VIDaaS
Design Architecture of Virtual Infrastructure with Database as a Service

Asif Akram and Christian Fernau
Oxford University Computing Services, Oxford University, Banbury Road, Oxford, U.K.

Keywords: Cloud, Cloud Computing, Software as a Service (SaaS), Database as a Service (DaaS).

Abstract: This paper presents design and deployment architecture of research project Virtual Infrastructure with Database as a Service (VIDaaS). VIDaaS supports common requirements of multi-disciplinary researchers and caters for different database types besides the traditional relational database. The ambitious and challenging goals of VIDaaS require innovative and flexible design and deployment architecture. The primary focus of this research paper is to discuss an architecture framework for the existing database system melded with the Cloud computing paradigm in order to improve the scalability and boost the performance of database systems. VIDaaS is based on reusable architectural model benefiting from Service-Oriented Architecture (SOA) and cloud design which is not limited to DaaS and can be applicable to any Software as a Service (SaaS).

1 INTRODUCTION

The Virtual Infrastructure with Database as a Service (VIDaaS) Project comprises two fundamental elements: the deployment of a 'Database as a Service' (DaaS) production-quality software service; and the implementation of a hybrid virtual infrastructure (VI) upon which DaaS, can be developed, refined, run and managed. The hybrid 'Infrastructure as a Service' (IaaS) component provides a virtual infrastructure upon which the DaaS and other 'Software as a Service' components can be hosted. This involves enabling seamless capacity management and the sharing of infrastructure and services between universities and/or other academic institutions.

The DaaS component is developed from the prototype delivered for the Supporting Data Management Infrastructure for the Humanities (Sudamih) Project to a full production service. Functionality is extended to meet common requirements of researchers beyond the humanities and to cater for different database types besides the traditional relational database. From the user perspective, it offers an intuitive means of creating, editing, querying and sharing databases through a simple Web interface.

The VIDaaS Infrastructure as a Service (IaaS) is secure, resilient, reliable, cost-effective and necessarily include access management and accounting tools. VIDaaS aims in delivering a platform between researchers and the low level infrastructure services found on the commercial cloud, which will avoid users being locked in to any one commercial cloud provider (Logicalis, Amazon , Salesforce) and proprietary technologies (Amazon SimpleDB , Microsoft SQL Azure , Amazon Relational Database Service , DabbleDB, Google BigTable). We reckon as a service provider of outsourced databases, VIDaaS can leverage an economy of scale by providing complex systems for many users. Users, in turn, could save large amounts of money on database experts, security solutions and software and hardware maintenance. The VIDaaS is developed from various open source java frameworks and is compliant to Java Enterprise Edition 6 (JEE6) specification.

2 SYSTEM ARCHITECTURE

VIDaaS consists of various reusable components and each component is an independent fully functional service. VIDaaS components deal with Identity and Access management; Payment and Billing; on demand Virtualization; and VIDaaS controller. The top level architecture diagram is shown in the figure 1.

Akram A. and Fernau C..
VIDaaS - Design Architecture of Virtual Infrastructure with Database as a Service.
DOI: 10.5220/0003932301020107
In *Proceedings of the 2nd International Conference on Cloud Computing and Services Science* (CLOSER-2012), pages 102-107
ISBN: 978-989-8565-05-1

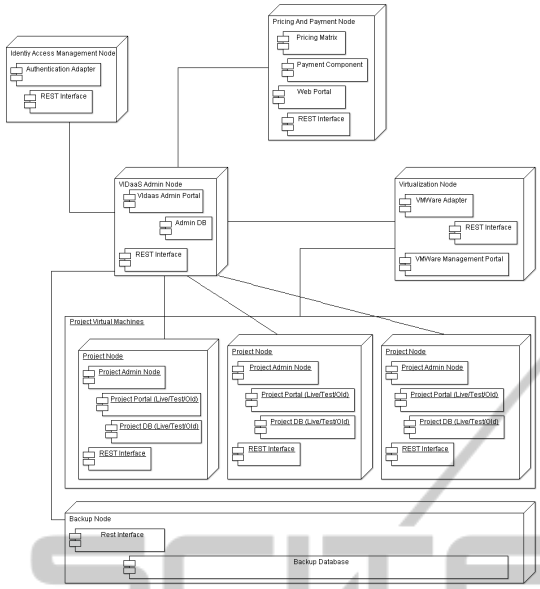

Figure 1: Top level VIDaaS deployment architecture.

The VIDaaS is based on "*split application functions and couple loosely*". The communication between components is via secure RESTful Web services. VIDaaS communication layer uses network-based interfaces and discourages inter-process communication or file based communication paradigms. This allows VIDaaS to effectively scale in the cloud, as demand grows, the system can scale each component independently instead of having to scale the entire application. The separation and reusability of functionality inherent in SOA make it an ideal architecture for the cloud. Each of the VIDaaS components is discussed below with the role it plays in the whole system.

2.1 VIDaaS Admin Node

The "VIDaaS Admin Node" glues together all VIDaaS components together in compliance with institutional policies and underlying resources. VIDaaS Admin Node typically resides on its own virtual node. In a production environment, it may be desirable to have several VIDaaS Admin Nodes to provide redundancy and resilience. The VIDaaS admin node itself consists of three sub-components i.e. *VIDaaS Admin Portal*; *VIDaaS Admin Database*; and RESTFul Interface.

2.1.1 VIDaaS Admin Portal

The VIDaaS Admin Portal is the main Web interface to access the whole system either as VIDaaS users or VIDaaS administrator. The VIDaaS Admin Portal allows VIDaaS users to administrate projects and databases owned by them and it also provides an interface for system administrators to configure and manage the whole VIDaaS system.

VIDaaS registered users can create new projects, administrate all projects owned by them, and view projects for which they have access rights. Once the user selects any specific project he/she is redirected to the project's own virtual machine/node (section: 2.4). Tasks related to any specific project i.e. creating, modifying, deleting and mirroring the database are performed on the project's node.

2.1.2 VIDaaS Admin Database

The VIDaaS Portal stores details of registered users and successfully created project in the local database hosted on the VIDaaS Admin node. The admin database stores all information about registered users; deployed projects, project members and their roles; and databases within each project. The VIDaaS Admin Database links users with their organisational '*Single Sign On*' facilities; maps users with projects and databases; automates projects backup and manages project/database recovery from system level failure.

The single place of storage for system information can be potentially single place of failure and can also result in a bottleneck. On the contrary, it also improves easy system management, monitoring, administration and the report generation. The future version of VIDaaS will support multiple Admin Nodes, each with its own local database for load balancing and resilience. VIDaaS will also provide out of box support to synchronise all VIDaaS Admin Nodes and their local databases.

2.1.3 RESTful Interface

The RESTful Interface of the VIDaaS admin node is the heart of VIDaaS level inter-component communication within the system. The VIDaaS Admin Node interacts with Identity Access Management either when user registers with VIDaaS or sign-in. Similarly through RESTful Interface, the VIDaaS admin node communicates with Virtualization Node and Billing and Payment Node whenever a new project is created, existing projects is modified or existing project is dismantled. The communication between VIDaaS Admin Node and Project Node/s takes place whenever the project database/s are created, modified or deleted. Finally, VIDaaS Admin Node communicates with the Backup Node at regular intervals to trigger the

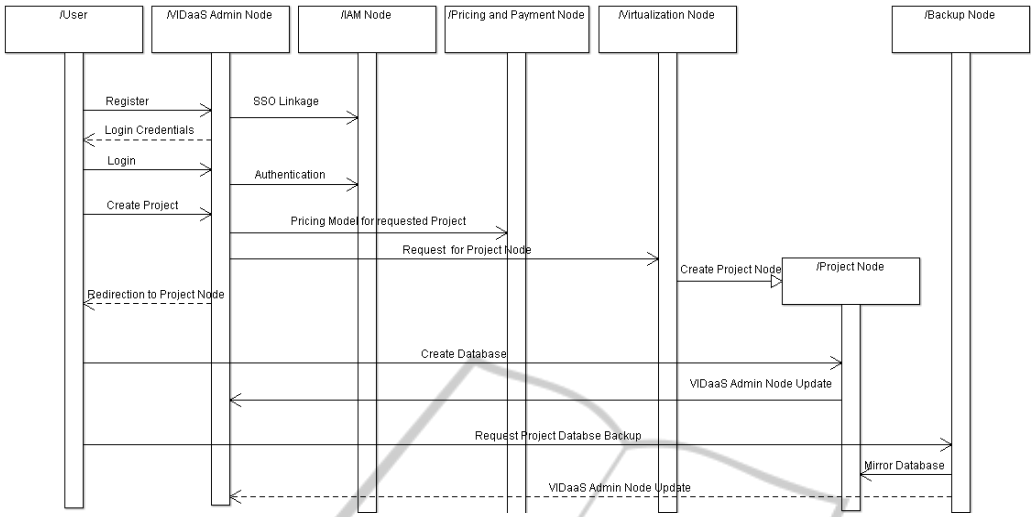

Figure 2: Communication Sequence Diagram between various components.

project's database backup. The simplistic use case of a user registration, login, project creation, project database creation and project database backup in the form of sequence diagram in shown in the Figure: 2. This top level sequence diagram highlights interaction between various VIDaaS components according to their roles.

2.2 Identity Access Management Node

The Identity and Access Management Node is the core of protecting user data from un-authorised access. The VIDaaS system authenticates users through Identity Access Management node so that only authorised users are able to access the services. The VIDaaS system facilitates database sharing and collaboration and supports various user roles within the system

User roles determine how the user authenticates with VIDaaS and what features of the VIDaaS system are accessible to the user.

It is the job of the IAM component of the VIDaaS project to enable a user to access VIDaaS services in as simple and unobtrusive way as possible whilst, at the same time, keeping all their project data completely secure from unauthorised access. The IAM implementation within VIDaaS needs to allow secure access to the VIDaaS system with the minimum interaction from administrators thus automating most of the service or initiated by the end user.

2.3 Virtualization Node

The Virtualization Node encapsulates features like creating, cloning and dismantling individual virtual machines. The Virtualization Node has been separated from the VIDaaS Admin node to aid with the abstraction of VIDaaS from the underlying virtualisation technologies. In current release the Virtualization Node only supports VMware vSphere Hypervisor (based on ESXi) with possibility of other hypervisor technologies such as Xen, KVM, Hyper-V, etc. The Virtualization Node has following three components:

- VMWare Adapter: The Virtualization Node accepts requests from the VIDaaS Admin Node to create and tear down projects. Since each project resides in its own virtual machine, when a new project is created by a user, it is the job of the Virtualization Node to create a new virtual machine and seed it with the relevant software to enable user access via VMWare adapter.
- REST Interface: In VIDaaS the Virtualization Node is only accessed via VIDaaS Admin Node to minimize potential security risks. All communication between VIDaaS Admin Node and the Virtualization Node is through REST Interface which itself uses secure sockets with RESTful Web services.
- VMWare Management Portal: Virtualization Node also has Web interface to interact with underlying hypervisor technology to create, modify and dismantle individual virtual machines. The Web interface to manage virtual machines is only available to VIDaaS administrators, though in future release all this functionality will only be available from the VIDaaS Admin Node.

The separation of Virtualization node from other components within VIDaaS deployment architecture can lead to potential support of various hypervisors in parallel to each other. Though the Virtualization Node is currently using VMware vSphere Hypervisor but the underlying machines are created in Open Virtualization Format (OVF). The OVF standard is not tied to any particular hypervisor or processor architecture.

2.4 Project Node

Each project resides in its own virtual machine called the Project Node. The Project Node is automatically created and made publically available, once the user creates new project via the VIDaaS Admin Node. Users can own more than one project and each project will reside in its own virtual machine. The Project Node has following three sub-components:

- Web interface;
- Database; and
- RESTful Web service.

The Web interface itself is logically split in two interfaces:

- Project Admin Interface: to create database/s; backup database/s; or delete database/s. The Project Admin interface also allows to add or remove other users (already registered with VIDaaS) as project members, update their role within the project; and
- Project Interface: to manage data in each individual database i.e. access, query and update the data within the database. The Project interface is only available when project owner's want their database/s publically available. The Project Interface reverse engineers the database and creates appropriate set of web pages for each table within the database. The generated web pages have a form to add new rows in the table, query database and display table contents in tabular format.

The virtual machine for each project has an instance of a PostgreSQL, installed by default. Users create empty database in the PostgreSQL installation via Project Admin interface.

The VIDaaS project is designed to cater two potential use cases, i.e. existing research groups with working databases and new research groups without databases. Research groups with working database can populate database in various ways either by uploading the existing Microsoft Access database, database in comma separated value format or the database dump from the existing database. The VIDaaS itself supports various versions of Microsoft Access database in a fairly uniform way.

The VIDaaS project has also developed Web based SQL Designer to design the database from scratch. The SQL Designer is developed using various JavaScript libraries and fully support drag and drop. The purpose of SQL Designer is not only to design the database from scratch but also modify the existing databases held by VIDaaS. Though, users can use their preferred existing desktop tools to modify the database and upload the modified database, but we encourage the use of SQL Designer.

The VIDaaS project also has an experimental tool not yet fully integrated to create the project database from publically available database. This tool pulls the database structure and its data from publically available databases and mirrors the gathered information as a local database. This experimental tool has support for different popular databases. This tool can also be used to mirror databases hosted within the VIDaaS to external servers.

The VIDaaS project specifically aims towards research groups and is designed with the notion changes in the research database will be required. VIDaaS introduces the concept of three versions of a single database i.e. live, old and test database. The newly created database by user is always a live database of a project within the VIDaaS system. When researchers plan significant modification to the original database (changes in the structure of the original database), then they have the option to create the "test" database by a single click. The "test" database is pure mirror of "live" database. The planned changes may make the database incompatible with existing database due to addition or removal of tables or columns, changes in tables and columns names, changes in columns data types or even due to changes in relations between various tables. These potentially incompatible changes highlight the significance of the "test" version of the database. Once researchers are happy with changes in the "test" database, they can easily make the "test" database into "live" database. Similarly, on reaching any major milestone researchers are suggested to make a backup copy of their live database as an "old" database. The manual mirroring of the database as "old" database is independent of the VIDaaS system level regular backups done according to the project configurations. Similarly, if the project's "live" database has data loss, data

corruption or any other unexpected problem, project owners can recover the database from the previous milestone saved as an "old" database. The figure 3 shows possible transitions between different versions of a single database.

Figure 3: Three potential versions of a database.

The RESTful interface of the Project node is mainly to communicate with VIDaaS Admin node. The communication is to update the VIDaaS Admin Node of changes made in the project. These changes include changes in project configurations, members and project databases i.e. creating or deleting database, creating or deleting different versions of database. Any change related to database design or data contained within the database is not propagated to VIDaaS Admin Node. The RESTful interface is also used to initiate the database recovery or backup from/to the Backup Node (discussed in next section).

2.5 Backup Node

The Backup Node is a system level backup facility of all databases managed by VIDaaS. The Backup Node is designed with elasticity in the mind. The VIDaaS architecture has provision to manage multiple Backup Nodes as demand grows. In VIDaaS, Backup Nodes are added to the system as required and are dismantled automatically when they are not required. Though, at any given time there is always one Backup Node within the VIDaaS.

The purpose of Backup Node is to provide recovery mechanism from the VIDaaS system level failures. The Backup Node also provisions for human mistakes i.e. project owners/administrators mistakenly deleted database, or wiped out all tables within the database. The system level backup keeps the backup of "live" version of each database hosted by VIDaaS on the Backup Node. The VIDaaS backup is triggered automatically based on the project configurations and Service Level Agreements (SLA).

As described earlier, each project is created by accessing VIDaaS Admin Portal and only VIDaaS Admin node has access to each registered project. The project owners select the backup policy of the database at the project registration time. Currently, VIDaaS only supports handful backup policies i.e. daily, weekly, fortnightly or monthly. The VIDaaS Admin node instantiates the system level backup of the project database according to the project configurations. The Backup Node doesn't run any live database and only archives the database dump in the flat file format.

2.6 Pricing and Payment Node

The Pricing and Payment Node, as name suggests is the charging module of VIDaaS. The Pricing and Payment Node is responsible for charging projects according to their usage of VIDaaS resources and processing the payment. This module is not yet developed and deployed.

3 VIDaaS TOOLS

VIDaaS has developed few tools to complement Database as a Service. These tools provide one stop solution to all database needs of researchers in the form of VIDaaS. Tools developed for VIDaaS are standard based and can be used either as a standalone application or as a VIDaaS integrated component. Few of the main tools integrated with VIDaaS are explained below:

3.1 SQL Designer

VIDaaS provides the tool to design the database from scratch to researchers starting new projects. The *SQL Designer* is user friendly, easy to use, innovative; standards compliant designer with drag and drop support. The SQL Designer can be effectively used by researchers with basic understanding of Structured Query Language (SQL). The SQL Designer also supports different type of relations between tables i.e. one-to-one, one-to-many, many-to-many, imposing and non-imposing. Columns within the table are represented by different colours to indicate primary, foreign and ordinary keys. The current version of SQL Designer produces SQL compliant with PostgreSQL. Tthe future will capable to generate SQL compliant with various popular database systems i.e. Oracle, MySQL, SQL Server, IBM DB2 etc. and users will have a choice to select the database of their choice.

3.2 Database Migration Tool

The VIDaaS also provides tool to migrate existing databases into the VIDaaS with minimum hassle. The database migration tools support most of popular databases available in the market. The only

requirement of this tool is the publically accessible database.

3.3 Form Builder

The VIDaaS also provides the tool to create the HTML form with drag and support. The purpose of the Form Builder is to create a form to support complex database queries from multiple tables. Project administrators are responsible for creating such forms and making them available along with project portal. Users will fill the form created through the Form Builder and users entered values will be translated into SQL queries.

3.4 Supporting Microsoft Access

The support for Microsoft Access database was crucial for the success VIDaaS and its wider acceptance within the research community of Arts and Humanity. VIDaaS has robust and easy to use support for various versions of Microsoft Access database. Researchers are only required to upload their existing databases to the VIDaaS. The uploaded Microsoft Access database is mirrored as a PostgreSQL database with a single click. The mirroring process maintains relational and data integrity of the uploaded Microsoft Access database.

4 FUTURE WORK

VIDaaS system is based on single tenancy database; which is simpler to implement and manage but is burden on cloud resources. For, future versions will investigate multi-tenancy database arrangement; where similar research databases share the same virtual machine. The multi-tenancy arrangement leads to the concept of "*Community Cloud*". The community cloud is used and controlled by a group of users with similar and shared interests. In our case community cloud means researchers from same domain tackling similar problem.

The Billing and Payment Node of VIDaaS is in the design stage and needs to be developed, tested and integrated with rest of the system.

The support for Document Based databases such as CouchDB or MangoDB is missing from VIDaaS and needs to be added. The aim of VIDaaS is an umbrella project to commission different types of databases on demand in cloud environment. We also need to bring diversity in our test users rather than confining VIDaaS to humanity and social science domain.

5 CONCLUSIONS

This paper outlines the architectural design and deployment strategy adopted for database as a service, in the VIDaaS project. The VIDaaS architecture is modular in nature, with different components encapsulating core functionalities in accordance with the Service Oriented Architecture. The modular design of VIDaaS has benefits during the development, deployment and testing stage of the whole project. The paper also briefly discusses various tools developed to enhance the effectiveness of the VIDaaS framework.

REFERENCES

Amazon.com, Inc. 2011, Amazon SimpleDB, viewed 12 December 2011, https://aws.amazon.com/simpledb/

Supporting Data Management Infrastructure for the Humanities, viewed 12 December 2011, http:// sudamih.oucs.ox.ac.uk/

Amazon.com, Inc. 2011, Amazon Relational Database Service (Amazon RDS), viewed 12 December 2011, https://aws.amazon.com/rds/

Virtual Infrastructure with Database as a Service, viewed 12 December 2011, http://vidaas.oucs.ox.ac.uk/

Chang F. 2006, 'Bigtable: A Distributed Storage System for Structured Data', OSDI'06: Seventh Symposium on Operating System Design and Implementation, http://research.google.com/archive/bigtable.html

DabbleDB (now purchased by twitter), Website not available on 12 December 2011

Internet2 2011, Shibboleth project, viewed 12 December 2011, http://shibboleth.net/

Logicalis Group 2011, viewed 12 December 2011, http://www.logicalis.com/

Martinez-Uribe, L 2009, Supporting Data Management Infrastructure for the Humanities (Sudamih), Oxford University Computing Services, Oxford, viewed 12 December 2011, http://sudamih.oucs.ox.ac.uk/

Microsoft Corporation 2011, SQL Azure, viewed 12 December 2011, http://www.windowsazure.com/ en-us/home/tour/database/

Salesforce.com 2011, database.com, viewed 12 December 2011, http://www.database.com/

Ragouzis, N 2008, Security Assertion Markup Language (SAML) V2.0 Technical Overview. OASIS Commit tee Draft, Document ID sstc-saml-tech-overview-2.0-cd-02 http://www.oasis-open.org/committees/ download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf

VMware, Inc. 2011, Palo Alto, viewed 12 December 2011, http://www.vmware.com/