Computing, Information Systems, Development Informatics & Allied Research Journal
Vol. 8 No. 1, March, 2017 - www.cisdijournal.net

Multi-Tenancy Approach: An Emerging Paradigm for Database Consolidation

Amadin F. I & Obienu A. C
Department of Computer Science
University of Benin
P.M.B. 1154, Benin City. Nigeria.
E-mail: frankamadin@uniben.edu, obienu.anayochukwu@physci.uniben.edu
Phone: 08037313900

ABSTRACT

In the past, organizations have tried to consolidate their data onto a single database. However, they have encountered a number of difficulties such as namespace problem, access control to multi-use databases and high operational cost. This paper considered multi-tenancy as it applies to database consolidation. Its relevance in the area of cost, access right and management within an organization. This paper will guide database administrators and developers to decide on the approach to employ in projects involving databases.

Keyword-: Database; Consolidation; Tenant; Multi-tenancy; Servers

CISDI Journal Reference Format
Amadin F. I & Obienu A. C. (2017): Multi-Tenancy Approach: An Emerging Paradigm for Database Consolidation
Computing, Information Systems, Development Informatics & Allied Research Journal. Vol 8 No 1. Pp 89-94.
Available online at www.cisdijournal.net

1. INTRODUCTION

In recent times, software maintenance is becoming a growing concern throughout the software community. Studies have revealed that software maintenance costs are requiring a greater share of the overall software budget. As a result, the organization especially the academic institutions have need to adapt to the environment that will mitigate these costs if they are to remain relevant. Moreover, several institutions are said to be at a crossroads due to the fact that they are operating in an era of dwindling financial resources that are not forthcoming thus affects the efficiency and effectiveness of their functions. Due to the emphasis on budget reductions, ways to reduce costs like consolidating multiple databases become important.

Databases consolidation has been a big trend in the industry for a while now. In the past, organizations have tried to consolidate their data onto a single database, but they have encountered a number of difficulties. For one, they experienced namespace problems, where the names of columns of one user database table may be identical to those in another. Moreover, it is difficult to effectively control access to multi-use databases. This is because different users require different tables, but establishing access rights for each user at the database level is time-consuming and error-prone (Jackson, 2012). Also, high operational cost still exists, which impedes organizational progress that this research wants to address.

As innovations and creativity continue, organizations take strategic interest in the tools and technologies that facilitate their daily activities, which in turn, has necessitated the search for better ways of consolidating data. This search for better options has resulted in the concept of Multi-tenancy. Multi-tenancy is a new concept, and is fundamental to providing the missing link in database consolidation (Morle, 2013; Amadin and Obienu, 2015).

2. MULTI-TENANCY APPROACH

One of the fastest growing business model adopted in the sales of software, based on the principle of outsourcing is Software as a Service (SaaS). With Software as a Service, a service provider hosts application or a software on its infrastructure and then delivers it as a service to multiple tenants. In general, the term 'multi-tenancy' is applied to software development, to indicate an architecture in which a single running instance of an application concurrently serves multiple clients (tenants). According to Olumuyiwa et al. (2015), A Multi-Tenant Database (MTD) is a way of deploying a Database as a Service (DaaS). It refers to a principle where a single instance of a Database Management System (DBMS) runs on a server, serving multiple clients (tenants) as depicted in figure 1. In other words, it is a database architecture that allows multiple databases (pluggable databases) to be consolidated into a container database without changing their applications.

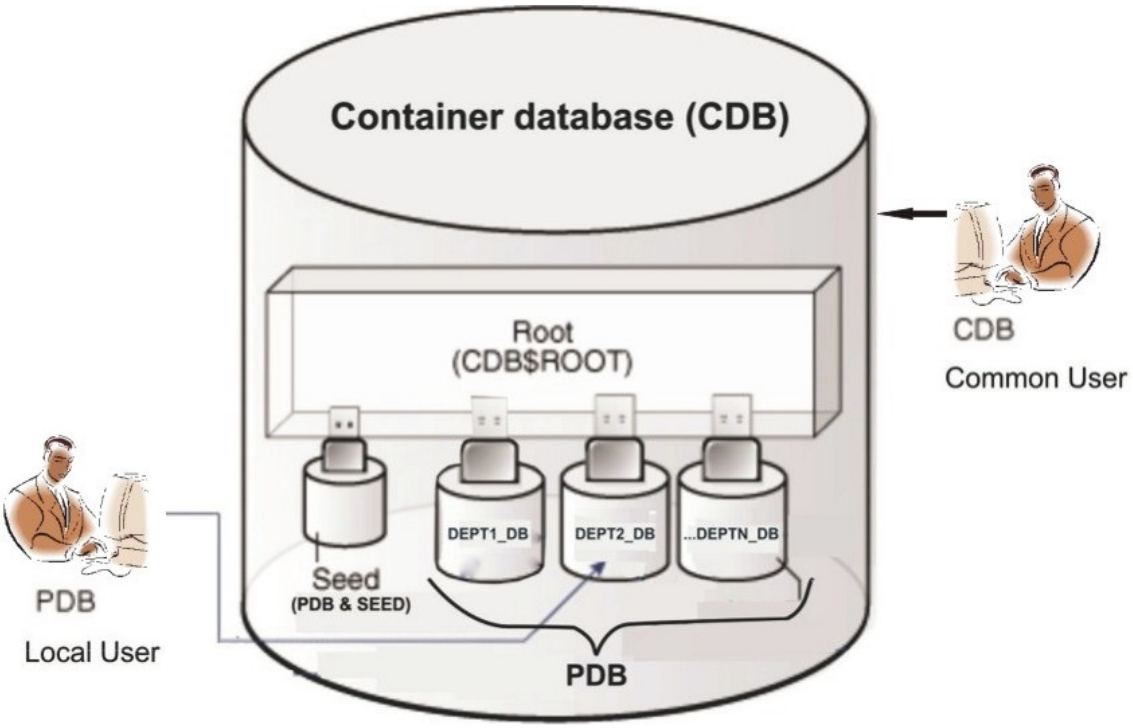

Figure 1: Multitenant Approach

The concept of Container Databases (CDBs) and Pluggable Databases (PDBs) is also a new concept, and is fundamental to providing the "missing link" in database consolidation (Morle, 2013). A CDB can contain one or more PDB, and a PDB is the actual 'database' from the viewpoint of the application. PDBs can be plugged into and unplugged from CDBs using simple commands, and they can be cloned and moved to other CDBs. All the PDBs that are plugged into a CDB share a single instance and can be resource-managed by a single set of controls within the CDB.

3. APPROACHES TO MANAGING MULTI-TENANT DATA

The typical character of Software as a Service applications is 'single-instance multi-tenancy', according to this feature, three main approaches have been proposed: Separate database, Shared Database, Separate Schemas; and Shared Database, Shared Schemas (Jacobs and Aulbach, 2007).

3.1 Separate Database

In this approach, System resources and application code are collectively shared among all the tenants on that particular server. Meanwhile, each tenant has its own set of data that remains logically isolated from data that belongs to all other tenants (see figure 2). Database security prevents any tenant from accidentally or maliciously accessing other tenants' data.

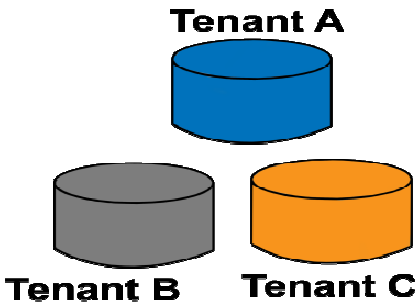

Figure 2: Separate Databases

It is easier to extend the application's data model in other to meet tenant's needs, and can simply restore a tenant's data from backups. However, it costs higher for the relatively high hardware and maintenance requirements. This approach is best suited for clients who are willing to pay extra for added security and customizability. For example, clients in fields such as social security or banking often have very strong data isolation requirements.

3.2 Shared Database, Separate Schemas

This approach involves accommodating multiple tenants in the same database, with each tenant having its own set of tables that are grouped into a schema created specifically for the tenant (see figure 3).

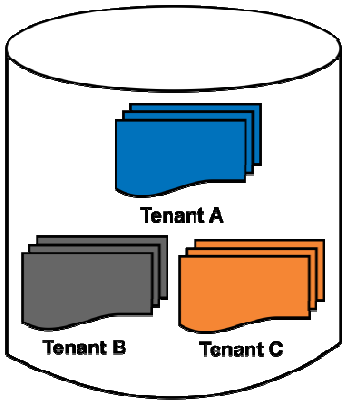

Figure 3: Shared Database, Separate Schemas

The separate-schema approach is relatively easy to implement, and tenants can extend the data model as easily as with the separate-database approach. This approach offers a moderate degree of logical data isolation for security-conscious tenants, though not as much as a completely isolated system would, and can support a larger number of tenants per database server. The separate schema approach is appropriate for applications that use a relatively small number of database tables, on the order of about 100 tables per tenant or fewer. This approach can typically accommodate more tenants per server than the separate-database approach can, so you can offer the application at a lower cost, as long as your customers will accept having their data co-located with that of other tenants.

3.3 Shared Database, Shared Schema

A third approach involves using the same database and the same set of tables to host multiple tenants' data (see figure 4). A given table can include records from multiple tenants stored in any order; a Tenant ID column associates every record with the appropriate tenant. The shared-schema approach is appropriate when it is important that the application is capable of serving a large number of tenants with a small number of servers, and prospective customers are willing to surrender data isolation in exchange for the lower costs that this approach makes possible. Note that, of the three approaches presented here, the shared schema approach has the lowest hardware and backup costs, because it allows you to serve the largest number of tenants per database server. However, because multiple tenants share the same database tables, this approach may incur additional development effort in the area of security, to ensure that tenants can never access other tenants' data, even in the event of unexpected bugs or attacks.

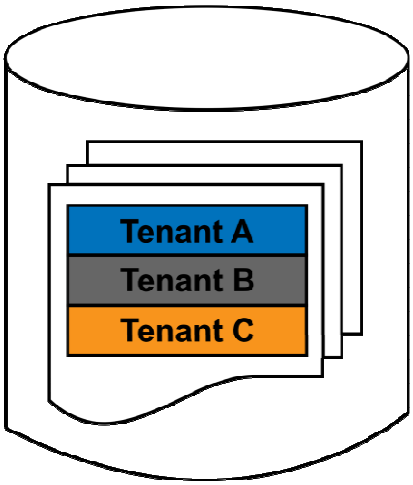

Figure 4: Shared schema

However, there are factors that help in determining the most suitable and appropriate approach of the multitenant database. The use of the system should be one of the influencing factors towards the decision. Elmore (2011) emphasize that the tenant application and usage requirements should be the primary consideration in deciding the right model of the multi-tenant database. Sometimes users (tenants) are not equipped with necessary information about this before taking a decision on what approach to adopt. Their decision is sometimes influenced by what vendors tell them. There is need to examine all these basic factors before approaching a service provider in order to make the right decision on this. Some of these factors are itemized by Keemti (2010), are Size of tenant database, Number of tenants, Number of users per tenant, Growth rate of tenants, Growth rate of tenant database, Security, Cost and Flexibility – ability to create multiple tables by tenants. All these are major consideration that must be considered to make a good decision about the adoption of a multi-tenant database model depending on the tenant individual system requirements.

4. TOWARDS MULTI-TENANT APPROACH IN DATABASE CONSOLIDATION

Multi-tenancy is a recent technology that is gaining wide awareness and acceptance in several fields due to its practical relevance in improving database consolidation. It has found its utility in several sectors such as in academic libraries, business organization, academic transcript processing and so on (Amadin and Obienu, 2015). Currently, most organizations encourage their enterprise units (divisions/departments) to be independent of each other. As a result, several databases are scattered over many machines, within the same organization with reasonably large size. The needed information to make vital decisions is scattered across multiple isolated information islands in a sea of computers and database systems. As a result, the information users do not know what data are available in each database.

They do not know where to find information that they believe are available. This situation makes it difficult, if not impossible, for information users to access information when needed. Moreover, these enterprises' databases are deployed on dedicated servers, which are not fully utilized during much of the time, thereby wasting both hardware and human resources (Curino et al., 2011; Ni et al., 2014, Amadin and Obienu, 2015). This current organizational structure contributes to the increase of Islands of Unreachability, of which many Users cannot access these islands of information, as well as leads to low utilization of dedicated servers. By consolidating hardware and sharing database and files, access to large pool of information, the costs of hardware, storage, and maintenance is greatly improved. Multi-tenancy provides the benefit of managing multiple databases as a single database, and yet retains the isolation and resource prioritization of individual databases.

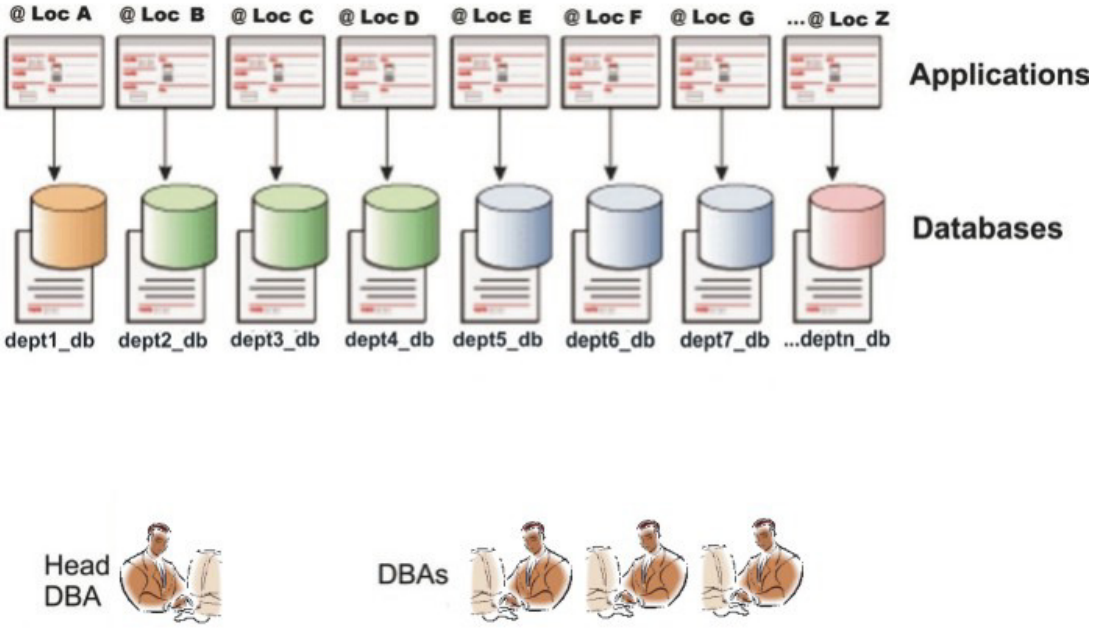

Figure 5: Database Environments before Consolidation

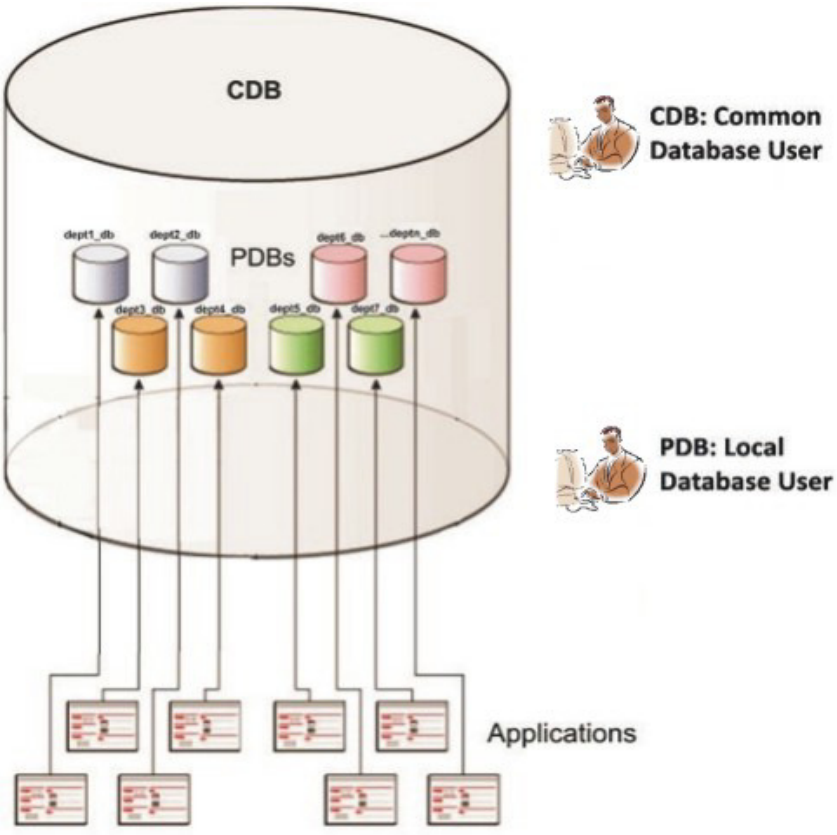

Figure 6: Consolidated Databases

For example, ***n*** PDBs on a single server share one set of database files and one database instance, thereby requiring less hardware and fewer personnel. Following the improvements in hardware technology, especially the increase in the number of CPUs, servers now are able to handle heavier jobs than before. For example, n servers may have one database each, with each database using 10% of hardware resources and 10% of an administrator's time. A team of database administrators must manage the System Global Area (SGA), database files, security accounts of each database separately, while the system administrators must maintain n different systems.

To demostrate the problem in reduced scale, Figure 5 depicts ***n*** databases at different locations, each with its own application and server. A head DBA oversees a team of ***x*** DBAs, each of whom is responsible for one or more databases. Figure 6 depicts the databases in Figure 5 after consolidation onto one database. The Database Administrator team is reduced, with one CDB administrator managing the CDB while few PDB administrators split management of the PDBs.

From the example above, it can be deduced that database consolidation will greatly improve access to information. Access to universal content can be achieved through a common user account. A user account is common, which means that it can connect to any container on which it has privilege. A database user can use a common user account to access information across various databases. As a result, the user can access large pool of information within the organization.

However, adoption the multi-tenancy approach for database consolidation has the following benefits:

i. Cost reduction
ii. Improve access to larger pool of information resources
iii. Monitoring and easier management of the physical database
iv. Fewer database patches and upgrades
v. High Utilization of Dedicated Database Servers
vi. Easier and prompt movement of data and code
vii. Support for Database Resource Manager

5. CONCLUSION

Database consolidation using the concept of multi-tenancy can lead to easier management and monitoring of the physical database, improve efficient utilization of dedicated servers, reduce the operational cost and improve greater access to a larger pool of information resources. This innovation has helped to remove the large-scale investments in software and hardware resources, in upgrading them regularly and also in expensive licences of application software used on in-house hosted database systems. Thus this dynamic approach introduced will help reduce the high computational overhead from the numerous separate databases involved as well as reduce the time spent. Moreover, the information user can now access data from one physical database rather than split attention among dozens or hundreds of database**.**

6. SUGGESTION FOR FURTHER RESEARCH

Multi-tenancy approach raises a number of potential directions for the future work as more research can be done on factors that help in determining the most and appropriate multitenant approach to adopt, as well as, areas where its utility can be found.

REFERENCE

1. Amadin, F. I and Obienu, A. C (2015): Consolidation of School E-Libraries Database Using Multitenancy Approach. Afr J. of Comp & ICTs. Vol 8, No. 4. Pp 11-18.
2. Elmore, A., Das, S., Agrawal, D. and Abbadi, A. (2011), “Towards an elastic and autonomic multitenant database” Proc. of NetDB Workshop. [online].
3. Jackson, J. (2012). ”How Oracle's pluggable databases will work” [Accessed 09 August 2016]. Available at http://news.idg.no/cw/art.cfm?id=FA2CCE67-BFB7-0211-C5BE6DDF6255CD59
4. Jacobs, D and Aulbach, S (2007), ‘Ruminations on Multi-Tenant Databases’. In Proc. of BTW Conf., pp.514–521, 2007.
5. Keemti, P. (2010), “Multi-tenant Database Architecture”. [Accessed 23 August 2015]. Available @ http://www.msdn.microsoft.com/eus/library/aa479086.aspx#mlttntda_topic1
6. Morle, J. (2013). “Database Consolidation with Oracle 12c Multitenancy”. [Accessed 23 August 2015]. Available at http://www.implementerslab.com/atifacts
7. Ni, J; Li, G; Wang, L., Feng, J; Zhang, J and Li, L (2014), ‘Adaptive Database Schema Design for Multi-Tenant Data Management’. Transaction on Knowledge and Data Engineering IEEE, 26(9), .2079-2093.
8. Olumuyiwa ,M. O, Mary, G.,& Kevan, A. B. (2015). ‘Predicting the Impact of the Factors That Influence the Adoption of Multi-Tenant Databases’. Proceedings of the International Conference on Computer and Information Science and Technology Ottawa, Ontario, Canada, May 11– 12, 2015 Paper No. 105