Oracle Applications 12.2.X - Editioning and Online Patching

Rajalakshmi Thiruthuraipondi Natarajan
rajalan11@gmail.com

Abstract

Oracle patching is a recurring periodic activity typically performed by the DBAs. In this activity, the patches released by Oracle are applied to the existing Oracle application or data base to fix the security vulnerabilities or to introduce new and advanced features. Prior to Oracle version 12.2.X, applying these patches would need the application and/ or database to be brought down, there by impacting application availability. With Oracle 12.2.X, Oracle introduced a new architecture and AD Online Patching (ADOP) and editioning concepts, where these patches can be applied without bringing down the application for the entire tenure, but for a brief cutover period. To enable online patching, the Oracle instance needs to be properly set-up and there are pre and post patching activities that need to be done to take full advantage of this ability. However, for all practical purposes, there are benefits and challenges in adopting the online patching strategy that the team needs to consider.

Keywords : Oracle upgrade, ADOP., Online Patching, Editioning, Oracle 12.2.X. Oracle Patching Strategy, Patching Cutover.

Introduction

Oracle releases patches for both its database and application every quarter to address the bugs, security vulnerabilities or to adopt to the new compliance requirement or to provide enhanced features. It is the responsibility of the application support in conjunction with the DBAs to analyze these patches for impact and relevance to the installed modules and apply them. To aid this activity, organizations maintain a brief downtime period, usually over the weekends, where the activity is minimal and apply the patches. However, occasionally, there are some pretty big patches that might need a longer downtime than scheduled to apply the patches or need ad-hoc need to apply a patch to address an imminent issue, which would result in additional impact to productivity.

With the release of version 12.2.X, Oracle made a significant overhaul of its architecture mainly focusing on the patching activity. With the new architecture, the DBAs can apply the patches to the so-called patch edition, without bringing down the application, which would be running on the run-time edition. The only downtime that would be needed is during the cut-over, where the runtime and the patch editions are swapped making the patched side available for usage. Unlike the downtime needed for patching, the cutover downtime is fairly constant and substantially low. To effectively do the online patching, there are preparation steps required and various stages during the course of online patching that needs to be done. This document explores these steps, and the margin of benefit provided by this novel ability while also analyzing the limitations and ease of use.

Oracle 12.2.X Editioning

Before we get into the online patching, we need to understand the editioning concept which is the base for on-line patching. Prior to 12.2.X, the Oracle technology stack contained only one sets of data base and file systems, which was used by the application at run-time. Hence, any patching activity would need the application, and the database brought down, since patching effectively modifies the objects and cannot be done when the application is in use as it might cause data and file corruption. Beginning 12.2.X, the database objects and application files in the server are duplicated and each set is called an "Edition". At any given point, one set is actively used in the application and is operational. This is called the "Run-Time Edition". The other set is dormant, more like an asynchronous shadow. These usually are exact replica of the runtime objects and files, and this set is called the "Patch Edition".

Though residing in the same instance, the runtime and patch editions are relatively independent of each other, i.e., anything that happens in the patch edition does not affect the runtime edition and vice versa and it is in this patch edition that all patches are executed while the system is still running. Once the patches are applied and verified successfully, there is brief cutover period, where the unpatched runtime edition is swapped with the patched edition and now the objects in the patch edition becomes the runtime edition and becomes operational. To the end-user, there is no visible change since both these editions are exactly the same to begin with.

This editioning concept applies to both database objects and the file system objects. At the database level, every schema, that needs to be eligible for online patching, needs to be edition enabled, which makes the objects under them eligible too. Similarly, in the file systems, where the forms, reports and other non-database objects are stored, have two copies each one referred as *fs1* and *fs2*, referring to file system one and two. At a given point of time, one these file systems will be runtime and other patch editions. Additionally, there also is a non-editioned objects and files systems that are common for both editions. These are generally objects and files that are not affected by patching, such as, inbound and outbound directories that cannot be switched or other files such as log and output that might be needed irrespective of the file systems.

ADOP

Application DBA Online Patching, generally referred to as ADOP is a utility introduced and extensively used in Oracle 12.2.X, which enables the DBAs to apply patches without bringing down the application or database. It leverages the Oracle Edition-based Redefinition (EBR) feature at the database level and the dual file system in the server level to apply the patches while the application is still running. The *adop* command, which is used for online patching is used to perform or skip each steps in online patching, like prepare, apply, finalize, cutover and cleanup. These different stages of online patching have their own purpose to ensuring the success of the patching activity. This online patching is not only used for Oracle standard objects, but can be used for custom objects too, provided they strictly abide by the online patching requirements.

Prior to execution, all necessary pre checks such as compatibility, synchronization, eligibility , etc., need to be done, without which the patching might not be successful. It also goes without saying that though the process of applying patch has changed, the due diligence of testing the patch both for application and functional impact needs to be tested thoroughly in lower instance before proceeding to production and any anomaly or ambiguity needs to be properly addressed along with the contingency plan in case of failure.

Online Patching

Prior to actually applying the patch using online patching, there are a few prep steps to be taken to ensure that the patching is successful. Not only this makes sure that the patching is successful, but also to make sure that none of the updates from the runtime edition is missed during cutover. As mentioned earlier, the patch and runtime editions need not be in sync, however, if the patch edition is to be made runtime, it should have all the runtime changes, else there is a chance that the new runtime might miss a few functionalities.

Following are the steps in involved in online patching. There are slight variations on how to trigger these based on single node and multi node environments, but the underlying operations are the same –

Validate - This is an optional step that can be run before any the start of online patching. This basically, checks the server to ensure that the necessary resources available for the patching to continue.

Prepare - This is the first step in the whole online patching process. As the name implies, it prepares the instance for the patching activity. It performs all the necessary checks to make sure that instance is poised for online patching. In also synchronizes the standard patches (not custom code) from runtime to patch edition if anything is missed. It also checks for any missed activity from the prior patching, such as clean-up (or any other step). It is in this step that the patch is downloaded and unzipped kept ready for execution.

Apply - In this step the patch is actually executed in the patch edition. There are several optional report generating parameters that can be passed to capture the patch application log. As long as the validate and prepare phase are run and no issues found, this is fairly a straight-forward activity.

Finalize - This is step that prepares the instance for final cutover. The key operation here is to ensure that the data base objects are recompiled and are valid in addition to other validations.

Cutover – This is the phase where the switch happens form patch to runtime edition, and it is in this phase that the application goes down. The services are brought down sequentially, like starting with workflow services, concurrent manager and other application services. The initial switch happens at the database level, where the patch edition is made the runtime edition and vice versa, then moves on the application servers, where the file systems undergo the same and the respective environmental values are set. Once the switch is successfully completed, the database and the application services are brought up.

Cleanup – This is technically the last step to be performed as part of the online patching activity. This can optionally be skipped, however, if skipped, it will get triggered the next time prepare is run. This step cleans up the any obsolete objects that were potentially modified as part of the patching activity and any connections to old editions. This can either be a done in a quick mode, where only the essential clean up is done, or the full mode, where the clean-up much more intensive and recovers as much storage as possible.

In addition to the above steps, there are other non-mandatory or special steps that might be needed under specific occasions such as -

abort – As the name suggests, this command is to terminate the patching process for whatever the reason. This can only be used in prepare and apply phase, as once apply phase is done, the patch is already in the patch edition of the system. In addition to terminating the patching activity, it also cleans up and residue left over by the in-progress patching so that the application can resume its original version.

fs_clone – This command is to clone the runtime edition to patch edition. As mentioned earlier, in prepare phase, only the missing standard patches are copied to the patch edition, but the custom objects are not. It is quite possible that the IT team adopts a mixed approach, where the standard patches are applied via online patching, but the customizations are directly applied to run edition, in this case to ensure that the customizations are not missed out as part of the patching activity, it is recommended that the fs_clone run before the start of the apply phase to keep the patch edition up-to date before applying the patch.

Analysis of the ADOP Usability Challenges

Undoubtedly, online patching is a big step forward in Oracle application maintenance activities, however, the question arises, how effective is it practically. Owing to the complexity of the process, Oracle enforces strict prerequisites before the instance can be used for online patching and any unaccepted deviations would render this functionality unavailable. While it is true that Oracle guides the implementation in every step of the way to make the upgraded 12.2.X instance to be ADOP ready, yet some steps can be skipped optionally, hindering the full ability of ADOP.

An instance that is properly upgraded to 12.2.X with all the instructions and recommendations, along with the properly configured database and application systems will be a perfect platform for online patching to run seamlessly. However, in an instance that's been there for long before upgrading to 12.2X, might not be maintained as clearly as intended.

The key area where the ADOP might not make sense is the custom objects migration. While Oracle extends the ability to do online patching even for custom objects in 12.2.X, these objects need to be packaged per Oracle standards to be able to do so along with the underlying customer schema in database and the custom directories in application editioned as per the guidelines. For instance, one of the ADOP requirement is that the PL/SQL codes be only in APPS schema and the tables be in its respective schema, however, there are several organizations, who might have adopted a approach of having all the custom code and tables in the custom schema or everything in APPS schema. This means that there is huge clean-up job for the implementation team or should compromise on the online patching strategy. Also, in reality, though the company's IT guidelines suggest rules for maintaining custom codes, in due course of time it usually is neglected either due to impending need or just oversight.

Most custom migration are few code changes, unless it is a major redesign, and it does not make sense to go through the rigorous six to eight step process to move them every time. Also, these custom codes migrations, usually do not need any down time can be planned to be migrated during off business hours without affecting the other areas makes it the conventional process more lucrative than the online patching.

Though Oracle has provided online patching ability, it has not removed the conventional patching option (yet!), which means the DBA team can opt to either patch using ADOP or the old way. Traditionally, IT teams have always factored in the application downtime in their operations, which is a good practice by itself, since it provides an opportunity to do tune and bring the systems back-up. This window in most cases is good enough to migrate the patches and any custom changes along with it and the bring the instance back in time. There are certain rase cases where the applications need to be down for longer than usual, which can be handled on exception basis.

Conclusion

Oracle Applications R12.2X, unlike prior version which had major functional changes, brought in Online patching concept, which is one the biggest technical change that Oracle has introduced in a long time. It is evident that this capability reduces the time taken for the patching activity and downtime needed. While this works great for large Oracle patches, which might need hours of downtime, it is debatable on how useful it is for relatively smaller patches and custom objects. With the level of preparation needed and the stringent prerequisites needed, it might not be a best fit for patches or changes needed a small downtime. Also with the limitation on what objects can get editioned, it might affect what patches can be editioned and what cannot. As this is a new concept introduced by Oracle, we can only hope that Oracles builds up on this to close these limitations and ease the criteria to do online patching.

Reference

[1]. Preparing for Online Patching, https://docs.oracle.com/cd/E26401_01/doc.122/e22961/T302934T608605.htm#:~:text=Introduction,utility%20before%20enabling%20online%20patching.,

[2]. Patching Procedures, https://docs.oracle.com/cd/E26401_01/doc.122/e22954/T202991T531065.htm#:~:text=Note:%20The%20adop%20utility%20invokes,revert%20to%20the%20previous%20edition.&text=If%20you%20do%20not%20want,command%20from%20a%20different%20shell.&text=This%20will%20promote%20the%20patch,Guidelines%20For%20Online%20Patching%20Cutover.

[3]. Understanding R12.2: How Editions Work on the Database, https://www.itconvergence.com/blog/understanding-r12-2-how-editions-work-database-objects/, Jul-2016

[4]. E-Business Suite – ADOP Basic Usage Training Videos (Doc ID 2103131.1)

[5]. Oracle Corporation, Understanding R12.2: How Editions Work on the Database - An Oracle Database capability for online application upgrades, https://www.oracle.com/a/tech/docs/ebr-technical-deep-dive-overview.pdf

[6]. Alfredo Abate, Patching in E-Business Suite 12.2: Edition Based Redefinition (EBR), https://www.oatug.org/blogs/alfredo-abate/2019/05/02/patching-in-ebs-122-edition-based-redefinition, May-2019

[7]. Oracle E-Business Suite 12.2 Patching Technology Components Guide (Doc ID 1355068.1), Feb-2019

[7]. Vivek Tiwari, Planning Tips and Best Practices for Upgrading to EBS 12.2, https://www.datavail.com/blog/planning-tips-and-best-practices-for-upgrading-to-ebs-12-2/, Jan-2017

[8]. Himanshu, Understanding Editioning in Oracle EBS R12.2, https://www.funoracleapps.com/2021/06/understanding-editioning-in-oracle-ebs.html, Jun-2021

[9]. Developing and Deploying Customizations in Oracle E-Business Suite Release 12.2 (Doc ID 1577661.1), Nov-2021