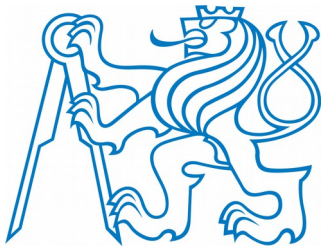
České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačové grafiky a interakce



Zpráva k projektu 493/2013/1

Konsolidace zálohování a archivace dat
Připojení k DÚ CESNET

Michal Strnad , Jan Kubr

Prosinec 2014

Úvod

Při řešení projektu bylo potřeba vyřešit připojení k datovým úložištím CESNET a instalaci námi zvoleného zálohovacího software.

Vytvoření Virtuální Organizace a základní nastavení

Pro účely tohoto projektu jsme se rozhodli využít služeb oddělení datových úložišť CESNET, které se poskytují studentům, akademickým pracovníkům a pracovníkům výzkumných organizací v ČR pro vzdělávací a výzkumné účely. Z přehledu protokolů a služeb [1] naším potřebám nejvíce vyhovovala služba pod označením Virtuální Organizace (dále jen VO).

VO slouží primárně pro skupinu lidí, která řeší společně nějaký projekt nebo je z jedné organizace. V rámci VO pak mohou mezi sebou sdílet zdroje, jež byly VO přiřazeny po dohodě s administrátory datových úložišť CESNET.

Datová úložiště nabízejí možnost stát se členem předpřipravené VO s kódovým označením VO Storage a v případě větších nároků na úložiště umožňují založit vlastní VO a nastavit ji dle potřeby žadatelů.

VO Storage je virtuální organizace, která sdružuje uživatele (účty), kteří budou ukládat data pouze pro sebe, bez možnosti je sdílet, či chtějí úložiště jen otestovat. Velikost datového prostoru je zde omezena na 500 GB. Z důvodu těchto nedostatků jsme zvolili cestu vlastní VO, která pokryje naše potřeby.

Vlastní VO má určeného správce (nebo i více správců) z vlastních řad, který vytváří skupiny a řídí členství. Speciálním typem účtu, který zde může být vytvořen, je tzv. servisní identita sloužící pro přístup zálohovacích aplikací a skriptů na úložiště.

Pro registraci do VO Storage i vlastní VO, je potřeba mít účet ve federaci identit eduID.cz. Jelikož všichni členové pracovní skupiny tohoto projektu jsou zaměstnanci nebo studenti Českého vysokého učení technického v Praze, které je členem federace eduID.cz, nebylo nutné využívat služeb Hostel [5] a bylo možné využít účty používané pro přístup ke službám univerzity.

Pro domluvení parametrů VO jsme se obrátili na mailinglist datových úložišť: du-support@cesnet.cz, kde jsme s pracovníky diskutovali naše potřeby a možnosti datových úložišť CESNET. DU CESNET nyní disponuje třemi geograficky oddělenými úložišti a to v Plzni, Brně a Jihlavě. Jelikož je plzeňské data centrum nejblíže, zvolili jsme Plzeň. Tím dosáhneme co nejmenší odezvy při přenosu. Náš předpoklad jsme si ověřili spuštěním skriptu [6]. Dále bylo potřeba zvolit migrační politiku [7]. Ta definuje co se s daty po uložení do adresáře odpovídajícího migrační politice stane. V plzeňském data centru jsou k dispozici cache_tape, tape, tape_tape a disk_only. Vzhledem k tomu, že úložiště chceme využít pro zálohy a archívy, je pro nás nejvýhodnější migrační politika s kódovým označením tape_tape. Pokud uživatel zkopíruje data do adresáře spadajícího pod tuto migrační politiku, budou data uložena na SATA discích. Pokud se na ně nějaký čas nepřistoupí nebo dojde k zaplnění svazku dojde k automatické migraci dat na nižší tiery. Konkrétně bude jedna kopie dat uložena na MAID discích a druhá na páskách.

Jakmile nám administrátoři DU předali naši VO, vytvořili jsme si servisního uživatele, kterého jsme používali pro přístup na úložiště.

Datové úložiště CESNET během října 2014 zprovoznilo většinu služeb přes protokol IPv6. Z důvodu dostupného 10 Gbps připojení přes IPv6 only jsme se rozhodli tuto možnost využít a použít

přenosové protokoly, které IPv6 podporují. Mezi tyto protokoly patří SSH, SCP, SFTP, rsync a NFS. Pro rozhodnutí, který protokol je pro naše potřeby nejlepší jsme použili přehled protokolů a služeb s jejich doporučeními [2] vypracovány pracovníky datových úložišť CESNET. Z tohoto přehledu nám vyšly nejlépe celkem dva protokoly a to rsync a NFS.

- Rsync je nástroj pro synchronizaci dat, kde není vyžadována historie záloh, jen přesná kopie. Využívá delta přenos, což znamená že s přenášejí jen změny v souborech, což podstatně šetří šířku pásma. Naopak kopírování nového souboru vyžaduje několik metadata operací, což ve výsledku velmi omezuje přenos těchto souborů na cca 40 souborů za vteřinu bez ohledu na jejich velikost a kapacitu přenosové linky.
Umožňuje zachovávat práva a informace o vlastníkovi souboru při přenosu na vzdálené úložiště v rozšířených atributech souborů [3].

- Protokol NFS ve verzi 4, které datové úložiště CESNET provozují nad autentizačním protokolem Kerberos, slouží pro připojení vzdáleného úložiště, tak aby se pro uživatele tvářil jako lokální svazek a mohl na něj použít všechny nástroje, které běžně používá pro práci s lokálními daty. Jeho nevýhodou je, že bez dalších nástrojů, nelze pomocí tohoto protokolu zachovat původního vlastníka souboru. Proto jsme se rozhodli tyto protokoly zkombinovat.

Prvotní konfigurace protokolu NFSv4 v kombinaci se systémem Kerberos bylo poněkud složitější, ale pomocí návodu [4] dostupného na webu oddělení datových úložišť se to nakonec podařilo zprovoznit.

Automatické připojení vzdáleného svazku po spuštění server jsme vyřešili přidáním následujícího řádku do souboru /etc/rc.local:
mount -t nfs4 -o sec=krb5,rsize=1048576,wsize=1048576 nfs.du1.cesnet.cz:/ /storage/cesnet/

Dále jsme si vytvořili keytab, jakožto náhradu za lístky. Keytab nám umožní získat jednodenní lístky s keytabu, aniž by bylo potřeba zadávat opětovně heslo.

Nastavení Baculy

Na serveru Bdrive jsme nainstalovali metabalíček "bacula", který obsahuje balíčky bacula-server, bacula-client, bacula-common. Pokud chceme v Bacule číst nebo zapisovat na libovolný volume, tak fyzický volume musí mít softwarový label, který značí že je volume připojen (mounted). Abychom label nemuseli ručně přidávat pokaždé, když budeme mít nový volume, nastavili jsme autolabeling, jenž tuto operaci provede zcela automaticky.

Hesla pro všechny komponenty jsou generována během instalace. Přesto jsme je z bezpečnostních důvodů změnili.

Instalací balíčku "bacula" v Debianu se instaluje i databáze SQLlite. Ta je však dobrá jen pro testování a již není vhodná pro větší množství dat. Při záznamech v řádu miliónu a více se její činnost velice zpomalí. Jelikož bychom se časem dostali na hranici jejich možností, rozhodli jsme se vyzkoušet alternativní databáze jako je MySQL a PostgreSQL. Obě databáze byly při počtu záznamů v miliónech násobně rychlejší než MySQL. Nakonec jsme se rozhodli pro PostgreSQL, který vykazoval o něco lepší výsledky.

Postup na instalace PostgreSQL

V Debianu při instalaci baculy přes balíčkovací systém chybí skript pro vytvoření samotné databáze, proto v tomto návodu provádíme i kroky v něm obsažené.

Vytvoříme uživatele 'bacula'

su - postgres
createuser bacula
Jako odpověď na každou otázku co nám příkaz zobrazí dáme N.

Nastaveni hesla

alter user bacula with encrypted password 'bacula';
Vytvořit databázi se spravným kódováním (není doporučované používat kódování UTF-8, jelikož by vytvořilo názvy souborů jež by nemuseli jít obnovit.

createdb -E SQL_ASCII -T template0 bacula
Upravit v souboru /usr/share/bacula-director/make_postgresql_tables jméno databáze (db_name=bacula)

Spustíme skript (musíme pod uživatelem postgres)

/usr/share/bacula-director/make_postgresql_tables // Musíme pod uživatelem postgres
Upravit v souboru /usr/share/bacula-director/grant_postgresql_privileges jméno databáze a uživatele (db_name=bacula, db_user=bacula)

Spustíme skript (musíme pod uživatelem postgres)

/usr/share/bacula-director/grant_postgresql_privileges
Upravíme nastavení Directoru v souboru /etc/baculad/bacula-dir.conf

```
Catalog {
  Name = MyCatalog
  dbname = "bacula"; dbuser = "bacula"; dbpassword = "bacula"
}
```

Případně můžeme dodat ještě definici ovladače

```
Catalog {
  Name = MyCatalog
  dbdriver = "dbi:postgresql"; dbaddress = 127.0.0.1; dbport =5432
  dbname = "bacula"; dbuser = "bacula"; dbpassword = "bacula"
}
```

Umístíme do /etc/postgresql/8.4/main/pg_hba.conf následující řádek:

```
local   bacula      bacula                        md5
```

Restartuje PostgreSQL

/etc/init.d/postgresql restart
A restartujeme Directora

/etc/init.d/bacula-dir restart

Optimalizace rychlosti

Během testů protokolů jsme zjistili, že SSH které používá rsync a sftp se chová tak, že vytvoří pro každé SSH spojení nový socket. Je však možné vytvořit perzistentní socket, který využívají všechna tato spojení. Výsledkem bude zrychlení, které bude tím větší, čím větší bude počet souborů. Nastavení perzistentního soketu se provádí následovně:

Přidáme do souboru ~/.ssh/config následující

Host host
ControlPath ~/.ssh/master-%r@%h:%p
ControlMaster no
'host' je doménové jmné serveru, kam budeme data přenášet.

Perzistentní připojení se vytváří pomocí příkazu

ssh -MNf username@server
Toto spojení běží na pozadí a nyní již můžeme použít rsync nebo sftp klasickým způsobem.

Literatura

[1] https://du.cesnet.cz/cs/navody/jsme_tady_poprve
[2] https://du.cesnet.cz/cs/prehled_protokolu_a_sluzeb_s_jejich_doporucenimi/start
[3] https://du.cesnet.cz/cs/navody/rsync/start
[4] https://du.cesnet.cz/cs/navody/nfs/start
[5] http://hostel.eduid.cz
[6] https://du.cesnet.cz/_media/cs/select_dc.sh
[7] https://du.cesnet.cz/cs/navody/home-migrace-plzen/start