reports through web service with photo and thumb	Lightweight application requires only data through web service that is compatible with typical devices like web cam, biometrics, scanner, and printer (no data store locally). Get complete details of citizen. (no data store locally) print reports (no data store locally)
getUser ID_with_DisposeToken	Authorize user through web service
getUser	Complete UserInformation so display user details gujarat portal filter data respective user district, Taluka
GetFamilyDetail	Verify selected BPL family central database through web service
Search_Card_Village_Wise	Verify selected BPL family central database through web service
ElectionCardByPart	Verify selected Voter Name and IDCARDNo. central database through web service (currntly only for Gandhinagar District)
Seald Authority	Gujarat circle pension case file and pension received pension at Rajasthan circle that get pension details of Gujarat circle through web service
Web Entry Data Populate	Search based on criteria (token no., retirement date, pensioner's name, division, office, birth date, pension type). Return pending case detail for inward from web entry module
GetFamilyDetailByFormno	Return all BPL family with FamilyID of selected village
GetVillageList	Return all BPL family with FamilyID of selected village
Device	Upload thumb impression and Photo to database server
Document, RegisterDocument	This service is used to insert update registration document details
Reports	All reports of gARVI through web service for smart client
UserInfo	User information
Master	Get master file
ConfigParam	Dynamic system parameter like stamp duty
Gujarat Portal Service for Index 2	Register document no. with year and sub registrar office (SRO) OfficeID
getDepartments	List of department with IDs
getOfficesByApplication	Give application-wise office list
getOfficesByDept	Give department-wise office list
getUser	Give details of user with office and role
getUserByApplication	Give details of application user
getUser ID_with_DisposeToken	Give user token and dispose the token
getUserWorkarea	Give work area and levels of user
getUsersByApplication	Give list of application users
isValidUser	Authenticate only valid user for application, if not valid then return error code
BillStatus	Linked with gujarat portal services
GetRequestRoR	For getting record of rights (RoR) from client server application RoR-VF7/12, VF8a
UploadRoR	XML to be inserted to central server
Authorization	Validate userID at district, Taluka

TABLE 1: Continued.

Web service	Functionality description
GetRoRData	To use data of land records
GetRegData	Sending registration data from portal to employment exchange database
Get Gujarat portal applications	Application received at portal—transfer to JSK application for further process
Different functionalities for printing record of rights (ROR) using smart client application	Data access through web service from client to central server and vice versa
GetAddresslocation	Capture address details from form
GetAddress	Doorno/Flatno, Housename/no, locality
GetContactDetails	Get H.No, mobile phone, land phone, emailid
UserLogin	User location and role/access permissions
GetBeneficiaryDetails	Beneficiary name, gender, social status, date of birth, income, native village, district, physical status, parents names, thumb and photo, aadhaarid
GetStudentStatus	Student scholarship status
SendScholarshipData	Send data to bank server securely
Feedback	Store entire feedback in database
FeedbackReply	Store reply feedback in database and generate unique ID
Sendemail	Fetch message and send message to the citizen through email
Sendsms	Fetch message and send message to the citizen through SMS
FeedbackReport	Report the feedback on such service for that district for the period
MessagingService	Capture address details, message details, and timestamp
LanguageSelect	Render all the parameters of parametric file in local language
RenderMap	Show location in map for that service
Backup	Store the transaction data of that service for that period in the specified file
Recover	Recovery of transaction data of that service into respective tables for the specified period
FinancialInclusion	Capture bank account details of the beneficiary
CapturePersonDetails	Capture permanent data of person
CaptureFamilyDetails	Capture family details from form
Get, welfareDetails	Person details and welfare schemes details
GetHealthStatus	Get health records and history
TrackFamily	Show parents details
PerformanceAttendance	Details of beneficiaries who availed the service during that period in that location
EpidemicAlert	Generate alerts to the stake holders in the location and neighbourhood
LandOwnership	Owner name, owner ID, property size, and market value
LoanOnLand	Mortgage details if any, financial institution details
KisanCreditCard	Beneficiary name, gender, social status, date of birth, income, native village, district, physical status, parents names, thumb and photo, aadhaarid, kisancreditid, and property ownership details
Inputs	Inputs benefits issue status
AgriMachineryRequest	Capture request details from form/SMS
AgriMachineryStatus	Availability status
AgriCommodityAvailability	Commodity availability status captured from form or SMS
AgriCommodityRequest	Capture customer and commodity details from SMS or form
alert agrimarket transaction	Send SMS to producer and customer about interested commodities

records applications at NIC which are currently functioning in silos.

6.4. Cloud Storage. Eucalyptus implements hierarchical storage according to distributed transaction semantics satisfying often conflicting user requirements for scale and storage. The storage offered can be based on ephemeral buckets or persistent volumes. For a given set of users belonging to a particular security group, Eucalyptus distributed file system Walrus allows use-cases as diverse as effective backup of data securely, snapshots of virtual machine states for persistence, seamless addition of load balancers and application servers through EBS snapshots, elastic IPs, and VLANs. For a richer user experience, the cloud can be accessed with client-side graphical user interface (GUI) tools such as HybridFox, RightScale, and EC2Dream [23].

6.5. Application Virtualization in Cloud. From a value-add standpoint, application virtualization is more than cost-effective hardware use and remote software hosting. Given an enterprise service registry, cloud layer abstracts enterprise infrastructure to dynamically provision network, storage, and applications according to user specifications. Cloud layer of architecture interfaces with other application layers through web services, thus enabling on-demand scalability, availability, and interoperability of applications.

7. Plan of Action

The follow-up plan of action is proposed for implementation

- (1) to ensure interoperability and integrate processes, create web service wrappers for existing application software; develop new applications within the framework of SOA (service oriented architecture) with the above-mentioned layered architecture;
- (2) web service lifecycle can be maintained with the Repositories, registries, and corresponding reporting mechanisms developed at central, state and district level;
- (3) simultaneously, Eucalyptus may be installed in each data centre (on a cluster of at least two servers) to create virtual machines clusters for local or remote access;
- (4) virtual platforms (such as PostgreSQL) can be installed on all virtual machines as desired and made available as services;
- (5) web service repositories can be hosted on virtual machines using virtual platforms in the data center;
- (6) such artifacts shall have preconfigured application servers, database servers, and so forth. Thus, we will have all the three components of a cloud infrastructure: IaaS, PaaS, and SaaS (web services).

8. Recommendations

The following recommendations may be considered as guidelines for future directions.

- (1) Setup private IaaS cloud at national, state data centers. The resulting virtual machines can then be provisioned to remote locations like villages, talukas, and even districts without incurring additional costs on infrastructure.
- (2) The virtual machines can be further utilized for controlling application hosting and data and server migration over the private cloud of NIC data centers.
- (3) The private cloud can also be used for BCP (business continuity planning), DR (disaster recovery), BPM (business process modeling), risk management, performance management, change management, and so forth.
- (4) Specialized e-governance applications involving data persistence across transactions and distributed data mining systems can be further explored.
- (5) Requirement-based updates to SOA governance for fulfilling implementation of on-demand BPM and BPEL in IaaS cloud.

9. Implementation

To date, Eucalyptus-based IaaS cloud is operational in the following NIC data centers:

- (1) Andhra Pradesh,
- (2) Assam,
- (3) Madhya Pradesh,
- (4) Sikkim,
- (5) Kerala,
- (6) Goa,
- (7) Nagaland,
- (8) Pune.

The following example applications have been migrated onto virtual server in the IaaS cloud.

- (1) Drupal content management system [24],
- (2) Disha content application service provider [25].

It is proposed to host other applications such as

- (1) e-procurement,
- (2) Integrated drug surveillance program.

These applications will be functioning in SaaS model over IaaS cloud. The web service repositories are simultaneously being created in various states. A typical list of web services in a state portal according to architecture discussed is given in Table 1.

10. Conclusion and Contribution

The Government of India, Department of Information Technology, has initiated national e-governance plan (NeGP) for the execution of e-governance projects in the country, both at central and state levels. National Informatics Centre (NIC) of the Department of Information Technology is providing network backbone and e-governance support to central government, state governments, UT administrations, districts, and other government bodies. SAN (storage area network), data centers, and SWANs (state wide area network) have been established in all 35 states/UTs through NIC as a part of NICNET.

By connecting all these data centers (SAN) into a private cloud, wherein all the computational resources such as the CPUs, disk storage system, specialized software systems, and so forth, will be provisioned to all the users connecting to the cloud. Using the enabling technologies enumerated, e-governance applications can be deployed as web services to provide interoperability, business continuity, transaction persistence, server provisioning, and so forth.

In this paper, we propose a novel layered architecture for major scalable, reliable, replicable, and economical e-governance applications over a novel e-governance grid-cloud architecture comprising web service repositories at three levels (central, state, and district levels) over the integrated e-governance grid of india (E-ggI). A private IaaS cloud layer is proposed for implementation over E-ggI using Eucalyptus. SaaS model applications are being migrated to the virtual servers in IaaS cloud over E-ggI.

Appendix

Table 1 lists web services that are already operational on the proposed cloud.

Acknowledgments

The authors thank the Director of General National Informatics Centre, New Delhi, for the permission to author and use the information provided in this paper. The authors also thank Dr. G.V.B. Subrahmanyam of Mahindra Satyam, Mrs. Annapurna, and Mr. H.R. Venkateshwar of NIC for their help in the preparation of this paper.

References

- [1] "Linux Virtualization," <http://www.centos.org/docs/5/html/5.2/Virtualization/>.
- [2] I. Foster, C. Kesselman, J. Nick, and S. Tuecke, "The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration," Open Grid Service Infrastructure WG, Global Grid Forum, June 2002.
- [3] I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the grid: enabling scalable virtual organizations," *International Journal of High Performance Computing Applications*, vol. 15, no. 3, pp. 200–222, 2001.
- [4] D. Nurmi, R. Wolski, C. Grzegorzcyk et al., "The eucalyptus open-source cloud-computing system," in *Proceedings of the 9th IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID '09)*, pp. 124–131, May 2009.
- [5] "National e-Governance Plan," <http://www.mit.gov.in/>.
- [6] M. Cannataro, A. Congiusta, A. Pugliese, D. Talia, and P. Trunfio, "Distributed data mining on grids: services, tools, and applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 6, pp. 2451–2465, 2004.
- [7] "WSRF Enabled Weka," <http://grid.deis.unical.it/weka4ws/>.
- [8] X. L. Li, W. Huang, and B. Yan, "Vega information grid: a suite of toolkit for building information sharing scenario," in *Proceedings of the 5th International Conference on Grid and Cooperative Computing (GCC '06)*, pp. 536–542, October 2006.
- [9] V. Stankovski, M. Swain, V. Kravtsov et al., "Grid-enabling data mining applications with DataMiningGrid: an architectural perspective," *Future Generation Computer Systems*, vol. 24, no. 4, pp. 259–279, 2008.
- [10] "Department of Information Technology Cyber Security Strategy," <http://www.mit.gov.in/content/cyber-security-strategy>.
- [11] W. M. P. van der Aalst, M. Dumas, A. H. M. ter Hofstede, N. Russell, H. M. W. Verbeek, and P. Wohed, "Life after BPEL?" in *Proceedings of the 2nd International Workshop on Web Services and Formal Methods (WS-FM '05)*, vol. 3670 of *Lecture Notes in Computer Science*, pp. 35–50, Springer, Versailles, France, 2005.
- [12] "Department of Information Technology Cyber Security Policy," http://www.mit.gov.in/sites/upload_files/dit/files/ncsp-060411.pdf.
- [13] I. Foster, "Globus toolkit version 4: software for service-oriented systems," in *IFIP International Conference on Network and Parallel Computing*, vol. 3779 of *Lecture Notes in Computer Science*, pp. 2–13, Springer, 2006.
- [14] "Science Clouds," <http://www.nimbusproject.org/>.
- [15] "OGSA-DAI," <http://www.ogsadai.org.uk/>.
- [16] S. Majithia, M. S. Shields, I. J. Taylor, and I. Wang, "Triana: a graphical web service composition and execution toolkit," in *Proceedings of the IEEE International Conference on Web Services (ICWS '04)*, pp. 514–524, IEEE Computer Society, 2004.
- [17] R. Buyya, S. Pandey, and C. Vecchiola, "Cloudbus toolkit for market-oriented cloud computing," in *Proceedings of the 1st International Conference on Cloud Computing (CloudCom '09)*, pp. 24–44, Springer, Beijing, China, December 2009.
- [18] P. Chapman, J. Clinton, R. Kerber et al., "Crispdm 1.0: Step-by-step data mining guide," 2000, <http://www.crisp-dm.org/>.
- [19] European Commission, "The user challenge benchmarking: the supply of online public services," Report of the 7th Measurement, 2007, <http://www.epractice.eu/files/media/media1673.pdf>.
- [20] E. Hovy, "Data and knowledge integration for e-Government," in *Digital Government: e-Government Research, Case Studies and Implementation*, H. Chen et al., Ed., vol. 17 of *Integrated Series in Information Systems*, pp. 219–232, Springer, Berlin, Germany, 2008.
- [21] Commission of the European Communities (CEC), "i2010 e-Government Action Plan: Accelerating e-Government in Europe for the Benefit of All," 2006, <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2006:0173:FIN:EN:PDF>.
- [22] United Nations Development Programme (UNDP), "e-Government Interoperability: A Review of Government Interoperability Frameworks in Selected Countries," 2007, <http://www.apdip.net/projects/gif/serieslaunch>.

- [23] "Eucalyptus Client Tools," <http://open.eucalyptus.com/wiki/ec2-compatible-tools>.
- [24] "Drupal Content Management System hosted on Cloud," http://informatics.nic.in/tech_update1.php.
- [25] "Disha Website," <http://disha.ap.nic.in/>.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

