WHITE PAPER

LEVERAGING DATABASE VIRTUALIZATION FOR TEST DATA MANAGEMENT

Vikas Dewangan, Senior Technology Architect, Infosys

Abstract

Database virtualization is an emerging trend in test data management (TDM) and is all set to bring about a revolution in the provisioning of non-production data. These tools offer efficient virtual cloning of production databases for software application development and testing, without having to physically replicate the entire database. Coupled with powerful features like taking point-in-time snapshots, write backs to the source database, and rolling back changes; these tools have the potential to bring in significant cost savings and efficiency improvements. In this paper, we explore the concept of database virtualization, industry tool features, and how these tools can be leveraged to improve TDM.

Infosys®
Navigate your next

Introduction

Virtualization technologies have been around for a while and were first introduced in the era of mainframe systems. However, it is only in the recent past, due to advances in technology, that virtualization has been gaining popularity. The term 'virtualization' refers to encapsulating a particular hardware or software resource so that multiple systems can make use of it. By means of virtualization, hardware resources can be utilized more effectively through consolidation, resulting in significant cost savings. Further, virtualization leads to savings in hardware costs; thus resulting in decreased energy consumption and increased green technology. Given these advantages, virtualization will continue to see a trend of increasing adoption in the years to come. There are a number of popular virtualization technologies today, such as server, storage, database and desktop virtualization. Out of these, database virtualization is the focus of this paper.

The field of test data management (TDM) concerns itself with provisioning of production-like data in non-production environments, such as those used for application development and testing and is thus a related area. Database virtualization is an emerging trend in TDM, with the potential of offering significant cost advantages. In this paper we will explore how database virtualization can be optimally leveraged to increase TDM effectiveness.

Traditional approach

The common approach taken by most enterprises today is to have physical copies of the production database in each non-production environment. Non-production environments include environments for software development, testing, and training.

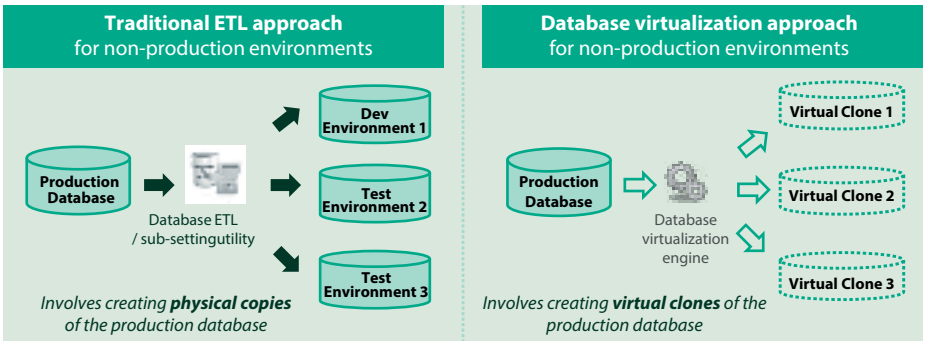
It is not uncommon to find even 10 – 12 such production copies in most enterprises. These physical copies may contain either an entire copy or a subset of the production database. With each additional environment needed there is an increase in physical storage requirements, database administration needs and the time required to setup these databases. Refreshing these non-production databases is a time-consuming and costly exercise. Often, due to contentions, the availability of development and test environments becomes a bottleneck in the software development life cycle and can lead to increased overall turnaround times. In summary, the traditional approach is complex and costly.

Database virtualization – a better approach

With the coming of age of database virtualization, a gamut of opportunities for optimization of non-production environments has opened up. Several enterprises have started taking cognizance of this trend and are actively exploring how best they can leverage this technology for deriving maximum business value.

So what exactly is database virtualization and how does it work? In the case of a database virtualization-based approach, the virtualization engine creates multiple virtual clones of the production database. These virtual copies of the database can be utilized in the same way as the actual physical copies. They will be transparent to the applications in these environments and will continue to function with these virtual database instances in exactly the same manner as they did with the physical non-production databases. There is no need to create subsets of production data and entire datasets can be made available to the relevant applications.

Presented below is a simplified view of the traditional data refresh process for non-production environments compared with a database virtualization-based approach. It should be noted that in some cases there may be an intermediate database between the production and non-production environments, which may be used for masking or sub-setting purposes.



Key features of database virtualization tools

Today, there are a number of vendors offering products in the database virtualization space. Let us take a look at some of the important features and benefits that these products (though not all) provide:

1. **Provision database clones:** The functionality of rapidly provisioning virtual database clones is at the heart of all database virtualization tools. These virtual clones can typically be created in a matter of hours, as opposed to physical databases, which may take up to several days to setup and load data

2. **Resource segregation:** Each virtual database instance is a private copy and is unaffected by changes made to other virtual clones ('Ring fencing')

3. **Lower resource utilization:** Database virtualization tools implement highly-optimized compression algorithms to minimize the in-memory footprint of each cloned database instance. Usually data that is duplicate across clones is maintained as a common block across instances, while modified data is stored privately for each clone, leading to optimization of the memory footprint

4. **Ease of refresh:** Unlike refreshing physical databases, refreshing an existing virtual database clone with the latest production data is relatively simple and straightforward. The clones being in-memory instances, the refresh process is much faster and efficient as there are no physical writes. Most database virtualization engines will refresh the virtual instance with changes only, rather than purging and refreshing the entire database

5. **Point-in-time snapshots:** Often, in test environments, there is a need to take snapshots of multiple databases at the same point in time to ensure consistency of the test data. In the traditional approach, extracting data from multiple databases seldom happens simultaneously and there is some time difference between data extraction processes. This leads to referential integrity issues across databases (as several transactions may have taken place during this time difference), which have to be fixed in non-production environments. With database virtualization engines, it is possible to obtain a virtual clone of each source database against a specific point-in-time. This results in maintaining the referential integrity and consistency of data across the provisioned virtual database instances.

6. **Synchronization with source database:** The updates made to the virtual database clones can be written back to the intermediate database if required. For example, this might be required if the data in the virtual clone has been augmented for testing purposes and a need arises to replicate this to other test environments. This feature can also be used to help maintain a 'gold copy' of the database.
7. **Self-service:** Virtualization tools, once setup, are easy to use and can reduce the dependency on database administrators (DBAs) when compared to the traditional approach where the physical copy of the database needs to be refreshed. Further, the TDM team as well as developers and testers, can be provided the facility to obtain, via self-service, virtual copies of the given source databases.
8. **Roll back to a previous state:** Most database virtualization tools provide a feature for rolling back the database state to a previous date. This is especially useful for testing teams. Often, test data gets corrupted or changed after a test cycle. By using the roll back feature, the test data can be restored to a previous consistent state to enable another round of testing.
9. **Roll forward:** After rolling back, some database virtualization tools provide the facility to roll forward, if required, to a specified date in the database timeline.
10. **Ease of administration:** Database virtualization products offer centralization of all administrative tasks leading to ease of administration. There is no need to account for differences in environment configurations in each non-production environment where the target database is to be deployed.

These features have the potential to bring significant cost savings and efficiency improvements from a TDM perspective.

TDM use cases

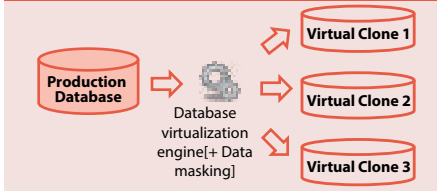
TDM processes can be broadly divided into data refresh and data provisioning processes. The data refresh process involves refreshing a non-production database with required datasets. The data provisioning process is concerned with mining this data and allocation of identified records to end users. Database virtualization is applicable

to the former process viz. data refresh. Let us look at some relevant cases of how it can be leveraged for TDM:

a) Database virtualization in non-production environments

As we have seen, in the traditional approach, each non-production database is a physical copy or subset of the production copy. With a database virtualization tool, the non-production environments are virtual database clones. An added advantage here is that if the data in any environment is corrupted –- let's say after a round of testing –- it is very easy to revert the data back to its original state. Some database virtualization tools provide support for data masking, which is a key requirement of most TDM implementations. A simplified depiction of this scenario is shown below:

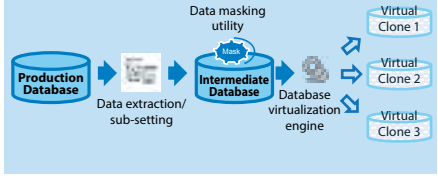
Non-production environment setupby virtual cloning of production database



b) Database virtualization with data masking

Data masking is often a key TDM requirement and not all database virtualization tools support it. In this case, an intermediate database can be setup, which will store the masked data and serve as the source database for the database virtualization engine. The intermediate database might be a full or partial copy of the production database. Data masking can be achieved by either using industry standard TDM tools or custom scripts.

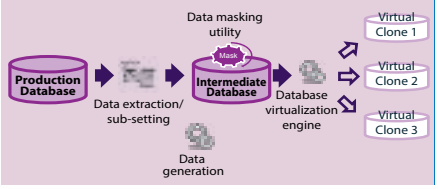
Non-production environment setup with data masking and virtual cloning



c) Database virtualization with data masking and synthetic data creation

In practice, there may be several data requirements from data consumers that cannot be fulfilled from production databases, as the data is unavailable. This may be the case, for example, in new development programs that may require data for testing. In this case, a synthetic data creation tool can be used to generate data, which in turn can be inserted into the intermediate database. As the intermediate database is being used as the source for the database virtualization engine, it will ensure that the virtual database clones reflect the generated data, in addition to the production data.

Non-production environment setup with synthetic data generation and virtual cloning



Benefits for TDM

The key benefits that database virtualization can provide for TDM are:

1. **Faster test data setup and refresh:** Database virtualization can be leveraged for rapidly provisioning a database clone from a source database; for example, production, disaster recovery, a gold copy etc. Similarly, refreshing a virtual database clone can be executed quite quickly.
2. **Cost savings:** Test databases, being virtual clones, do not require physical storage space. Also, database virtualization tools do not maintain full copies of the data — only the changes in data are tracked and logged. This results in significant savings in hardware costs.
3. **Increased reuse:** Most database virtualization tools provide point-in-time snapshots. Thus, if there is a need, for example, to repeat a round of testing or a specific test case, it is easy to roll back the database to a previous state.

4. **Increased test environment availability:** As virtual database clones have a significantly lower resource utilization as compared to physical databases, it is feasible to have larger number of test databases at any given point in time. This can significantly eliminate environment contentions and increase productivity.
5. **Enable TDM for DevOps:** DevOps teams can rapidly provision data for dev and test environments using self-service features, thus enabling continuous integration.

Best practices

While considering database virtualization for TDM, the following best practices should be followed:

1. **Conduct product fitment:** There are several robust database virtualization products available in the market today, with significantly varying feature-sets and technology compatibility. Hence, it is an imperative to carry out a comprehensive product fitment analysis before selecting the right product, keeping in mind the current and long term needs.
2. **Protect sensitive data:** Production databases often contain sensitive information that if exposed can result in serious data breaches. It is imperative to have a mechanism in place to mask sensitive information so that the virtual database clones provisioned for development and testing are properly desensitized. One strategy could be to leverage a database virtualization product that inherently supports data masking or even setting up a staging database that has desensitized data as the source database for the virtualization engine (instead of using the production database directly as the source).
3. **Plan for data reusability:** As database states can be easily rolled back and rolled forward, a data reuse strategy should be put in place to effectively utilize this feature for relevant scenarios (for example, regression testing cycles).
4. **Enable self-service:** Relevant development / testing team members should be provided access to create virtual database clones to decrease the dependency on DBAs.
5. **Monitor performance metrics:** Key database performance metrics, such as SQL query response time, need to be monitored before and after virtualization, so that corrective action may be taken accordingly.

Database virtualization product examples

Listed below are a few database virtualization tools available in the market today, which can create virtual database clones. As this technology is still maturing and evolving, the databases supported by these vendors are likely to increase soon:

#	Tool	Databases supported
1	Delphix DaaS	Oracle, SQL Server, Sybase, DB2, PostGres, MySQL
2	Actifio Copy Data Virtualization	Oracle, MS SQL Server
3	NetApp FlexClone	Oracle
4	EMC XtremIO	Oracle, MS SQL Server
5	VMware vFabric Data Director	Oracle, SQL Server, MySQL

Conclusion

As compared to the traditional approach, database virtualization offers significant opportunities in terms of cost optimization and efficiency improvements in non-production environments. The features offered by these tools are specifically useful for databases used for development and testing purposes. The key benefits of leveraging database virtualization for TDM are faster test data setup, cost savings in disk space and increased reusability of test data. Moreover, the ability to provide self-service, coupled with features like ease of refresh, taking point-in-time snapshots, write backs to the source database and ability to roll back any changes made can be a game-changer in the effective management of test data. Enterprises would be well advised to take a serious look at database virtualization tools, irrespective of whether they have a formal TDM implementation in place.

For more information, contact askus@infosys.com

Infosys®
Navigate your next