Česká zemědělská univerzita v Praze

Technická fakulta

Porovnání možností zálohování databáze Oracle 12c Standard Edition a výběr vhodného řešení

Bakalářská práce

Vedoucí práce: Ing. Tomáš Vokoun
Autor práce: Klára Bělinová

Praha 2018

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Technická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Klára Bělinová

Informační a řídicí technika v agropotravinářském komplexu

Název práce

Porovnání možností zálohování databáze Oracle 12c Standard Edition a výběr vhodného řešení

Název anglicky

Comparison of Database backup options and suitable solution choice for Oracle 12c Standard Edition

---

Cíle práce

Hlavním cílem je z dostupných zdrojů seznámit s metodami zálohování databáze a následně zjistit která z těchto metod je nejefektivnější z pohledu vysoké dostupnosti v případě výpadku v ORACLE.
Dílčím cílem je implementace vysoké dostupnosti v ORACLE pomocí vybrané metody zálohování databáze.

Metodika

Z pohledu vysoké dostupnosti v případě výpadku v ORACLE popsat vybrané metody zálohování databáze (EXPDP, RMAN, STBY, EXP, FLASHBACK DATABASE), vysvětlit jak jednotlivé metody fungují, jejich silné a slabé stránky, jejich porovnání mezi sebou a následně vhodnou výběrovou metodou rozhodnout která zálohovací metoda je nejvíce vyhovující z pohledu vysoké dostupnosti při výpadku. Při studiu odborných zdrojů budou použity metody analytické a syntetické.

Doporučený rozsah práce

40

Klíčová slova

Zálohování databáze, STBY, Oracle, EXPORT, IMPORT

Doporučené zdroje informací

Expert Consolidation in Oracle Database 12c
Oracle Backup and Recovery: All about Oracle Backup and Recovery
Oracle Database 12c Backup and Recovery Survival Guide
Oracle Database 12c Oracle RMAN Backup & Recovery by Robert Freeman
Oracle Data Guard: Standby Database Failover Handbook
1001 tipů a triků pro SQL

Předběžný termín obhajoby

2017/18 LS – TF

Vedoucí práce

Ing. Tomáš Vokoun

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 30. 11. 2017

Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 30. 1. 2018

prof. Ing. Vladimír Jurča, CSc.

Děkan

V Praze dne 10. 03. 2018

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Porovnání možností zálohování databáze Oracle 12c Standard Edition a výběr vhodného řešení" jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autorka uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.03.2018 _____

Poděkování

Ráda bych touto cestou poděkovala vedoucímu mé bakalářské práce Ing. Tomáši Vokounovi, za věcné připomínky, užitečné rady, konzultace, které mi během psaní práce pomohli. A samozřejmě děkuji i své rodině a příteli, kteří mě během tohoto období podporovali.

Abstrakt: Většina organizací v dnešní době potřebuje uchovávat velké množství dat, která jsou nenahraditelná. Ztráta těchto dat, může mít fatální následky pro organizaci. Proto je důležité aplikovat opatření, které zminimalizují ztrátu dat při výpadku. Každá organizace si opatření přizpůsobí svým potřebám. Nejpoužívanějším je zálohování těchto dat, které zajistí jejich minimální, případně žádnou ztrátu. V první části práce bude uveden popis vybraných metod zálohování databází Oracle. A následně se tyto varianty porovnají mezi sebou, aby se vybrala nejefektivnější metoda při možném výpadku. V poslední části bude vybraná metoda implementována a budou s ní prováděny testy, jak rychle bude uveden provoz do normálu a případně o kolik dat organizace může přijít.

Klíčová slova: Zálohování databáze, STBY, Oracle, EXPORT, IMPORT

Comparison of Database backup options and suitable solution choice for Oracle 12c Standard Edition

Abstract: Most organizations nowadays need to store a large amount of data that is irreplaceable. Loss of this data may have fatal consequences. Therefore, it is important to apply measures that minimize data loss in the event of failure. Each organization will adapt the measures to their needs. The most used method is backing up these data to ensure minimum or no loss of data. The first part of the thesis will describe the selected Oracle database backup methods. And then these variants are compared to each other in order to choose the most efficient method of backing up data in a possible of failure. In the last part, the selected method will be implemented and tested how quickly system can be returned back in normal and how many data may organization lose.

Keywords: Backup database, STBY, Oracle, EXPORT, IMPORT

Obsah

1 Úvod

Dnes většina organizací potřebuje udržovat velké množství dat a tato data jsou zpravidla pro příslušnou organizaci nenahraditelná. Může se jednat o snadno doplnitelná data, například data o zaměstnancích organizace nebo o nabízených službách. Ale také se jedná o nenahraditelná data pro podnik, kdy při jejich ztrátě, může být způsobena například finanční újma, kterou organizace nemůže dopustit. Vzhledem k významu těchto dat, která jsou dnes i jednou z mocných zbraní v konkurenčním prostředí, je nezbytně nutné tato data uchovávat takovým způsobem, aby za žádných okolností nedošlo k jejich nenávratné ztrátě. A když by mělo k takové ztrátě dat dojít, tak jen v případě, že pro organizaci by v konečném důsledku bylo ekonomicky výhodnější než zálohu uchovávat či ztracená data doplnit jiným způsobem.

Proto je pro organizace velmi důležité, aby implementovaly určité databázové řešení, které je nejlepším způsob pro uchovávání dat, tak aby byla minimální, ne-li žádná ztráta dat při možném výpadku. Pro tuto implementaci je potřeba nejlépe zaměstnat databázového administrátora, aby pak měl za úkol celkovou správu databáze i uživatelů a jeho nejdůležitější úloha byla bezesporu právě zálohování a udržování databáze v žádoucím stavu.

Na výběr tématu mé bakalářské práce měl vliv můj profesní obor, který se zabývá programováním v PL/SQL na databázích Oracle. Proto provažuji za přínosné popsat vybrané metody zálohování a obnovy prostředí databáze Oracle, prakticky otestovat mnou vybranou metodu zálohování a následně nasimulovat výpadky produkčních systémů, pro ověření integrity dat.

V první části práce je uveden popis vybraných metod zálohování a obnovení databází Oracle, tak jak je uvádí literatura a dokumentace. Následující část je věnována porovnání těchto metod mezi sebou z pohledu vysoké dostupnosti při výpadku. V poslední části bude uvedena implementace metody vyplývající z porovnání jako nejvýhodnější a její testy při výpadku produkční databáze, mající za úkol zmapovat, jak rychle bude možné obnovit provoz a o kolik dat se při užití vybrané metody nenávratně přijde.

2 Cíl práce

Bakalářská práce se bude zabývat problematikou zálohování databází Oracle, z pohledu vysoké dostupnosti při výpadku. Hlavním cílem bude z dostupných zdrojů popsat vybrané metody zálohování databáze Oracle a z tohoto seznámení udělat porovnání metod mezi sebou na úrovni vysoké dostupnosti při výpadku, které zahrnují jak zálohování databáze, tak i její obnovení, a to s minimální ztrátou dat. A následně z nich vybrat tu nejefektivnější.

Dílčím cílem bude tuto vybranou metodu implementovat a provést potřebné simulační testy, pro potvrzení správnosti výběru a to z hlediska vysoké dostupnosti při výpadku produkční databáze.

3 Metodika práce

Metodika bakalářské práce bude založena na studii odborných informačních zdrojů a pro tuto studii bude použita analytická a syntetická metoda. Na začátku práce bude vysvětlen pojem zálohování databází Oracle a k čemu slouží.

Tato práce bude rozdělena na teoretickou a praktickou část. Kde v teoretické části budou jednotlivě popsány vybrané metody zálohování Oracle a následně pomocí porovnáním vybrána nejefektivnější metoda pro zálohování databází Oracle.

Praktická část bude přímo zaměřena na vlastní implementaci vybrané metody. Následně po dokončení implementace, budou provedeny simulační testy výpadků produkčních systému, aby se potvrdila správnost vybrané metody. Tyto výsledky budou vyhodnoceny v této části, jako nová kapitola pro zhodnocení výslednů.

4 Úvod do zálohování

4.1 Definice zálohování

V oblasti databází se zálohování nebo proces zálohování týká kopírování a archivace dat, takže může být použito k obnově originálu v případě ztráty, poškození nebo jiné potřeby práce s daty uloženými v minulosti. Tato kopie může obsahovat důležité části databáze, například řídicí soubor a datové soubory. (2)

Zálohy mají dva odlišné účely:

- Primárním účelem je obnovení dat po jejich ztrátě, ať již při vymazání nebo poškození dat.
- Sekundárním účelem záloh je obnovení dat z dřívější doby podle uživatelsky definovaných pravidel uchovávání dat, která jsou obvykle konfigurována v rámci zálohování a určují, jak dlouho je požadována kopie dat.(7)

Zálohy jsou rozděleny do fyzických záloh a logických záloh. Fyzické zálohy jsou kopie fyzických souborů používaných při ukládání a obnově databáze. Tyto soubory zahrnují datové soubory, řídicí soubory a archivované REDO logy. Nakonec je každá fyzická záloha kopií souborů, které ukládají informace o databázi do jiného umístění, ať již na disku nebo na paměťové médium offline, jako je například páska. Logické zálohy obsahují logická data, jako jsou tabulky a uložené procedury. Může se použít ORACLE Data Pump pro export logických dat do binárních souborů, které se můžou později importovat do databáze. K doplnění fyzických záloh lze použít logické zálohy.(4)

Každý správce ORACLE databází má hlavní úkol navrhnout, implementovat a spravovat strategii zálohování a obnovy. Obecně řečeno, cílem strategie zálohování a obnovy je ochrana databáze před ztrátou dat a rekonstrukce databází po ztrátě dat.

Obvyklé úlohy správy zálohování zahrnují následující:

1. Plánování a testování reakcí na různé druhy poruch
2. Konfigurace prostředí databáze pro zálohování a obnovu
3. Nastavení plánu zálohování
4. Sledování prostředí pro zálohování a obnovu
5. Odstraňování potíží se zálohováním
6. Obnova dat ze ztráty dat v případě potřeby

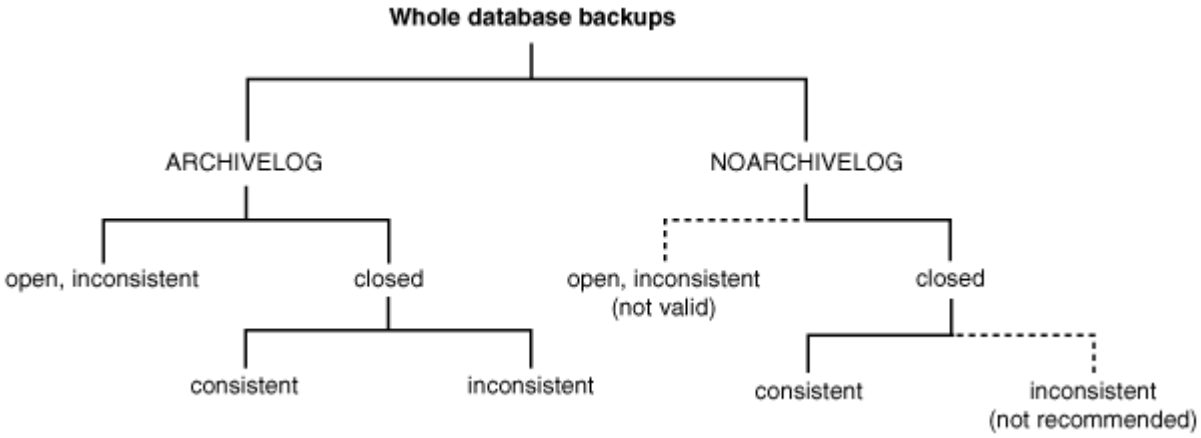
Správce ORACLE databází může také plnit další povinnosti týkající se zálohování a obnovení (4):

1. Ochrana dat, která zahrnuje vytvoření kopie databáze pro dlouhodobé ukládání
2. Přenos dat, který zahrnuje přesun dat z jedné databáze nebo jednoho hostitele do druhého.

4.2 Celá a částečná záloha

4.2.1 Zálohování celé databáze

Při provádění celé zálohy dojde k zálohování databáze jako celku. To znamená, že se uloží veškeré datové soubory a řídící soubor. Tento způsob je nejběžnější typ zálohy. Celá záloha databáze může být provedena buď v ARCHIVELOG (archivní) nebo v NOARCHIVELOG (nearchivní) modu. Před provedením celé zálohy databáze musí být rozmyšleny důsledky zálohování ARCHIVELOG a NOARCHIVELOG modu.(5)(1)



Obrázek 1 Možnosti zálohování celé databáze

4.2.2 Zálohování TABLESPACE

Záloha TABLESPACE je záloha datových souborů, které tvoří tabulkový prostor. Například pokud TABLESPACE USERS obsahuje data files 2, 3 a 4, pak záloha TABLESPACE USERS zálohuje tyto tři datové soubory.

Zálohování TABLESPACE, ať už online nebo offline, je použitelné pouze v případě, že databáze pracuje v ARCHIVELOG modu. Důvodem je, že je nutné obnovení tabulkového prostoru v souladu s ostatními tabulkovými prostory v databázi.(5)

4.2.3 Zálohování datového souboru

Při zálohování datového souboru je zálohován jen jeden datový soubor. Zálohování datového souboru, není tak běžné jako zálohy TABLESPACE, které platí v ARCHIVELOG databázích. Podmínky pro to, aby záloha datového souboru byla použitelná pro databázi v NOARCHIVELOG modu, jsou:

- Každý soubor dat v tabulkovém prostoru je zálohován. Není totiž možné obnovit databázi, pokud nejsou zálohovány všechny datové soubory.
- Datové soubory jsou pouze pro čtení nebo offline-normální.(5)

4.2.4 Zálohování řídicích souborů

Zálohování řídicího souboru je zásadním aspektem zálohování a obnovy. Bez řídicího souboru nelze databázi připojit nebo otevřít.

Může se instruovat RMAN, aby automaticky zálohoval řídicí soubor při každém spuštění úloh zálohování. Protože AUTOBACKUP používá výchozí název souboru, RMAN může tuto zálohu obnovit i v případě, že repozitář RMAN není k dispozici. Proto je tato funkce mimořádně užitečná při scénáři obnovy po havárii.

Ruční zálohování řídicího souboru může být provedeno pomocí následujících metod:

- Příkaz RMAN BACKUP CURRENT CONTROLFILE - provádí binární zálohu řídicího souboru buď jako záložní sada, nebo jako kopie obrazu.
- Příkaz SQL ALTER DATABASE BACKUP CONTROLFILE - vytváří binární zálohu ovládacího souboru.
- Příkaz SQL ALTER DATABASE BACKUP CONTROLFILE TO TRACE - exportuje obsah kontrolního souboru do souboru skriptu SQL. Skript lze použít k vytvoření nového řídicího souboru. Zálohování souborů trasováním má jednu hlavní nevýhodu, a to, že tyto zálohy neobsahují žádné záznamy archivních logů, záloh a kopií RMAN. Z tohoto důvodu jsou preferovány binární zálohy.(5)(1)

4.2.5 Archivované zálohy protokolu REDO

Archivované logy jsou nezbytné pro obnovení nekonzistentní zálohy. Pokud se obnoví záloha pomocí posledního protokolu, musí být k dispozici každý protokol vytvořený mezi těmito dvěma body. Jinými slovy nelze obnovit protokol 100 do protokolu

200, pokud chybí protokol 173. Pokud chybí protokol 173, je možné zastavit obnovu v protokolu 172 a otevřít databázi s touto RESETLOGS volbou.

Vzhledem k tomu, že archivované logy jsou nezbytné pro obnovu, měly by se pravidelně zálohovat. Pokud je to možné, pak je pravidelně zálohovat na pásky.(5)

4.3 Konzistentní a nekonzistentní zálohy

Konzistentní záloha je taková záloha, ve které jsou všechny soubory zálohovány a obsahují všechny změny až na stejné číslo změny systému (SCN). To znamená, že soubory v záloze obsahují všechna data odebraná ze stejného okamžiku. Na rozdíl od nekonzistentní zálohy nevyžaduje obnovovací proces, protože její data jsou v pořádku a nepotřebují opravit.

Nekonzistentní zálohování je zálohování jednoho nebo více databázových souborů, které se uskuteční během otevření databáze nebo po abnormálním vypnutí databáze. (12)

4.3.1 Přehled konzistentních záloh

Konzistentní zálohování databáze nebo části databáze je záloha, ve které jsou všechny datové soubory čtení, zápisu a kontrolní soubory kontrolovány stejným SCN.

Jediný způsob, jak vytvořit konzistentní celou zálohu databáze je vypnout databáze s NORMAL, IMMEDIATE či TRANSACTIONAL volbou a provést zálohu, zatímco databáze je uzavřena. Pokud databáze není čistě vypnutá, například instance selže nebo je vydán příkaz SHUTDOWN ABORT (=přeruší všechny relace, příkazy SQL klientů jsou ukončeny a také samotní klienti odpojeni a ukončí procesy na pozadí.), datové soubory databáze jsou vždy nekonzistentní - pokud databáze není pouze pro čtení.

ORACLE zpřístupňuje řídicí soubory a datové soubory stejnému SCN během kontrolního bodu databáze. Jediné tabulkové prostory v konzistentní záloze, které mají starší SCN, jsou pouze normální tabulkové prostory pouze pro čtení a offline, které jsou stále v souladu s ostatními soubory dat v záloze, protože nebyly provedeny žádné změny.

Důležitým bodem je, že se může obnovit databázi po obnovení konzistentní celé databáze zálohování bez nutnosti obnovy, protože data jsou již konzistentní: není zapotřebí žádná akce, aby byla data v obnovených datových souborech správná. Proto se může obnovit roční synchronizovanou zálohu databáze bez provedení obnovy médií a obnovení instance ORACLE. Samozřejmě, když se obnoví konzistentní záloha celé databáze bez

použití REDO logů, ztratí se všechny transakce, které byly provedeny od doby, kdy byla záloha provedena.

Konzistentní záloha celé databáze je jedinou platnou možností zálohování pro databáze pracující v NOARCHIVELOG modu, protože jinak je nutné obnovení pro konzistenci. V NOARCHIVELOG modu ORACLE archivuje REDO logy a tak nemusí být na disku uloženy požadované REDO logy. Důsledná celá záloha je také platnou možností zálohování databází pracujících v ARCHIVELOG modu. Po obnovení tohoto typu zálohování a archivaci protokolů, je možnost buď okamžitě otevřít databázi a ztratit transakce, které byly provedeny po provedení zálohy, nebo použití archivovaných protokolů k obnovení těchto transakcí. (12)

4.3.2 Přehled nekonzistentních záloh

Nekonzistentní záloha je záloha, ve které jsou soubory, které jsou zálohovány, neobsahují všechny změny provedené u všech SCN. Jinými slovy chybí některé změny. To znamená, že soubory v záloze obsahují data pořízená z různých časových okamžiků. K tomu může dojít, protože datafiles jsou modifikovány jako zálohy. Obnova systému ORACLE činí nekonzistentní zálohování konzistentní čtením všech archivovaných a online REDO logů, počínaje nejstarším SCN v libovolné hlavičce datového souboru a aplikací změn z protokolů zpět do datových souborů.

Pokud databáze musí být v provozu 24 hodin denně, sedm dní v týdnu, pak není jiná možnost, než provést nekonzistentní zálohy celé databáze. Zálohování datových souborů online se nazývá online zálohování. To vyžaduje, aby byla spuštěna databáze v ARCHIVELOG modu.

Pokud se spustí databáze v ARCHIVELOG modu, není třeba zálohovat celou databázi najednou. Například pokud databáze obsahuje sedm tabulkových prostorů a pokud se zálohuje kontrolní soubor i jiný tabulkový prostor každý večer, pak se v týdnu zálohují všechny tabulkové prostory v databázi i řídicí soubor. Tato rozložená záloha se může považovat za zálohu celé databáze. Pokud však musí být obnovena taková rozložená záloha, je třeba obnovit pomocí všech archivovaných REDO logů, které byly vytvořeny od doby, kdy byla provedena nejdříve záloha. (5)(12)

4.4 Online a offline záloha

Hlavním rozdílem mezi Online a Offline zálohou je, zda je databáze přístupná koncovým uživatelům. (12)

4.4.1 Online záloha

Online záloha nebo také HOT záloha se provádí, když je instance databáze otevřená. Uživatelé tak mají přístup k datům i ve chvíli, kdy se provádí zálohování a takto prováděná záloha je vždy nekonzistentní. Obnovováním je tedy databáze převedena do konzistentního stavu pomocí archivních a online REDO logů. Z tohoto důvodu není možné používat nekonzistentní zálohy na databázích běžící v NOARCHIVELOG modu. (12)

4.4.2 Offline záloha

Offline, neboli COLD zálohy, se provádí, když je instance databáze ve stavu SHUTDOWN. Díky tomu je záloha vždy konzistentní. Pokud se provádí pomocí nástroje RMAN, musí být databáze otevřená ve stavu MOUNT. (12)

4.5 Důležité soubory

- data files – vlastní data databáze
- control file – soubor s globálním nastavením databáze; obsahuje seznam datových souborů, archivovaných logů,…
- log files
- archived logs – archivované REDO logy
- current logs – REDO logy aktuálně používané databází
- server parameter file (SPFILE) – soubor s nastavením parametrů databáze (ALTER DATABASE SET …)(12)

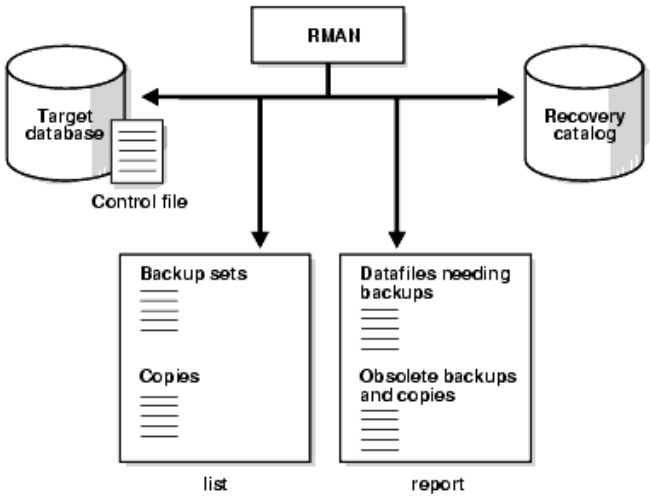
5 Teoretická východiska

5.1 RMAN

Aplikace Recovery Manager (RMAN) je klient ORACLE Database, který provádí zálohování a obnovu úloh v databázích a automatizuje správu strategií zálohování. Velmi zjednodušuje zálohování, obnovu a obnovu databázových souborů.

Prostředí RMAN se skládá z nástrojů a databází, které hrají roli při zálohování dat. Minimální prostředí pro RMAN musí obsahovat následující součásti (3):

- Cílové databáze
  - Databáze ORACLE, ke které je RMAN spojeno s TARGET klíčovým slovem. Cílova databáze je databáze, na které RMAN provádí operace zálohování a obnovy. RMAN vždy udržuje METADATA o svých operacích v databázi v řídicím souboru databáze. METADATA RMAN jsou známé jako METADATA RMAN úložiště.
- RMAN klient
  - Spouštěcí soubor ORACLE Database, který interpretuje příkazy, řídí relace serveru k provádění těchto příkazů a zaznamenává jeho činnost do řídicího souboru cílové databáze. Spouštěcí soubor RMAN je automaticky nainstalován s databází a je obvykle umístěn ve stejném adresáři jako ostatní spustitelné soubory databáze. Například je umístěn klient RMAN v systému Linux $ORACLE_HOME/bin.



Obrázek 2 Seznamy a zprávy RMAN

RMAN zálohuje data na konfigurované výchozí zařízení pro požadovaný typ zálohy. RMAN ve výchozím nastavení vytváří zálohy na disku. Je-li povolena oblast rychlé obnovy, a pokud se nezadá FORMAT parametr, RMAN vytvoří zálohy v oblasti obnovy a automaticky jim dává jedinečné názvy.(3)

Pokud se zadá BACKUP AS COPY, pak RMAN zkopíruje každý soubor jako image copy, což je bit-for-bit kopie databázového souboru vytvořeného na disku. Kopie image jsou shodné s kopiemi vytvořenými pomocí příkazů operačního systému. Pomocí programu RMAN je možné vytvářet kopie snímku v době, kdy je databáze otevřená.(11)

5.1.1 Vytváření přírůstkových záloh

Pokud se zadá BACKUP INCREMENTAL, pak RMAN vytvoří přírůstkové zálohování databáze. Přírůstkové zálohy zachycují na úrovni bloku změny na úrovni databáze provedené po předchozí přírůstkové záloze. Přírůstkové zálohy jsou obecně menší a rychlejší než zálohování databází. Obnova s přírůstkovými zálohami je rychlejší než samotné použití REDO logů.

Výchozím bodem pro přírůstkovou strategii zálohování je úroveň 0 přírůstkové zálohy, která zálohuje všechny bloky v databázi. Přírůstková záloha na úrovni 0 je v obsahu shodná s hodnotou úplné zálohy, na rozdíl od úplné zálohy je však záloha úrovně 0 považována za součást přírůstkové strategie zálohování.

Přírůstková záloha úrovně 1 obsahuje pouze bloky změněné po předchozí přírůstkové záloze. Neexistuje-li záloha databáze úrovně 0, pak jí RMAN automaticky vytvoří.

Záloha úrovně 1 může být kumulativní přírůstková záloha, která zahrnuje všechny bloky změněné od poslední zálohy úrovně 0, nebo diferenční přírůstková záloha, která zahrnuje pouze bloky změněné od poslední přírůstkové zálohy. Přírůstkové zálohy jsou ve výchozím nastavení rozdílné.

Během obnovování program RMAN nejprve obnoví zálohu úrovně 0, poté automaticky provede přírůstkové zálohování a podle potřeby obnoví protokoly. Tím se opětovně použijí změny provedené v databázi od zahájení zálohování.(17)

5.1.2 Aktualizace záloh

RMAN postupně aktualizovaná funkce zálohování je efektivní strategie přírůstkové zálohování. Strategie má tyto hlavní rysy:

- Strategie vyžaduje kopii datového souboru na úrovni 0 jako základ. Tato kopie má buď systémově definovanou, nebo uživatelem definovanou značku.
- Periodicky jsou vytvořeny diferenční zálohy úrovně 1 se stejnou značkou jako kopie datového souboru úrovně 0. BACKUP FOR RECOVER OF COPY příkaz určuje, že přírůstkové zálohy obsahují pouze bloky, co se změnily od poslední přírůstkové zálohy se stejnou značkou.
- Pravidelně jsou přírůstkové zálohy aplikovány na kopii datového souboru úrovně 0. Vzhledem k tomu, že kopie datového souboru byla aktualizována novějšími změnami, vyžaduje nyní méně obnovy médií. (17)

5.1.3 Ověření databázových souborů a záloh

Příkaz VALIDATE se používá ke zjištění, že všechny databázové soubory existují, jsou v správném umístění a nejsou fyzicky poškozeny. CHECK LOGICAL kontroluje poškozené logické bloky.(3)

5.2 STANDBY

STANDBY databáze je velmi konzistentní kopie produkční databáze. Vytváří kopii produkční databáze, může vytvořit až třicet STANDBY databází a zahrnout je do konfigurace Oracle Data Guard. Po vytvoření aplikace Oracle Data Guard automaticky zálohuje archivní logy z produkční databáze a následné je obnoví do STANDBY databáze.

Podobně jako u produkční databáze může být STANDBY databáze buď databáze Oracle o jediné instanci, nebo Databáze Oracle RAC.

5.2.1 Typy databází STANDBY režimu

Fyzická STANDBY databáze

Poskytuje fyzicky identickou kopii produkční databáze, s identickou strukturou na disku jako má produkční databází. Schéma databáze, včetně indexů, je stejné. Fyzická STANDBY databáze je synchronizována s produkční databází prostřednictvím REDO logů, které jsou převzaty z produkční databáze a obnoveny do fyzické STANDBY databáze.(6)

Od verze ORACLE Database 11 g Release 1 (11.1) může fyzická STANDBY databáze zálohovat a obnovovat REDO, zatímco je otevřená pro přístup pouze pro čtení.

Fyzická STANDBY databáze může být proto použita současně pro ochranu dat a reportovací účely.

Kromě toho, od data vydání ORACLE Database 11 g Release 2 (11.2.0.1) lze fyzickou STANDBY databázi použít pro instalaci vhodných jednorázových oprav, nastavení patch updates sady aktualizací (PSU) a kritických patch updates (CPU) dříve než bude nasazeno na produkční databázi. (9)

Logická STANDBY databáze

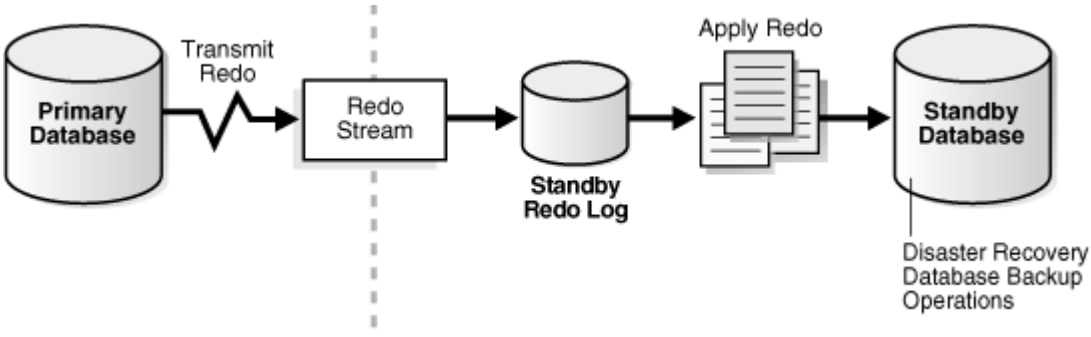
Obsahuje stejné logické informace jako produkční databáze, i když fyzická databáze a struktura dat může být odlišná. Logická STANDBY databáze je synchronizována s produkční databází pomocí aplikace SQL, která převádí obdržené REDO logy z produkční databáze do SQL příkazu a pouští je na STANDBY databázi.

Flexibilita logické STANDBY databáze umožňuje upgradovat software ORACLE Database (sady patchů a nové verze ORACLE Database) a provádět další údržbu databáze v rolovacím módu s téměř žádným zdržením.(14)

Snímek STANDBY databáze

Stejně jako fyzická nebo logická STANDBY databáze snímek STANDBY databáze přijímá a archivuje REDO data z produkční databáze. REDO logy obnovené do snímku STANDBY databází se nepoužijí, dokud REDO data ze snímku STANDBY databáze nejsou aplikována dřív než je databáze převedena opět do fyzické STANDBY databáze, to tedy znamená, že se úměrně zvyšuje čas obnovy databáze o čas, který se stráví aplikováním REDO logů.

Snímek STANDBY databáze se nejlépe používá ve scénářích, které vyžadují dočasný, aktualizovatelný snímek fyzické STANDBY databáze. Může například použít volbu ORACLE Real Application Testing pro uvolnění pracovního vytížení produkční databáze a následně ji přehrát pro testovací účely do snímku STANDBY databáze. Protože REDO obnovené do snímku STANDBY databází se nepoužijí, dokud se nepřevedou zpět do fyzického STANDBY režimu, čas potřebný k obnovení z produkční databáze při selhání je úměrný množství REDO dat.(14)



Obrázek 3 Typická konfigurace Oracle Data Guard

5.2.2 Oracle Data Guard

Oracle Data Guard zajišťuje vysokou dostupnost, ochranu dat a obnovu po havárii pro podniková data. Oracle Data Guard poskytuje komplexní nastavení služeb, které vytvářejí, udržují, spravují a monitorují jednu nebo více STANBY databází, které umožňují produkční databázi přežít havárii a poškození dat. Oracle Data Guard udržuje tyto záložní databáze jako kopie produkční databáze. Pokud se produkční databáze stává nedostupná kvůli plánovanému nebo neplánovanému výpadku, může Oracle Data Guard přepnout libovolnou STANDBY databázi na produkční roli a minimalizovat čas výpadku. Oracle Data Guard lze používat s tradičními technikami zálohování, obnovy a clusterů, aby byla zajištěna vysoká úroveň ochrany dat a dostupnost dat. Služby Oracle Data Guard využívají také další funkce ORACLE, jako jsou Oracle Streams a Oracle GoldenGate, pro efektivní a spolehlivý přenos REDO logů z produkční databáze do jednoho nebo více vzdálených hostitelů.(6)

S Oracle Data Guard mohou administrátoři volitelně zlepšit výkony v produkční databázi tím, že uvolní zálohovací a reportovací operace náročné na CPU do STANDBY systémů.(10)

5.2.3 Služby Oracle Data Guard

Redo Transport Services

Řídí automatizovaný přenos REDO logu z produkční databáze do jednoho nebo více archivních hostitelů.

Aplikovat služby

REDO data jsou aplikována přímo z REDO logů STANDBY databáze, které jsou přesunuty v reálném čase. Pokud nejsou soubory pro REDO logy nakonfigurovány, pak REDO data musí být před použitím nejdříve archivována na STANDBY databáze.

Přechody rolí

Změnit roli databáze ze STANDBY databáze na produkční databázi nebo z produkční databáze na STANDBY databáze pomocí operace přepnutí nebo převzetí služeb při selhání.(6)

5.3 EXPDP

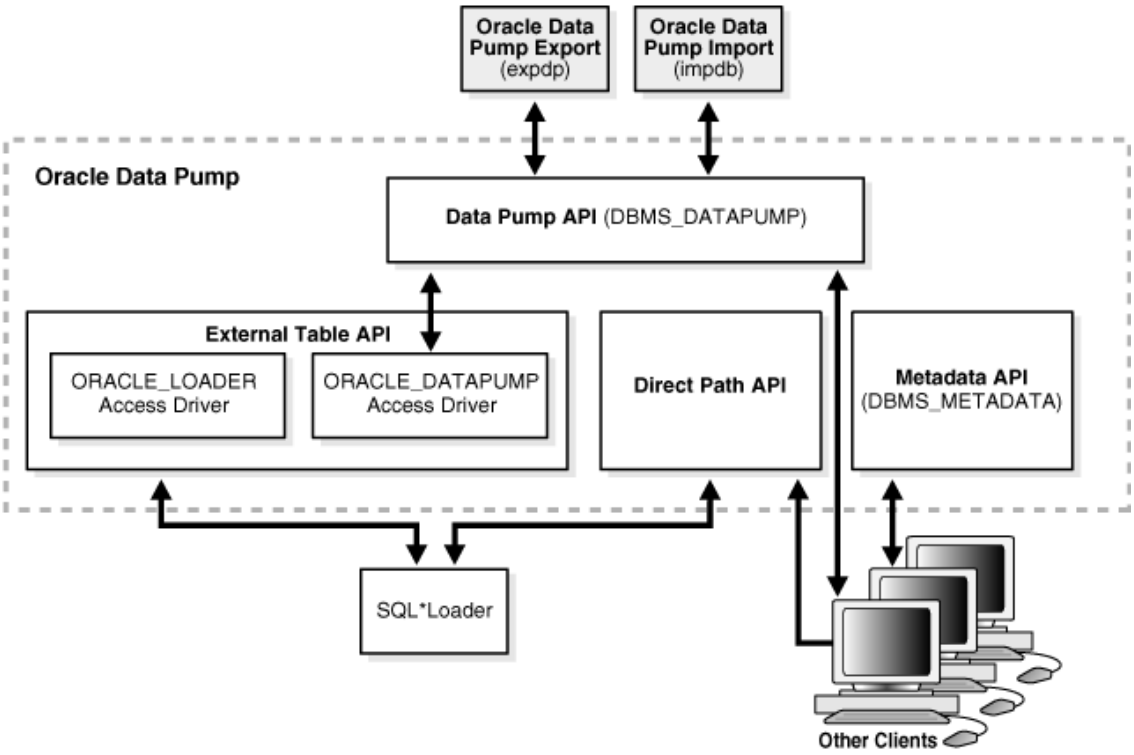
Data Pump Export je nástroj pro vyložení dat a METADAT do souborů operačního systému, který se nazývá dump file set. Dump file set lze importovat pouze pomocí nástroje Data Pump Import. Lze importovat do stejného systému nebo lze přesunout do jiného systému a aplikovat jej.

Dump file set je tvořen jedním nebo více soubory disku, které obsahují data tabulky, METADATA objektů databáze a řídicí informace. Soubory jsou psány ve vlastním binárním formátu. Během operace importu nástroj Data Pump Import používá tyto soubory k vyhledání všech databázových objektů v sadě souborů výpisu.

Protože Dump file sety jsou spíše spuštěny serverem než klientem, musí správce databáze (DBA) vytvořit adresářové objekty, které definují umístění serveru, do kterých jsou soubory zapisovány.

Data Pump Export umožňuje určit, že úloha by měla přesunout podmnožinu dat a METADAT určených režimem exportu. To se provádí pomocí datových filtrů a filtry METADAT, které jsou zadávány pomocí parametrů exportu.

Data Pump Export je spuštěn pomocí EXPDP příkazu. Vlastnosti exportní operace jsou určeny parametry exportu, které specifikujete. Tyto parametry lze zadat buď na příkazovém řádku, nebo v souboru parametrů.(13)



Obrázek 4 Architektura ORACLE Data Pump

Export nabízí různé režimy pro vykládání různých částí databáze. Dostupné režimy:

- Full mode
- Schema mode
- Table mode
- Tablespace mode
- Transportable tablespace mode

5.3.1 Datové filtry

Filtrování dat je implementováno pomocí parametrů QUERY a SAMPLE parametrů, které určují omezení řádků tabulky, které mají být exportovány.

Filtrování dat může také nastat nepřímo kvůli filtrování METADAT, kteáé mohou obsahovat nebo vyloučit objekty tabulky spolu s přidruženými datovými řádky.

Každý datový filtr lze zadat jednou pro tabulku v rámci úlohy. Pokud se pro konkrétní tabulku i pro celou úlohu použijí různé filtry se stejným názvem, má přednost parametr filtru dodaný pro konkrétní tabulku.(13)

5.3.2 Filtry METADAT

Filtrování METADAT je implementováno pomocí parametrů EXCLUDE a INCLUDE parametry. Parametry EXCLUDE a INCLUDE parametry se navzájem vylučují.

Filtry METADAT určují soubor objektů, které mají být zahrnuty nebo vyloučeny z operace exportu nebo importu. Například se může požádat o úplný export, ale bez specifikací balíčků.

Závislé objekty identifikovaného objektu se zpracovávají společně s identifikovaným objektem. Například pokud filtr určuje, že index má být zahrnut do operace, budou zahrnuty také statistiky z tohoto indexu. Stejně tak, pokud je tabulka vyloučena filtrem, filtry vyloučí také indexy, omezení, granty a spouštěče na tabulce.

Pokud je pro typ objektu zadáno více filtrů, použije se na ně implicitní operace AND. To znamená, že objekty vztahující se k úloze musí předat všechny filtry použité na jejich typy objektů.

Stejný název filtru METADAT lze v rámci úlohy zadat vícekrát.(7)

5.4 EXP

Nástroj EXPORT poskytuje jednoduchý způsob přenosu datových objektů mezi databázemi ORACLE, a to i v případě, že se nacházejí na platformách s různými konfiguracemi hardwaru a softwaru.

Exportní soubor je Dump file binárního formátu ORACLE, který je obvykle umístěn na disku nebo pásku. Dump file lze přenést pomocí FTP nebo fyzickým přenesením (v případě pásky) do jiného místa. Soubory lze pak aplikovat s nástrojem IMPORT pro přenos dat mezi databázemi, které se nacházejí v systémech, které nejsou připojeny prostřednictvím sítě. Soubory lze kromě běžných postupů zálohování použít také jako zálohy.

EXPORT Dump file lze číst pouze pomocí nástroje ORACLE IMPORT. Verze nástroje IMPORT nemůže být novější než verze nástroje EXPORT použitého k vytvoření souboru výpisu.(9)

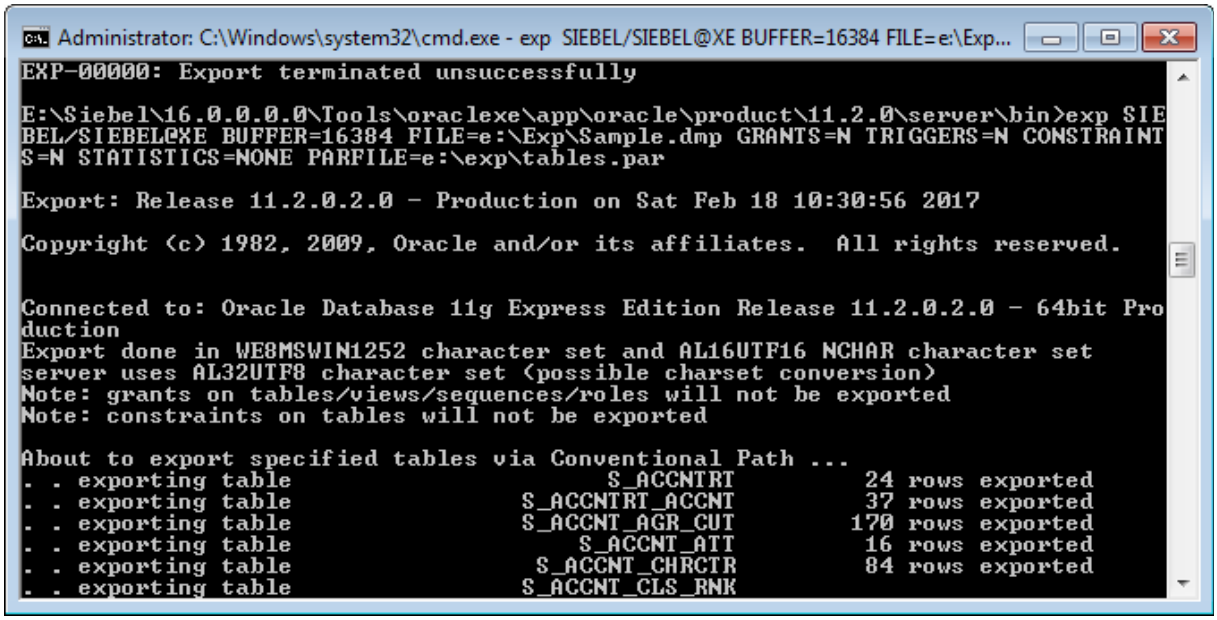
Může také zobrazit obsah exportního souboru bez vlastního importu. Pro zobrazení se používá parametr SHOW.

Zahájení exportu a zadání parametrů se provádí některým z následujících způsobů:

- Položky příkazového řádku
- Soubory parametrů
- Interaktivní režim (15)

5.4.1 Čtyři režimy provozu

1. Full
    - Exportuje úplnou databázi. Tento režim můžou používat pouze uživatele s rolí EXP_FULL_DATABASE. Parametr FULL.
2. Tablespace
    - Umožňuje privilegovanému uživateli přesunout sadu tabulek z jedné databáze ORACLE do jiné. Parametr TRANSPORT_TABLESPACE.
3. User
    - Umožňuje exportovat všechny objekty, které patří uživateli (např. Tabulky, granty, indexy a procedury). Import privilegovaného uživatele v uživatelském režimu může importovat všechny objekty ve schématech určené uživateli. Parametr OWNER.
4. Table
    - Umožňuje exportovat konkrétní tabulky a oddíly. Privilegovaný uživatel může tabulky kvalifikovat zadáním schématu, který je obsahuje. Pro libovolnou tabulku, pro kterou není název schématu zadán, exportujte výchozí název schématu vývozce. Parametr TABLES.(13)



Obrázek 5 Export databáze

5.5 FLASHBACK DATABASE

Databáze FLASHBACK je podobná tradičnímu obnovení v čase (point-in-time). Umožňuje vrátit databázi do stavu v době nedávné minulosti. Zálohování metodou FLASHBACK je mnohem rychlejší než obnovení typu point-in-time, protože nevyžaduje obnovení datových souborů ze zálohy a vyžaduje méně změn z archivovaných REDO logů.(16)

Může se použít FLASHBACK Database k potlačení většiny nechtěných změn v databázi, pokud jsou datové soubory nepoškozené. Může vrátit databázi k předchozímu stavu a vrátit zpět účinky ALTER DATABASE OPEN RESETLOGS příkazu.

Databáze FLASHBACK používá svůj vlastní protokolovací mechanismus, který vytváří snímky a uloží je do oblasti rychlého obnovení. Databáze FLASHBACK lze použít pouze v případě, že jsou k dispozici protokoly zpětného snímku. Pro využití této funkce se musí předem nastavit databáze, aby se vytvořily záznamy zpětného vyhledávání.

Pro aktivaci FLASHBACK Database, se konfiguruje oblast rychlého obnovení a nastaví se cíl pro uchovávání minulosti. Tento cíl určuje, jak daleko se může databáze převést do minulosti.

Od této doby v pravidelných intervalech databáze kopíruje snímky každého pozměněného bloku v každém datovém souboru do protokolu zpětného snímku. Tyto blokové snímky mohou být později znovu použity k rekonstrukci obsahu datového souboru pro libovolný okamžik, kdy byly snímky zachyceny.

Použije-li se FLASHBACK Database k přetočení databáze do minulého času, příkaz určuje, které bloky se po cílovém čase změnily a obnoví je z protokolů zpětného snímku. Databáze obnovuje verzi každého bloku, který je bezprostředně před cílovým časem. Databáze pak používá REDO logy k opětovnému použití změn, které byly provedeny poté, co byly tyto bloky zapsány do protokolu zpětného snímku.

Snímky pro obnovení musí být k dispozici na disku nebo pásce, po celou dobu, po kterou probíhá zálohování. Pokud je například cíl uchovávání snímku 1 týden, musí se zajistit, aby byly online a archivované REDO logy, které obsahují všechny změny za poslední týden. V praxi je obvykle potřeba opakovat snímky mnohem déle, než je cíl pro uchování zpětného snímku, který podporuje obnovu v daném okamžiku. (8)

5.5.1 Omezení databáze FLASHBACK

Protože databáze FLASHBACK funguje tak, že zruší změny datových souborů, které existují v okamžiku spuštění příkazu, má následující omezení:

- Databáze FLASHBACK může pouze vrátit zpět změny datového souboru vytvořeného databází ORACLE. Nemůže být použita k opravě selhání médií nebo k nápravě při náhodném vymazání datových souborů.
- Databáze FLASHBACK nelze použít k vrácení operace zmenšení datového souboru.
- Pro obnovení datového souboru, který jste vynechali, nelze použít samotnou databázi FLASHBACK
- Pokud je databázový kontrolní soubor obnoven ze zálohy nebo znovu vytvořen, veškeré informace o nahromaděném záznamu o zálohování jsou smazány. FLASHBACK databáze se nemůže vrátit k určitému okamžiku před obnovením nebo opětovným vytvořením kontrolního souboru.
- Při použití FLASHBACK databáze s cílovým časem, ve kterém probíhá NOLOGGING operace, je pravděpodobné blokování poškození v databázových objektech a datových souborech ovlivněných NOLOGGING operací.(16)

6 Porovnání uvedených metod a vysvětlení výběru

V této kapitole bude provedeno porovnání vybraných a dříve popsaných metod pro zálohování a také definována vhodná metoda pro zálohování z pohledu vysoké dostupnosti databáze při výpadku. Cílem tohoto srovnání bude vyzdvihnout klady a zápory, které vybrané metody řešení mají při použití na dané úrovni.

6.1 Kritéria pro porovnání metod

Existují různá kritéria pro výběr vhodného zálohovacího systému. Neexistuje žádný jednotný přístup nebo návod. Zpravidla se používá kombinace několika metod, jejichž výběr závisí obvykle na možnostech organizace, jestli je potřeba 24/7 provoz, investice do úložiště, zda systém produkuje data pořád nebo občas, apod. Tudíž každá organizace používá jinou vhodnou metodu pro své systémy na základě vlastních požadavků.

Pro porovnání různých metod zálohování bylo zvoleno kritérium vhodné z pohledu vysoké dostupnosti v případě výpadku produkční databáze. Toto kritérium nebude přímo jako hlavní faktor porovnávání, ale bude sloužit především jako vodítko, které umožní zasadit informace a porovnání možností do kontextu praktického využití. Pro výběr bude použita analyticko-syntetickou metodiku.

6.2 Samotné porovnání metod mezi sebou

V této kapitole se nejprve přistoupí k porovnání metod z pohledu dostupnosti dat. Porovnání se bude zabývat především možností vytvoření zálohy a jejím uložením na konkrétní médium, Dalším bodem bude samotné obnovení dat a to z důvodu, že možnost snadného vytvoření záloh nemusí znamenat i snadné obnovení a pro konečné uživatele je časová náročnost obnovy a tedy dostupnost dat důležitým kritériem pro výběr zálohovací metody. Poslední část kapitoly by byla věnována shrnutí poznatků získaných z předešlého porovnání a výběru vhodného řešení.

Prvním požadavkem, který byl výše nastíněn, je možnost provedení zálohy dat a její uložení na konkrétní média, ať už fyzické nebo cloudové. V dnešní době existuje možnost zálohovat celé virtuální servery pomocí na to uzpůsobeného virtualizačního softwaru. Nicméně tyto metody jsou časově i kapacitně náročné, navíc pro provoz 24/7 s nedostatečnou integritou dat a s následnou možnou nekonzistencí dat uložených v

databázi. Pro podchycení těchto nedostatků disponuje Oracle vlastním zálohovacím systémem RMAN.

Zmíněný RMAN umožňuje zálohování a obnovení databáze na úrovni jednoho datového bloku. Tato funkce je vhodná, pokud jsou například poškozeny pouze některé datové bloky databáze a díky tomu je možnost obnovit pouze poškozený blok bez nutnosti obnovení celé databáze. Také je vhodná z pohledu nižších požadavků na kapacitu úložných médií, což pro organizaci je jeden z podstatných faktorů pro výběr vhodné metody zálohování.

RMAN umožňuje použití více principů zálohování a jejich použití je závislé na požadavcích na RTO, RPO a na HW. Z pohledu vysoké dostupnosti by bylo nejefektivnější využití vytváření kompletních záloh celé DB v krátkých časových intervalech se zapnutým archivním logováním, nicméně to vytváří obrovské nároky na diskovou kapacitu a proto se obvykle volí kompromis v podobě přírůstkových záloh, které v sobě udržují jen změny, které byly provedené po hlavní záloze, která je pak následně k celé obnově také potřeba.

Vlastnosti této zálohovací metody spolu s archivním módem databáze zaručují vysokou míru RPO při výpadku, nicméně míra RTO je naopak vyšší vzhledem k nutnosti obnovy všech dat ze vniklých záloh.

Dále Oracle nabízí možnost online přenášení všech vzniklých změn v databáze na vzdáleného hosta. Tato uvedená metoda klonuje archivní logy produkční databáze na jiné umístění a tam je obnovuje do naklonované databáze z produkce. Takto duplikovaná databáze se nazývá STANDBY a zároveň umožnuje, aby byla spuštěna jen ve čtecím modu a tím mohla být využívána například pro spouštění dotazů (reportů), tak aby se nezatěžovala produkční databáze. U STANDBY je také snaha o minimalizaci ovlivnění výkonu na produkční databázi při replikaci, má přímý přenos archivních logu z produkčního serveru a také zajišťuje velmi nízkou míru selhání, protože všechny součásti systému jsou duplikovány. V této metodě je potřeba kapacita na médiu pouze pro archivní logy, které databáze generuje při nastaveném archivním módu a kapacitu pro duplicitní databázi, jiné nároky na kapacitu nemá. V této metodě je zálohování prováděné online pomocí služby Data Guard, je to proces zálohování společně s obnovou dat. Jakmile je vytvořený nový archivní log, proces ho okamžitě pošle přes službu pomocí funkcionality Data Guard do STANDBY databáze a následně jej naaplikuje, tím je zajištěna vysoká

konzistence dat, jelikož při výpadku produkční databáze jsou archivní logy již naaplikovány.

Další možností je využití Oracle FLASHBACK, ten umožnuje pouze zálohovat celou databází, není zde možnost udělat jakýkoliv přírůstek. V době kdy se spustí zálohování pomocí FLASHBACK se udělá snímek celé databáze ke konkrétnímu času, nicméně to vytváří velké kapacitní nároky na použité médium. Dále zde není možnost vrátit pouze vybrané objekty, ale je potřeba aplikovat celou databázi a při běhu obnovování není po celou dobu databáze k dispozici. Z toho důvodu může být proces obnovení velmi časově náročný. A pokud jsou zálohy uloženy na páskách, může tento proces být ještě delší.

U metody Datová pumpa (EXPDP) se pracuje se skupinou souborů nazývaných dump file sety, všechny jeho úlohy běží na serveru pomocí serverových procesů. Přístup k úložišti, kde jsou dump file sety uchovávány, vynucuje bezpečnostní model, který lze využívat pomocí práv k řízení přístupu k těmto souborům. Data pump také umožňuje přenášet data mezi dvěma databázemi a to pomocí databázových linků, které tvoří most mezi dvěma databázemi a umožňuje jejich komunikaci a odpadává tak nutnost vytváření souborů na disku.

Jako alternativa ještě existuje starší metoda pro dump file sety EXP, ta funguje pouze na jednom souboru a má přístup k souborům jak na server, tak i na klienta (bez použití adresářů ORACLE). Stejně jako novější model dokáže vyexportovat celou databázi do jednoho souboru a následně se z něj databáze obnoví. Nevýhodou jsou, vzhledem k nutnosti vytvářet zálohy celé databáze, velké nároky na kapacitu diskových prostorů.

Obě tyto metody vyexportují databázi pouze v konkrétním stavu v určitém čase a nedochází tak k průběžné synchronizaci. Při spuštění exportu se udělá bod, ke kterému se vytvoří záloha, ale produkční databáze běží dál.

Druhým požadavkem, který byl nastíněn, je obnovení dat. Pokud se mluví o dostupnosti dat z pohledu zálohování, nejedná se pouze o proces zálohování a ukládání dat, ale také o možnost přístupu k těmto datům, pokud by nastala potřeba tyto data obnovit. Každá organizace má samozřejmě jiné nároky na tuto dostupnost a tudíž i hledá rozdílná řešení.

V první řadě Oracle RMAN poskytuje obnovení celé databáze a to jednak z jedné samostatné zálohy celé databáze a jednak ze zálohy rozšířené o zálohované změnové přírůstky. Následné obnovení při výpadku spolu se ztrátou dat z produkční databáze, nastane pomocí RMAN obnovení cele databáze a dotočení k ní přírůstkové zálohy, které byly vytvořeny od poslední velké zálohy. A data, která byly vytvořeny od posledního vytvořeného archivního logu a výpadkem, jsou nadobro ztraceny.

Jak již bylo zmíněno, STANDBY databáze obnovuje již při zálohování dat archivních logů. Při využívání této metody lze takřka vyloučit ztrátu jakýchkoliv dat, protože okamžitě vytvoření archivního logu je i obnoven na vzdálenou databázi. Zbývají pouze nedoplněné REDO logy, které ještě nebyly archivovány, ale pro jejich udržení Oracle umožnuje funkci ukládání na více odlišných umístění a tím omezit jejich ztrátu na minimum.

Existují měrné jednotky RTO (Recovery Time Objective) a RPO (Recovery Point Objective), které ukazují na dostupnost dat při výpadku. Ukazatel RTO vyjadřuje množství času potřebné pro obnovení dat a celého nedostupného systému. A ukazatel RPO vyjadřuje, do jakého bodu v minulosti lze data obnovit. Jinak řečeno, o kolik dat při výpadku organizace přijde.

RMAN RTO oproti STANDBY vyšší, protože RMAN jen zálohuje databázi v nastavený interval a obnovení pak musí proběhnout celé databáze. Kromě případu, že se poškodí pouze jeden datový blok.

Metoda FLASHBACK umožňuje vrátit databázi v čase do nastaveného časového omezení. Má možnost bez problému přistupovat k historickým datům pomocí konstrukcí SQL. Dotazují se na historická data v různých časových úsecích, jak data existovaly v různých časových bodech. Přístup na historická data je pouze ve čtecím modu, už se nedají jakkoliv měnit.

Následná obnova probíhá obnovením kompletního snímku databáze, při výpadku se přijde o data, která byla vytvořena mezi posledním a budoucím snímkem databáze.

EXPDP a EXP pracují na podobném principu jako FLASHBACK, vytvoří zálohu celé databáze do jednoho souboru, jak bylo zmíněno a tento soubor použije při obnovení celé databáze. Tyto metody se používají především pro malé systémy bez větší fluktuace dat, kdy se již ztracená data dají jednoduše znovu doplnit.

Z výše uvedeného porovnání vyplývá, že každá ze zvolených metod má své klady a zápory. Ale vzhledem k tomu, že chceme dosáhnout co nejnižších hodnot RTO a RPO při

simulaci stálého provozu, je z probíraných metod nejefektivnější STANDBY, která udržuje databázi konzistentní v téměř aktuálním čase a pro její přepnutí postačí provést minimum krátkodobých úkonů a obnovení minimálního objemu dat a tím snížit hodnotu RTO.

7 Praktická část práce

Pro implementaci a zrealizování testů nad vybranou metodou zálohování je zapotřebí nasimulovat potřebnou infrastrukturu, která umožní dosáhnout relevantních výsledků, ze kterých se dají získat výsledky měřitelné metrikou RPO a RTO. K tomu se dá využít jakýkoliv virtualizační nástroj s podporou lokální sítě, podporovaný OS a samotný software ORACLE se simulací provozu (generování dat).

Pro tyto potřeby lze využít virtualizační nástroj Oracle VM VirtualBox, který jednak splňuje požadavky pro jednoduchou správu a jednak pro propojení dvou virtualizovaných systémů, jako OS lze použít Windows Server a pro DB se využije nejvyšší stabilní verze ORACLE 12c.

Oba servery budou identicky postavené včetně struktury složek, což nám poslouží pro snadnější správu STANDBY serveru. Jako propojení se využije interní lokální síť.

Na produkčním prostředí bude navíc prováděna simulace provozu každých 30 vteřin.

Parametry obou prostředí:

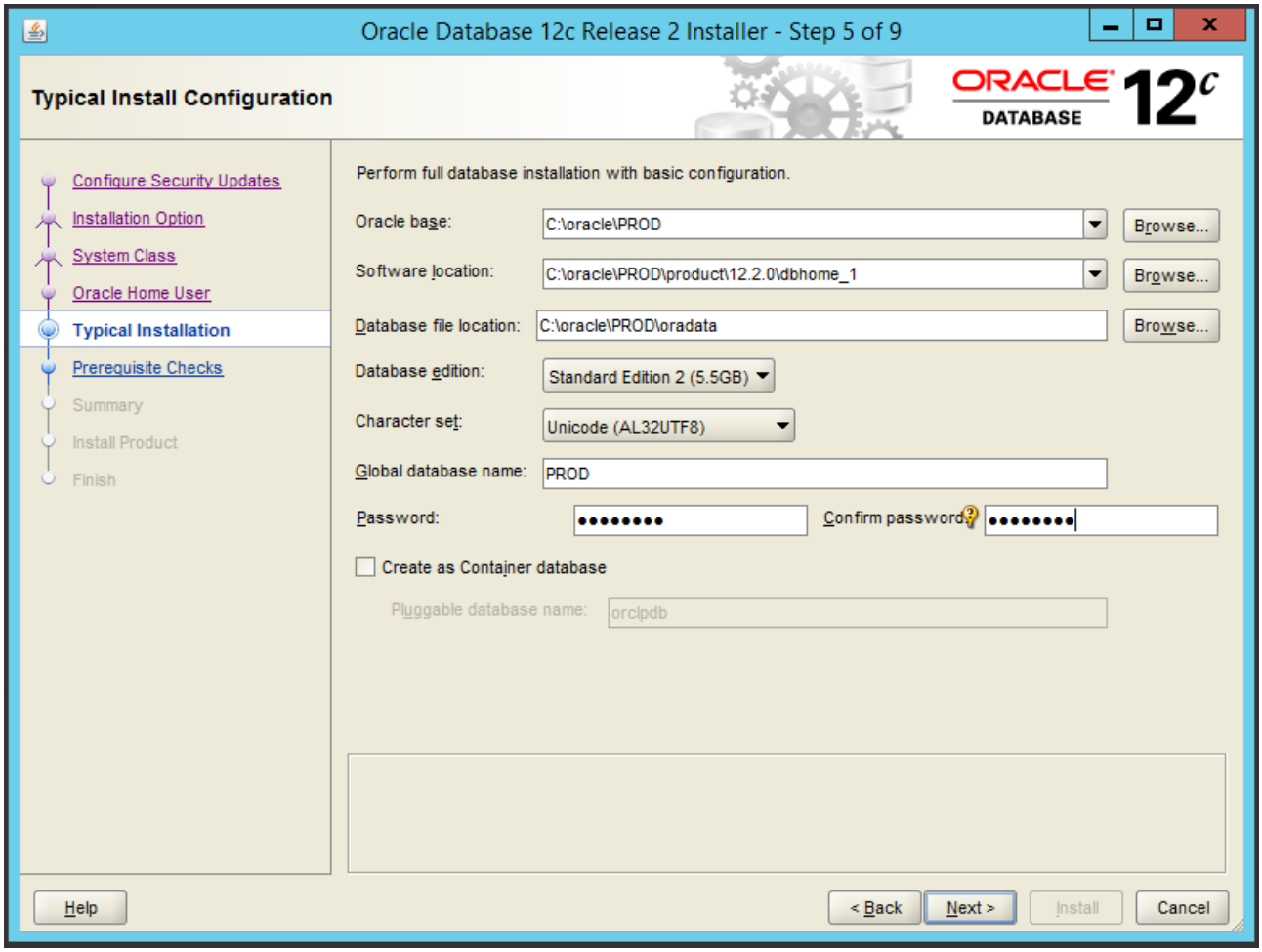
- Virtualizace prostředí pomocí Oracle VM VirtualBox
- Diskový prostor o velikosti 50GB
- 4 GB RAM
- OS Windows Server 2012 64x
- Interní lokální síť
- DB software ORACLE 12c Standard Edition
- Databáze o velikosti 1GB

7.1 Příprava prostředí

Pro přípravu prvního serveru, na kterém poběží produkční prostředí, je třeba nainstalovat základní software ORACLE a následně vytvořit instanci databáze. Tyto dva kroky, se můžou spojit při instalaci pomocí volby v ORACLE instalátoru „Create and configure a database“, která všechny potřebné úkony provede v jednom kroku.

Doporučuje se při instalaci nastavit umístění Oracle base do vhodné struktury, například podle firemních standardů. S touto cestou se nastaví Software location a Database file location. Důležité je vybrat Database edition, to se řídí podle zakoupené licence od společnosti ORACLE. A jako poslední se musí nastavit Global database name

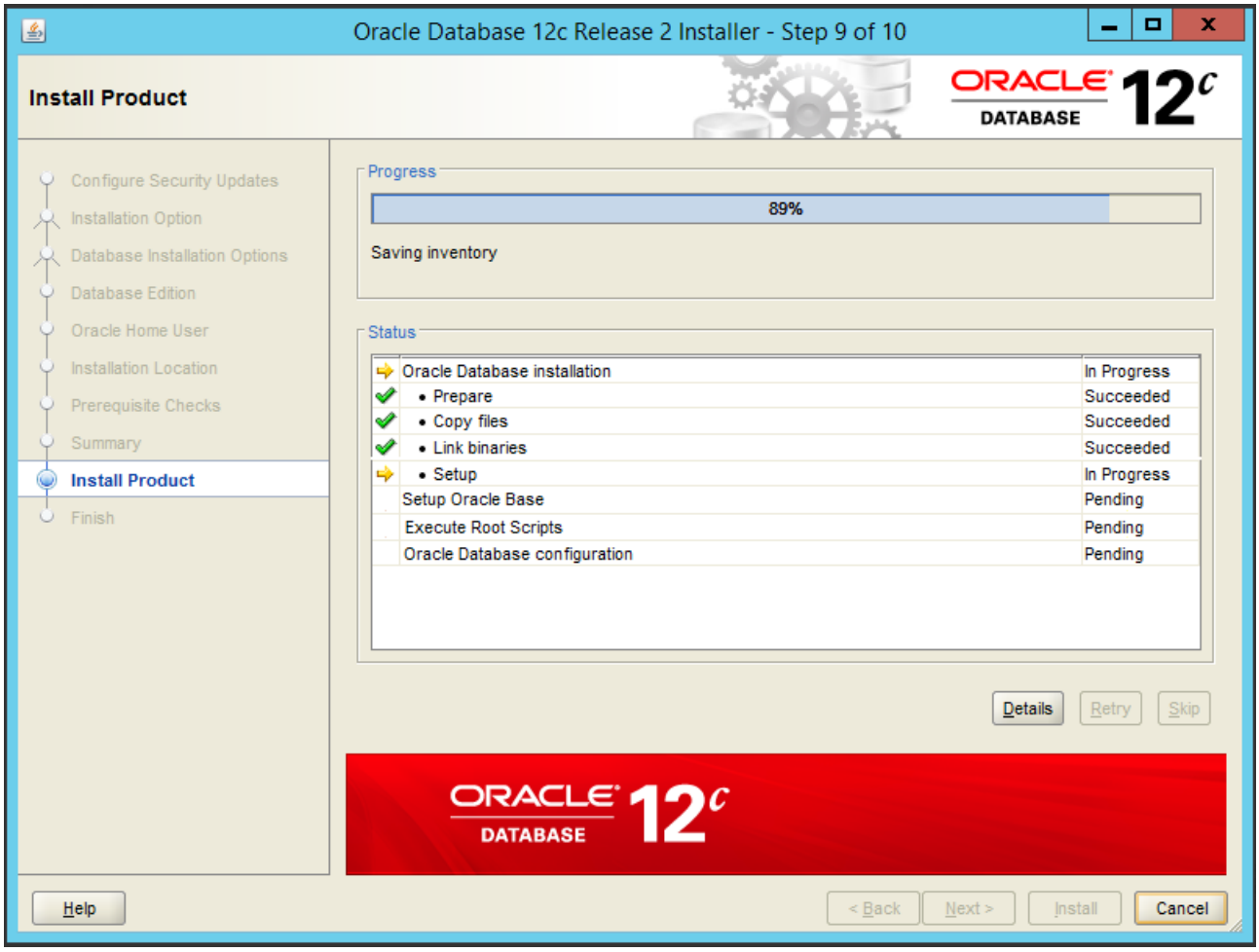
(SID) a Password, toto heslo později slouží pro přihlášení do administrátorského účtu databáze, jako jsou SYS a SYSTEM.



Obrázek 6 Oracle instalátor - nastavení parametrů

Po nastavení potřebných parametrů se spustí kontrola nastavení systému. Zkontroluje se, zda je nainstalovaný veškerý software potřebný pro ORACLE, jakým je například Java. Provedená kontrola vypíše souhrn nastavení před instalací, který si může uživatel prohlédnout a zkontrolovat, zda je korektní. Případně ho může před spuštěním instalace pozměnit. Následně spustí instalaci ORACLE - tato instalace může chvíli trvat. V rámci instalací jsou provedena také všechna potřebná nastavení, aby bylo hned po nainstalování možné se připojit do databáze. Lze se připojit pomocí nástroje SQL*Plus, který je součástí ORACLE. SQL*Plus je základní nástroj pro správu databáze, nemá vlastní GUI, používá se z příkazové řádky. Lze v něm spouštět pět druhů textových příkazů, jakou jsou SQL a PL/SQL výrazy, interní příkazy SQL*Plus, (SET, SHOW…), externí příkazy uvozené znakem „!“ a jako poslední komentáře. Spustí se příkazem „SQLPLUS“ spolu s uvedením role, na kterou se má připojit, případně lze do příkazu přidat ještě uživatele, kam se má

připojit, (standardně se připojuje na uživatele SYS). Do databáze se lze připojit také přes developer nástroj. Tento nástroj je program, který si uživatel nainstaluje sám a používá ho pro připojení a následnou správu nebo vývoj.

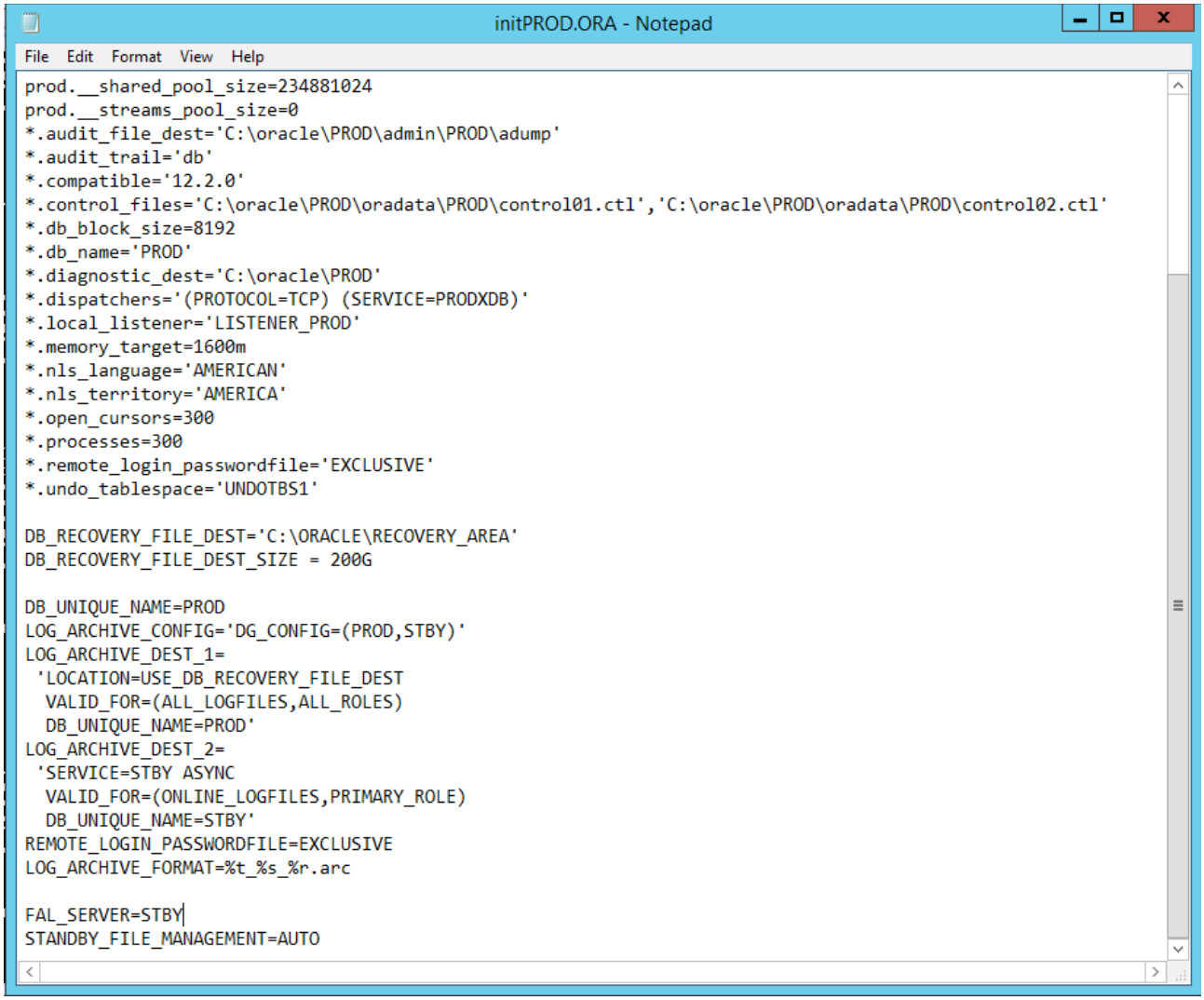


Obrázek 7 Oracle instalátor - průběh instalace

Následují kroky, pro do nastavení databáze, vlastní instalace nastavila jen nutné věci pro spuštění a následnou práci s databází. Ostatní nastavení musí provést sám uživatel, a to podle určitých kritérií (například podle standardů Oracle, efektivnosti, …).

Nastavují se parametry, se kterými se spouští databáze. K jejich nastavení slouží parametrizační soubor PFILE a SPILEF. Rozdíl mezi PFILE a SPFILE je v tom, že PFILE je statický textový soubor na straně klienta, který se standardně edituje pomocí textových editorů (například notepad, PSPad,…) a uživatel jej může nastavovat sám. SPFILE je oproti prve uvedenému v binární podobě a lze jej modifikovat pouze příkazem „ALTER SYSTEM SET…“; zmíněný způsob snižuje lidskou chybovost, protože dokud není příkaz syntakticky správně, nemůže se provést a spustit databázi a tím změnit funkčnost systému. SPFILE lze vytvořit také jako kopie PFILE pomocí příkazu v nástroji

SQL*Plus a to „CREATE SPFILE FROM PFILE“, který převede textovou formu na binární. A naopak lze vytvořit PFILE z SPFILE „CREATE PFILE FROM SPFILE“, převedením binární formy na textovou a to umožní uživateli změnit nastavení ručně. Uživatel si posléze například vytvoří znovu SPFILE z PFILE s jeho novým nastavením.



Obrázek 8 PFILE soubor

Databáze se standardně spouští pomocí SPFILE, ale lze ji spustit i pomocí PFILE, a to tím že ho startovacímu procesu vnutí přidáním cesty na PFILE následujícím způsobem: „STARTUP PFILE=C:\oracle\PROD\product\12.2.0\dbhome_1\database\init.ora“.

Pro nastavování zálohování a archivování archivních logů je dobré nastavit umístění pro RECOVERY_AREA. To se nastavuje přes parametr DB_RECOVERY_FILE_DEST v PFILE. RECOVERY_AREA je nastavené umístění na disku (adresář), souborový systém nebo skupina disků, která poskytuje centralizovanou paměť pro soubory zálohování a

obnovy. Oracle vytvoří archivní logy a záznamy zpětného vyhledávání do nastavené oblasti RECOVERY_AREA. Tato oblast funguje také jako mezipaměť disku pro pásky.

Nedílnou součástí je také parametr DB_RECOVERY_FILE_DEST_SIZE, která určuje jak velká má být oblast RECOVERY_AREA.

V PFILE se nastaví ještě parametry LOG_ARCHIVE_DEST, které se používají pouze pro databázi v ARCHIVELOG režimu nebo pro obnovování z archivních REDO logů. Je to sada parametrů sloužící pro nastavení umístění, kam se budou ukládat archivní logy. Každý takový parametr musí obsahovat atribut LOCATION nebo SERVICE určující místní adresáře na disku nebo vzdáleně přístupnou databázi. Všechny ostatní atributy jsou volitelné.

Po nastavení PFILE je dobré otestovat jeho funkčnost, tím že se s ním spustí databáze - až po úspěšném nastartování databáze se může vytvořit SPFILE ze zmíněného PFILE.

Následně po nastavení PFILE a SPFILE se instance databáze spustí a pro archivování vytvořených REDO logů se uvede do režimu ARCHIVELOG. V režimu ARCHIVELOG databáze vytváří kopie REDO logů a uloží je do předem nastaveného umístění RECOVERY_AREA. Do tohoto umístění se uloží až po naplnění REDO logu případně při jeho ručním přepnutí, kdy se vytvoří nový REDO log s tím, že ten předchozí zůstane nenaplněný do své maximální velikosti.

V okamžiku, kdy se databáze přepne do ARCHIVELOG režimu, se ihned spustí generování archivních REDO logů. Tyto archivy budou přesouvány na server STANDBY a následně na databázi aplikovány.

Do ARCHIVELOG režimu se může databáze uvést jen v nastartovaném módu MOUNT. Tento mód umožňuje spustit instanci Oracle a připojení databáze bez jejího otevření. V tomto stavu se nedá do databáze připojit přes nástroj developer Do tohoto módu se lze dostat přes příkaz „STARTUP MOUNT“, a to z vypnuté databáze nebo z již spuštěné instance v NOMOUNT. Režim NOMOUNT způsobí, že instance nebude při startu ani v režimu MOUNT, ani otevřená. Po uvedení do módu MOUNT se spustí příkaz „ALTER DATABASE ARCHIVELOG“ pro již zmíněný režim ARCHIVELOG, a v ten okamžik se začnou tvořit archivy REDO logů.

Těmito popsanými kroky se databáze dostane do stádia, kdy je připravena na otevření. Samozřejmě lze databázi nastavovat daleko více, ale pro účely testování výpadku produkčního prostředí a znovu uvedené do provozu se zajištěním vysoké integrity dat, toto

nastavení postačí a databáze se tedy uvede do otevřeného stavu příkazem „ALTER DATABASE OPEN“.

Nastaví se lokální názvy připojení v souboru tnsnames.ora pro zjednodušení komunikace a propojení mezi dvěma servery. Je to konfigurační soubor, který obsahuje názvy síťových služeb mapovaných na Oracle LISTENER. Název síťové služby je alias mapovaný na síťovou adresu databáze v síťovém descriptoru. Klienti a databázové servery (které jsou klienty jiných databázových serverů) používají název síťové služby při vytváření spojení s aplikací. Ve výchozím nastavení se soubor tnsnames.ora nachází v $ORACLE_HOME\network\admin, lze ho také uložit do jiného umístění například jako adresář určený systémovou proměnnou prostředí TNS_ADMIN s uvedenou cestou na soubor nebo hodnotou v registru.

Oracle LISTENER je samostatný proces naslouchání, který běží na databázovém serveru. Přijímá příchozí požadavky na připojení ke klientům a spravuje provoz těchto požadavků na databázový server. LISTENER je nakonfigurován s jednou nebo více adresami naslouchacího protokolu, informacemi o podporovaných službách a parametry, které řídí jeho chování. Konfigurace LISTENERu je uložena v konfiguračním souboru s názvem listener.ora.

Protože všechny konfigurační parametry mají výchozí hodnoty, je možné spustit a používat LISTENER bez konfigurace a má výchozí název LISTENER, nepodporuje při spuštění žádné služby a naslouchá na následující adrese protokolu TCP / IP: (ADDRESS = (PROTOCOL = tcp) (HOST = jméno_hostitele) (PORT = 1521)). Tento proces je potřebný pouze na straně databázového serveru a je řízen nástrojem LSNECTL.

V této chvíli, je databáze dostatečně nastavena a lze se připojit přes developer nástroj, kde se začnou řádně zakládat schémata, objekty, a podobně. Do developera se lze přihlásit jen na administrátorské uživatele, například SYS. SYS uživatel má neomezené oprávnění, všechny tabulky a pohledy pro Data Dictionary (datový slovník) jsou uloženy v tomto schématu. Základní tabulky a pohledy v SYS jsou zásadní pro provoz Oracle databáze. A proto jsou zpracovávány pouze systémem, nikdy ne uživatelem nebo správcem databáze, zachovává se tím integrita. Ve schématu SYS je povoleno vytvářet objekty, ale je doporučeno žádné objekty nevytvářet.

Ze schématu SYS se vytvoří vlastní schéma potřebné pro následnou simulaci provozu a testování výpadků. Bude sloužit pro vytváření objektu simulace. Nejprve se založí TABLESPACE příkazem „CREATE TABLESPACE ‚name‘…“. TABLESPACE je

prostor, kam Oracle logicky uchovává data, tabulky nebo indexy, fyzicky jsou ukládány v datových souborech.

Vytvořený TABLESPACE se použije jako výchozí prostor pro nové schéma, přičemž přiřazení se provádí již při založení, pomocí volitelného parametru „DEFAULT TABLESPACE ‚name'…".

Aby bylo možné se do schématu připojit a vytvářet v něm objekty, musí se přes uživatele SYS přiřadit privilegia na možnost přihlášení do nového schématu a následné možnosti v něm vytvářet objekty. Na to se využije příkaz "GRANT ‚privileges' TO ‚name_schema'", který privilegia uživateli přiřadí.

```
-- Založení Tablespace
CREATE TABLESPACE PROD
DATAFILE 'C:\oracle\PROD\oradata\PROD\PROD.dbf' SIZE 100M
AUTOEXTEND ON NEXT 50M MAXSIZE 1000M;

-- Založení User
CREATE USER PROD
IDENTIFIED BY Heslo123
DEFAULT TABLESPACE PROD
QUOTA UNLIMITED ON PROD
TEMPORARY TABLESPACE TEMP;

-- Přidání privilegii na připojení a vytváření objektu
GRANT CONNECT, RESOURCE TO PROD;
```

V tomto stavu je databáze připravena na vytvoření simulace provozu. Aby bylo možné zjistit časy RPO bude nutné spustit na databázi jednoduchý script, který bude do tabulky postupně vkládat řádky. Založí se nejprve tabulka, do které se budou plnit data, která se budou zapisovat do REDO logů. Postačí tabulka o dvou sloupcích, kde bude sloupeček s datem a časem vložení záznamu a náhodně generovaného stringu. Pro plnění tabulky v tomto případě postačí založit Proceduru, která bude každé spuštění generovat 10.000 nových záznamů do tabulky.

Procedura, Funkce nebo Package je skupina PL/SQL příkazů, které se můžou spouštět podle názvu. Tyto balíčky jsou uloženy v databázi a tím se umožňuje, aby byl kód zapsán a testován jednou a poté byl přístupný pro libovolné aplikace.

```
--Založení Tabulky
CREATE TABLE DummyDataTable
(
Date_log        DATE,
Data            CLOB
);

--Založení Procedury
CREATE OR REPLACE PROCEDURE DummyDataProcedure IS
BEGIN
  FOR i IN 1..10000
    LOOP
      INSERT INTO DummyDataTable(Date_log, Data)
      VALUES(SysTimeStamp, DBMS_RANDOM.string('A', 100));
    END LOOP;
END DummyDataProcedure;
/
```

Pro vytvoření jobu je potřeba pod uživatelem SYS nastavit nově vytvořenému schématu privilegia na zakládání jobu. Bez tohoto privilegia nebude možné job založit. Nyní stačí založit Job, který tuto Proceduru bude spouštět, co bude plnit tabulku daty. Tento Job se nastaví na náhodně vybraný časový interval spouštění. Joby slouží ke spouštění různých úloh nebo procesů v určitých časových intervalech. Tyto intervaly se mohou nastavit trvale (například každou Středu v 9:00 hodin ráno) nebo se může nastavit pomocí Funkce, která bude vracet generovaný čas pro spuštění podle potřeb uživatele, který nastavil různé podmínky, jaký čas se má vrátit (například jednou se spustí ve Čtvrtek v 18:00 hodin večer a další spuštění v Pátek v 01:00 hodin ráno). Jobu se dá nastavit také Auto Drop (=po běhu smazat), přidat komentář, od jakého data a času se má spouštět nebo do jakého data a času se má spouštět a spoustu dalších parametrů.

Pro spuštění simulace se nastaví spouštění jobu každých 30 vteřin., tím bude zajištěna simulace provozu produkčního prostředí.

```
--Přiřazení privilegií
GRANT EXECUTE ON dbms_scheduler TO PROD;
GRANT CREATE ANY JOB TO PROD;

-- Založení Jobu
BEGIN
  dbms_scheduler.create_job(job_name           => 'PROD.DUMMYDATA',
                            job_type       => 'PLSQL_BLOCK',
                            job_action     => 'DummyDataProcedure;',
                            start_date     => SYSDATE,
                            repeat_interval => 'FREQ=SECONDLY;INTERVAL=30',
                            end_date       => to_date(null),
                            job_class      => 'DEFAULT_JOB_CLASS',
                            enabled        => TRUE,
                            auto_drop      => FALSE
END;
/
```

Po přípravě primárního serveru je možné začít s nastavením druhého serveru, kam se bude replikovat produkční databáze. Na tento server postačí nainstalovat jen samotný software Oracle bez automatické konfigurace databáze. K instalaci se využije stejný Oracle instalátor, jako se využil při přípravě prvního serveru s tím rozdílem, že se zvolí možnost „Install database software only“ a následně ještě „Single instance database installation“, která nainstaluje pouze software Oracle.

Jakmile se dokončí instalace, vytvoří se stejná struktura složek, jako je na produkčním serveru, aby mohly být soubory správně zduplikovány na STANDBY.

Z produkce se zkopíruje PFILE a passwordfile „pwdPROD.ora“, soubor hesel sloužící k připojení ke vzdálené správě přes uživatele s rolí SYSDBA.

Služba pro instanci se vytvoří ručně, vzhledem k tomu, že neproběhla klasická konfigurace databáze. Vytvoří se pomocí utility ORADIM s parametry SID, STARTMODE, který určuje, zda se má služba spustit automaticky nebo ručně. Dále s parametrem SYSPWD, což je heslo pro SYSDBA a jako poslední je nutno zadat cestu k PFILE.

V okamžik, kdy jsou oba servery připravené, se musí nastavit síť mezi nimi, aby k sobě mohly přistupovat, a povolí se porty na Firewallu.

Následně je potřeba udělat kopii produkce, kterou lze provést několika způsoby. V tomto případě je zvolena duplikace pomocí nástroje RMAN, která je po správné přípravě serveru z pohledu replikace nejjednodušší. Pro tohle je zapotřebí vytvořit nové řídící soubory na server, kam bude zduplikována produkční databáze. V těchto souborech jsou

obsaženy informace o struktuře databáze a umístění potřebných systémových Tablespace (tabulkové prostory) nebo REDO logů. Při běhu duplikace se přesunou také soubory REDO logů nazvaných REDO.log a přesune defaultní systémové tabulkové prostory, jako jsou SYSTEM01.dbf, SYSAUX01.dbf, TEMP01.dbf, UNDOTBS01.dbf, USER01.dbf a PROD.dbf, který byl vytvořen pro simulaci provozu. Tyto tabulkové prostory jsou uloženy standardně v $ORACLE_HOME\oradata\$ORACLE_SID.

Systémové tabulkové prostory jsou nedílnou součástí konfigurace Oracle. Uchovávají všechny tabulky Data Dictionary v databázi a jsou vytvořeny v době vytváření databáze. Tyto tabulkové prostory jsou vždy online a musí v tomto stavu být pro otevření databáze.

Dále nastartovat produkční instanci v režimu Mount a STANDBY instanci v Nomount režimu. Po tomto kroku se využije nástroj RMAN, který duplikaci provede. Spustí se příkaz na připojení obou databází pomocí RMAN a to následovně: „RMAN TARGET=SYS/*****@PROD AUXILIARY=SYS/*****@STBY;“. U tohoto příkazu se nesmí splést pořadí zadaní přihlašovacích údajů. První musí být údaje pro produkční instanci a druhé v pořadí STANDBY instance. Pokud se toto zadaní zamění, začne se obnovovat produkce ze STANDBY, což zapříčiní rozbití produkce.

Dále pro spuštění duplikace se spustí příkaz „DUPLICATE TARGET DATABASE FOR STANDBY FROM ACTIVE DATABASE DORECOVER NOFILENAMECHECK;“, tím začne proces duplikace.

Proces duplikace vytvoří přesnou kopii produkční instance a nastaví jí roli DATABASE_ROLE na „PHYSICAL STANDBY“.

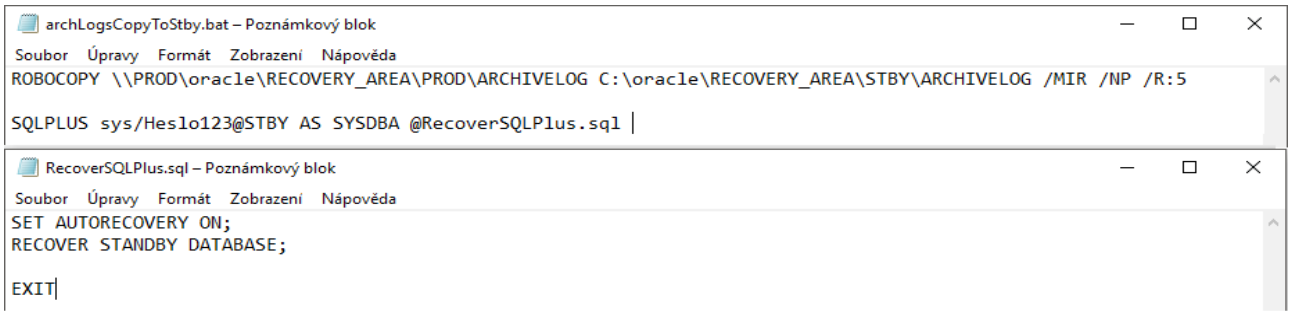
Jakmile se dokončí duplikace databáze, měla by jít instance nastavit do režimu MOUNT STANDBY a otevřít pro čtení. Použijí se na to příkazy „ALTER DATABASE MOUNT STANDBY DATABASE“ pro nastavení do režimu MOUNT STANDBY a „ALTER DATABASE OPEN READ ONLY“ pro otevření databáze pro čtení. Ve stavu pro čtení by STANDBY databáze měla být do okamžiku výpadku produkční databáze. Dokud nenastane situace výpadku.

Následný krok je, nastavit automatické kopírování a aplikování archivních logů z produkce. Pro kopírování je potřeba, aby se obě složky s archivními logy synchronizovaly - k tomu se využije utilita ROBOCOPY a díky funkci zrcadlení se zajistí, aby obě složky byly synchronní. Ideální varianta by byla ještě zároveň kontrolovat, zda se

archivní logy aplikovaly. A na základě toho je promazávat na produkčním serveru po záloze. Nicméně pro tyto testovací účely, je tato metoda nepotřebná.

Na aplikaci zkopírovaných logů se pro tento případ využije funkcionalita Oraclu „RECOVER STANDBY DATABASE;“, spouštěná přes SQL*Plus. Recovery Standby Database při aplikování archivních logů kontroluje, které logy ještě nejsou aplikovány a ty poté aplikuje. K příkazu se přidá ještě „SET AUTORECOVERY ON;“, což nastaví automatickou obnovu všech dostupných archivních logů. Proces musí probíhat v nastartovaném režimu Mount Standby Database.

Oba tyto procesy se mohou spustit v jednom běhu. Udělají se dva scripty, v prvním se spustí kopírování logů, následně se připojí do SQL*Plus a spustí druhý script, který spustí příkazy pro aplikování a sám se odpojí. Tento proces se nastaví jako job serveru „Task scheduler“, a bude se spouštět každých 10 minut.



Obrázek 9 Script pro synchronizování databází

7.2 Simulace výpadků

Tato část se bude zabývat simulací výpadků produkčního prostředí a znovu-obnovení standardního provozu na dříve připraveném prostředí. Před každým výpadkem, bude zjištěno poslední datum ze simulační tabulky, pro srovnání se STANDBY databází. V ideálním případě se údaj o datu bude na obou prostředích shodovat.

Budou se provádět statistiky ztrátovosti dat a časové náročnosti pro znovu obnovení provozu pro jednotlivé testy.

7.2.1 Výpadek produkční databáze

Nasimulovat výpadek produkce lze několika způsoby, například vypnutím služby pro instanci nebo vypnutím databáze pomocí příkazu „SHUTDOWN“. V tomto případě se využije příkaz SHUTDOWN, který databázi korektně zastaví bez jakéhokoliv rizika poškození.

Následně je zapotřebí STANDBY vypnout, vypnout službu na obnovování z produkce a aplikovat zbylé archivní logy, které ještě nebyly aplikovány. Spolu s tím se musí zkopírovat REDO logy produkce a vytvořit nové řídící soubory pro STANDBY, které budou databázi od té chvíle označovat jako produkční prostředí a spouští archivní mód pro uchovávání všech proběhlých změn.

Záložní databáze se spustí v MOUNT režimu a zkontroluje se, zda má roli „PRIMARY“, k tomu stačí spustit SELECT „SELECT DATABASE_ROLE FROM V$DATABASE;“. Pokud je správně brána jako produkční, spustí se „RECOVER DATABASE“, který obnoví přesunuté REDO logy zkopírované z produkce. Po správném obnovení je možné databázi otevřít a začít na ní pracovat jako v běžném provozu. V tuto chvíli se spustí vytvořený job, který zajišťoval simulaci provozu, aby došlo k vygenerování dat na původní STANDBY databázi pro dokončení simulace.

Jakmile bude produkce opravena, muže se na ní přepnout zase zpět. V testovaném případě stačí spustit. Ještě před tím, se musí zastavit STANDBY a zkopírovat z ní nové archivní logy, které se na ní během provozu vygenerovaly. K tomu je nutné zkopírovat REDO logy a vygenerovat nové řídící soubory pro produkci, bez čehož by nešla databáze spustit, protože by hlásila starou verzi od REDO logů.

V tu chvíli se spustí RECOVER DATABASE, aby se obnovily archivní logy a REDO logy. A následně se databáze otevře a je obnoven standardní provoz.

Na závěr je nutné uvést STANDBY do původního stavu. To se provede vytvořením nových Controlfilů z produkce označených jako STANDBY a to příkazem „ALTER DATABASE CREATE STANDBY CONTROLFILE AS 'C:\oracle\tmp\control01.ctl';“. Tyto logy se obnoví na záložní server pomocí „RESTORE STANDBY CONTROLFILE FROM 'C:\oracle\tmp\control01.ctl';“ a spustí se v režimu MOUNT STANDBY DATABASE a následně se otevře v režimu pro čtení a znovu spuštění služby na obnovování z produkce.

Před testem bylo zapotřebí zjistit poslední datum ze simulační tabulky na produkci a STANDBY a následně také po testu přepnutí na STANDBY. Toto je nezbytně nutné pro následné zjištění hodnoty RPO, předpokládá se, že v tomto testu bude RPO rovno nule, tedy časy na obou databázích se budou rovnat.

Výsledek testu před aplikací logů je:

- Hodnota pro produkci ID bylo 10:03:53.106000
- Hodnota pro STANDBY bylo 10:02:02.137000

Výsledek testu po aplikaci logů je:

- Hodnota pro STANDBY bylo 10:03:53.106000

Výsledek byl tedy úspěšný, hodnoty se rovnaly a bylo prokázáno, že při tomto výpadku nebyla ztracena žádná data.

7.2.2 Výpadek produkčního serveru

Výpadek produkčního serveru se nasimuluje vypnutím, tím nastane nedostupnost na databázi i na server. Pro tento test nebudou nastaveny dvě cesty pro archivní a REDO logy, aby bylo zjištěno, jak velká ztráta dat bude.

Po nasimulovaném výpadku serveru, se rovnou přepne záložní server na produkční, protože nemáme k dispozici žádné archivní logy nebo REDO logy ze serveru. Vypne se služba na obnovování z produkce a STANDBY se nastartuje v NOMOUNT režimu a vygenerují se nové řídící soubory, aby byla databáze označena jako PRIMARY, označení se zkontroluje SELECTem „SELECT DATABASE_ROLE FROM V$DATABASE;“. Poté se databáze otevře a spustí se provoz.

Provoz bude pokračovat na neúplných datech, protože na server se v této chvíli nedá dostat a provoz nemůže v tomto testovaném případě čekat na zprovoznění serveru. Pokud by byla situace, kdy lze počkat na zprovoznění, není potřeba přepínat na záložní databázi, protože po zprovoznění bude dostupná také produkční databáze.

Jakmile bude server plně funkční, přepne se zpět na produkci. Přepnutí proběhne vypnutím STANDBY databáze a překopírováním nových archivních logů a REDO logů. Následně se vygenerují nové řídící soubory produkce a spustí Recovery database. V tuto chvíli se produkce otevře a je znovu obnoven standardní provoz se ztrátou dat. Záložní databáze se uvede zpět do STANDBY role.

Před testem se zjistilo poslední datum simulační tabulky na produkci a STANDBY,pro zjištění měrné metriky RPO.

Výsledek testu je:

- Hodnota pro produkci ID bylo 10:38:49.359000
- Hodnota pro STANDBY bylo 10:37:01.574000

Porovnání ukázalo, že hodnota RPO je nenulová, v tomto případě je hodnota 1 minuta a 48 vteřin.

7.2.3 Simulace plánované údržby

STANDBY databáze může sloužit také jako přechodný provoz, při údržbě produkčního prostředí. Produkce se korektně vypne, zkopírují se z ní chybějící archivní logy s REDO logy. Následně se vygenerují nové řídící soubory pro STANDBY a obnoví zbytek dat. V tento moment se databáze otevře a přechodně se na ní spustí provoz.

Po skončení údržby na produkčním prostředí, se provoz přesune zpět. Zkopírují se logy, vygenerují nové řídící soubory a obnoví data, to obnoví provoz na produkci. A jako poslední se uvede STANDBY do původního stavu před přepnutím.

Nedojde ke ztrátě žádných dat a nebude potřeba udělat delší odstávku než jen pár minut pro přepnutí databází.

8 Výsledky a jejich hodnocení

Testy proběhly celkem tři, přičemž mezi testy bylo zapotřebí kopii produkce opětovně postavit respektive provést plnou kopii z produkční databáze do záložní.

Po dokončení prvního testu byla zjištěna měřená metrika RPO, tak jak bylo očekáváno, tedy nulová.

Měrná metrika RTO pro tento test byla pouhých pár minut, tedy 10 minut. Samozřejmě u větších databází, například více než 1TB, bude tato měrná metrika RTO podstatně vyšší, protože při tomto testu byla využita databáze o velikosti 1GB. Následné přepnutí zpět na produkční databázi proběhlo ve stejném čase. K tomuto času se připočítává i přepínání aplikací, které jsou k databázi připojeny. Čas, který se díky těmto aplikacím připočte, je závislý na jejich počtu a době jejich startu a vypnutí. Důležité je také vzít v potaz reakční dobu databázového administrátora.

Varianta	RTO	RPO
Výpadek produkční databáze	10 min	0

Tabulka 1 Výsledky pro simulaci výpadku produkční databáze

Pro druhý test výpadku celého produkčního prostředí, byla zjištěna hodnota RPO 1 minuta a 48 vteřin. Takové ztrátě se lze vyhnout generováním archivních logů a plnění REDO logů na dvě umístění, nejlépe na umístění na produkčním serveru a na vzdáleném hostovi, například záložním serveru.

Pro archivní logy se dvě umístění nastavují v PFILE jako parametr LOG_ARCHIVE_DEST_n, místo znaku „n“ se vloží pořadí. A pro REDO logy se umístění nastaví v Controlfilech.

Časová náročnost měřená měrnou metrikou RTO pro tento výpadek, zabere také pouhých pár minut. V tomto případě do necelých 10 minut jako u předchozího výpadku se stejným použitým prostředím. Poté nějaký čas, který zabere zprovoznění serveru a databáze na něm.

Varianta	RTO	RPO
Výpadek produkčního serveru	10 min	108 vteřin

Tabulka 2 Výsledky pro simulaci výpadku produkčních systémů

A u posledního testu bylo prokázáno, že při údržbě je RPO nulové a RTO vcelku odpovídá RTO předchozím dvěma testům.

Varianta	RTO	RPO
Simulace plánované údržby	10 min	0

Tabulka 3 Výsledky pro simulaci plánované údržby

9 Závěr

V práci bylo zjištěno, že nejbezpečnější metoda zálohování je STANDBY databáze, která splňuje požadavky na vysokou dostupnost při výpadku. Při této metodě je zajištěna vysoká integrita dat, pokud dojde k výpadku produkčních systémů měrnou metrikou RPO. Také poukazuje na časovou náročnost obnovení provozu měrnou metrikou RTO, jak z pohledu přepnutí z produkčních systémů, tak i z pohledu přepnutí na produkční systémy zpět ze STANDBY.

Toto tvrzení potvrzují i provedené testy v této práci, které byly provedeny na vytvořeném simulačním prostředí. Provedené testy, poukazují také na naměřenou hodnotu RTO a RPO, podle kterých se dalo jednoznačně určit, zda použitá metoda splňuje vysokou dostupnost při výpadu systémů.

Verze Oraclu Enterprise Edition nabízí pro STANDBY funkčnost Oracle Data Guard, který poskytuje komplexní nastavení služeb, které obstarávají jednu nebo více STANDBY databází. Tyto databáze umožňují zachovat provoz při výpadku a poškození dat. Oracle Data Guard udržuje záložní databáze jako přesné kopie produkční databáze. Pokud produkční systémy vypadnou, může Oracle Data Guard přepnout libovolnou STANDBY na produkční roli a minimalizovat čas výpadku.

Verzi, která je použita v této práci je Standard Edition a ta tuto možnost bohužel nemá, proto musel být použit jiný způsob vytváření přesně kopie instance. Využila se vestavěná funkcionalita ROBOCOPY a následně příkaz používaný v SQL*Plus na obnovu archivních logu. Z těchto příkazů se vytvořil script, který byl nastaven na opakované spouštění v intervalu 10 minut.

Zvolený způsob neposkytuje tak velkou integritu dat, jakou má Oracle Data Guard, ale nejde o tak velký rozdíl. Ten je ve skutečnosti pouze v tom, že Oracle Data Guard je online a pokud se tedy vytvoří nový archivní log, okamžitě se aplikuje na STANDBY. Alternativně zvolený způsob oproti tomu log neaplikuje okamžitě, ale až v nastaveném intervalu. Proto je zde riziko, pokud není nastaveno ukládání logů na vzdáleného hosta, že při výpadku produkčního serveru nebude možné překopírovat poslední archivní logy a následně je tedy nebude možná ani aplikovat a dojde ke ztrátě dat.

Před samotnou stavbou infrastruktury pro dosažení vysoké dostupnosti při výpadku je dobré si ujasnit zmiňované hodnoty RTO a RPO, které pro kombinaci unikátního systému a metody zálohování jednoznačně řekne, zda splňují požadavky organizace nebo nikoliv.

10 Seznam literatury

(1) Darl Kuhn. *Pro Oracle Database 12c Administration*. Apress, 2013. ISBN 9781430257288.

(2) Francisco Munoz Alvarez, Aman Sharma. *Oracle Database 12c Backup and Recovery Survival Guide.* Packt Publishing, 2013. ISBN 9781782171218.

(3) FREEMAN, Robert G. a Matthew HART. *Oracle database 12C Oracle RMAN backup and recovery*. New York: McGraw-Hill Education, 2016. ISBN 9780071847445.

(4) Ignatius Fernandez. *Beginning Oracle Database 12c Administration: From Novice to Professional*. Apress, 2015. ISNB 9781484201930.

(5) James Seo. *Oracle Backup and Recovery: All about Oracle Backup and Recovery*. Slowthinking Co., Ltd., 2014.

(6) KUMAR, Bipul. *Oracle Data Guard: Standby Database Failover Handbook.* North Carolina: Rampant TechPress, 2004. ISBN 9780974599380.

(7) Martin Bach. *Expert Consolidation in Oracle Database 12c*. Apress, 2014. ISBN 1430244291.

(8) Matthew Hart, Scott Jesse. *Oracle Database 10g High Availability with RAC, Flashback & Data Guard*. McGraw Hill Professional, 2004. ISBN 9780072258288.

(9) Michael New, Edward Whalen, Matthew Burke. *Oracle Enterprise Manager Cloud Control 12c Deep Dive*. McGraw Hill Professional, 2013. ISBN 9780071790574.

(10) NASSYAM, Basha Emre Baransel. *Oracle data guard 11gr2 administration beginner's guide.* Packt Publishing, 2013. ISBN 9781849687911.

(11) Backing Up the Database [Online]
https://docs.oracle.com

(12) BackUp [Online]
https://docs.oracle.com

(13) Data Pump Export [Online]
https://docs.oracle.com

(14) Duplicate [Online]
https://docs.oracle.com

(15) EXP [Online]
https://docs.oracle.com

(16) Flashback [Online]
https://docs.oracle.com

(17) Recovery Manager Architecture [Online]
https://docs.oracle.com

Seznam obrázků