White Paper

Managing Many Databases as One: Pluggable Databases from Oracle

Sponsored by: Oracle Corp.

Carl W. Olofson
November 2013

IDC OPINION

For more than a decade, IDC has been pointing out that the most pressing problem facing IT organizations is the complexity crisis: the mounting numbers of deployed software components of the IT environment, with their dedicated physical systems including servers, storage, and networking. As these combinations of elements multiply, more time and expense go into managing them, coordinating their servicing, upgrading, and replacement; tuning them; and performing general maintenance.

Nowhere is this more true than with database servers. Tremendous amounts of time and effort are required to manage and administer the rising numbers of database instances, even though many, if not most, may be running on the same underlying database management software. Not only does the multiplicity of database servers cost time and effort to manage, but the complexity multiplies the likelihood of human error, which can compromise database availability, performance, and reliability. Also, when database administrators (DBAs) find their time consumed by routine maintenance tasks, they are unable to deliver the kind of adaptability and flexibility in database services that the enterprise requires to remain competitive.

DBAs and IT managers are limited in the range of responses they can apply to this problem. What is required is a capability within the relational database management system (RDBMS) itself to support flexible management and redeployment of multiple databases while enabling hardware sharing and bulk administration. Oracle Multitenant, new with Oracle Database 12c, offers such a capability, enabling the following:

- Support for consolidating multiple databases, each with its own service levels, managed in a single container that may be managed as if they were on separate servers yet pool physical resources, thus reducing complexity while increasing hardware utilization
- Movement of databases from container to container without impacting applications, thereby giving DBAs and IT managers the flexibility to alter their deployment configuration at will for maximum efficiency at all times
- Performing routine maintenance operations, such as patching the software and applying upgrades, in bulk against all the databases within a container at one time, saving weeks, and even months, of time on the operations schedule

- Quickly cloning databases for test purposes using versioning to avoid complete duplication of the data and using masking to protect sensitive data, thereby enabling a more dynamic development and test environment by spinning up test instances in minutes rather than days or weeks

IN THIS WHITE PAPER

This white paper reviews the challenges facing today's enterprise IT managers and DBAs in managing the rising numbers of databases while trying to control cost, reduce errors, and deliver more and better service to users in the form of responsiveness, flexibility, and agility. It considers various approaches to this problem and focuses on a core RDBMS-centric response based on enabling users to consolidate databases systematically through an approach involving a two-level architecture of a database container (or superinstance) and multiple pluggable databases. The paper then considers how Oracle Database 12c, with its multitenant architecture, delivers this capability.

SITUATION OVERVIEW

The Database Management Complexity Crisis

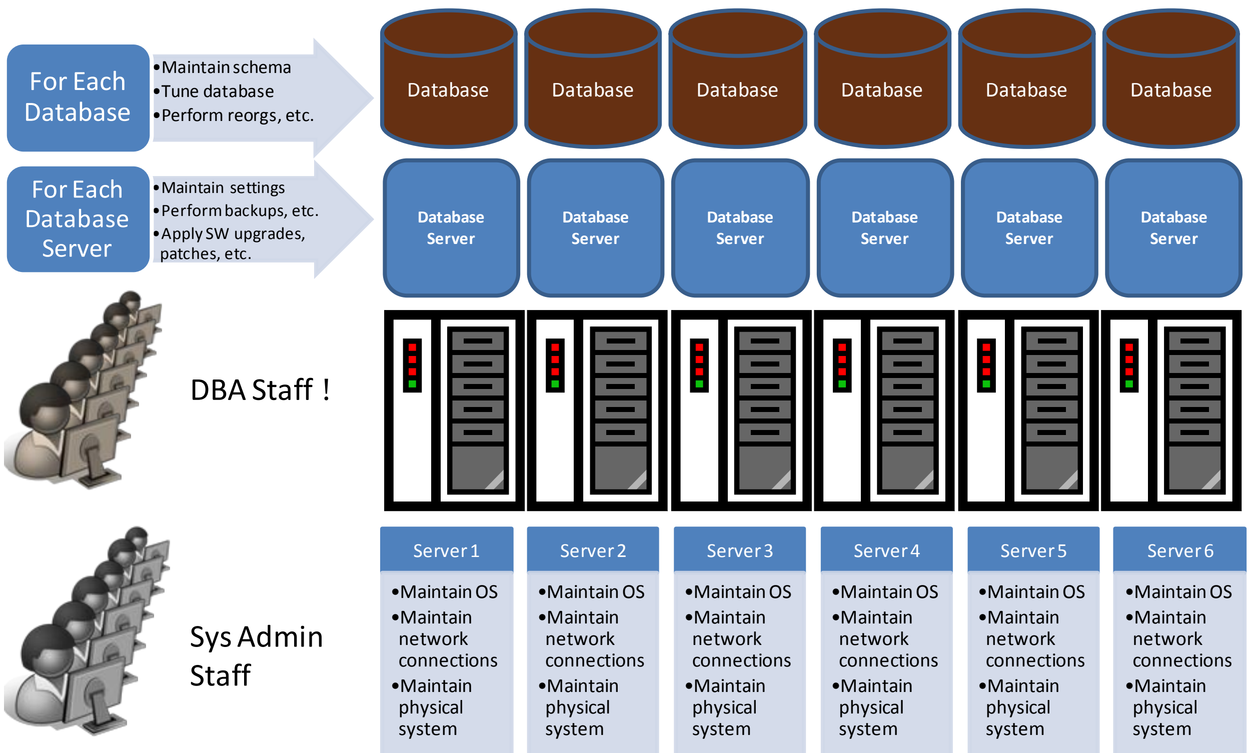
Since the beginning of the client/server era, the number of databases in most medium-sized to large enterprises has grown steadily and has reached the point where database management has become unmanageably complex. Most enterprises have multiple RDBMSs in play, but most have standardized on one RDBMS for their most critical transactional and analytical data management. The idea behind such standardization is simplification. But while contract management has simplified to some extent, the physical management of the database servers remains complicated and time consuming, and the problem is getting worse.

Even though the database servers are running the same core software, and may even be deployed on similar systems, they must each be managed separately, which means scheduling staff time and physical resources for upgrades and parallel testing each time a patch or upgrade occurs. Also, like all the databases in the environment, they are locked into the servers on which they run. Moving them would involve a cumbersome changeover process and require changes to the applications as well. This not only undermines any efforts at greater efficiency but also impedes flexibility and agility in database deployment.

Figure 1 provides a simplified view of how the proliferation of databases on dedicated servers ties up staff time.

FIGURE 1

How Staff Tasks Are Allocated: Dedicated Database Server Scenario $



Source: IDC, 2013

So Many Databases, So Little Time

In study after study, IDC has found that the chief complaint of DBAs is that they spend so much of their time performing routine maintenance and tuning operations that they have little left over for assisting users with application enhancement and addressing the business data needs of the enterprise. In effect, DBAs are forced to spend most of their time performing tedious but time-consuming technical work that the business management level of the enterprise does not understand and only a fraction on tasks that would gain business-level recognition: saving the company money and enhancing business opportunities through the smarter use of enterprise data. A major reason for this is that, with all the database instances in the datacenter, DBAs must perform the same routine tasks over and over for each instance. They are also unable to make operational adjustments that can improve the responsiveness and flexibility of the databases in response to things like periodic or seasonal changes in demand.

The Operational Cost of Many Databases

When most enterprises calculate the total cost of managing their databases, they take into account only license and maintenance fees, hardware costs, and depreciation. They tend to overlook the staff cost, yet multiple total cost of ownership (TCO) studies conducted by IDC have found that more than 60% of the average cost of running a database over any given five-year period is represented by staff cost.

Most enterprises have fixed staffing budgets and so must make do with what they have. Thus that staff resource is spent largely on activities such as remapping data to storage, reindexing databases, repartitioning databases, unloading and reloading databases, and applying and testing patches and software upgrades. While these are necessary activities in any database environment, it is also the case that as the number of databases increases, the staff time required to manage them increases on a linear scale as DBAs perform the same tasks over and over on each database instance. This is because the databases are all separate and cannot share key resources. Such sharing would enable the management of some aspects of database operation on a collective basis rather than an individual basis.

The Software Configuration Nightmare

As the number of databases increases, so does the maintenance effort. In addition, when enterprises apply upgrades that may impact applications, those application maintenance activities must be coordinated with the specific databases they use, so there is a careful maintenance scheduling effort involved. The complexity of that effort is multiplied by the number of applications and databases involved.

Also, because application changes require the creation of a development or test database, copied in whole or in part from production, and because systems resources are not unlimited, the creation of development and test databases must be scheduled for the resources involved. Thus even application developments that could be done in parallel and have no interdependency often are queued up like cars waiting at the traffic volume control light to enter a California freeway.

Databases Locked onto Servers

Because databases are assigned to fixed servers without the practical option of reassignment, those servers must be configured for the maximum demand that the database might apply. So, for example, if one has a set of accounting databases that run very hot during the week or two *after* a quarter- or year-end close, but not so much the rest of the time; another set of databases for sales that run very hot for a week or two *before* the end of the quarter or end of the year to capture last-minutes sales; and another set of databases used for analytics involved in periodic marketing campaigns that run very hot only *during* those campaigns, there is no convenient way for those databases to pool resources so that those that need them can get them at times when others are lightly loaded. Instead, all those database servers are provisioned for peak loads and run at, say, 20–30% capacity the rest of the time.

Ineffective Methods for Consolidation and Dynamic Provisioning

To deal with these problems, users have attempted various ways of consolidating multiple databases onto a single server or clustered servers. These attempts include running multiple database instances on one server as discrete processes, consolidating schemas onto a single database instance, and using a hypervisor technology to run each database instance in its own virtual machine. Each approach is accompanied by daunting problems.

Running Multiple Database Instances as Processes on a Server

If one deploys multiple database instances on one server so that the server supports managing each instance in its own process and resource space, administration problems quickly pile up. For one thing, the systems and database administrators must coordinate their tuning efforts to ensure consistent performance and efficient resource allocation. For another, most RDBMS software is designed on the assumption that the RDBMS has sole control of the machine. Running the same RDBMS software in multiple discrete processes can lead to severe resource contention and even deadlocks.

Consolidating Schemas on a Single Instance

Another approach, one that better utilizes the systems optimization features of the RDBMS, is to blend multiple database schemas onto a single instance. The issue here is that it becomes next to impossible to tune the constituent databases in this blended system effectively. The techniques involved are sophisticated, arcane, and error prone. Usually, the best bet is to tune the whole system for the performance requirements of the most demanding constituent database, which will probably mean massively overprovisioning the server.

Using External Hypervisors

Another technique involves the use of hypervisor technology. By running multiple database instances on a server, each in its own virtual machine, one can fairly simply manage and tune each instance. The problem here is that the hypervisor itself adds overhead to performance-sensitive databases, and the RDBMS again, not knowing it is running in a virtual machine, makes optimization choices that can tend to defeat the load-balancing features of the hypervisor. These issues make this approach suitable mainly for databases that either are not performance sensitive or are test or development copies of production databases.

Cloud Computing

Cloud computing is a datacenter configuration and provisioning strategy that creates flexibility while reducing management overhead and complexity. Instead of fixed systems dedicated to specific databases and applications, databases and applications may be moved around among systems as needed. Instead of managing each resource in isolation, the resources can be managed in groups. When applied in a datacenter, this approach forms a private cloud. Resources may be added and removed at will, creating not just scalability but elasticity.

Core Principles of Cloud Computing

The basis for cloud computing is an approach that involves establishing flexibility (the ability to move resources around), reliability (continuous operation), and elasticity (the ability to increase or decrease resources for a given application as needed), which is achieved through resource virtualization and standardization. In this way, hardware resources are made fungible so they can be reassigned at will. Cloud computing should also enable administrators to manage hardware and software in groups instead of one at a time. In addition, public clouds, and even some private clouds, make use of multitenancy, wherein multiple tenants can access shared resources but are managed at their own performance and security levels, with strict isolation of data.

Applying Cloud Computing to Databases

To apply the cloud computing architecture, application developers have built their applications as interactive components deployed across many servers on a grid, so they may be run on as many or as few systems as necessary to meet the demand for their services, with dynamic load balancing and the adding or removing of server and storage resources as part of the picture.

Databases do not lend themselves to this approach in general because of their need to control both system and storage resources in an exclusive way in order to guarantee performance. Deploying a database in a cloud computing environment in such a way as to fully exploit the capabilities outlined previously requires support within the RDBMS software itself. Such software must support the following key capabilities:

- The movement of a database instance from one machine to another without major administrative effort and without change to the application (
- Backing a server or a cluster of servers to support multiple separate applications, each with its own service-level agreement (SLA), with the administration and tuning of each being no more complicated than if it were running on its own dedicated server
- The application of software patches and upgrades to the server or cluster of servers as a unit rather than requiring such patches or upgrades to be applied to each database instance running on these servers
- The ability to allocate new servers as needed and, once the software has been installed and configured, to allow existing database instances to be moved to the new server with no noticeable disruption to applications or users
- Support for multitenancy within the RDBMS, where multitenancy is desired (Most cloud providers offering multitenancy deliver it through a software layer above the database, which can compromise both performance and security.)

Benefits of a Private Cloud Database Environment

Deploying databases in a private cloud computing architecture would yield the following benefits:

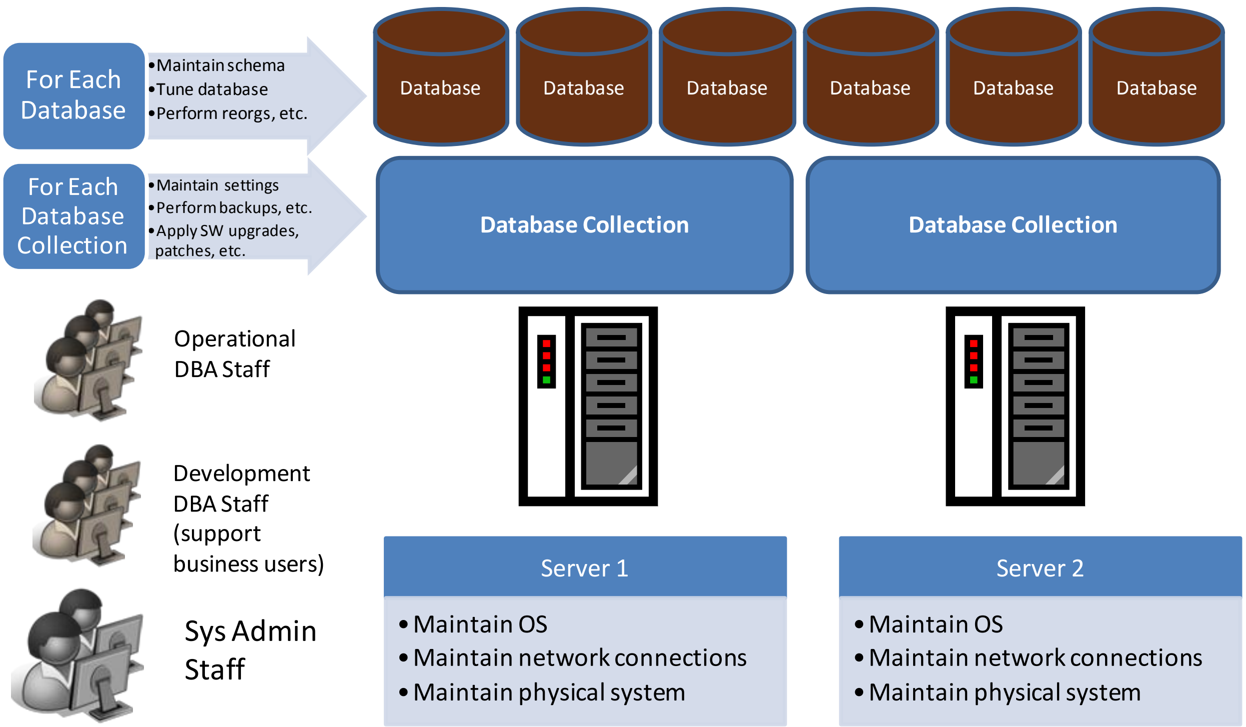
- Reduced overall administrative cost because software patches and upgrades may be applied for a given collection of databases (Thus if one has 5 database installations containing a total of 100 databases, instead of needing to apply the patches and upgrades 100 times, DBAs could apply them only 5 times and shift the constituent databases around during the process to minimize the impact on users.)
- Simple yet discrete tuning, allowing DBAs to focus on more high-value tasks because each database may be tuned as if it were standalone yet realize the efficiency of pooled resources within a collection of databases
- Shorter batch windows required for backup because all the databases in a given collection are backed up in one job, with optimization to realize efficiency in such a bulk operation, yet each database remains individually recoverable
- Much higher system utilization rates because databases that peak at different times can be adjusted and shifted around within a collection, ensuring even utilization of the hardware resources that collection requires (For instance, a sales database might be given more resources, and the accounting database fewer resources, as the quarter nears its end, and then the sales

database is dialed back and the accounting database expands for the quarter-close processing. The marketing database might be shifted to a separate server when its needs expand to support a special marketing campaign. All these things are done without any noticeable impact on applications or users.)

Figure 2 illustrates how staff time can be much more effectively used when databases can be combined into collections.

FIGURE 2

How Staff Tasks Are Allocated: Database Collection Scenario $



Source: IDC, 2013

Oracle Multitenant

Oracle Database 12c introduces a key feature designed to deliver the ability to deploy databases using a cloud computing architecture. This feature is called Oracle Multitenant.

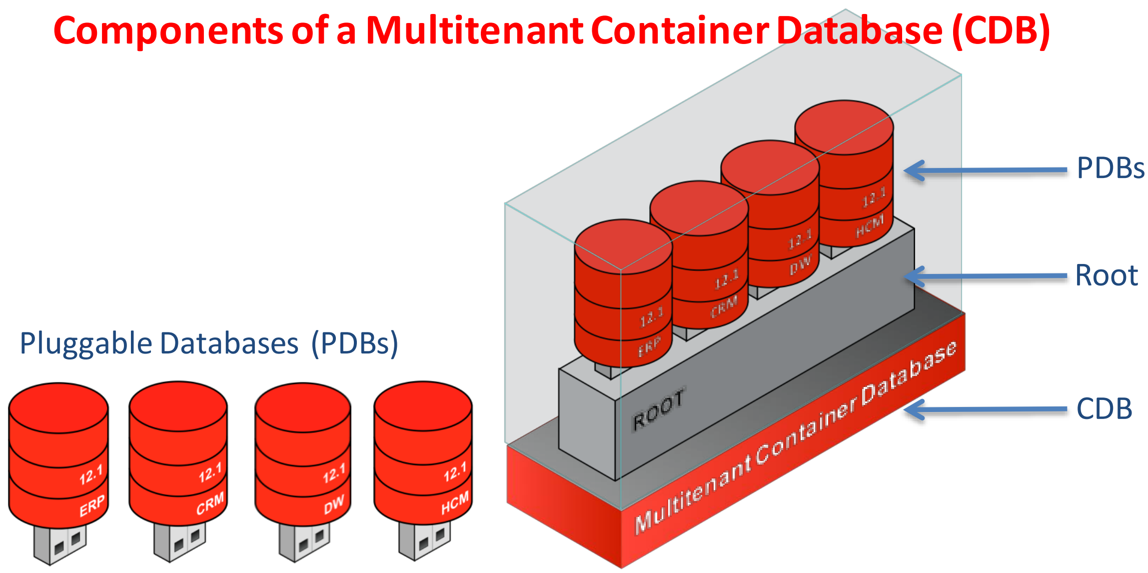
Container Databases and Pluggable Databases

At the heart of this technology is the idea that the physical aspects of database administration are handled at the level of a container database, which is how Oracle Database manages database collection, as discussed previously. This facility is designed to hold multiple pluggable databases within it. Each pluggable database may be logically administered separately, with tuning options that may be applied in familiar ways, yet the databases share the physical resources managed by the container database. These resources may be shifted around on a priority basis among the pluggable databases within a container database.

Figure 3 shows how Oracle illustrates this architecture.

FIGURE 3

Oracle Architecture for Pluggable Databases $



Source: Oracle Corp., 2013

Moving and Copying Pluggable Databases

A pluggable database may be moved from one container to another simply by unplugging it from one and plugging it into another. For users upgrading from Oracle Database 11g, the path to consolidating multiple databases on a single server or cluster is as simple as this: upgrade each database to the Oracle Database 12c release and plug each database into the container database. Then, it's just a matter of balancing the priorities of the databases at the container level.

Manage Software at the Container Database Level

When software patches and upgrades come in, they may be applied at the container level. If you have 10 pluggable databases in a container, upgrading the one container database also upgrades all 10 pluggable databases at once.

Manage Performance at the Pluggable Database Level

Each database may be tuned in the usual ways, applying the knowledge and techniques well understood by Oracle DBAs. Of course, the performance of pluggable databases is tempered by their priority within the container database, but once the right tuning options have been set, adjusting performance and resource demands among the pluggable databases to deal with periodic shifts in demand becomes quite simple.

Support for Multitenancy

Oracle Multitenant delivers an architecture that enables users to establish multitenancy if desired. Unlike schemes that involve assigning tenants to different database servers, or that filter database requests and use hidden qualifying columns, this approach delivers native Oracle Database support while ensuring that each tenant has a database that is isolated from every other tenant, logically and, optionally, physically, with its own security scheme and performance settings.

Benefits of Oracle Multitenant

Oracle Multitenant delivers the benefits of database management in a cloud computing architecture:

- Overall administrative costs are reduced as most administrative tasks are done (at the container database level. This can reduce the total administrative effort by as much as 10 times and also reduces the chance of human error.
- Simple yet discrete tuning, with adjustments through container database priority setting, means much less low-level tuning and much more time for DBAs to spend on high-value tasks.
- Shorter batch windows are required for backup because backup jobs run at the container database level. Database recovery is similarly affected; databases can be recovered at either the individual pluggable database level or the container level. There are even greater benefits for managed collective recovery of container databases when this feature is combined with Oracle Active Data Guard.
- System utilization rates are much higher because individual databases don't need to be deployed on separate systems, each of which is allocated for peak usage. The result is a manifold savings in hardware resources — fewer servers and less memory. Yet the ability to move pluggable databases among container databases delivers far more flexible scalability.
- Since core code and system management is done at the container level, pluggable databases can be spun up and maintained in much less time than single instances, allowing better responsiveness to changing business needs. (

Most $ administrative tasks are done at the container database level. This can reduce the total administrative effort by as much as 10 times and also reduces the chance of human error.

Oracle Multitenant delivers an additional benefit not directly related to the cloud computing paradigm, which is that one can rapidly spin off test or development "clones" of production databases. Because they run off a common base that has been snapshotted (production data changes are not seen by the test environment, and test data changes are not seen by the production system), these can be spun up in minutes rather than hours or days as would be required with non-container database technology. In addition, when combined with Data Masking, the DBA can ensure that all sensitive data in the database has been masked or altered in the test or development view.

Customer Comments

We spoke with several beta customers in the course of researching this white paper. Each commented on how easy Oracle Multitenant was to learn and use as well as the general quality of the software in this release. Notable details are as follows:

- Logical Technologies supports a large number of Oracle customers in Australia and has been working with Oracle Database 12c for nearly 12 months. It found that moving the databases to pluggable databases was a transparent process, with "no perceptible difference in the way they worked." The company put 5 pluggable databases in 1 container database and, with no extraordinary changes to them, realized a 30% improvement in performance. It noted "dramatic savings" in DBA time as well.
- ARAMARK uses Oracle's full suite of applications for ERP and CRM. The company visited the Oracle lab to run a beta test of the upgrade but, because of a technical problem on its end, could not bring a database ready to load on the test system, so it performed the upgrade on a laptop running a copy of its production database. The upgrade ran so well and so smoothly that the system was running flawlessly and the subsequent conversion to pluggable databases on the same system was "seamless."
- Postbank is the largest retail bank in Germany with 14,000,000 customers. The bank supports many of its ongoing banking operations using Oracle Database. Prior to Oracle Database 12c, it had attempted to consolidate databases (as many as 25) on a single server but found the administration and tuning to be complex. It tested Oracle Multitenant for a massive database consolidation, moving 12 databases to 1 container. The bank was so pleased with the test that it set up a container with 20 pluggable databases, replacing the former configuration. It found that it could reduce buffer caches by 25% and enjoy the same performance. Formerly, it had 20 executable environments on a server, 1 for each database, but now it just has 1. Instead of having 20 environments to patch, now just it just has 1.

FUTURE OUTLOOK

Every IT organization should be preparing for a move to cloud computing. It is an architecture that offers such compelling benefits in terms of efficiency, flexibility, and reliability that such a move should be seen as inevitable. The question then becomes how does database technology fit into the mix?

Obviously, Oracle Database 12c with the new multitenant architecture represents an important step in this direction. Other technologies in the areas of systems management, application architecture, and application program deployment are likely to complement what Oracle has done.

Oracle has taken an important leadership step here. The multitenant feature represents the kind of technology that will lead IT into the next phase of computing.

CHALLENGES/OPPORTUNITIES $

Other RDBMS vendors will no doubt seek to address the cloud issues discussed here by various means. Customers will need to keep their eyes and minds open as developments progress. A key challenge for Oracle will be to adjust to market opportunities by making this and other important technology innovations available to the widest possible range of customers at an affordable price. The opportunity in doing so is in influencing datacenter configurations so as to ensure very strong brand loyalty across a spectrum of users for years to come.

CONCLUSION

The biggest challenge facing enterprise datacenters today is the near paralysis caused by the mounting complexity of systems and software. The approach involving virtualization and elasticity that leads to cloud computing is clearly the best way to alleviate this issue, but the problem is thorny when it comes to dealing with databases. Oracle Database 12c with Oracle Multitenant offers an option that represents a smooth transition for existing Oracle Database users and delivers the kind of flexibility and resource manageability necessary to address the complexity crisis and prepare the datacenter for the private cloud, yet does so in a way that is familiar to Oracle DBAs, leveraging their existing skills.

Users should consider the following:

- Evaluate the complexity of your existing environment; how many databases do you manage currently using Oracle Database, and how much effort is involved in managing them?
- What are your utilization rates for database servers?
- How difficult is it to change configurations to meet the changing resource requirements of production databases for peak processing periods?
- How long does it take to spin up a test or development database instance, and do you force projects to queue up for the instance?
- Are you using an external virtualization technology to achieve database consolidation, and if yes, are the performance and complexity of operations acceptable?

If the answers to these questions indicate significant expense and brittleness of operations, then consider upgrading to Oracle Database 12c and using Oracle Multitenant to unlock your deployment flexibility potential.

About IDC

International Data Corporation (IDC) is the premier global provider of market intelligence, advisory services, and events for the information technology, telecommunications and consumer technology markets. IDC helps IT professionals, business executives, and the investment community make fact-based decisions on technology purchases and business strategy. More than 1000 IDC analysts provide global, regional, and local expertise on technology and industry opportunities and trends in over 110 countries worldwide. For more than 48 years, IDC has provided strategic insights to help our clients achieve their key business objectives. IDC is a subsidiary of IDG, the world's leading technology media, research, and events company.

Global Headquarters

5 Speen Street
Framingham, MA 01701
USA
508.872.8200
Twitter: @IDC
idc-insights-community.com
www.idc.com

Copyright Notice

External Publication of IDC Information and Data — Any IDC information that is to be used in advertising, press releases, or promotional materials requires prior written approval from the appropriate IDC Vice President or Country Manager. A draft of the proposed document should accompany any such request. IDC reserves the right to deny approval of external usage for any reason.

Copyright 2013 IDC. Reproduction without written permission is completely forbidden.