

Diplomová práce

Synchronizace a replikace geodat v prostředí Esri platformy

odkaz na pdf verzi

https://github.com/makkuska/diplomka/blob/master/text/latex_text/Solanska_DP.pdf?raw=true

Úvod

1 CÍLE PRÁCE

2 POUŽITÉ METODY A POSTUPY PRÁCE

3 TEORETICKÁ VÝCHODISKA

3.1 Vymezení pojmů

3.2 Replikace

3.3 ArcGIS produkty

3.4 Použité programové prostředky

3.4.1 PostgreSQL 9.x (PostGIS)

3.4.2 Microsoft SQL Server Express 2008

3.4.3 ArcSDE geodatabase

3.5 Nástroje pro replikaci v PostgreSQL

3.5.1 Slony-I

3.5.2 Streaming replikace

3.5.3 pgpool

4 NÁVRH A IMPLEMENTACE REPLIKACE

4.1 Návrh replikačního řešení pro katedru

4.2 Příprava prostředí pro konfiguraci replikace

4.3: Připojení produktů ArcGIS k databázi PostgreSQL

4.4 Implementace replikace

4.4.1 Slony-I replikace

Inicializace

Přidání další tabulky do replikace

4.4.2. Streaming replikace

4.5 Rozložení zátěže mezi replikačními uzly

5 TESTOVÁNÍ

6 DISKUZE

7 ZÁVĚR

ZDROJE

Úvod

Dnešní trend je ukládat a ponechávat stále více dat pouze v digitální podobě. Mnoho dokumentů už se vůbec netiskne do papírové podoby, tím spíš pokud dnes existují elektronické podpisy, díky kterým je tištěná verze naprosto zbytečná. S přibývajícím počtem dat je však třeba řešit komplikace, které počítačová data přinášejí. Počítačová experti řeší například otázky, kam ukládat tak velké množství dat, jak data efektivně aktualizovat, jak zabránit poškození dat ať už způsobených lidským faktorem či fyzickým poškozením hardware. V případě, že se poškodí disk, můžeme často během okamžiku přijít o všechna data, někdy však pro ztrátu dat stačí pouze stisknout tlačítko na klávesnici. Určitě už se Vám nejednou stalo, že jste se nemohli přihlásit do svého účtu na internetu z důvodu přetížení serveru. I to je problém, který velké množství dat a velký počet uživatelů přináší. Jak tedy pracovat s těmito objemy, jak zabránit komplikacím, které mohou poškodit či zcela zničit celou dosavadní práci, a jak zrychlit celý proces práce s daty?

Řešením velkého počtu výše uvedených problémů může být ukládání dat do databáze a jejich následná replikace. Replikací je myšlena pokročilá funkce, která zajišťuje kopii dat na více serverů. Nabízí ji většina dnešních databázových serverů, zajišťuje větší robustnost databáze a vysokou dostupnost dat. Replikaci lze využít ve všech odvětvích, které pracují s daty. Výjimkou tedy není ani geoinformatika, která pracuje s velkými objemy dat, které navíc nesou informaci o geografické poloze. Právě reprezentace geografické polohy, skrze textový zápis souřadnice daných bodů, může způsobit razantní zvýšení velikosti dat. Například u webových dat se navíc musí řešit častý přístup k databázi, protože každé posunutí výřezu či přiblížení, resp. oddálení výřezu mapy, je samostatným dotazem, který musí kapacita serveru zvládat. Při představě, že si uživatel bude posouvat výřez mapy po 50m, může to způsobit velkou zátěž pro server. V tomto případě je potřeba řešit replikaci z důvodu rozložení zátěže.

Z mého pohledu data středně velkého až velkého projektu je vhodnější ukládat do databáze než jiných formátů typu shapefile, Microsoft Access nebo obyčejného tabulkového procesoru. Nabízí nám to sofistikované uložení dat, snadné propojení jednotlivých vrstev, snadnou přenositelnost dat, možnost relačního propojení dat nebo efektivní vyhledávání. Replikace samotná se poté využívá pro kopii dat a následnou aktualizaci změn, která v databázi nastanou.

Replikaci ocení uživatelé pracující na společném projektu, distribuovaná pracoviště i společnosti s velkým množstvím důležitých dat, jejichž kopie je rozhodující pro jejich fungování. Dobrým příkladem využitelnosti replikace je také nový trend využívání offline aplikací v mobilních telefonech. Databáze se vždy replikuje do mobilního telefonu, kde může fungovat offline a vždy, když se klient připojí na internetovou síť, aplikace kontroluje zda není na serveru novější verze databáze a pokud je, zkopíruje pouze změny, které proběhly od posledního stahování. (Jako příklad z geoinformatického prostředí bych uvedla diplomovou práci Dalibora Janáka, který řeší replikaci databáze lezeckých cest do mobilní aplikace.)

Databázové systémy nabízí širokou škálu nastavitelnosti, která umožňuje přizpůsobit replikaci danému řešení.

1 CÍLE PRÁCE

Cílem diplomové práce je provést rešerši v oblasti dostupných replikačních řešení a na jejím základě prakticky otestovat proces synchronizace a replikace geodat, které je možností v kombinaci s ArcGIS produkty.

V teoretické části práce budou podrobně definovány pojmy týkající se zálohování dat, především však synchronizace a replikace, dále detailně rozebrána replikace ve všech možných variantách nastavení, tedy jednosměrná, dvousměrná, synchronní, asynchronní, kaskádová, logická a fyzická. Dále rozbor zahrne celé portfolio produktů od desktop řešení, přes možnosti ArcGIS serveru až po cloudový ArcGIS online.

V rešerší části budou diskutovány dva databázové server, SQL Server a PostgreSQL, oba podporované ArcGIS produkty a jejím základě pak vybrát jeden, na kterém pak proces replikace bude prakticky testován.

Praktická část se bude zabývat návrhem replikačního řešení, které zahrne požadavky a možnosti katedry a bude brát v úvahu její způsoby využívání databáze. Na základě rešerše pak bude vybráno replikační řešení, připraveno testovací prostředí na základě všech výše zmíněných kritérií a na konec i praktickému testování výše zmíněných procesů.

Postupnými opakovanými procesy budou sledovány dílčí parametry procesu (rychlost procesu, úplnost, chybovost, podporované formáty).

Můj jeden odstaveček - něco jako - jak vidím vlastní přínos do tématu.

?? a popsány prostředky, které se na platformě Esri k těmto procesům využívají. Budou popsány možnosti, požadavky a předpoklady pro úspěšnou realizaci.

2 POUŽITÉ METODY A POSTUPY PRÁCE

Napsat úplně nakonec, v rozsahu 1 stránky.
Jaká data, jak probíhal postup.

POZNÁMKY

- Jakými kritérii se hodnotí, zda replikace probíhá správně - it-literatura.
- co budu sledovat - jestli databáze (databáze + SDE) umí replikovat jednoduchá i složitější geodata, jestli se to přeneslo, rychlost (Postgre x SQL), jaká bude kvalita (např. dle zachovaného tvaru, stejného počtu bodů, objemu), verifikace - jestli vrátí info o tom, že to proběhlo
- Replikace rastrů - je to možné? Zkoušel to už někdo?
- 4 úrovně použití replikace:
 1. uživatel ji nepotřebuje
 2. uživatel ji využívá skrze desktop (ale stále je zapotřebí server licence)
 3. přistupuje se síťově
 4. přistupuje se online
- zdroje: OGC, knihy, BostonGIS.com, dbsvět.cz (Co umí server 2008 - česky), novinky 2012 (opravy, co je nového)
- rozhraní pro Postgre, SQL -- jak se ovládá ArcCatalog, v čem se nastavuje ArcSDE, SQL Tools
- “ArcCatalog pomáhá organizovat a spravovat veškerá data GIS, tj. mapy, glóby, datové sady, geodatabáze, modely, metadata a služby (Esri, 2006).”

3 TEORETICKÁ VÝCHODISKA

Databáze je soubor vzájemně propojených datových složek, který slouží pro efektivní ukládání dat \citep{Oppel2009}. Ta jsou ukládána ve formě tabulek, entit a atributů, a jsou vzájemně propojeny logickými vazbami, které se nazývají \{it relace\} \citep{Connolly2005}. Toto logické uložení vazeb mezi tabulkami umožňuje efektivní manipulaci s daty, rychlé vyhledávání i komplexní analýzu \citep{Momjian2001}.

Základy \{it relační databáze\} položit v roce 1970 matematik E. F. Codd, který relačnímu modelu přidal i srozumitelné příkazy vycházejících z běžné angličtiny, které jsou dnes známy jako jazyk \{it SQL\} (Structured Query Language) \citep{Zak2001}. V dnešní době je možné setkat se také s pojmy objektová a objektově-relační databáze, které přebírají řadu vlastností z oblasti objektového programování.

Obvykle se rozlišují pojmy databáze, který odkazuje na obecný koncept, a pojem databázový systém nebo přesněji \{it systém řízení báze dat\} \footnote{angl. Database Management System (DBMS)}, což je konkrétním počítačovým program, který zajišťuje fyzické uložení dat. Moderní SŘBD jsou navrženy na principu klient/server, kdy databáze běží jako služba na pozadí a čeká na dotazy od klientů. Server umožňuje uživatelům přístup k databázi, vytváření a aktualizování dat, stejně jak jako vyhledávání či analyzování \citep{Connolly2005}. Uživatel s databází komunikuje skrze jazyk SQL většinou v kombinaci s vlastním jazykem daného databázového systému.

Pro uložení dat malého projektu je samozřejmě možno použít i jiného formátu určeného pro ukládání dat, například tabulkového procesoru, formátu XML, či moderního JSON. Pro komplexní správu dat velkého projektu je však databáze pro své relace více než vhodná.

Prostorová databáze, někdy také zvaná geodatabáze, není nic jiného než databáze přidaná o datový typ určený pro ukládání prostorové informace o prvku, prostorové indexy a sadu funkcí vhodných pro správu prostorových dat. Více informací o prostorových databázích v kapitole \odkazKapitola{PostgreSQL} PostgreSQL 9.x (PostGIS) a \odkazKapitola{MSSQL} MS SQL Server 2008.

Z toho vyvstává otázka, co jsou prostorová data, také zvaná geodata. Z pohledu společnosti Esri se jedná se prvky, které nesou informaci o geografické poloze, zakódovanou informaci o tvaru (bod, line, polygon) a popis geografického jevu. Tato geodata jsou uložena ve formátu, který je možno použít v geografickém informačním systému \citep{Esri2006}. Příkladem takového formátu může být vektorový Esri shapefile, Esri coverage, GML, KML, GeoJSON nebo rastrový Erdas Image a GeoTIFF. Dalším způsobem je již zmíněná databáze, do níž se vektorová data ukládají ve specifickém tvaru daném standardem OGC \footnote{OGC standardy jsou kontrolované konsorciem Open Geospatial Consortium, \newline zdroj \url{http://www.opengeospatial.org/ogc}} Simply Feature for SQL 1.2.1, který specifikuje způsob uložení dat v digitální podobě. Simple Features je založen na 2D geometrii s~možností lineární interpolace mezi lomovými body. To umožňuje vložení následujících prvků:

```
\begin{itemize}
\item bod - POINT(0 0)
\item linie - LINESTRING(0 0, 1 1, 1 2)
\item polygon - POLYGON ((0 0,4 0,4 4,0 4,0 0),(1 1, 2 1, 2 2, 1 2,1 1))
```

```

\item série bodů - MULTIPOINT((0 0),(1 2))
\item série linií - MULTILINESTRING((0 0,1 1,1 2),(2 3,3 2,5 4))
\item série polygonů - MULTIPOLYGON(((0 0,4 0,4 4,0 4,0 0),(1 1,2 1,2 2,1 2,1 1)), ((-1 -1,-1
-2,-2 -2,-2 -1,-1 -1)))
\item geometrická kolekce, která může obsahovat různé geoprvky (body, linie i polygony) -
GEOMETRYCOLLECTION(POINT(2 3),LINESTRING(2 3,3 4))\footnote{zdroj
http://postgis.net/docs/manual-2.1/using\_postgis\_dbmanagement.html#RefObject}
\end{itemize}

```

První slovo specifikace určuje druh prvku (point, linestring, polygon, multipoint, ~...), následují v závorce vypsání souřadnice lomových bodů. Za tím ještě může následovat volitelný parametr kód souřadnicového systému.

Hodnoty lze dále vkládat přes Well-Known Binary (WKB) nebo Well-Known Text (WKT) reprezentaci. PostGIS funkce pro vkládání geometrie vypadá následovně:

```

\begin{itemize}
\item ST\_AsBinary(geometry) pro bitový zápis WKB
\item ST\_AsText(geometry) pro WKT text
\end{itemize}

```

Příklad uložení linie do databáze s jedním lomovým bodem v souřadnicovém systému WGS84:

```
(LINESTRING(15.96876 50.85494, 17.21879 49.60494, 18.27348 49.85350), 4326))
```

(1)

Esri ctí vektor, raster, jeho uložení v databázi (feature dataset - ukázka) udělat printscreen

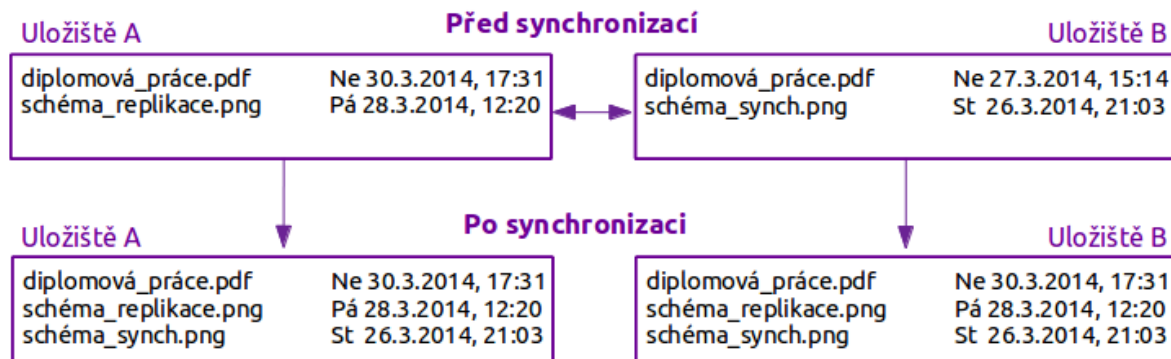
3.1 Vymezení pojmů

Pro lepší porozumění textu této práce je potřeba definovat pojmy replikace, synchronizace a verzování, včetně popisu toho, jak jsou dané pojmy chápány v produktech ArcGIS. Je vhodné upozornit, že výše zmíněné procesy jsou v literatuře často užívány lehce odlišně. Některé zdroje pojmy replikace a synchronizace rozlišují, jiné je naopak považují za synonyma.

Všechny dotyčné pojmy úzce souvisí se zálohováním dat, tedy kopírováním dat mezi dvěma a více uložišti. To, co tyto pojmy spojuje, je totiž vždy, v nějaké míře, zabránění ztráty dat, ať už chybou či fyzickým poškozením disku. Dané pojmy se poté liší například konkrétním způsobem provedení zálohy, či konkrétním důvodem pro použití daného procesu.

Synchronizace je obecnější pojem, který může být považován za nadmnožinou replikace. V případě, že existují dva datové zdroje, které je potřeba v daný okamžik sjednotit, je možno mluvit o synchronizaci souborů či datových složek. U souborů se shodným názvem se porovnává čas posledního zápisu, velikost nebo obsah souboru, naopak soubory, které shodu nezaznamenají, jsou jednoduše zkopírovány. Synchronizací se tedy dá proces označit v okamžiku, kdy existují nejméně dva datové zdroje a smyslem synchronizace je porovnat tato uložišť a dostat je do stejného stavu. To může například přispět snazší spolupráci více uživatelů nad stejnými daty nebo uživateli, který pracuje

na více počítačích.



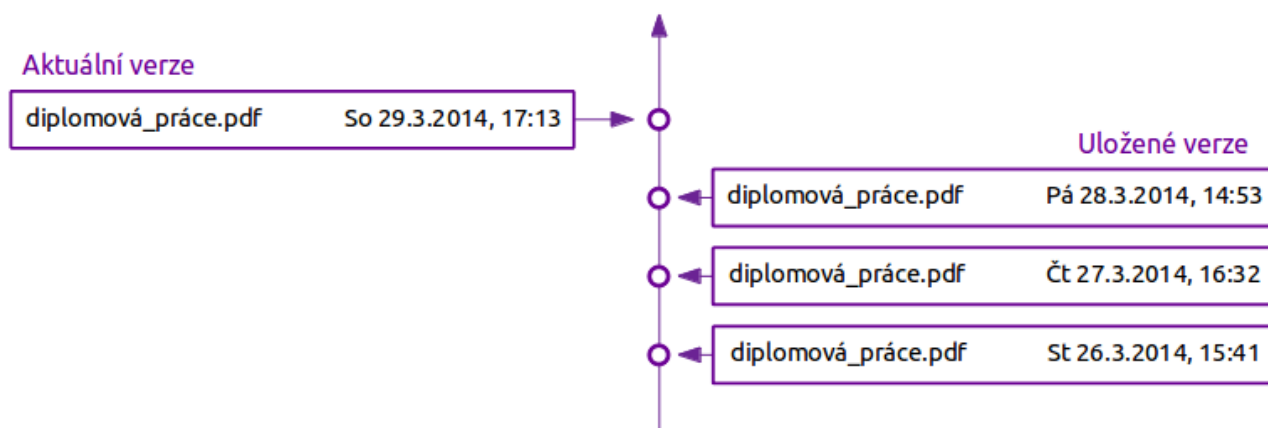
Obr. č: Příklad obousměrné synchronizace dat mezi dvěmi datovými uložišti

Replikace naopak začíná s daty existujícími pouze na jednom uložišti. Často je tento proces používán právě ve spojitosti s databázemi, kdy je kopie dat (také replika) tvořena z důvodu snížení zátěže serveru, či ochrany dat. Replikace je tedy často vyžadována z jiných důvodů než synchronizace a pro zajištění konzistence dat používá jiných technologií. V případě, že je již kopie vytvořena, je poté možno mluvit i o synchronizaci dat, protože replika průběžně kontroluje, zda na hlavním serveru nedošlo ke změně, a pokud ano, dané změny zkopíruje. Více se replikací zabývá kapitola \odkazKapitola{kReplikace} Replikace.

Oba procesy je možno použít jednostranně, tedy kopírovat data pouze z jednoho uložiště na druhé a nikoliv opačně, nebo oboustranně, kdy se datové zdroje kopírují navzájem mezi sebou.

Specifickým způsobem zálohy dat je verzování, kdy se data na záložním datovém uložišti nepřepisují, ale systematicky ukládají v takzvaných verzích tak, aby se uživatel mohl snadno kdykoliv vrátit k předchozím stavům souborů. Smyslem verzování je zachovat všechny zvolené stavy práce, čímž se verzování liší od zálohování, kde stačí mít aktuální kopii daných dat. To, co je zde popsáno jako verzování, se v produktech ArcGIS nazývá archivování dat \citep{Law2008}.

Verzování může probíhat ručně, poloautomatizovaně či plně automatizovaně díky speciálním nástrojům pro správu verzí, kterých je na internetu dostupná celá řada. Oblíbeným verzovací systémem programátorů je Git\footnote{více na <http://git-scm.com/>}, open-source nástroj pro správu verzí, který pomáhá při práci s malými i velkými projekty a podporuje týmovou spolupráci. Umožňuje vrátit jednotlivé soubory nebo celý projekt do předchozího stavu, porovnávat změny provedené v průběhu času, zjistit, kdo naposledy upravil něco, co nyní možná způsobuje problémy, kdo vložil jakou verzi a mnoho dalšího \citep{Chacon2009}. Git je vhodný zejména pro textové soubory, protože dokáže analyzovat části textu, či programového kódu a zvýraznit místa, která se změnila.

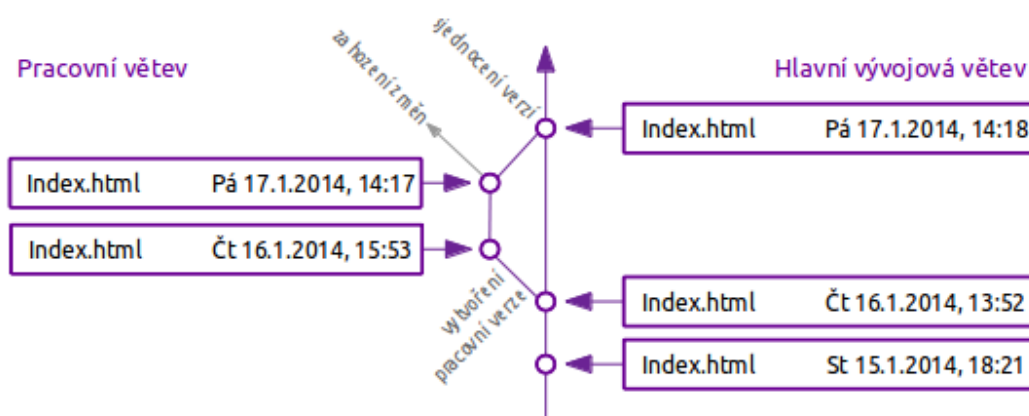


Obr. č: Příklad verzování souboru

Samotná databáze verzování dat neumožňuje. Nejsnazší způsob, jak získat verzi dat, je skriptem `\texttt{dump}`, který exportuje databázi do souboru. V MS SQL Serveru je tento proces nazýván Snapshot, tedy snímek databáze nebo také snímková replikace. Takový soubor se poté může verzovat podobným způsobem jako jakýkoliv jiný binární soubor typu shapefile. A to samé platí i pokud v databázi ukládáme geodata.

Proto byl vytvořen verzovací systém také pro prostorová data, který vychází ze systému Git a nese název GeoGIT. Umožňuje uživatelům uchovávat změny v souborech shapefile, SpatialLite a z databáze PostGIS (PostgreSQL), stejně tak jako vrátit se k jakékoliv z předchozích verzí.

Verzování může být chápáno také jako vytvoření pracovní verze. V případě, že programový kód či data jsou plně funkční či správná, ale je potřeba je aktualizovat, testovat či jinak měnit, pak je vhodné vytvořit tzn. pracovní verzi, aby nedošlo k poškození té aktuální. Jedná se o kopii aktuálního stavu, na které je možno pracovat a zkoušet. V případě, že práce nedopadne podle představ, je možno změny zahodit, pokud je tomu naopak, je možno pracovní verzi sjednotit s platnou verzí. Tento způsob verzování umožňuje Git i GeoGIT a takto chápe pojem verzování i společnost Esri.



Obr. č: Příklad verzování souboru s použitím pracovní větve

3.2 Replikace

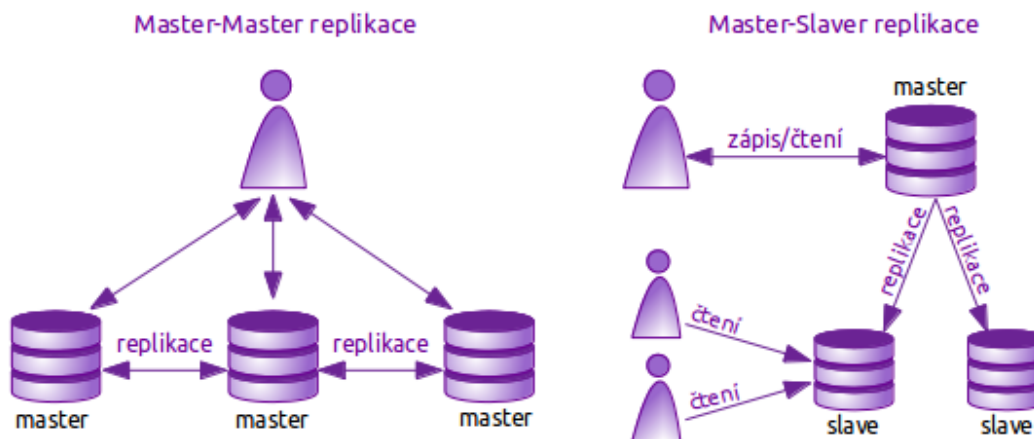
Replikace je proces, u kterého jsou data a databázové objekty kopírované z jednoho databázového serveru na druhý a poté synchronizovány pro zachování souladu obou databází. Synchronizací v tomto případě myslíme kopírováním všech změn, které v databázi nastanou. Použitím databáze je možno data distribuovat na různě vzdálená místa nebo mezi mobilní uživatele v rámci počítačové sítě a internetu \citep{Microsoft2013}.

Mnohé moderní aplikace se musí zabývat velkým počtem přístupů do databáze, což může v některých případech způsobovat problémy. Buď je server přetížen počtem připojení a data tedy přicházejí k uživateli pomalu, nebo dokonce úplně vypadne.

Mezi časté důvody použití databázové replikace tedy patří zajištění dostupnosti dat \footnote{angl. High Availability}, resp. snížení pravděpodobnosti, že data nebudou dostupná, což může být způsobeno již zmíněným výpadkem serveru nebo například fyzickou ztrátou dat \citep{ObeHsu2012}. Další důvodem je rozložení zátěže přístupů do databáze mezi více serverů, takže nebude docházet ke zpomalení výkonu hlavního serveru ani k situaci, že data nebudou dostupná kvůli jeho výpadku \citep{BellKindahlThalmann2010}. Databáze je často zálohovaná, například skriptem dump a i to může server zpomalit. Vhodným řešením je tedy nejdříve vytvořit kopii dat na jiný datový server a až poté proces zálohování spustit.

Všechny databáze zapojené do procesu replikace jsou v odborné literatuře nazývané uzly, angl. node. Tyto uzly dohromady tvoří replikační cluster \footnote{volně přeloženo jako skupina serveru zapojených do replikace}. Při správně nastavené replikaci, by v clusteru nikdy neměly být méně než tři uzly. Může se totiž stát, že vypadne jeden ze dvou uzlů, čímž dojde, ikdyž jen na krátkou chvíli, k situaci, že data nebudou v daný okamžik zálohovaná.

Uzly v replikačním clusteru mohou mít jednu ze dvou základních rolí, nejčastěji nazývaných master a slave. Master server nebo pouze master je server, který poskytuje data k replikaci, má práva na čtení i zápis a probíhají tedy na něm veškeré aktualizace. Je možno se setkat také s pojmenováním Primary server, Provider, Sender, Parent nebo Source server. Naprosto jiný pojem zavádí MS SQL Server, který tento zdrojový server nazývá Publisher (česky Vydavatel). Druhý databázový server je nejčastěji nazýván slave, Standby, Reciever, Child nebo Subscriber (česky Odběratel). Poslední pojem je také používán MS SQL Serverem. Na tento server, který je dostupný vždy jen pro čtení dat, se data a aktualizace kopírují, není však možné na něj změny zapisovat \citep{RiggsKrossing2010}.

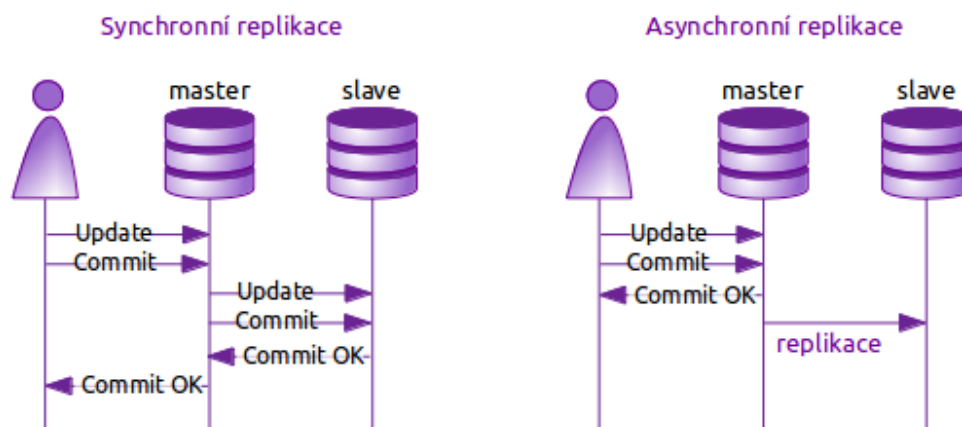


Obr. č: Srovnání master-master a master-slave replikace

Podle počtu master a slave serverů v replikačním clusteru, se rozlišuje zda se jedná o jednosměrnou nebo obousměrnou replikaci. Tzv. master-master replikace umožňuje zapisovat do všech uzlů v replikačním clusteru, což může být praktické například při použití databáze offline [\odkazObrazek{svrovnaniM-M-S}](#). Změny se tedy synchronizují mezi všemi databázovými uzly. Tento způsob však nese značné komplikace, je potřeba řešit konflikty změn ve stejných datech a je relativně náročný na údržbu. Tato práce se zabývá použitím druhé způsobu, tzv master-slave replikace. Tato replikace používá vždy jen jeden master server v clusteru a dva a více slave servery. Kopie dat tedy probíhá jednosměrně, vždy z master na slave servery. Podle Bella a kol. (2010) mají moderní aplikace často více čtenářů než zapisovatelů, proto je zbytečné, aby se všichni čtenáři připojovali na stejnou databázi jako zapisovatelé a zpomalovali tím jejich práci [\cite{BellKindahlThalmann2010}](#). Z toho důvodu je tedy použití master-slave replikace více než vhodné.

Při návrhu replikace je potřeba zamyslet se také nad tím, zda bude synchronní či asynchronní. Synchronní replikace neumožní, aby na master serveru proběhla nová transakce, dokud se poslední transakce úspěšně neprovede na slave serveru [\cite{Boszormenyi2013}](#). Tento přístup zajistí, že žádná data nebudou v průběhu transakce ztracena. V některých případech tento způsob může zbytečně zpomalit rychlost přístupu do databáze, protože je nutno čekat na každou nedokončenou transakci. Zároveň může způsobit snížení dostupnosti databáze, protože v případě, že se například přeruší spojení mezi servery, nemůže na masteru proběhnout žádný další dotaz. Ale jistě si najde své opodstatnění například při bankovních transakcích, kde je potřeba, aby všechny operace proběhly na obou stranách. V tomto případě je užití tohoto způsobu zcela nezbytné.

Druhým způsobem je asynchronní replikace, při které se nová data mohou zapisovat na master server, přestože ještě nedošlo k replikaci stávajících dat na slave server [\cite{ObeHsu2012}](#). To je sice za běžného provozu rychlejší, v některých případech však může způsobit nekonzistenci dat, například když proběhne transakce na master serveru, který však spadne dřív, než se změna zapíše na slave. V takovém případě se slave změní na master server, ale zároveň se nikdy nedozví o transakci, o které má uživatel informace, že proběhla v pořádku.



Obr. č: Rozdíl mezi synchronní a asynchronní replikací

pozn. Update představuje SQL příkaz pro vložení nebo aktualizace dat, Commit znamená potvrzení příkazu (typicky u transakcí) a Commit OK potvrzení úspěšného zapsání do databáze.

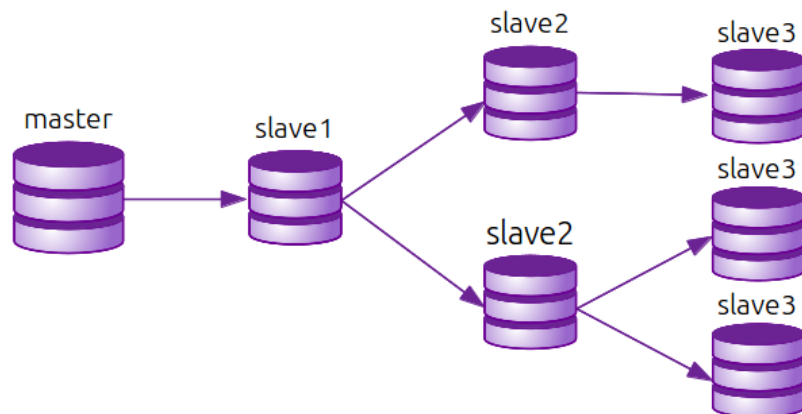
Replikace v PostgreSQL umožňuje plnou kopii dat z databáze i pouze výběr některých tabulek. Více o možnostech a způsobech nastavení replikace v kapitole [odkazKapitola}](#) a PRAKTICKÁ ČÁST :)

Dále je možno rozlišovat replikaci pole toho, zda je logická nebo fyzická. Výsledek obou typů má naprosto identický výsledek, přesto se mírně liší.

Fyzická replikace kopíruje data na druhý server v binární podobě. Tím, že se kopírují celé složky dat, je na slave serverech zajištěna identická replika. Protože se kopírují binární data, která mají jasné danou strukturu, je potřeba mít na obou serveru stejnou platformu a architekturu. Tento způsob je velice spolehlivý a často snazší na konfiguraci.

Naopak logická přenáší SQL příkazy tak, jak byly použity na master serveru a ty poté proběhnou na slave serverech. Tím se nasimuluje průběh změn dat na hlavním serveru a zajistí se konzistence dat. Tento způsob je více flexibilní, umožňuje výběr jen několika databází nebo tabulek a není závislý na architektuře ani operačním systému [Boszormenyi2013](#).

Posledním diskutovaným pojmem je kaskádová replikace, která umožňuje připojit repliku k jinému slave serveru místo k hlavnímu master serveru. Tento způsob může být výhodných především z těchto dvou důvodů. Řekněme, že se kaskádová replikace používá při existenci většího počtu slave serverů v clusteru, třeba sta. V případě, že by se všechny repliky připojovaly k hlavnímu serveru, došlo by u něj k razantnímu zpomalení jeho výkonu. Kaskádová replikace může být praktická také v okamžiku, kdy se data přenáší na velkou vzdálenost, třeba do Číny. V případě, že mají v Číně dvě repliky, je zcela zbytečné, aby se obě kopie přenášely na tak velkou vzdálenost, když druhá replika se může připojit k první a mít data s mnohem menším zpožděním.



Každý databázový server (myšleno SŘDB) si volí terminologii a konkrétní nastavení mírně odlišně. Tato kapitola se snaží popsat chápání replikace co v největší míře obecně s ohledem na použití tohoto pojmu v PostgreSQL. Zcela jinou terminologii, ikdyž založenou na stejných principech, zavádí MS SQL Server, který používá pojmy transakční replikace pro master-slave replikace a slučovací replikaci pro master-master replikaci.

výhody replikace arcsde

3.3 ArcGIS produkty

V názvu práce se objevuje spojení Esri platforma, čímž jsou chápány produkty společnosti Esri. Esri je americká společnost zabývající se vývojem software zaměřeného na geografické informační systémy. Manželé Dangermondovi ji založili v roce 1969^{footnote}{Více info <http://www.esri.com/about-esri/history>}.

Z hlediska chápání Esri má GIS tři roviny. První je to GIS jako prostorová databáze reprezentující geografické informace, dále sada map zobrazující prvky a vztahy mezi prvky na zemském povrchu a zároveň i software pro GIS jako sada nástrojů pro odvozování nových informací ze stávajících. Esri tyto tři pohledy na GIS propojuje v software ArcGIS jakožto kompletní GIS, který se skládá z katalogu (kolekce geografický datových sad), map a sad nástrojů pro geografické analýzy.

Esri vytváří integrovanou sadu softwarových produktů ArcGIS, které poskytují nástroje na kompletní správu GIS a přizpůsobují produkty různým úrovním nasazení. Výběr produktu záleží na tom, zda zákazník požaduje jedno nebo více uživatelských systémů, zda se má jednat o stolní systém nebo server, popř. zda má být dostupný prostřednictvím internetu. Nabízí také produkty vhodné pro práci v terénu \citep{Esri2006}.

Základními produkty^{footnote}{Názvy jednotlivých produktů použitých v tomto odstavci jsou platné od verze ArcGIS 10.1. Starší verze ArcGIS používají jiné názvy, jejichž přehled je možný na stránkách firmy ARCDATA Praha <http://www.arcdata.cz/produkty-a-sluzby/software/arcgis/prejmenovani-arcgis/>.} jsou stolní systémy ArcGIS for Desktop ve verzích Basic, Standard, Advanced^{footnote}{Zdroj

<http://www.esri.com/software/arcgis/about/gis-for-me>}, dále serverové verze ArcGIS for Server (pro Linux a Windows) ve třech úrovních funkcionality (Basic, Standard, Advanced) a dvou úrovních kapacity serveru (Workgroup a Enterprise). Další produkt ArcGIS for Mobile, ve verzích ArcPad, ArcGIS for Windows Mobile a ArcGIS for Smartphone and Tablet, je určený především pro práci v terénu. A v neposlední řadě verze dostupná skrze internet ArcGIS Online. K tomu všemu Esri přidává velké množství extenzí a další verzí^{footnote} {Kompletní seznam na oficiálních webových stránkách Esri <http://www.esri.com/products> nebo <http://www.arcdata.cz/produkty-a-sluzby/software/arcgis/>}.

Produkt	Verze		
ArcGIS for Desktop	Basic	Standard	Advanced
ArcGIS for Server	Basic	Standard	Advanced
ArcGIS for Mobile	ArcGIS for Windows Mobile	ArcPAD	ArcGIS for Smartphone and Tablet
ArcGIS Online			

Obr. č: Verze programu ArcGIS platné od verze 10.1.

Dle \cite{Law2008} je nativním formátem produktů ArcGIS geodatabáze a jsou rozlišovány tři druhy geodatabáze. Ani v jednom případě se však nejedná o databázi v pravém slova smyslu, tak jako ji chápeme v \odkazKapitola{PostgreSQL} a \odkazKapitola{MSSQL}. V každém případě však tyto způsoby umožňují uložení, přístup a správu dat. U prvních dvou typů, personální a souborové geodatabáze, se data ukládají do jednoho binárního souboru, kde jsou však ukládána ve stejné struktuře jako v plnohodnotném databázovém serveru. Do takového geodatabáze můžeme uložit více než jednu vrstvu, což je výrazný rozdíl oproti formátu shapefile. Výhodou je dále možnost použití relací, sofistikované dotazování a v neposlední řadě i snadná přenositelnost, protože takováto databáze bude vždy jen jeden soubor obsahující několik vrstev. Oproti tomu shapefile, který obsahuje jen jednu vrstvu, je tvořen minimálně 4 soubory. Oba tyto typy podporují pouze jednoho editujícího uživatele a mnoho uživatelů s právem čtení. Nepodporují dlouhé transakce ani verzování.

databáze		souborová .gdb ¹	personální .mdb ¹
datové uložení/ databázový server		lokální souborový systém	MS Access
licence		ArcGIS for Desktop (všechny verze)	ArcGIS for Desktop (všechny verze)
operační systém		Windows (možná i jiné)	Windows
požaduje ArcSDE		ne	ne
vlastní datový typ		ne	ne
víceuživatelská editace		ano, ale s limity	ne
počet editorů		1 pro každý dataset nebo tabulku ²	1 ²
počet čtenářů		více než 1 ²	více než 1 ²
replikace	typ replikace	souborová replikace	souborová replikace
	master server	ne ¹	ne ¹
	slave server	ano	ano
verzování		ne	ne
velikostní limity		1TB pro každý dataset ²	2GB ²

Obr. č: Přehled rozdílů personální a souborové geodatabáze používané programem ArcGIS

zdroje: 1. <http://www.esri.com/software/arcgis/geodatabase/single-user-geodatabase>
2. <http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#//003n0000007000000>

Tato práce se více zaměřuje na třetí typ, technologii ArcSDE, kterou v některých materiálech nazývají "geodatabáze ArcSDE". Nejedná se o geodatabázi, ale spíše o zprostředkovatele komunikace mezi programem ArcGIS a databázovým server. Umožňuje víceuživatelský přístup, verzování i replikaci \citep{Esri2006}. Tato technologie využívá jako datové uložení některý z již existujících databázových serverů, např. níže popsané PostgreSQL nebo MS SQL server. Touto technologií se více bude zabývat kapitola \odkazKapitola{ArcSDE} ArcSDE geodatabase.

3.4 Použité programové prostředky

3.3.1 PostgreSQL 9.x (PostGIS)

PostgreSQL je objektově-relační databázový systém s otevřeným zdrojovým kódem dostupný na většině platform. Je volně k dispozici pro použití, modifikaci a znovu rozšíření způsobem, který si sami zvolíme. Jedná se o robustní, výkonný, bezpečný, kompatibilní a interoperabilní software s podporou a dobře komentovaným zdrojovým kódem. Vyhovuje standardům SQL od verze SQL 2008 a nabízí velké množství pokročilých funkcí. PostgreSQL je založen na architektuře klient-server, to znamená, že server pořád běží a čeká na dotazy klienta \citep{Momjian2001}.

S vývojem databázového serveru PostgreSQL začala University of California v Berkley již více než před 20 lety. Nyní je vyvíjen a udržován velkou komunitou nezávislých vývojářů. Používá licenci TPL (The PostgreSQL Licence), která je mírně odlišná od open-source licence BSD (Berkeley Distribution Software), ze které vychází \citep{RiggsKrossing2010}.

Řadí se mezi nejpokročilejší databáze díky schopnosti pracovat s velkými objemy dat, díky své rychlosti a funkcionalitě může soupeřit i s populárními komerčními systémy jako je Oracle, IBM DB2, Microsoft SQL Server 2008 a dalšími \citep{PostgreSQL2012}.

Samotné PostgreSQL neobsahuje datové typy a funkce vhodné pro správu prostorových dat. K tomu je nutné přidat nástavbu PostGIS, která rozšiřuje databázi PostgreSQL o podporu geografických dat. PostGIS implementuje specifikaci „Simple Features for SQL“ konsorcia OGC. PostGIS umožňuje ukládání geometrických objektů (bod, linie, polygon), použití prostorových funkcí pro určení vzdáleností, délky linií, výměr a obvodu ploch, výběr indexu při spojení prostorových a atributových dotazů a mnoho dalších.

PostGIS používá dva základní prostorové datové typy *geography* a *geometry*. Typ *geography* ukládá souřadnice v kartézských rovinných souřadnicích, kterým odpovídá souřadnicový systém WGS84. Je zejména vhodný pro malá území. Při výpočtu vzdálenosti dvou bodů tento datový typ vrátí jako výsledek nejkratší vzdálenost v kilometrech v rovině. Typ *geometry* data ukládá v polárním rovinném systému a umožňuje nastavit souřadnicový systém podle potřeb. Výsledkem dotazu na vzdálenost dvou bodů tedy bude úhel ve stupních. Po převodu do metrické soustavy dostaneme nejkratší vzdálenost na kouli. Při výběru datového typu může být rozhodující například počet funkcí, kterých typ *geometry* poskytuje mnohem více než *geography*, nebo velikosti daného území \citep{OpenGeo2012}.

Existuje také další nástavba PostGIS Raster, která rozšiřuje ukládání a manipulaci s rastrovými daty, nástavba PostGIS Topology pro topologickou správu vektorových dat a pgRouting pro síťové analýzy. PostGIS je podporován velkou řadou software zabývajících se správou geografických dat, což také umožňuje snadnou přenositelnost a použitelnost jednotlivých nástaveb (příklad software podporujících PostGIS: QGIS, GvSIG, GRASS, ArcGIS).

PostGIS používá mnoho běžně používaných knihoven jako GEOS (Geometry Engine Open Source) pro implementaci jednoduchých prostorových prvků a metod pro topologii, PROJ4 pro převod mezi kartografickými projekcemi nebo GDAL/OGR (Geospatial Data Abstraction Library) pro převod mezi různými vektorovými i rastrovými formáty \citep{ObeHsu2011}. PostGIS 1.5. obsahovala přes 800 funkcí, typů a prostorových indexů \citep{ObeHsu2012}. Aktuální verze PostGIS\footnote{Aktuálně na <http://postgis.refractory.net/>} je 2.1.

PostgreSQL podporuje replikaci i synchronizaci bez nutnosti další instalace. - funguje RS jen změnou konfigurační soubor po instalaci, nebo je potřeba ovlivnit už před instalací? Replikace v PostgreSQL - co umí, jaké má nástroje

Od verze ArcGIS 9.3. je PostgreSQL oficiálně podporovanou databází pro ukládání geodat v produktech ArcGIS. Při instalaci je pouze potřeba zajistit kompatibilitu verzí. Pro verzi ArcGIS 10.1 jsou podporované verze PostgreSQL 9.0 a PostGIS 1.5., pro ArcGIS 10.1 SP1\footnote{Service Pack 1} je to PostgreSQL 9.1.3 a PostGIS 2.0 \citep{OSGEO2013}\footnote{Zdroj a další informace na stránkách PostgreSQL <http://trac.osgeo.org/postgis/wiki/UsersWikiPostgisarcgis> nebo ArcGIS <http://resources.arcgis.com/en/help/system-requirements/10.1/index.html #/015100000075000000>}. Databáze PostgreSQL se dá v ArcGIS produktech použít dvojím způsobem. Buď jen jako úložiště dat bez přidání geografického datového typu, nebo včetně datového typu, tedy včetně PostGIS knihovny. ArcSDE podporuje pouze datový typ PostGIS Geometry a přidává vlastní datový typ Esri St_Geometry. Výhodou používání Esri St_Geometry je nezávislost na zvoleném databázovém systému, tedy snazší přenositelnost celého řešení.

Práce byla testována na verzích PostgreSQL^{footnote}{Více na <http://www.postgresql.org/>} 9.1.4 a PostGIS 2.0.

3.4.2 Microsoft SQL Server Express 2008

Microsoft SQL Server (dále MS SQL Server) je relační databázový systém vyvíjený společností Microsoft dostupný pro různé verze operačního systému Windows. Dodává se v mnoha verzích, které lze nainstalovat na různé hardwarové platformy na základě odlišných licenčních modelů \citep{Whalen2008}. Podle Leitera (2009) SQL Server nabízí 8 základních verzí: Enterprise, Standard, Workgroup, Web, Express, Express Advanced Edition, Developer Edition a Compact Edition. Enterprise edition podporuje naprosto vše, co SQL Server nabízí, naopak verze Express, která je dostupná zdarma, obsahuje omezení některých funkcí a proto je vhodná spíše pro malé nebo začínající projekty \citep{Leiter2009}.

Prostorová data jsou implementována jako CLR rozšíření a přidávají databázovému serveru dva prostorové datové typy geometry a geography. Rozdíl mezi datovými typy je podobný jako u PostgreSQL. První jmenovaný slouží k reprezentaci dat (bodů, linií, polygonů) v rovině, naproti tomu datový typ geography slouží ukládání stejných dat na povrchu zeměkoule. Oba typy pracují ve dvou dimenzích (nebere se v potaz výška). Podporuje také indexování dat, index je tvořen standardním B stromem \citep{Cincura2009}.

SQL Server je podporován a používán ArcGIS produkty od začátku jeho vývoje. Verze ArcGIS Enterprise může být propojena s jakoukoliv uživatelem zvolenou a zakoupenou licencí databázového systému. Verze ArcSDE Desktop a Workgroup používají verzi Express, která je dostupná zdarma a podporuje většinu základních funkcí. Replikaci plně podporuje verze Enterprise, ostatní verze ji podporují pouze s omezenými funkcemi. Avšak již zmiňovaná verze Express, která je podporována ArcSDE Desktop a Workgroup, může být použita pouze slave server, tedy odběratelem replikovaných dat, není tedy možné do takovéto databáze připojené do replikačního clusteru zapisovat. Nemůže být tím, kdo poskytuje data k replikaci \citep{Whalen2008}. Stejně jako u PostgreSQL platí, že si uživatel může zvolit, zda použije datový typ, který je součástí ArcSDE, nebo ten, který je implementován do SQL Serveru.

- Replikace v MSSQL - co umí, jaké má nástroje?
- SQL omezeně používá souřadnicové systémy?
- OGR?

3.4.3 ArcSDE geodatabase

ArcSDE je technologie firmy Esri pro správu geoprostorových dat uložených v relačních databázových systémech. Jedná se o otevřenou a interoperabilní technologii, která podporuje čtení a zápis mnoha standardů. Využívá jako své nativní datové struktury standard konsorcia OGC Simple Feature a prostorový typ ISO pro databázové systémy Oracle, IBM DB2 a Informix. Poskytuje vysoký výkon a je přizpůsobena velkému počtu uživatelů \citep{Esri2006}.

ArcSDE je prostředník pro komunikaci mezi klientem (př. ArcView) a SQL databází (př. PostgreSQL). Umožňuje přístup a správu dat v databázi, současnou editaci jedné databáze více

uživatelé, zajišťuje prostorový datový typ (St_Geometry), dále integritu dat, dlouhé transakce a práci s verzemi \citep{Law2008}.

Technologie ArcSDE vyžaduje dvě úrovně: databázovou a aplikační, která se skládá z ArcObjects a ArcSDE. Databázová úroveň zajišťuje jednoduchý, formální model pro uložení a správu dat ve formě tabulek, definici typů atributů (datových typů), zpracování dotazů či víceuživatelské transakce \citep{Law2008}. ArcSDE podporuje databázové systémy IBM DB2, IBM Informix, Oracle, Microsoft SQL, PostgreSQL \citep{Esri2013a}.

Existují tři úrovně ArcSDE databáze: desktop (ArcSDE Desktop), skupinová (ArcSDE Workgroup) a podniková (ArcSDE Enterprise). Každá verze má jiné parametry a umožňuje různou úroveň editace, \odkazTabulka{sde}.

databáze		ArcSDE		
		Desktop	Workgroup	Enterprise
datové uložště/ databázový server		SQL Server Express	SQL Server Express	PostgreSQL, Oracle, SQL Server a další
licence		ArcGIS for Desktop (všechny verze)	ArcGIS for Server Workgroup	ArcGIS for Server Enterprise
operační systém		Windows	Windows	všechny operační systémy
požaduje ArcSDE		ano	ano	ano
vlastní datový typ		ne	ne	ano
víceuživatelská editace		ne	ano	ano
počet editorů		1	10	bez limitu
počet čtenářů		3	10	bez limitu
replikace	typ replikace	databázová replikace	databázová replikace	databázová replikace
	master server	ne	ne	ano
	slave server	ano	ano	ano
verzování		ano	ano	ano
závislost na sítích		lokální síť	lokální síť, internet	lokální síť, internet
velikostní limity		10GB	10GB	záleží na velikosti serveru

Obr. č: Přehled verzí ArcSDE, jejich parametrů a možností

zdroj: <http://www.esri.com/software/arcgis/geodatabase/multi-user-geodatabase>

Od verze ArcGIS 9.2 je ArcSDE Desktop spolu s databázovým systémem MS SQL Server Express součástí licence produktů ArcGIS for Desktop Standard a Advanced. Takovou databázi mohou současně používat 4 uživatelé, z toho jen jeden může databázi editovat, jsou však omezeni velikostí databáze.

Součástí licence ArcGIS for Server Workgroup je ArcSDE Workgroup, která se liší od verze Desktop především tím, že počet uživatelů, kteří mohou současně editovat nebo prohlížet databázi, je zvýšen na deset.

Nejvyšší úroveň, ArcSDE Enterprise, je možno získat s licencí ArcGIS for Server Enterprise, která uživatelům přináší nejméně omezení. Mohou si vybrat z několika komerčních i nekomerčních

databázových systémů, počet uživatelů není omezen, stejně jako velikost databáze.

K ArcSDE a vybrané databázi je možno přistupovat přes ArcCatalog, není tedy potřeba instalace dalšího software nebo zkušenost s administrací databáze \citep{Esri2006}.

Replikaci a synchronizaci dat umožňují pouze ArcSDE Enterprise a Workgroup \citep{Esri2013b}. Jak už bylo zmíněno v předchozí kapitole \odkazKapitola{MSSQL} Microsoft SQL Server Express 2008, SQL Server Express je možný použít v replikačním clusteru pouze jako slave server. Vzhledem k tomu, že proces replikace je implementován při do ArcObjects a ArcSDE, nezáleží na konkrétním databázovém systému \citep{Law2008}.

Správa transakční geodatabáze - Esri 2006 str. 109
--

3.5 Nástroje pro replikaci v PostgreSQL

PostgreSQL nabízí hned několik nástrojů pro řešení replikace. Je možno použít zabudovanou streaming replikaci, která je dostupná od verze PostgreSQL 9.0 nebo některou z extenzí, například Slony-I, pgpool, Skytools nebo Postgres-XC. Tato kapitola se dále bude zabývat a porovnávat nativní streaming replikace s extenzí Slony-I a pgpool.

3.5.1 Slony-I

Jak píše \cite{Boszormenyi2013} je Slony-I jeden z nejrozšířenějších externích nástrojů pro replikaci pro PostgreSQL. Zároveň se také řadí mezi nejstarší, plně používán je v PostgreSQL již od verze 7.3. a je velmi dobře podporován i dalšími externími řešeními pro PostgreSQL, například programem PgAdmin3, který nabízí správu dat pomocí grafického rozhraní \citep{Boszormenyi2013}.

Jedná se o trigger-based replikaci, což znamená, že je ke každé existující tabulce přidán trigger, který zajistí replikaci každé změny, která v databázi nastane. Z toho také vyplývá, že se jedná o logickou replikaci, kdy je možné replikovat pouze změny v datech, tzv. DDL změny, tedy SQL příkazy INSERT a UPDATE, nikoli strukturu databáze, příkazy typu CREATE/DROP TABLE, ALTER TABLE. Každá změna struktury se tedy musí provést ručně, což se může jevit jako nevýhodné. Nese to ale i své klady, například možnost výběru pouze některých tabulek. Vytváří si totiž tzv. *replikační set*, do kterého se zapíše pouze ty tabulky, které je potřeba replikovat.

Další výhodou, a to zvláště v porovnání se streaming replikací, je možnost replikace dat mezi různými verzemi PostgreSQL bez ohledu na platformu a architekturu. Naopak spíše za nevýhodu je považováno, že si vytváří ke každé tabulce vlastní schéma, do kterého se ukládají replikovaná data, což způsobuje redundanci dat.

Slony-I replikace je z principu asynchronní, zpoždění je v řádu vteřin nebo v desítkách. Umožňuje Hot Standby mode, kdy je možno použít repliku na dotazy, i kaskádovou replikaci. Slony-I má vlastní konfigurační nástroj a samotná replikace funguje díky vlastnímu replikačnímu démonu, který běží stále, registruje změny a kopíruje je na slave servery.

3.5.2 Streaming replikace

Streaming replikace je nativní řešení do PostgreSQL implementováno od verze 9.0. Jedná se o log-shipping replikaci, což znamená, že jsou změny zapsány nejdříve vždy do transakčního logu v PostgreSQL nazvaného WAL (Write Ahead Log) přímým zápisem na disk a až poté potvrzeny jako úspěšné. Tento způsob zajišťuje datům naprosté bezpečí, protože kdyby došlo k chybě a změny se nezapisovaly na disk, ale pouze do cache, mohlo by dojít k jejich ztrátě. Zároveň to zajišťuje kopii jak dat, tak i struktury databáze. Existuje pouze jeden transakční log pro jednu instalaci PostgreSQL, proto se replikují vždy všechny databáze a není možné výběru jen několika tabulek, tak jako u Slony-I \citep{Boszormenyi2013}. Protože replikace probíhá pomocí transakčního logu, je nutné použití stejné verze PostgreSQL, stejné platformy i architektury na všech uzlech replikačního clusteru.

Streaming replikace umožňuje jak synchronní, tak asynchronní replikaci, dále Hot standby mode i kaskádovou replikaci.

schémata výhody x nevýhody

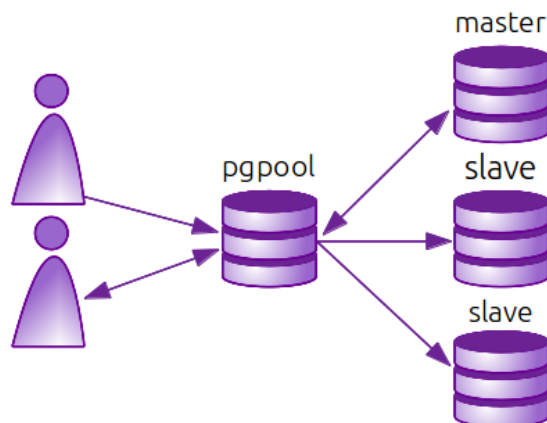
3.5.3 pgpool

Nástroj pgpool, který je stejně jako Slony-I extenzí pro PostgreSQL, je dalším z nástrojů, které je možno v použít pro replikaci dat, umožňuje však i mnohé další funkce. Je prostředníkem pro komunikaci mezi klientem a PostgreSQL serverem a jeho hlavní úlohou je zvýšení efektivity a rychlosti práce s databází.

Jednou ze základních výhod použití pgpool je možnost sdílení spojení klienta s databází, což v praxi znamená, že se vytvoří několik spojení, která i po skončení dotazu zůstanou otevřená a připravená pro další použití. Nemusí se tedy navazovat spojení při každém požadavku, ze strany klienta, což velice zrychlí provoz a zajistí plynulost užívání databáze. Umožňuje také paralelní dotazování, tedy složitý dotaz rozdělí mezi více uzlů, což velice sníží čas vykonání daného dotazu.

Zároveň je nástrojem pro optimalizaci nastavení replikace, která je potřebná hned z několika důvodů \citep{pgpool2014}. V případě, že by v replikačním clusteru bylo třeba deset různých serverů, aby nedošlo k přetížení, bylo by nutné dát každému uživateli přístup k jinému serveru, nebo přístupy do databáze manuálně rozkládat skrze složité programové řešení. pgpool tohle vše zajišťuje sám.

Navenek se jeví jako jakákoliv jiná databáze, do které se připojí všichni uživatelé bez ohledu na jejich práva či požadavky. On poté rozděljuje dotazy mezi uzly v replikačním clusteru dle aktuální zátěže \citep{Boszormenyi2013}. Zároveň, pokud má uživatel přístup k zápisu i čtení, umí na základě jeho aktuálního SQL příkazu, rozhodnout, zda jej připojí k master nebo slave databázi, \odkazObrazek{pgpool-obr}. Tento způsob velmi zjednoduší jak administraci databáze, tak nastavení pro běžného uživatele, který se připojí pouze do jedné databáze a víc se nestará, zda z databáze pouze čte, nebo do ní i zapisuje.



Obr. č: Zjednodušené schéma pgpool

4 NÁVRH A IMPLEMENTACE REPLIKACE

Po řešeršní části byl pro nasazení replikace zvolen databázový server PostgreSQL ze dvou hlavních důvodů. Jedná se o plnohodnotný databázový systém dostupný zdarma se všemi nástroji, je multiplatformní a od verze ArcGIS 9.3 je plně podporován produkty ArcGIS. Dále byla zvolena jednosměrná master-slave replikace ve variantách synchronní, asynchronní i kaskádová.

Tato kapitola podrobně popisuje zvolené řešení, které bylo vytvořeno dle potřeb katedry, a konfiguraci vybraných nástrojů. Všechny procesy byly prakticky otestovány.

Vycházelo se z předpokladů, že katedra ... viz rámeček

Současný stav správy dat (databází) na katedře.

Data a uživatelé dat → z toho se bude vycházet při tvorbě replikačního řešení.

Možnosti a technické vybavení katedry.

schéma současného stavu

Z toho vyplývá: 4.1 Návrh replikačního řešení

4.1 Návrh replikačního řešení pro katedru

schéma, popis a vysvětlení výběru

4.2 Příprava prostředí pro konfiguraci replikace

Na začátku bylo potřeba připravit testovací prostředí hned s několika závislostmi. Hlavní používaný software je PostgreSQL s extenzemi PostGIS, Slony-I a pgpool. Informace o instalacích jednotlivých komponent jsou dostupné na jejich webových stránkách, ve Windows si stačí stáhnout instalační balík PostgreSQL, který umožňuje instalaci databázového systému včetně se všech výše zmíněných extenzí. Pro grafickou administraci databáze je doporučený, ale ne povinný, program PgAdminIII, který je také multiplatformní. Většina konfigurace zde bude popisována skrze příkazový řádek, neznamená to však, že nemá ekvivalentní použití skrze grafické rozhraní.

Všechny technologie byly testovány na Debian-based Linux, Slony-I také na Windows XP.

U všech typů replikace je potřeba mít databázového uživatele s právem pro replikaci, pod kterým bude daný proces probíhat. Je možné vytvořit nového uživatele a nastavit mu tato práva nebo použít již existující účet `postgres`, který jako SUPERUSER obsahuje také práva pro replikaci. Je však potřeba mu hned na začátku po instalaci změnit heslo.

Změna hesla uživatele postgres

```
ALTER ROLE postgres password 'kgigis';
```

Vytvoření nového uživatele

```
ADD ROLE replikace REPLICATION;  
ALTER ROLE replikace ENCRYPTED PASSWORD 'kgigis';
```

Předtím, než k dojde samotné konfiguraci replikace, je potřeba vytvořit totožnou repliku dat hlavního serveru. U systémů, které nejsou stejné, není možné dosáhnout totožné kopie na úrovni binárních data, protože adresářové struktury se mohou lišit. V takovém případě je možné zajistit kopii dat například skriptem `\texttt{pg_dump}` na master a `\texttt{pg_restore}` na slave serveru.

Příklad exportu dat z databáze dat skrze:

```
pg_dump > /tmp/dump.sql  
pg_restore /tmp/dump.sql
```

Naopak, pokud se má jednat o kopii dat a konfiguračních souborů mezi dvěma databázemi stejného systému a architektury, je možné kopírovat celou složku, tedy přímo binární data. Je mnoho způsobů, jak toho dosáhnout, například klasickým kopírováním skrze skript `\texttt{cp}`, resp. `\texttt{scp}` u vzdálených složek, nebo skriptem `\texttt{rsync}`. Kopírování dat za běhu databáze ještě vyžaduje použití příkazu `\texttt{SELECT pg_start_backup}`, který zajistí, že bude archivovat transakční log do té doby než proběhne příkaz `\texttt{SELECT pg_stop_backup}`. Tím nepříjdeme o žádné změny.

Příklady možných způsobu zkopírování dat na repliku:

```
SELECT pg_start_backup('backup', true)
```

```
a) scp -rv root@158.194.94.93:/var/lib/postgresql/9.1/main  
/var/lib/postgresql/9.1/main  
nebo
```

```
b) rsync -ave ssh root@158.194.94.93:/var/lib/postgresql/9.1/main  
/var/lib/postgresql/9.1/main
```

```
SELECT pg_stop_backup()
```

Další možností je použití skriptu určeného pro zálohování dat v PostgreSQL `pg_basebackup`. Tento příkaz mimo jiné umožňuje kopírování dat za běhu replikace.

Použití pg_basebackup pro vytvoření repliky:

```
pg_basebackup -D /var/lib/postgresql/9.1/main/ -U replication -h 158.194.94.93
```

Kopie souborové struktury dat je potřeba především pro streaming replikaci. Tento krok je velice důležitý pro správný chod replikace. V případě, že se data nespávně zkopírují, není možné replikaci zprovoznit.

Pokud se začíná s prázdným databázovým systémem, je kopírování dat vždy velmi rychlé a je tedy vhodné nastavit replikaci dřív, než se data začnou přidávat do databáze. V případě, že už databáze naplněná daty je, není problém replikaci nastavit, jen je třeba počítat s delším časem kopírování dat a větší opatrností při konfiguraci.

Aby bylo možné pracovat s databází, je nejdříve nutné chápat význam jednotlivých konfiguračních souborů a mít přehled o souborové struktuře PostgreSQL. Vzhledem k tomu, že si ji každý systém uzpůsobuje podle sebe, nezbyvá než po instalaci PostgreSQL nastudovat, kde se jaký soubor nachází. Naštěstí existuje tabulka `\texttt{pg_settings}`, která uchovává veškeré informace o nastavení databáze.

Příklad SQL příkazu, spuštěného na serveru geohydro, který vypíše umístění jednotlivých souborů a složek:

```
SELECT name, setting FROM pg_settings WHERE category = 'File Locations';
```

name	setting
data_directory	/var/lib/postgresql/9.1/main
external_pid_file	/var/run/postgresql/9.1-main.pid
hba_file	/etc/postgresql/9.1/main/pg_hba.conf
config_file	/etc/postgresql/9.1/main/postgresql.conf
ident_file	/etc/postgresql/9.1/main/pg_ident.conf

Jak je možno vidět, existují tři hlavní konfigurační soubory:

- *postgres.conf*, který ovládá obecné nastavení jako výchozí uložště, kterým IP databáze naslouchá, velikost alokované paměti a další,
- *pg_hba.conf*, který povoluje konkrétním uživatelům přístup z určitých IP adres,
- *pg_ident.conf*, který slouží k mapování uživatel operačního systému na uživatele PostgreSQL `\citep{ObeHsu2012}`.

Je vhodné zajistit také práva k jednotlivým složkám a souborům. Vzhledem k tomu, že databáze zapisuje do složky s daty (*data_directory*), musí mít postgres, i po zkopírování celé datové struktury na jiný server, práva pro zápis.

A v neposlední řadě je potřeba zajistit konektivitu obou, resp. všech serverů v replikačním clusteru. S tím souvisí i nutnost nastavení povolení přístupu z IP adresy slave serverů, kterou je možno zajistit skrze konfigurační soubor *pg_hba.conf*. Následující příklad ukazuje možné nastavení souboru *pg_hba.conf* na master serveru. Povoluje uživatelům *market* a *replication*, přihlášených z dané IP adresy, přistupovat na master server a číst, resp. replikovat data.

# host	DATABASE	USER	ADDRESS	METHOD
host	all	market	80.188.74.1/32	md5

4.3 Připojení produktů ArcGIS k databázi PostgreSQL

4.4 Implementace replikace

4.4.1 Slony-I replikace

Inicializace

Jak už bylo zmíněno v kapitole [odkaz](#) Kapitole{Slony}, Slony-I neumožňuje replikovat strukturu databáze, proto je vhodné mít na začátku připravenou databázi tak, aby do její struktury později ideálně vůbec nemuselo zasahovat. Je tedy možné si vytvořit úplně novou databázi nebo použít již existující databázi, možno i naplněnou daty. Poté je nutno připravit druhý server do stejného stavu a to ideálně pomocí skriptu dump na master a pg_restore na slave serveru.

Pak se můžeme přihlásit k odběru replikačního setu.

- omit copy - vyvaruje se toho, že smaže všechno na slavu a začne kopírovat znovu
- umožňuje swichover, musí se všechno přerušit a pak příkazy log set a move set servery
- prohodit
- umožňuje failover - přesunutí mastera na slave (příkaz failover - povýšení slave databáze na master)

Pro názornost vytvoříme dvě identické databáze a tabulky na obou serverech. Názvy databáze můžou a nemusí být stejné. Pokud je databáze replikována v rámci stejné instalace PostgreSQL, je potřeba, aby názvy byly odlišné.

Vytvoření databáze stejného názvu na obou serverech pomocí SQL příkazu CREATE DATABASE:

```
postgres=# CREATE DATABASE repl1;  
CREATE DATABASE
```

Slony-I nabízí kopii jen výběru několika tabulek z databáze. Ty však musí mít na začátku inicialice replikace stejný název, stejnou strukturu a stejná data. Pokud už tabulka existuje, je možno ji aplikovat na druhý server pomocí skriptu dump na master a pg_restore na slave serveru. Důležité je, aby obě tabulky měly *primary key*, bez toho replikace nemůže probíhat.

Vytvoření tabulek stejného názvu na obou serverech pomocí SQL příkazu CREATE TABLE:

```
repl1=# CREATE TABLE repl_name (id serial, name text, primary key(id));  
NOTICE: CREATE TABLE will create implicit sequence "repl_name_id_seq" for  
serial column "repl_name.id"  
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index  
"repl_name_pkey" for table "repl_name"  
CREATE TABLE
```


Slony-I má vlastní konfigurační jazyk, pomocí kterého se nastavují konkrétní požadavky na replikaci. Na začátku se díky němu nastaví inicializace replikace, později je možno jejím prostřednictvím přidat do replikace další tabulku, či změnit strukturu databáze. Dále existují dva skripty *slonik*, který se vždy spouští jednorázově a vykonává požadavky definované v konfiguračním souboru, a *slon*, musí být spuštěn stále na masteru i slavu, pokud chceme, aby replikace běžela.

První skript spuštěný skriptem *slonik* na masteru vytvoří cluster (init cluster) a další na slave přidá slave server k existujícímu clusteru.

Založení souboru pro inicializaci clusteru (ini cluster) *init_master.txt* v */etc/postgresql/9-1/main* (na master):

```
cluster name = second_cluster;
# define nodes
node 1 admin conninfo = 'dbname=repl1 host=localhost user=postgres
password=tfgt';
node 2 admin conninfo = 'dbname=repl2 host=localhost user=postgres
password=tfgt';

# init cluster
init cluster ( id=1, comment = 'master Node');

# group tables into sets
create set (id=1, origin=1, comment='Repl tables');

set add table (set id=1, origin=1, id=1, fully qualified name =
'public.repl_names', comment='sample table');

store node (id=2, comment = 'slave node', event node=1);

store path (server = 1, client = 2, conninfo='dbname=repl1 host=localhost
user=postgres password=tfgt');
store path (server = 2, client = 1, conninfo='dbname=repl2 host=localhost
user=postgres password=tfgt');

store listen (origin=1, provider=1, receiver=2);
store listen (origin=2, provider=2, receiver=1);
```

Parametr *create set* zakládá tzn. replikační set, do kterého se poté příkazem *set add table*, přidávají tabulky, které chceme replikovat.

Spuštění skriptu *slonik* pro inicializaci clusteru:

```
root@gis01:/etc/postgresql/9.1/main# slonik master.txt
```

Vytvoření souboru pro přidání serveru (*subscribe set*) do existujícího clusteru *init_slave.txt* v */etc/postgresql/9-1/main* (na slave):

```

cluster name = second_cluster;

# define nodes

node 1 admin conninfo = 'dbname=repl1 host=localhost user=postgres
password=tfgt';
node 2 admin conninfo = 'dbname=repl2 host=localhost user=postgres
password=tfgt';

#Nastavení závislosti provider=master=node1, receiver=slave=node2
try {
    subscribe set (id=1, provider=1, receiver=2, forward=yes);
}
on error {
    exit 1;
}
echo 'Subscribed nodes to set 1';

```

Spuštění skriptu slonik pro přidání uzlu do clusteru:

```

root@gis01:/etc/postgresql/9-1/main# slonik slave.txt
slave.txt:15: Subscribed nodes to set 1

```

To, že se vytvořit cluster a tabulky do něj byla přidána, jde vidět v výpisu tabulky, kde nově přibýly trigger, které sledují změny, které v tabulce nastanou.

```

repl1=# \d repl_test
               Table "public.repl_test"
  Column      | Type          | Modifiers
-----+-----+-----
 id           | integer       | not null default
nextval('repl_test_id_seq'::regclass)
 name        | text          |

```

Indexes: "repl_test_pkey" PRIMARY KEY, btree (id)
Triggers: _first_cluster_logtrigger AFTER INSERT OR DELETE OR UPDATE ON
repl_test FOR EACH ROW EXECUTE PROCEDURE
_first_cluster.logtrigger('_first_cluster', '1', 'k')
Disabled triggers: _first_cluster_denyaccess BEFORE INSERT OR DELETE OR
UPDATE ON repl_test FOR EACH ROW EXECUTE PROCEDURE
_first_cluster.denyaccess('_first_cluster')

v pgAdmin se podívat, jak se vytvořily Nody, pokud nějaký chybi, je třeba doplnit, až pak spouštět slon

Když je takto vytvořen cluster a všechny repliky jsou do něj přidány, je možno spouštět skript slon, který zajišťuje samotnou replikaci. Spustíme ho následujícím příkazem, kde za slovem slon následuje

název clusteru, který byl definován hned v prvním řádku konfiguračních souborů init_master.txt a init_slave.txt, dále v uvozovkách IP počítače, název databáze a uživatelské jméno, pod kterým probíhá replikaci. Je důležité, aby po spuštění nevypisovalo žádné chyby, jinak někde nastala chyba a je potřeba zkontrolovat všechny parametry konfiguračních souborů.

```
root@gis01:/etc/postgresql/9.1/main# slon second_cluster 'host=localhost
dbname=repl1 user=postgres'
root@gis01:/etc/postgresql/9.1/main# slon second_cluster 'host=localhost
dbname=repl2 user=postgres'
```

spousteni ze slozky /etc/postgresql/9.1/main !!

Kontrolu toho, že replikace správně běží je možno otestovat tak, že se pokusíme přidat nový prvek na slave server a on musí zahlásit chybu, že je replikován a není tedy možné přidávat záznamy, nebo přidáním dalšího záznamu na master a on by se měl hned zkopírovat na slave.

```
repl2=# INSERT INTO repl_names (name) VALUES ('Rob');
ERROR:  Slony-I: Table repl_names is replicated and cannot be modified on a
subscriber node - role=0
```

```
pozn.:
pg_hba.conf:
pridani host    all          all          127.0.0.1/32      trust
-- nemyslim si, ze je potreba
```

6) when you're working from root, you should copy .pgpass to /root/ folder

```
cp /home/webgis/.pgpass /root/
```

```
.pgpass pridán vsude .. do /etc/postgres/9-1/main i do /root/ .. práva 0600
```

Přidání další tabulky do replikace

Přidání další tabulky či jakákoliv změna struktury je trochu složitější. Nejdříve je potřeba vytvořit soubor, který bude obsahovat SQL příkaz, který chceme provést, a poté spustit skript slonik, který zavolá soubor s SQL příkazem a vykoná jej na všech uzlech clusteru.

Vytvoření souboru deploy_ddl.sql ve složce /tmp/:

```
CREATE TABLE repl_address (id serial, name text, primary key(id));
```

Vytvoření skriptu slony_dll.txt, který umožní přidání tabulky za chodu replikace:

```
cluster name = second_cluster;
# define nodes
node 1 admin conninfo = 'dbname=repl1 host=localhost user=postgres'
```

```
password=tfgt';
node 2 admin conninfo = 'dbname=repl2 host=localhost user=postgres
password=tfgt';

execute script (
  SET ID = 1,
  filename = '/tmp/deploy_ddl.sql',
  event node = 1
);
```

Spuštění skriptu slonik, pro vykonání souboru, který volá SQL příkaz. Skript vypíše podrobný výpis, který oznamuje, že se tabulka vytvořila.

```
root@gis01:/etc/postgresql/9-1/main# slonik slony_ddl.txt
DDL script consisting of 2 SQL statements
DDL Statement 0: (0,67) [ CREATE TABLE repl_address (id serial, name text,
primary key(id));]
slony_ddl.txt:6: NOTICE: CREATE TABLE will create implicit sequence
"repl_address_id_seq" for serial column "repl_address.id"
slony_ddl.txt:6: NOTICE: CREATE TABLE / PRIMARY KEY will create implicit
index "repl_address_pkey" for table "repl_address"
DDL Statement 1: (67,69) [
]
Submit DDL Event to subscribers...
```

Takto se tabulka přidá do databáze, nikoliv však do replikačního clusteru. K tomu je potřeba vytvořit další soubor a vykonat ho skriptem slonik.

Vytvoření souboru pro přidání tabulky do replikačního clusteru add_to_set.txt:

```
#přidava do cluster second_cluster
cluster name = second_cluster;
# define nodes
node 1 admin conninfo = 'dbname=repl1 host=localhost user=postgres
password=tfgt';
node 2 admin conninfo = 'dbname=repl2 host=localhost user=postgres
password=tfgt';

#definuju nový set (id 2)
create set (id=2, origin=1, comment='Repl tables 2');

set add table (set id=2, origin=1, id=3, fully qualified name =
'public.repl_address', comment='Second table');

#přidám set (id=2)
subscribe set (id=2, provider=1, receiver=2);
#spojuji set id2 k setu id1
```

```
merge set(id=2, add id=1, origin=1);
```

Spuštění skriptu slonik

```
root@gis01:/etc/postgresql/9-1/main# slonik add_to_set.txt
```

Podobně je možné provést také smazání tabulky pomocí parametru DROP SET.

4.4.2. Streaming replikace

- Failover - pg_ctl promote, v ten okamžik se ukončí aplikování logu a zápis do databáze, nebo pomocí trigger file (když zjistí existenci, provede failover), standby si ho hlídá

Samotné nastavení asynchronní replikace není nijak náročné. V první fázi je potřeba konfigurace postgresql.conf na master serveru. Stačí editace pouhých dvou parametrů, aby byla replikace v provozu.

```
wal_level = hot_standby
max_wal_senders = 1
```

Stejně tak je potřeba konfigurovat postgresql.conf na slave serveru. wal_level a max_wal_level_sender může a nemusí zůstat stejný jako na masteru. Pokud však má být slave připraven zastoupit master server v případě jeho výpadku, pak je vhodné, aby hodnoty byly nastaveny stejně. Co je potřeba nastavit je hot_standby a hot_standby_feedback.

```
wal_level = hot_standby
max_wal_senders = 1
hot_standby = on
hot_standby_feedback = on
```

Tím však nastavení replikace nekončí. Ve složce s daty je potřeba vytvořit recovery.conf, který ponese informace o master serveru, viz příklad.

```
standby_mode='on'
primary_conninfo='host=158.194.94.93 user=replication password=kgigis'
```

To, že replikace běží lze zkontrolovat několika způsoby. Na slave server není možné přidat žádná data:

```
INSERT INTO test VALUES (5, 'javor');
ERROR:  cannot execute INSERT in a read-only transaction
```

Dále tak, že si na master serveru vypíšeme pg_stat_replication:

```
SELECT username, application_name, client_addr, state, sync_state FROM
```

```
pg_stat_replication ;
```

```
username | application_name | client_addr | state | sync_state
-----+-----+-----+-----+-----
replication | gis | 80.188.74.1 | streaming | sync
(1 row)
```

Poslední parametr udává, zda se jedná o synchronní nebo asynchronní replikaci.

Inicializace synchronní streaming replikace vyžaduje v `postgresql.conf` na master serveru přidání hodnoty parametru `synchronous_standby_names`:

```
synchronous_standby_names = 'synchronous_gis'
```

a dále v `recovery.conf` na slave serveru přidání parametru `application_name`, který nese stejnou hodnotu jako výše zmíněný parametr.

```
standby_mode='on'
primary_conninfo='host=158.194.94.93 application_name=synchronous_gis
user=replication password=kgigis'
```

Inicializace kaskádové streaming replikace vyžaduje nejméně tři servery v clusteru. Ale nejedná se o nic složitějšího než o změnu konfigurace souboru `recovery.conf` na replice, kde se hodnoty masteru vymění za hodnoty slavy, ze kterého chceme na danou repliku kopírovat.

```
standby_mode='on'
primary_conninfo='host=158.194.94.18 user=replication password=kgigis'
```

Přehled všech parametrů týkající se replikace včetně vysvětlení jejich smyslu lze najít v oficiální dokumentaci [footnote{http://www.postgresql.org/docs/9.1/static/runtime-config-replication.html}](http://www.postgresql.org/docs/9.1/static/runtime-config-replication.html).

4.5 Rozložení zátěže mezi replikačními uzly

- vytvoření hesla (více viz PostgreSQL Replication, str. 152)

- **pg_md5 -p**

password: <your password>

vypíše heslo v hashi, které se překopíruje do souboru **pcp.conf** ve formátu **username:[password encrypted in md5]**

př. market:098f6bcd4621d373cade4e832627b4f6

- soubor **pcp.conf** ve složce `/etc/pgpool/pcp.conf` — pokud tam není, tak hledat `locate pcp.conf.sample`

- konfigurace pgpool (více viz PostgreSQL Replication, str. 149)

- pokud chceme nastavovat master/slave Mode, pak je ideální překopírovat šablonu přednastavenou pro tento mód

- ve složce `/etc/pgpool/` soubory `pcp.conf`, `pgpool.conf`, `pg_hba.conf`
 - `locate pgpool.conf.sample-master-slave.gz` — překopírovat jej pod názvem **pgpool.conf** do `/etc/pgpool/`
 - v souboru **pgpool.conf** editovat řádky:

```

listen_addresses = 'localhost'
port = 9999          # 9999 pak bude port, který se budou do databáze připojovat klienti, místo
5432
pcp_port = 9898      # 9898 je port pro admina
-----
# tato část nastavuje jednotlivé nody (databáze)
# node0 — master server — p 5436
backend_hostname0 = 'localhost'          # Host name or IP address to connect to for backend
0
backend_port0 = 5436 # Port number for backend 0
backend_weight0 = 1          # Weight for backend 0 (only in load balancing
mode)
backend_data_directory0 = '/var/lib/postgresql/9.3/main'
backend_flag0 = 'ALLOW_TO_FAILOVER'
# node1 — slave1 server — p 9311
backend_hostname1 = 'localhost'
backend_port1 = 9311
backend_weight1 = 1
backend_data_directory1 = '/var/lib/postgresql/9.3/slave1'
backend_flag1 = 'ALLOW_TO_FAILOVER'
# node2 — slave2 server — p 9322
backend_hostname2 = 'localhost'
backend_port2 = 9322
backend_weight2 = 1
backend_data_directory2 = '/var/lib/postgresql/9.3/slave2'
backend_flag2 = 'ALLOW_TO_FAILOVER'
• spuštění pgpool
  ○ /etc/init.d/pgpool2 start|status|stop
    ■ pustí demona
  ○ pgpool -f /etc/pgpool2/pgpool.conf -F /etc/pgpool2/pcp.conf -n -d
    ■ -d debug (=včetně výpisu chyb), -n nespouští jako demon (= v pozadí)
    ■ bez -d -n je příkaz stejný jako init.d skript
• pcp_node_count
  ○ market ~ # pcp_node_count 30 localhost 9898 market test
  ○ 3
• pcp_node_info
  ○ market ~ # pcp_node_info 30 localhost 9898 market test 0
  ○ localhost 5436 1 0.333333
  ○ market ~ # pcp_node_info 30 localhost 9898 market test 1
  ○ localhost 9311 1 0.333333
  ○ market ~ # pcp_node_info 30 localhost 9898 market test 2
  ○ localhost 9322 1 0.333333
• pcp_pool_info
  ○ market ~ # pcp_pool_status 30 localhost 9898 market test
• co nastavit
  ○ delay_threshold
    ■ log_standby_delay = 'if_over_threshold' # if_over_threshold, none
    ■ delay_threshold = 10000000

```

- `delay_threshold` defines the maximum lag a slave is allowed to have to still receive reads. The setting is defined in bytes of changes inside the XLOG. So, if you set this to 1024 a slave is only allowed to be 1 KB of XLOG behind the master. Otherwise it will not receive read requests.
- Tam kde potřebuješ aby data byla na replice velice brzo (typicky když vytváříš profil na webu), nastavíš menší delay.

5 TESTOVÁNÍ

`pg_bench`

V *pg_bench* tj. při maximálním vytížení (výchozí konfigurace a IO s zablokovanou write cache (obyčejný disk v notebooku)) je režie asynchronní replikace cca 6-10%, a režie synchronní replikace cca 40% - v běžném provozu bych očekával režii znatelně nižší - pravděpodobně budete mít na svém serveru jistě IO s aktivní write cache.

zdroj <http://postgres.cz/wiki/Replikace>

6 DISKUZE

Proč PostgreSQL proč ne třeba MySQL, které taky podporuje ukládání prostorových dat? Protože PostgreSQL je zároveň podporováno produkty ArcGIS.

7 ZÁVĚR

Tato práce navrhuje možné použití replikace s použitím databázového serveru PostgreSQL.

ZDROJE

Knihy

(Bell, Kindahl, Thalmann, 2010) BELL, Charles, Mats KINDAHL a Lars THALMANN. *MySQL high availability*. 1. ed. Sebastopol, CA: O'Reilly Media, Inc, 2010. ISBN 978-059-6807-306.

(BÖSZÖRMENYI2013) BÖSZÖRMENYI, Zoltan. PostgreSQL replication: understand basic replication concepts and efficiently replicate PostgreSQL using high-end techniques to protect your data and run your server without interruptions. Birmingham: Packt Publishing, c2013, vii, 230 s. ISBN 978-1-84951-672-3.

(Connolly, 2005) CONNOLLY, Thomas. *Database Systems: A Practical Approach to Design, Implementation, and Management*. 4th ed. Harlow: Addison-Wesley, 2005, 1374 s. ISBN 03-212-1025-5.

(Esri, 2006) ESRI. *ArcGIS 9: Co je ArcGIS 9.2?*. United States: Esri Press, US, 2006. ISBN 15-894-8166-6.

(Chacon, 2009) CHACON, Scott. *Pro Git*. Praha: CZ.NIC, c2009, 263 s. Edice CZ.NIC. ISBN 978-80-904248-1-4.

(Leiter, 2009) LEITER, Chris. *Beginning Microsoft SQL server 2008 administration*. Indianapolis, IN: Wiley Pub., 2009, p. cm. ISBN 978-047-0440-919.

(Momjian, 2001) MOMJIAN, Bruce. *PostgreSQL: introduction and concepts*. Boston, MA: Addison-Wesley, 2001, xxviii, 461 p. ISBN 02-017-0331-9.

(Obe, Hsu, 2011) OBE, Regina a Leo HSU. *PostGIS in action*. London: Pearson Education [distributor], 2011, 492 s. ISBN 19-351-8226-9.

(Obe, Hsu, 2012) OBE, Regina a Leo HSU. *Postgresql: Up and Running*. Sebastopol, CA: O'Reilly, 2012, 164 s. ISBN 978-144-9326-333.

(Oppel, 2009) OPPEL, Andrew J. *Databases: a beginner's guide*. New York: McGraw-Hill, 2009, xviii, 478 p. ISBN 00-716-0846-X.

(Riggs, Krossing, 2010) RIGGS, Simon a Hannu KROSING. *PostgreSQL 9 administration cookbook: solve real-world PostgreSQL problems with over 100 simple, yet incredibly effective recipes*. Birmingham: Packt Publishing, 2010, 345 s. ISBN 978-1-849510-28-8.

(Vieira, 2009) VIEIRA, Robert. *Professional Microsoft SQL server 2008 programming*. Indianapolis, IN: Wiley Pub., 2009, 893 s. Wrox professional guides. ISBN 04-702-5702-4.

(Whalen, 2008) WHALEN, Edward a kol. *Microsoft SQL Server 2005: velký průvodce administrátora*. Vyd. 1. Překlad Jakub Mikulaščík, David Krásenský. Brno: Computer Press, 2008, 1080 s. Administrace (Computer Press). ISBN 978-80-251-1949-5.

Internetové zdroje

(Činčura, 2009) ČINČURA, Jiří. *Databázový svět* [online]. 2009 [cit. 2013-08-12].

(Esri, 2013a) ESRI. *A quick tour of working with databases in ArcGIS*. ArcGIS Help 10.1 [online]. 2013 [cit. 2013-08-02]. Dostupné z:

http://resources.arcgis.com/en/help/main/10.1/index.html#/A_quick_tour_of_working_with_databases_

[in_ArcGIS/019v00000008000000/](#)

(Esri, 2013b) ESRI. *Preparing data for replication*. ArcGIS Help 10.1 [online]. 2013 [cit. 2013-08-02]. Dostupné z:

http://resources.arcgis.com/en/help/main/10.1/index.html#/Preparing_data_for_replication/003n000000z5000000/

(Law, 2008) LAW, Derek. *Enterprise Geodatabase 101: A review of design and key features for GIS managers and database administrators*. Esri: Understanding our world. [online]. 2008, [cit. 2013-06-18]. Dostupné z: http://www.esri.com/news/arcuser/0408/entergdb_101.html

(Microsoft, 2013) MICROSOFT. *SQL Server - Replication*. Microsoft [online]. 2013 [cit. 2013-08-27]. Dostupné z: [http://technet.microsoft.com/en-us/library/ms151198\(v=sql.100\).aspx](http://technet.microsoft.com/en-us/library/ms151198(v=sql.100).aspx)

(OpenGeo, 2012) OPENGEO. *Introduction to PostGIS* [online]. 2012b [cit. 2012-08-08]. Section 17: Geography. Dostupné z: <http://workshops.opengeo.org/stack-intro/openlayers.html>

(OSGEO, 2013) OSGEO. *PostGIS and ArcSDE/ArcGIS Articles*. *PostGIS Tracker and Wiki* [online]. 2013 [cit. 2013-08-08]. Dostupné z: <http://trac.osgeo.org/postgis/wiki/UsersWikiPostgisarcgis>

(PostgreSQL, 2012) POSTGRESQL. *FAQ - PostgreSQL wiki* [online]. 2012 [cit. 2012-08-08]. Dostupné z: <http://wiki.postgresql.org/wiki/FAQ>