Technická zpráva CESNETU číslo 1/2004

Perun – systém pro řízení přístupu uživatelů k prostředkům MetaCentra

Aleš Křenek   Zora Sebestianová   Jiří Sitera

24. března 2004

Abstrakt

Systém Perun je nástroj zaměřený především na správu uživatelských účtů a souvisejících prostředků v *META Centru*. V této technické zprávě popisujeme jeho architekturu, jednotlivé komponenty, funkční návaznosti, konkrétní datové schéma, uživatelské a administrátorské www rozhraní, a návaznost na informační služby *META Centra*. Zvláštní pozornost je věnována rozšiřitelnosti – jsou detailně popsány vstupní a výstupní rozhraní rozšiřujících modulů, včetně konkrétních příkladů.

Obsah

1 Úvod

Infrastruktura projektu *META Centrum*[1] vznikající od roku 1996 vyžadovala nasazení systému pro správu uživatelských účtů a souvisejících prostředků. Požadavky na jeho funkcionalitu byly vyhodnoceny jako natolik specifické, že bylo přikročeno k vývoji vlastního systému.

První verze systému Perun [RKM99] byla nasazena v roce 1997. Její monolitická architektura se ale brzy ukázala jako nevýhodná – hledání chyb v implementaci bylo obtížné, rozšiřování (resp. modifikace jen určitých částí) komplikovaná, a vynucená implementace v jazyce C na řadu úloh těžkopádná. Proto bylo přikročeno k zásadní revizi architektury a vývoji verze 2.0, ve které byly jednotlivé služby jednoznačně vymezeny a implementovány samostatnými, nezávisle laditelnými programy (zpravidla skripty v jazyce Perl).

V současnosti používaná a v následujícím textu popsaná verze 2.2 pokročila ještě dále v modularizaci nahrazením specializovaných databázových triggerů pro vyhodnocení změn (vynucujících implementaci v PL/SQL) sadou univerzálních triggerů a samostatných plánovacích skriptů.

2 Architektura systému

2.1 Přehled

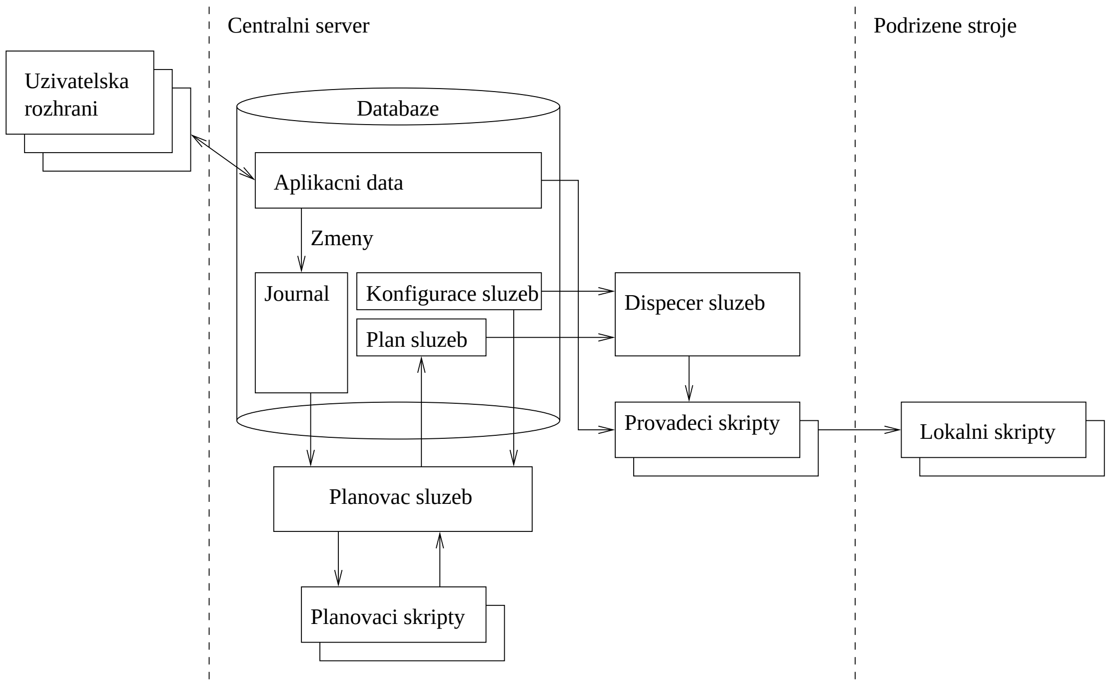
Systém Perun spravuje konfiguraci určitých prostředků. Cílem systému je udržet přesný obraz reality spravovaných prostředků a usnadnit manipulaci s nimi. V současné verzi systému se jedná o správu strojů, clusterů a uživatelských účtů na nich. Protože v rámci *META Centra* je používaná autentizace prostřednictvím Kerbera, spravuje systém také kerberovské realmy a množinu jeho principálů. Ke sdílení souborů se používá systém AFS a tedy rovněž tímto systémem definované a požadované prostředky jsou spravovány systémem Perun.

Aktuální, resp. správci požadováná konfigurace těchto prostředků je udržována v relační databázi se schématem navrženým tak, aby odráželo konkrétní požadavky, integritní omezení apod. (konkrétní schéma viz 2.7).

Spravované prostředky jsou členěny do tzv. *služeb* (tento přístup byl v principu převzat z architektury systému Moira, [RGL88]). Službou rozumíme logicky související část konfigurace spravovaných prostředků, kterou je žádoucí a má smysl měnit atomicky (např. soubor `/etc/passwd` na jednom počítači), a to včetně zohlednění implementačních omezení. Služby jsou tedy směřovány k určitému cíli, na kterém má být provedena úprava, odrážející změnu spravovaných prostředků, zanesenou v databázi (konkrétní služby jsou popsány v 3).

Systém je primárně postaven na tzv. *push modelu* toku dat a řízení (viz obr. 1). Veškeré změny v řízených konfiguracích prostředků jsou správcem provedeny v databázi. Systém změny zpracovává do aktualizací zasažených služeb tímto postupem:

[1] `http://meta.cesnet.cz/`



Obrázek 1: Blokové schéma architektury systému

- požadovaná změna v databázi
- detekce a vyhodnocení změny dat
- naplánování služby
- propagace služby

Uživatel (správce prostředků) tedy pouze aktualizuje (pomocí uživatelského rozhraní, viz 5) data v databázi, mapování na služby a jejich aktualizace je provedena transparentně.

Veškeré změny v klíčových tabulkách popisujících konfiguraci řízených prostředků jsou databázovými triggery ukládány do speciální tabulky JOURNAL. Její struktura už je nezávislá na konkrétním obsahu a struktuře aplikačních dat.

Každá změna v aplikačních datech (tj. vytvoření záznamu v tabulce JOURNAL) je zachycena *Plánovačem služeb*. Ten podle konfigurace zavolá příslušný plánovací skript, jehož výsledkem je naplánování aktualizace jedné nebo více služeb (a případné další parametry). Kontrolu nad naplánovanými službami nadále přebírá *Dispečer služeb*, který zajišťuje jejich aktualizaci prostřednictvím prováděcích skriptů. Dispečer také zajišťuje zotavení z dočasných chyb.

Na cílových stanicích pracuje démon, který zde v případě potřeby zajišťuje finální provedení služby (viz 2.5).

2.2 Sledování změn

Ke sledování změn v datových tabulkách zdrojů *META Centra* slouží tabulka JOURNAL. Tato tabulka zachycuje, jaké změny byly v databázi provedeny, kdo je provedl a kdy.

Záznamy jsou jednoznačně identifikovány trojicí čísel:

- *sess* – číslo ze speciální sekvence, přiřazováno jednou v rámci každého připojení k databázi,
- *stmt* – pořadové číslo SQL příkazu v rámci daného připojení,
- *line* – pořadové číslo záznamu ovlivněného tímto příkazem.

Administrátor systému má k dispozici skript, kterým vytváří pro aplikační tabulky dvojici triggerů **j_*jméno-tabulky*_stmt** a **j_*jméno-tabulky***.

První z těchto triggerů (pracující „before insert,update,delete“ na úrovni SQL příkazu, tj. pouze jednou, i když se příkaz týká více záznamů) volá proceduru *init_stmt*, jež inicializuje číslo připojení (sess), když se jedná o nové připojení uživatele, dále přiděluje číslo SQL příkazu (stmt) a zjišťuje jméno uživatele, který změnu provádí.

Druhý trigger (pracující „after insert,update,delete“ pro každý záznam) využívá proceduru *next_line*. Tato přiděluje číslo řádku *line* každé jednotlivé změně. Trigger pak zapisuje takto očíslované změny, včetně jména tabulky a času, do JOURNAL takovým způsobem, že pro každý měněný nebo vkládaný sloupec aplikační tabulky je vytvořen jeden záznam v tabulce JOURNAL (viz struktura záznamu v JOURNAL dále). Při rušení v aplikační tabulce vkládá trigger do JOURNAL jediný záznam.

Obě zmíněné procedury jsou součástí databázové package *journal_pkg* vytvořené současně s tabulkou JOURNAL.

Struktura záznamu v JOURNAL je následující:

- identifikace záznamu, složená z čísla sezení *sess*, čísla skupiny změn *stmt* a pořadí změny *line* (viz výše)
- čas, kdy byla změna provedena
- identifikace uživatele, který provedl změnu
- jméno tabulky, ve které změna proběhla
- typ změny (INSERT/UPDATE/DELETE)
- jméno sloupce, který byl změněn (pro INSERT a UPDATE)
- novou hodnotu sloupce (pro INSERT a UPDATE)
- identifikátor řádku v dotyčné tabulce (starého pro DELETE, měněného pro UPDATE, nového pro INSERT)

Jak bylo naznačeno výše, tabulka JOURNAL slouží především pro následné zpracování službami. Nicméně je využívána také pro dohledání původních hodnot některých sloupců, dále k zjištění, zda některá změna opravdu proběhla tak, jak se uživatel domnívá. Také umožňuje zjistit, kdy a kým byla zavedena určitá, například nežádoucí, změna.

2.3 Plánovač služeb

Plánovač služeb slouží k tomu, aby mohla být určitou změnou v aplikačních datech naplánována a následně vyvolána příslušná služba. Program plánovač služeb se jmenuje ***pjournal*** a pracuje jako démon. Jeho činnost je možné zhruba rozdělit do tří fází.

2.3.1 Průchod tabulkou JOURNAL

Na zápis do JOURNAL je navázán tzv. databázový alert, který v případě, že je transakce potvrzena, vzbudí plánovač služeb. Kromě toho je plánovač služeb z důvodů vyšší robustnosti ještě pravidelně buzen časovačem. Takto vzbuzený plánovač služeb prohledá tabulku JOURNAL počínaje za posledním záznamem zpracovaným při minulé aktivaci. Nalezené záznamy porovná proti konfigurační tabulce, aby určil, zda se jedná o významnou změnu, ke které je definována služba. Konfigurační tabulka PREP_SERVICES obsahuje následující údaje:

- typ změny (INSERT/UPDATE/DELETE)
- jméno změněné tabulky
- jméno plánovacího skriptu pro přípravu plánování

Pokud testovaná změna odpovídá záznamu v konfigurační tabulce, je tedy významná a je zapsána do pracovního pole. Po zpracování všech aktuálních záznamů jsou v pracovním poli vyhledány změny stejného typu, tj. změny, které mají být zpracovány týmž plánovacím skriptem a tyto jsou zapsány do dočasného souboru. Tento dočasný soubor obsahuje identifikaci záznamu v JOURNAL (*sess*, *stmt* a *line*), jméno tabulky a typ změny. Jeden řádek v tomto souboru odpovídá jedné databázové operaci nad danou databázovou tabulkou. Potom je postupně pro všechny vytvořené soubory spušten plánovací skript, na jehož ukončení plánovač služeb čeká.

2.3.2 Plánovací skripty

Plánovací skript dostane jako první vstupní parametr jméno dočasného souboru, naplněného plánovačem služeb. Struktura tohoto souboru je následující:

- identifikátor záznamu v JOURNAL (viz 2.2 struktura JOURNAL), tj. trojice čísel sess, stmt, line.
- jméno aplikační tabulky

- typ změny (insert/update/delete)

všechny položky oddělené mezerami.

Pro předání výstupů zpět plánovači služeb dostane druhý parametr se jménem dalšího dočasného souboru.

Existuje několik různých plánovacích skriptů, které se liší podle měněné tabulky a typu změny.

Specifická logika konkrétního plánovacího skriptu určí, která služba, eventuálně služby, mají být pro dotyčný typ změny naplánovány, dále určí cíl (stroj,cluster,realm,buňku), pro který má být služba provedena. Pokud má uživatel, který změnu v aplikačních datech provedl, stanovenu nějakou konkrétní prodlevu, je tato použita pro výpočet času, na který má být služba naplánována, jinak je plánována s implicitní prodlevou 30 minut. Tyto údaje, tedy čas plánování, služba a její cíl jsou plánovacím skriptem uloženy do výstupního souboru (viz výše). Jeho struktura je následující:

- plánovaný čas zpracování služby
- jméno služby
- jméno cíle (jméno stroje/jméno clusteru/jméno realmu/jméno buňky)
- typ cíle (H=stroj/C=cluster/R=realm/V=buňka)

Po naplnění výstupního souboru plánovací skript skončí. Konkrétní příklad plánovacího skriptu pro fiktivní službu je uveden v části 3.3.

2.3.3 Naplánování služby

Po ukončení plánovacího skriptu pokračuje v činnosti opět plánovač služeb, který přečte data z dočasného souboru a testuje proti konfigurační tabulce pro plánování, zda je služba aktuálně aktivní. Je-li, doplní z konfigurační tabulky nezbytné další údaje a službu naplánuje do tabulky PLAN_SERVICES. V případě, že služba byla naplánována na aktuální čas, pošle uživatelský signál dispečeru služeb (viz 2.4).

Plánovač služeb pak pokračuje ve zpracování dalšími záznamy v pracovním poli, vytvořeném při průchodu tabulkou JOURNAL (viz 2.3.1). Až projde celé pracovní pole pokračuje v procházení tabulky JOURNAL až do posledního aktuálně uloženého záznamu (provedení plánovacích skriptů je relativně dlouho trvající operace a mezitím mohly v tabulce JOURNAL přibýt další záznamy). Po zpracování všech záznamů se plánovač služeb uspí a očekává další databázový alert, nebo vzbuzení časovačem.

Struktura tabulky PLAN_SERVICES je následující:

- služba
- host, clust, realm, cell – identifikace cíle, vyplněn je právě jeden z těchto sloupců
- čas, kdy má být spuštěno zpracování služby

- stav zpracování (viz 2.4.1 a 2.4.3)
  - `plan` – plánovaná, ještě nespuštěná
  - `proc` – rozpracovaná, spuštěn služební skript, ještě neskončil
  - `canc` – zrušená, zahrnuta do zpracování jiného záznamu
  - `open` – otevřená, zpracována s chybou, bude opakována později
  - `done` – ukončená, bezchybně zpracovaná
  - `err` – ukončená s chybou i po vyčerpání povolených pokusů
- zbývající počet pokusů o provedení služby
- čas za který má být služba znovu naplánována při průběžné chybě
- chybová zpráva, pokud došlo k chybě v průběhu provedení služby
- číslo zpracovatelského procesu – služebního skriptu

Jak bylo popsáno výše jsou položky služba, host/clust/realm/cell a čas pro plánování stanoveny plánovacími skripty, stav zpracování je nastaven na *plan*. Počet opakování a prodleva jsou doplněny pro danou kombinaci služba-cíl z konfigurační tabulky CONF_SERVICES, která má následující strukturu:

- služba
- host, clust, realm, cell – identifikace cíle
- příznak aktivní/neaktivní
- počet možných opakování služby v případě chyby
- prodleva, čas za který má být služba znovu naplánována při průběžné chybě
- směr prohledávání tabulky PLAN_SERVICES (A/D)
- jméno služebního skriptu

Ostatní položky jsou využity až v průběhu zpracování služeb (viz 2.4).

2.3.4 Parametry plánovače služeb

Plánovač služeb (démon ***pjournal***) může být spuštěn s následujícími parametry:

`-d` vypisuje debugovací výpisy (bez hodnoty)

`-f` uchovává dočasné soubory pro komunikaci s plánovacími skripty (bez hodnoty)

`-t` bude zpracovávat z JOURNAL pouze změny určité tabulky

`-c` zpracuje záznamy až od zadaného data

`-a` nastavení timeout pro časovač (default 300 s)

`-s` pracuje až od určitého čísla sezení

`-m` pracuje až od určitého čísla skupiny změn
Je-li zadáno `-s` a není zadáno `-m` bere se `-m=0`. Je-li zadáno `-m` a není zadáno `-s` bere se `-s=0`

Při spuštění bez těchto voleb nebude démon vypisovat žádné trasovací výpisy, dočasné soubory budou po přečtení smazány, bude zpracovávat záznamy o všech tabulkách (viz konfigurace), bude použita defaultní hodnota pro časovač a začne pracovat od začátku tabulky JOURNAL.

2.4 Dispečer služeb

Dispečer služeb je určen ke spouštění naplánovaných služeb a k vyhodnocení úspěšnosti jejich činnosti. Dispečer služeb je program jménem **pdispatch2**[2] a pracuje jako démon. Jak bylo zmíněno již v 2.3.3, program je aktivován uživatelským signálem nebo časovačem (dále 2.4.1), kromě toho pak signálem o ukončení synovského procesu (viz 2.4.3). Jeho činnost je do určité míry ovlivňována konfigurační tabulkou CONF_SERVICES, jejíž struktura byla popsána v kapitole 2.3.3.

2.4.1 Spuštění služby

Při aktivaci signálem nebo časovačem dispečer služeb prochází tabulku PLAN_SERVICES a vyhledává záznamy s plánovaným časem spuštění menším či rovným aktuálnímu času. Tyto záznamy musí být ve stavu naplánovaný *plan* nebo otevřený *open* (viz též struktura tabulky PLAN_SERVICES). V konfigurační tabulce se navíc ověří, zda je služba aktuálně aktivní.

Směr prohledávání PLAN_SERVICES se řídí parametrem *ord* z tabulky CONF_SERVICES, tj. postupuje se od prvního (hodnota *ord* je 'A') nebo od posledního ('D') takto nalezeného záznamu pro danou kombinaci služba-cíl. Zpracován bude pouze první resp. poslední takto nalezený záznam, ostatní jsou označeny za zrušené *canc*, neboť jejich zpracování je pokryto zpracováním tohoto jediného záznamu.

Standardním postupem je průchod od posledního záznamu. Důsledkem je provedení služby až podle posledního naplánování. Obecně je tento postup výhodnější – je-li v databázi provedeno více změn v intervalech menších, než je prodleva naplánování, je ovlivněná služba provedena pouze jednou, a aktualizace zahrne všechny změny. Ovšem při větší frekvenci změn může docházet k jistému druhu strádání – daná služba je stále přeplánovávána na později, ale k provedení nedojde. Navíc pro některé služby tento způsob nemusí vyhovovat. Pro tyto případy je připraven druhý směr průchodu tabulkou. Při tomto způsobu

[2] Jméno programu je reliktem z přechodného období, kdy vedle sebe existovaly dvě jeho verze.

zpracování se zpracovávají záznamy, který jsou naplánovány na čas menší nebo stejný, jako je aktuální čas. Záznamy naplánované na pozdější časy zůstanou prozatím nezpracovány.

Z konfigurační tabulky CONF_SERVICES je získáno jméno programu služebního skriptu a tento skript je spuštěn (viz 2.4.2).

Zpracovávaný záznam v tabulce PLAN_SERVICES je označen jako rozpracovaný – *proc* a je zde uloženo *pid* procesu služebního skriptu.

Dispečer služeb pak nečeká zde na ukončení služebního skriptu a pokračuje vyhledáním dalšího aktuálního záznamu, který zpracuje výše popsaným postupem. Současně tak může být spuštěno několik služebních skriptů. Jejich maximální počet je limitován jedním z parametrů spouštění služebního skriptu (viz 2.4.4). Po zpracování posledního aktuálního záznamu, nebo vyčerpání maximálního počtu současně spuštěných služebních skriptů, program usne a očekává vzbuzení některým ze signálů nebo časovačem.

2.4.2 Služební skripty

Služební skripty dostávají jako jediný parametr název cíle, protože vždy zpracovávají celou množinu dat příslušných k dané kombinaci služba-cíl. Služební skripty jsou specifické pro konkrétní služby (nikoli konkrétní cíle), vzájemně se velmi liší, některé například vytvářejí soubory a přenáší je na cílové stroje, kde spustí výkonný skript, který provede finální zpracování, jiné přímo vytvářejí například nové kerberovské realmy nebo AFS volumy. Pro komunikaci s cílovými stroji se využívá program *pupdate*, který zajistí autentizaci uživatele vůči cílovému stroji a přenesení potřebných souborů (viz 2.5). V podstatě mohou služební skripty zajišťovat jakoukoliv činnost, která je vázána na změnu aplikačních dat.

Každý služební skript má k dispozici vlastní pracovní adresář pro kombinaci služba-cíl. Adresářová struktura, za jejíž zprávu odpovídá dispečer služeb, je vytvořena následujícím způsobem:

$PERUN_ROOT/spool/*cíl*/*jméno cíle*/*jméno služby*.

Do souborů v takto vytvořených adresářích jsou rovněž přesměrovány výstupy `stdout` a `stderr` ze služebních skriptů. Pokud ještě neexistuje, je potřebný adresářový strom vytvářen automaticky dispečerem služeb před spuštěním služebního skriptu.

V souladu s obecnými konvencemi znamená nulová návratová hodnota skriptu úspěšné provedení.

Skript také předpokládá odpovídající nastavení určitých proměnných v prostředí, zejména nutných pro připojení k databázi. Kromě standardních ORACLE_HOME a TWO_TASK je to ORA_USER, ve které skripty předpokládají hodnotu `login/password`.

2.4.3 Ukončení služby

Byl-li dispečer služeb aktivován signálem o ukončení synovského procesu, vyhodnocuje úspěšnost provedení služebního skriptu.

Podle *pid* služebního skriptu je v PLAN_SERVICES nalezena právě ukončená služba. Když byla ukončena úspěšně, je záznam v PLAN_SERVICES označen za *done* a zůstává uložen pro eventuální pozdější kontroly. Pokud ale došlo v průběhu provádění služebního skriptu k chybě, je záznam v PLAN_SERVICES označen za *open* a položka opakování *recur* (viz struktura PLAN_SERVICES) je snížena o jednu. Provedení služby je naplánováno za dobu stanovenou v konfigurační tabulce pro danou kombinaci služba-cíl. Bylo-li provádění služby už vícekrát opakováno bez úspěchu a počet možných opakování tak byl snížen až na nulu, je daný záznam v PLAN_SERVICES označen za chybný, *err*.

Po ukončeném vyhodnocení služby program zpracovává eventuální další ukončení služebních skriptů. Pokud žádné další ukončené nejsou, proces usne a očekává vzbuzení některým ze signálů nebo časovačem.

2.4.4 Parametry dispečera služeb

Dispečer služeb (démon ***pdispatch2***) může být spuštěn s následujícími parametry:

`-d` – vypisuje debugovací zprávy (bez hodnoty)

`-a` čas v sekundách – časový interval pro vzbuzení časovačem (default 30 s)

`-n` počet – maximální počet současně spuštěných služebních skriptů (default 4)

Bez těchto voleb bude démon pracovat bez trasovacích výpisů a použije defaultní hodnoty pro časovač a počet současně spuštěných úloh.

2.5 Program *pupdate* a výkonné skripty na cílových strojích

Program ***pupdate***, spouštěný některými služebními skripty (viz 2.4.2), je určen k zabezpečenému přenosu vygenerovaných datových souborů na řízený stroj a spuštění tzv. *výkonného skriptu*. Program *pupdate* je spouštěn s těmito parametry:

`pupdate` *cílový-stroj jméno-služby datové soubory ...*

Protějškem na řízeném stroji je démon ***perund*** spouštěný na specifickém portu z inetd. Jeho lokální konfigurační soubor určuje, které služby je dovoleno na tomto stroji spouštět a ve kterém adresáři (zpravidla lokálním) leží výkonné skripty. Spojení s *pupdate* je nejprve kerberovsky autentizováno (dovoleno je také pouze lokálně konfigurovatelnému principálu), přeneseny datové soubory, a spuštěn výkonný skript. Jeho úkolem je provést příslušnou aktualizaci služby, např. přidání záznamů do `/etc/passwd` nebo vytvoření domovských adresářů. Skript je spouštěn s identitou superuživatele, proto představuje potenciální bezpečnostní riziko, a jeho návrhu je tedy třeba věnovat patřičnou pozornost.

Potenciální bezpečnostní rizika jsou vůbec důvodem vlastní implementace *pupdate-perund* a nikoli použití standardních nástrojů (ssh apod.). Vůči externím útokům nejcitlivější složka – *perund* – představuje cca. 800 řádků jednoduchého a přehledného kódu. Apriorní bezpečnostní audit takového kódu je ve srovnání s programy určenými pro obecné použití daleko jednodušší a tedy i účinnější. Dalším argumentem je bezprostřední a striktní omezení možných akcí prováděných s idenitou superuživatele na několik málo vyjmenovaných skriptů. Posledním důvodem je „security by obscurity“ – prozatím není pravděpodobné, že by se *perund* stal terčem masových útoků a hledání známých chyb, jak je tomu v případě obecně používaných programů téměř pravidlem.

2.6 Příklad funkce služby

Pro ilustraci nyní uveďme příklad zpracování založení nového účtu na stroji **lhun**:

- vložení záznamu do ACCOUNTS — uživatel
- zápis o insert-do-accounts do tabulky JOURNAL — automaticky triggerem J_ACCOUNTS
- vzbuzení **plánovače služeb** databázovým alertem — automaticky triggerem po commitu transakce:
  - výběr záznamu o insert-do-accounts z JOURNAL
  - kontrola oproti konfigurační tabulce PREP_SERVICES zde nastaveno:

    ```
    TYP TBL          PROG
    --- ------------ -------------------------------------
    I   ACCOUNTS     plan/acc_plan.pl
    ```

  - změna je nakonfigurovaná, tedy spuštěn plánovací skript acc_plan.pl
    - plánovací skript vyhodnotí, že mají být naplánovány služby *fs,passwd,amd* (jejich seznam je zakódován v plánovacím skriptu) pro stroj **lhun**
  - naplánovány služby *fs, passwd* pro stroj **lhun** do PLAN_SERVICES, ke spuštění za 15 minut (aktuální čas 17-OCT-03 16:30:00):

    ```
    SERVICE HOST  CLUST REALM CELL  SCHEDULE           STAT RECUR DLY
    ------- ----- ----- ----- ----- ------------------ ---- ----- ---
    MESSAGE                                                 PID
    -------------------------------------------------------- ---------
    fs      lhun                    17-OCT-03 16:45:00 plan     5  30

    passwd  lhun                    17-OCT-03 16:45:00 plan     5  30
    ```

    Služba *amd* není aktivní – viz dále konfigurační tabulka CONF_SERVICES.

- **dispečer služeb** pracující nad PLAN_SERVICES se vzbudil časovačem:
  - nenašel nic ke spuštění
    n-krát

- po uplynutí 15 min. – dispečer služeb našel ke zpracování služby *passwd, fs* pro stroj **lhun**
- kontrola oproti konfigurační tabulce CONF_SERVICES, zde nastaveno:

  ```
  SERVICE HOST  CLUST REALM CELL  ENABLE  RECUR DLY ORD PROG
  ------- ----- ----- ----- ----- ------  ----- --- --- -----------
  passwd  lhun                       Y        5  30  D  gen/passwd
  amd     lhun                       N        5  30  D  gen/amd
  fs      lhun                       Y        5  30  D  gen/pbs
  ```

  služba *amd* není pro **lhun** aktivní, naplánované služby budou provedeny.
- spuštění služebního programu *passwd*
  - vytvoření souborů `passwd` a `group`, obsahujících seznam potřebných informací o uživatelích a seznam UNIXových grup a jejich členů.
  - zavolání ***pupdate***
  - překopírování souborů `passwd` a `group` na stroj **lhun**
  - spuštění výkonného skriptu ***passwd*** na vytvoření nového `/etc/passwd` a `/etc/group`.
- služba skončila úspěšně – vyznačení v tabulce PLAN_SERVICES – status *done*
- spuštění služebního programu *fs*
  - vytvoření souborů potřebných pro rekonstrukci filesystemu.
  - zavolání ***pupdate***
  - překopírování souborů na stroj **lhun**
  - spuštění výkonného skriptu ***fs*** pro kontrolu a doplnění filesystemu
- služba skončila neúspěšně:
  - snížení počtu dalších možných opakování služby o 1, položka *recur*
  - status služby změněn na *open*, otevřený, neukončený
  - opakování spuštění služby naplánováno za 30 minut (dle CONF_SERVICES)
  - zapsání chybové zprávy
  - update záznamu v PLAN_SERVICES

- stav v tabulce PLAN_SERVICES pak vypadá následovně:

```
SERVICE HOST  CLUST REALM CELL  SCHEDULE           STAT RECUR DLY
------- ----- ----- ----- ----- ------------------ ---- ----- ---
MESSAGE                                                 PID
------------------------------------------------------- ---------
fs      lhun                    17-OCT-03 17:16:00 open     4  30
ther errors: connect(): Connection timed out               33315
amd     lhun                    17-OCT-03 16:45:00 plan     5  30
```

```
passwd  lhun                    17-OCT-03 16:45:00 done      5  30
                                                                33319
```

2.7 Struktura databáze

Obsah databáze lze rozdělit na aplikační a pracovní data. Aplikační část databáze modeluje spravované prostředky, uživatele, jejich účty apod. Pracovní část dat slouží k zachycení vývoje dat, ke konfiguraci a parametrizaci některých činností systému.

Databázoví uživatelé (tj. administrátoři systému, viz 2.7.3 a 5.1) mají možnost vytvářet vlastní, na společné schéma navazující tabulky ve svém vlastním jmenném prostoru. Takové tabulky ovšem nebudou obecně přístupné plánovacím a služebním skriptům ani dalším uživatelům, proto je jejich využití přesahující ad-hoc řešení konkrétních problémů problematické.

Určení jednotlivých databázových tabulek a význam jejich sloupců jsou zachyceny přímo v databázových komentářích a celá struktura databáze je přístupná přes webový interface (viz 5.5.1). Následující text popisuje data podle jejich vzájemné souvislosti.

2.7.1 Aplikační data

2.7.1.1 Uživatelé

Údaje o uživatelích jsou především uchovávány v tabulce PEOPLE. Zde jsou k číselnému identifikátoru uživatele přiřazeny obvyklé personální údaje jako jméno, příjmení, titul, rodné číslo, adresa, e-mail a telefon.

Příslušnost uživatele k organizaci je zachycena buďto vyplněním položky *org*, která nese identifikátor organizace ukazující do tabulky organizací (viz dále), nebo, není-li organizace v systému známa, je její název uveden v položce *org_other*. V tabulce PEOPLE jsou také uloženy informace o příslušnosti k buňce, příslušnost k ACL (viz 2.7.2.1), doplňující informace atd.

K této tabulce se váže tabulka LOGINS, ve které je uloženo jedno i více lognames pro každého uživatele – to dovoluje oddělit fyzickou identitu uživatele od potenciálně více identit elektronických.

Tabulka UIDS nese UNIXové a AFS identifikátory uživatelů v jednotlivých buňkách. Existence uživatele v AFS je zaznamenána v tabulce PTS_USERS.

Nepřímo s tabulkou uživatelů souvisí ještě tabulka ORGANIZATIONS, která uchovává seznam organizací spolupracujících s *META Centrem*, resp. jejichž členové mají v *META Centru* účty.

2.7.1.2 Přihlášky

Tabulka REGISTRATIONS obsahuje údaje o přihláškách uživatele. Kromě identifikátoru uživatele je to číslo přihlášky, typ přihlášky, stav zpracování přihlášky, doba platnosti, případně požadovaná disková kapacita na AFS, příslušnost k nějakému projektu *META Centra* a doplňující popisné údaje.

Souvisejícími tabulkami jsou HOST_REGISTRATIONS, která registruje požadavky na zřízení účtu na strojích a HOST_EXTRAS, která obsahuje nestandardní požadavky pro přihlášky na stroje. Tabulka HREG_EXTRAS navazuje zvláštní požadavky pro stroj k přihláškám.

Obdobou jsou tabulky CLUST_REGISTRATIONS, která registruje požadavky na zřízení účtu na clusterech a CLUST_EXTRAS, která obsahuje nestandardní požadavky pro přihlášky na clustery. Tabulka CREG_EXTRAS navazuje zvláštní požadavky pro cluster k přihláškám.

2.7.1.3 Účty

Informace o uživatelských účtech jsou uloženy v tabulce ACCOUNTS. Tato tabulka obsahuje:

- identifikátor uživatele (klíč do tabulky LOGINS)
- název stroje nebo clusteru, kde je účet zřízen (klíče do HOSTS nebo CLUSTS)
- dobu platnosti účtu
- příslušnost ke skupině
- uživatelův shell
- příznak dočasného zákazu účtu (Y/N)
- nepovinné údaje
  - filesystem, kde se nachází domovský adresář, a jméno adresáře (klíč do DIRECTORIES)
  - AFS domovský volume a AFS buňku volumu
  - explicitní klíč a volby pro generování amd mapy
  - poznámku

  Tyto údaje podléhají dodatečným integritním omezením – není-li určen domovský adresář na nějakém filesystemu, je vyžadován AFS volume.

2.7.1.4 Stroje

Informace o strojích, registrovaných v systému nese tabulka HOSTS. Tabulka obsahuje především identifikátor stroje, což je v podstatě jeho stručný název, příslušnost k buňce, příslušnost ke kerberovskému realmu, úplné jméno stroje a řídící ACL (viz 2.7.2.1).

Dále příznak, zda je stroj pod správou systému Perun, či cizí. Tyto vyjmenované údaje jsou povinné.

Doplňující údaje jsou verze operačního systému a popisy.

Další doplňující informaci nese tabulka SHELLS_ON_HOSTS, která uchovává seznam používaných shellů na daném stroji.

2.7.1.5 Clustery

V tabulce CLUSTS jsou uloženy informace o skutečných i fiktivních clusterech spravovaných systémem.

Systém Perun rozlišuje clustery „homogenní“ a „nehomogenní“. Ke každému ze členů „homogenního“ clusteru se přistupuje zcela identickým způsobem, jakoby jeho stroje byly naprosto stejné. Ke členům „nehomogenního“ clusteru se přistupuje v podstatě jako k jednotlivým různým strojům, které byly za nějakým účelem sdruženy do clusteru.

Tabulka obsahuje identifikátor clusteru, což je jeho stručný název, úplné jméno clusteru a řídící ACL (viz 2.7.2.1). Nepovinné jsou popisné údaje a priorita pro případ kolize (např. při vytváření účtu).

Další doplňující informaci nese tabulka SHELLS_ON_CLUSTS, která uchovává seznam používaných shellů na daném clusteru.

Protože clustery jsou složeny z jednotlivých strojů, je tato skutečnost zachycena v tabulce CLUSTS_HOSTS, která obsahuje pro každý stroj, jenž přísluší do určitého clusteru jeden záznam, nesoucí jméno tohoto stroje a jméno příslušného clusteru.

Obdobně je koncipována tabulka CLUSTS_CLUSTS, umožňující sdružit clustery do jednoho nadřazeného clusteru.

2.7.1.6 Realmy

Tabulka REALMS uchovává kerberovské realmy. Obsahuje povinně identifikátor realmu, což je jeho zkrácený název, dále úplný název a příslušnost k ACL (viz 2.7.2.1).

K této tabulce se váže tabulka LOGINS_IN_REALMS (také pod synonymem PRINCIPALS), která zachycuje příslušnost uživatelů k jednotlivým realmům.

2.7.1.7 Buňky

Buňky se v systému využívají jednak pro AFS, jednak jako administrativní jednotky. Záznamy o nich se vytvářejí do tabulky CELLS. Tato tabulka obsahuje povinně identifikátor buňky, příslušnost ke kerberovskému realmu a úplné jméno buňky. Nepovinně pak identifikátor kontaktní osoby pro administrativní účely.

K buňkám se vážou AFS volumy, které jsou ukládány do tabulky VOLUMES. Kromě buňky a názvu volumu je v této tabulce uchovávána vazba na uživatele, přidělená velikost diskové kapacity, případně stroj a partition, kde se má volume vytvořit službou *vol*. Nepovinně též příslušnost k projektu a popis. K tabulce VOLUMES se ještě ukládá informace o AFS mountpointech v tabulce AFS_MOUNTS.

2.7.1.8 Filesystemy

V tabulce FILESYSTEMS se uchovávají filesystemy. Zde je potřeba rozlišit, že filesystem, definovaný pro jednotlivý stroj, je skutečný, existující filesystem, kdežto filesystem definovaný pro cluster je vlastně pouze šablonou, jak má být filesystem vytvořen na každém ze strojů clusteru.

Povinné údaje této tabulky jsou pouze identifikátor, čili jméno filesystemu, jméno stroje, kde se filesystem nachází nebo jméno clusteru, kde se filesystem nachází replikovaně na každém stroji, dále absolutní cesta k filesystemu. Dále se udává účel, k němuž filesystem slouží (home/scratch/extra) – touto hodnotou se řídí www rozhraní i služební skripty.

K filesystemům se úzce váže tabulka QUOTAS, která uchovává diskové kapacity na daném filesystemu, přidělené určitým uživatelům.

Další související tabulkou jsou DIRECTORIES, kde jsou popsány jednotlivé adresáře daného filesystemu a jejich vlastníci.

Tabulka EXPORT_PARAMS se využívá zejména k ukládání parametrů pro NFS export.

2.7.1.9 Skupiny

Skupiny jsou uloženy v tabulce GROUPS včetně řídícího ACL (viz 2.7.2.1), případně popisu účelu, pro který je ta která skupina zřízena.

Tabulka GROUP_MEMBERSHIPS zachycuje příslušnost uživatelů ke skupině.

Tabulky GROUPS_ON_HOSTS a GROUPS_ON_CLUSTS definují primární skupiny (v /etc/passwds) pro stroje a clustery. Obdobně tabulka GRP_CLUSTS_HOSTS definuje, na kterých strojích a clusterech se má daná skupina vytvořit (v /etc/group).

Potenciální existence skupiny v rámci AFS (tj. jako pts skupina) je zaznamenána v tabulce PTS_GROUPS.

2.7.1.10 Projekty

Tabulka PROJECTS udržuje informace o projektech *META Centra*. Kromě identifikátoru, tedy zkráceného názvu projektu, obsahuje jeho úplný název, popis, řešitelskou organizaci, příslušnost k ACL (viz 2.7.2.1) a další, doplňující informace.

Řešitelé a hlavní řešitel daného projektu jsou definováni tabulkou PEOPLE_PROJECTS.

Tyto tabulky jsou v současnosti využívány marginálně, komplexní správa projektů je předmětem plánovaných rozšíření v nejbližší době.

2.7.2 Pracovní data

2.7.2.1 Řízení přístupu

Mezi nejdůležitější pracovní data patří ACL (Access Control List), uložený v tabulce ACLS. Byly vytvořeny za účelem zpřístupnění určitých skupin dat pouze určitým uživatelům-administrátorům. Jak bylo uvedeno výše, v řadě aplikačních tabulek je tedy vyznačena příslušnost k určitému ACL. Pokud dotyčný uživatel, který s daty právě pracuje, je v daném ACL oprávněn (tabulka ACLS_LOGINS), jsou pro něj data přístupná, pokud ne, jsou nepřístupná. Takto je zajištěno, aby každý uživatel-administrátor mohl manipulovat pouze s daty, za něž nese odpovědnost.

2.7.2.2 *Vyhodnocení změn a plánování služeb*

Další důležité pracovní tabulky jsou JOURNAL, která monitoruje změny v aplikačních datech a PLAN_SERVICES, která drží údaje o plánování a realizaci služeb (jak bylo podrobně popsáno v kapitolách 2.2 – 2.4). Sem patří rovněž tabulky PREP_SERVICES a CONF_SERVICES, které slouží ke konfiguraci služeb.

2.7.2.3 *Specifické tabulky pro služby*

Tabulky CLUST_PARAMS a HOST_PARAMS nesou mimo jiné také některé další parametry pro výkon služeb.

Tabulka UPDATE_DELAY obsahuje časový parametr uživatele, který udává zpoždění služeb po provedení změn uživatelem (viz 2.3.2).

Čistě interní použití mají tabulky pro dočasné uchování změn LOGINS_IN_REALMS_CHANGE a VOLUMES_CHANGE.

2.7.3 Přístup k datům

Všechny tabulky v databázi patří systémovému uživateli PERUN. S touto databázovou identitou také pracují oba démoni (pjournal, pdispatch), a všechny plánovací a služební skripty.

Pro přístup ostatních uživatelů k tabulkám byla vytvořena public synonyma se stejným jménem. Uživatelé mohou přímo z tabulek jenom číst.

V odstavci 2.7.2.1 byly popsány ACL pro přístup administrátorů. Nad všemi tabulkami jsou vytvořeny pohledy, které zpřístupňují pouze ta data, ke kterým má daný databázový uživatel právo zápisu (odvozené přímo nebo zprostředkovaně od příslušné hodnoty ve sloupci ACL). Tyto pohledy tedy umožňují oprávněným uživatelům jak čtení, tak zápis. Jméno takto vytvořeného VIEW se skládá ze jména původní tabulky a přípony „_RW“ např. ACCOUNTS_RW.

3 Implementované služby

Jak bylo ukázáno výše, služby jsou základní výkonnou složkou systému Perun. V současné době jsou v systému implementovány následující služby:

Služby	**Změny v tabulkách**	**Plánovací skript**	**Služební skript**	**Soubory**	**Výkonný skript**
passwd	ACCOUNTS	acc_plan.pl	passwd	`passwd` `group`	passwd
pwd_send			pwd_send	`passwd` `group`	passwd
fs	DIRECTORIES	dir_plan.pl	fs	*stroj*.`home` *stroj*.`scratch`	fs
	QUOTAS	quot_plan.pl	fs	*stroj*.`home` *stroj*.`scratch`	fs
fs_send			fs_send	*stroj*.`home` *stroj*.`scratch`	fs
amd	ACCOUNTS	acc_plan.pl	amd	`home`	amd
pbs	ACCOUNTS	acc_plan.pl	pbs	`/host/`*jméno-stroje* `/clust/`*cluster* `/clust/`*cluster*`.hosts`	pbs
krb	PRINCIPALS	lgrlm_plan.pl	krb		
vol	VOLUMES	vol_plan.pl	vol		

Jak je patrné z tabulky, lze služby rozdělit na dva typy: takové, které pomocí programu *pupdate* (viz 2.5) přesouvají data na cílový stroj a tam vyžadují spuštění výkonného skriptu (první 4 služby v tabulce) a naopak takové, které provádějí cílové úpravy přímo – *služby přímé* (poslední 2 služby).

Pro přehlednost uvádíme také tabulku cílů jednotlivých služeb:

Služby	**Cíl služby**
passwd	stroj nebo cluster
pwd_send	stroj
fs	stroj nebo cluster
fs_send	stroj
amd	stroj
pbs	stroj nebo cluster
krb	kerberovský realm
vol	buňka

3.1 Služby s výkonným skriptem

3.1.1 Služba *passwd* a *pwd_send*

Služba ***passwd*** se aktivuje jakoukoliv změnou uživatelského účtu. Plánovací skript **acc_plan.pl** službu naplánuje na stroj nebo cluster, kde je účet zřízen.

Po aktivaci služby, tedy spuštění služebního skriptu **passwd**, se vytvoří soubory `passwd` a `group` pro daný stroj.

Pro nehomogenní clustery není tato služba plánována na cluster, ale na každý z jeho strojů zvlášť – počítá se s tím, že datové soubory se pro ně budou lišit.

Pro homogenní clustery se vytvoří tyto soubory pouze jedenkrát pro všechny stroje, neboť jsou všechny stejné, a potom se naplánuje na aktuální čas jejich rozeslání službou ***pwd_send*** na každý ze strojů clusteru.

Soubor `passwd` obsahuje navíc čas expirace účtu a e-mailovou adresu každého uživatele, který má na daném stroji účet.

Na cílovém stroji je spuštěn výkonný skript **passwd**, který na základě přenesených souborů `passwd` a `group` vytvoří nový `/etc/passwd`, `/etc/group`, `/etc/mail/aliases` a `/etc/sendmail/returns`. Pokud tedy na cílovém stroji existovali uživatelé, kteří nejsou zaneseni v databázi, budou přepsáním výše zmíněných souborů zrušeni.

3.1.2 Služba *fs* a *fs_send*

Služba ***fs*** se aktivuje změnami (insert a update) adresářů, nebo změnami diskových kapacit přidělených uživatelům. Plánovací skript **dir_plan.pl** (změny v DIRECTORIES) i **quot_plan.pl** (změny v QUOTAS) službu naplánují na stroj nebo cluster, kde je umístěn adresář či přidělena disková kapacita. Adresář na homogenním clusteru je chápán jako jeho replika na všech jeho uzlech.

Po spuštění služebního skriptu **fs** se nejprve vytvoří soubory, které popisují filesystemy na cílovém stroji. Tyto soubory popisují adresáře, jejich vlastníky a grupy, přístupová práva a přidělenou kapacitu.

Obdobně jako u služby *passwd*, vytváří se, je-li cílem nehomogenní cluster (viz 2.7.1.5), tyto soubory pro každý ze strojů clusteru, je-li cílem homogenní cluster, vytváří se soubory pouze jedenkrát a jejich rozeslání se provede pomocí služby ***fs_send*** (obdoba *pwd_send*).

Na cílovém stroji je spuštěn výkonný skript **fs**, ten zkontroluje filesystemy a jejich velikosti, založí adresáře, které chybí, a překopíruje do nich případné vzorové soubory (např. `.profile`). Pro existující adresáře pouze upraví kvóty, jinak je ponechává beze změny.

3.1.3 Služba *amd*

Služba ***amd*** se aktivuje stejně, jako služba *passwd* vytvořením nebo změnou uživatelského účtu. Plánovací skript **acc_plan.pl** službu naplánuje na stroj, kde je účet zřízen. Služební skript **amd** vytvoří příslušné amd mapy, na cílovém stroji je spuštěn výkonný skript **amd**, který původní mapovací soubory nahradí novými, které byly předány pomocí služby ***amd***.

3.1.4 Služba *pbs*

Služba ***pbs*** předává PBS serveru informace o dostupnosti uživatelských účtů na jednotlivých strojích, které jsou pochopitelně nezbytné pro správné plánování úloh.

Služba se aktivuje vytvořením nebo změnou uživatelského účtu. Plánovací skript **acc_plan.pl** službu naplánuje na stroj nebo cluster, kde je dotyčný účet zřízen.

Spuštěný služební skript **pbs** vytvoří v aktuálním adresáři (viz 2.4.2) podadresář `host` nebo `clust` a do souboru s úplným jménem stroje nebo clusteru uloží logname všech uživatelů, kteří mají na tomto stroji účet. Pro clustery je ještě vytvořen soubor se jménem clusteru a příponou ".hosts", který obsahuje seznam všech strojů v clusteru.

Tyto soubory jsou pak odeslány na PBS-server, jehož jméno je pro daný stroj či cluster uloženo jako parametr *server* v tabulce HOST_PARAMS resp. CLUST_PARAMS (viz 2.7.2.2).

Na PBS-serveru je spuštěn výkonný skript **pbs**. Ten zpracuje předané soubory a připraví je k využití pro PBS. Při tomto zpracování nejsou existující data zničena, pouze doplněna dalšími, dosud chybějícími.

3.2 Služby přímé

3.2.1 Služba *krb*

Služba ***krb*** se aktivuje změnami v příslušnosti uživatelů k realmům. Služba je plánovacím skriptem **lgrlm_plan.pl** naplánována vždy, jedná-li se o typ změny delete, pro insert pouze tehdy, pokud se jedná o zavedení skutečného uživatele do realmu, nikoliv jen o zabrání místa („placeholder“) pro uživatele, aby se zabránilo duplicitám. Pro update je rozhodování o naplánování služby poněkud komplikovanější, zkráceně však lze říct, že se služba ***krb*** naplánuje vždy, když se jedná o změnu uživatele z „placeholder“ na skutečného uživatele, nebo naopak. Plánovací skript takto vyhodnocené změny uloží do dočasné tabulky LOGINS_IN_REALMS_CHANGE v podobě příkazů pro služební skript.

Při provádění služby služebním skriptem **krb** se zavede, zruší nebo změní uživatel v cílovém kerberovském realmu na základě záznamů v dočasné tabulce. Pokud by v dotyčném kerberovském realmu existoval uživatel, který není zanesen v databázi a cílem zrovna prováděné služby není právě jeho zrušení, zůstane nedotčen.

3.2.2 Služba *vol*

Služba ***vol*** se aktivuje změnami v AFS volumech. V současné době plánovací skript **vol_plan.pl** aktivuje službu pouze pro insert a update (viz tabulka PREP_SERVICES). Plánovací skript **vol_plan.pl** zapíše také do pracovní tabulky VOLUMES_CHANGE jméno buňky, jméno volumu a příkaz pro vytvoření „c“.

Služební skript **vol** projde záznamy se stavem „c“ v tabulce VOLUMES_CHANGE pro danou buňku, zjistí jméno měněného volumu, otestuje, zda takový volume existuje, pokud ne, založí jej a namontuje (mountpoint z tabulky AFS_MOUNTS).

Dále kontroluje existenci a případně vytváří pts uživatele odpovídajícího vlastníku vytvořeného volume v dané buňce a odpovídajícím způsobem nastavuje AFS přístupová práva. Ostatní data zůstávají beze změn.

3.3 Rozšiřitelnost služeb

Počet služeb a jejich funkce není nikterak omezena a mohou být libovolně přidávány. Pro zavedení nové služby stačí pouze rozhodnout, na základě jaké změny se bude celé její zpracování spouštět, doplnit službu do konfiguračních tabulek a napsat dva programy pro naplánování a pro provedení služby, tedy plánovací a služební skript, případně výkonný skript pro cíl služby.

Pro ilustraci si nyní uveďme příklad vytvoření fiktivní služby ***birthday***, která způsobí, že se při přihlášení uživatele ke kterémukoliv stroji zobrazí gratulace k narozeninám, pakliže je právě jeho den narození. V tom případě musíme provést následující kroky:

- Zvolíme tabulku, jejíž změny budou podstatné pro spuštění služby. **Zde zvolíme tabulku ACCOUNTS** a to jen vložení nového účtu, další změny účtů nemohou mít vliv na datum narození.

- Vytvoříme záznam v tabulce PREP_SERVICES pro insert:

```
TYP TBL          PROG
--- ------------ -------------------------------------
I   ACCOUNTS     plan/birth_plan.pl
```

- Vytvoříme plánovací skript **birth_plan.pl** (viz tabulka PREP_SERVICES), který naplánuje službu ***birthday*** na stroj, kde má uživatel nový účet (zde např. na stroji `kiril`):

```
#!/usr/bin/perl -w
use DBI;
my $fi_out=$ARGV[0];
my $fi_in=$ARGV[1];
my ($user,$pwd) = split '/',$ENV{ORA_USER};
my $lin;
open (FOUT,">$fi_out") or die "$fi_out: $!\n";
open FIN, $fi_in or die "$fi_in: $!\n";
my $dbh = DBI->connect('dbi:Oracle:',$user,$pwd,
                       { RaiseError=>1, AutoCommit=>0 });
# radky vstupniho souboru
JOUR: while ($lin=<FIN>) {
    chomp($lin);
    ($sess,$stmt,$line,$tbl,$typ)=split(/\s/,$lin);
# nalezeni puvodniho radku z~JOURNAL
    ($rr, $stamp,$person) = selectrow_array $dbh q{
        select  rowidtochar(rr),
                   to_char(stamp,'J')*86400+to_char(stamp,'SSSSS'),
        from journal where sess=? and stmt=? and line=? and tbl=? }
```

```
        ,undef,$sess,$stmt,$line,$tbl;
    if (! $rr) {  warn "No JOURNAL rows found \n"; next JOUR;   }
# nalezeni identifikace uzivatele a stroje, kde je jeho novy ucet
    ($persid,$host) = selectrow_array $dbh q{
        select id,host from accounts where rowidtochar(rowid)=? },undef,$rr;
    if (! $persid) {  warn "No ACCOUNTS rows found \n"; next JOUR;   }
# prodleva v~planovani podle zpracovatele, stanoveni casu naplanovani
    my $dly = selectrow_array $dbh q{
        select delay from update_delay where person=? },undef,$person;
    if (! defined($dly)) { warn "No UPDATE_DELAY rows found\n"; $dly=30; }
    $stamp=$stamp+$dly*60; # cas zmeny + prodleva (v tabulce v~min)
# do vystupniho souboru - zapis casu planovani,jmena sluzby, jmena stroje
    if($host) { print FOUT "$stamp birthday $host H\n";} # jen pro stroje
} # vsechny radky
close (FOUT);
disconnect $dbh;
close (FIN);
```

- Přidáme záznamy pro službu ***birthday*** do CONF_SERVICES pro všechny stroje, kde si přejeme službu v budoucnu uplatnit (zde např. na stroje *kiril* a *lhun*):

```
SERVICE  HOST    CLUST REALM CELL  ENABLE  RECUR DLY ORD PROG
-------- ------- ----- ----- ----- ------  ----- --- --- -------------
birthday kiril                       Y         5  30  D  gen/birthday
birthday lhun                        Y         5  30  D  gen/birthday
```

- Napíšeme služební skript **birthday** (viz CONF_SERVICES), který vytvoří soubor `birthday.data`, jenž obsahuje lognames, jména, příjmení a data narození všech uživatelů zadaného stroje, zavolá program *pupdate* (viz 2.5), přesune data na cílový stroj a vyžádá si spuštění výkonného skriptu **birthday**:

```
#!/usr/bin/perl -w
use DBI;
my $host = $ARGV[0];
my ($user,$pwd) = split '/',$ENV{ORA_USER};
my $dbh = DBI->connect('dbi:Oracle:',$user,$pwd,
         { RaiseError=>1, AutoCommit=>0 })   or die " Connect";
open BRNO,">birthday.data" or die " open birthday.data $!\n";

# vyber vsech uzivatelu na zadanem stroji
my $acc = $dbh->prepare(qq{
```

```
                select first_name,last_name,birth_no,logname
                from people p,accounts a,logins l
                where p.id=a.person and p.birth_no is not null and
                      a.person=l.person and a.login=l.seq and a.host=? });
$acc->execute($host);
# pro kazdeho uzivatele
while (my $acc = $acc->fetch) {
        my $rok=substr($$acc[2],0,2);  # rodne cislo
        my $mes=substr($$acc[2],2,2);
        my $den=substr($$acc[2],4,2);
        if ($mes>50) {$mes=$mes-50;}   # devcata
      # logname : prijmeni : krestni jmeno : datum narozeni
        print BRNO "$$acc[3]:$$acc[0]:$$acc[1]:$rok-$mes-$den\n";
}
close BRNO;
# nalezeni hostname
my $hname = selectrow_array $dbh q{
               select hostname from hosts where id = ?},undef,$host;
$dbh->disconnect;
# presun na cilovy stroj, aktivace vykonneho skriptu
exec "pupdate", $hname, "birthday", "birthday.data";
```

- Na cílové stroje připravíme ke spouštění výkonný skript **birthday**, který projde `.profile` soubory všech uživatelů a pokud v nich ještě není gratulační sekvence, přidá ji na konec souboru:

```
#!/usr/bin/perl -w
my ($lin);
if (! $ARGV[0]) die "data file required\n";
open (BRNO, $ARGV[0]) or die "err open input file\n";

# vsichni uzivatele stroje
while ($lin=<BRNO>) {
        chomp($lin);
        my ($logname,$jm,$prij,$datnar) = split (':',$lin);
        my ($rok,$mes,$den) = split ('-',$datnar);
      # nalezeni charakteristicke skupiny znaku
        my $je = system "fgrep '###>>>###' /home/$logname/.profile";
        if (! ($je>>8)) {
            open (PROF,">>/home/$logname/.profile") or die "open profile:$!\n";
          # gratulacni sekvence
            my $gratul=qq{
```

```
                    ###>>>###
                    mes=`date +%m`
                    den=`date +%d`
                    if [ \$mes -eq $mes ] && [ \$den -eq $den ]
                    then echo H A P P Y   B I R T H D A Y
                    fi
                    ###<<<### };
                print PROF $gratul;
                close PROF;
            }
    }
    close BRNO;
```

Gratulace se pak vypíše při přihlášení uživatele ke stroji, má-li zrovna narozeniny.

Z výše uvedené ukázky je vidět, že bez velké námahy lze tímto způsobem realizovat víceméně libovolnou činnost.

3.4 Výstup pro LDAP

Výstup pro LDAP se bude realizovat jako další služba ***metaldif***. Jako významná změna bude nadefinováno vložení, oprava a zrušení v tabulce ACCOUNTS. Záznam v tabulce PREP_SERVICES pak bude vypadat takto:

```
TYP TBL          PROG
--- ------------ -------------------------------------
I   ACCOUNTS       plan/meta_plan.pl
U   ACCOUNTS       plan/meta_plan.pl
D   ACCOUNTS       plan/meta_plan.pl
```

Plánovací skript *meta_plan.pl* naplánuje službu ***metaldif*** pro cluster *meta*.
V konfigurační tabulce CONF_SERVICES bude třeba vytvořit jediný záznam:

```
SERVICE  HOST    CLUST REALM CELL  ENABLE  RECUR DLY ORD PROG
-------- ------- ----- ----- ----- ------  ----- --- --- -------------
metaldif         meta                 Y        5  30  D  gen/metaldif
```

Služební skript *metaldif* vytvoří kompletní LDIF pro cluster *meta* a odešle jej stanovenému cílovému stroji.

Vyhodnocení úspěšnosti služby ***metaldif*** bude probíhat stejně, jako u všech ostatních služeb.

4 Informační služby *META Centra* – publikování informací

Informační služby *META Centra* čerpají informace ze systému Perun a umožňují přístup k nim přes jednotné rozhraní (adresářové služby, konrétně LDAP). Tato kapitola se zabývá publikováním informací o uživatelích *META Centra* a to zejména z hlediska struktury publikovaných dat. Obecnější informace o adresářových službách *META Centra* a jejich využití lze nalézt v ([Sit01]), v přípravě je technická zpráva popisující aktualizaci dotyčných informačních služeb včetně popisu změn důležitých z uživatelského pohledu ([Sit03]).

4.1 Základní změny v rámci aktualizace informačních služeb

V rámci aktualizace informačních služeb došlo k těmto základním změnám důležitým z hlediska systému Perun:

- *Změna schématu* – úprava a rozšíření LDAP schématu pro prezentaci údajů o uživatelích. Používané schéma je rozšířením standardního schématu o specifické atributy. Cílem provedené úpravy je využít co nejvíce standardních schémat a tím zvýšit kompatibilitu. Využito bylo zejména schéma `eduPerson` z projektu Internet2 (v oblasti informací o lidech) a rfc2307 (v oblasti informací o uživatelských kontech).

- *Propagace češtiny* – situace s využíváním české diakritiky u klientů adresářových služeb není stále úplně růžová, použité řešení je kompromisní a umožňuje zachovat kompatibilitu a přitom zpřístupnit klientům odpovídající skutečně české informace.

- *Mechanismus distribuce dat z databáze systému Perun* – tato úprava, nezávisle na struktuře dat, umožní zabezpečit propagaci změn databáze do LDAPu v reálném čase. Pro některé klienty a aplikace může tato funkcionalita znamenat značné zlepšení služeb informační infrastruktury.

4.2 Schéma dat

LDAP schéma dat pro zobrazení informací o uživatelském účtu (a zároveň člověku) je popsáno v tabulce. Sloupec schéma je označení schématu ve kterém je příslušný atribut definován. Pokud je zde uvedeno `MetaPerson`, jedná se o specifický atribut definovaný pro účely *META Centra* (více viz [Sit03]).

Použité třídy (objectclass): `inetOrgPerson, eduPerson, MetaPerson, posixAccount, shadowAccount`.

Jméno atributu, alias	schéma	význam
commonName, cn	inetOrgPerson	jméno a příjmení osoby
displayName	inetOrgPerson	jméno a příjmení osoby
surname, sn	inetOrgPerson	příjmení osoby
givenName	inetOrgPerson	jméno osoby
telephoneNumber	inetOrgPerson	telefonní číslo
mail	inetOrgPerson	e-mail adresa (dle RFC 822)
postalAddress	inetOrgPerson	„snail mail" adresa; může obsahovat několik řádek oddělených znakem `$`
organizationName, o	inetOrgPerson	Plné jméno organizace, ke které uživatel přísluší
uid	rfc2307	identifikace člověka jakožto uživatele – uživatelské jméno
homeCell	MetaPerson	identifikace domovské buňky v rámci *META*
homeDirectory	rfc2307	cesta k domovskému adresáři v globálním jmenném prostoru
loginShell	rfc2307	implicitní shell uživatele
userPassword	rfc2307	položka password pro passwd záznam uživatele, v našem případě `*`
gidNumber	rfc2307	Unix číslo primární skupiny uživatele, odpovídá příslušné položce `passwd`
uidNumber	rfc2307	Unix číslo uživatele, odpovídá příslušné položce `passwd`
status	MetaPerson	stav uživatele (platný/blokovaný) – viz níže
shadowExpire	rfc2307	předpokládaná platnost uživatelského konta v *META Centru*; formát `/etc/shadow`, tj. počet dnů od 1. ledna 1970
eduPersonPrincipalName	eduPerson	V případě *META Centra* plné jméno Kerberos principalu uživatele.

Poznámky:

- `displayName` je u nás totéž co `cn`. Někteří klienti používají `displayName` ve výpisech seznamů lidí (tel. seznam) namísto CN, protože CN může obsahovat ledaco jiného než jméno a příjmení. Může obsahovat také prostřední jméno a tituly.
- `uidNumber` může být dostupné pouze u některých uživatelů; pro použití s NSS_LDAP je vyplnění `uidNumber` nezbytné.
- Jak `uidNumber` tak `gidNumber` jsou hodnoty obecně lokální vůči buňce *META Centra*, tj. jeden uživatel může mít v každé buňce jiné Unix UID a GID. Tento fakt je vyjádřen pomocí tzv. voleb atributu (attribute options nebo také user tags). Atribut má tedy více hodnot, přičemž hodnoty jsou označeny volbou určující pro kterou buňku platí. Aktuální použité volby jsou `x-meta-`<*buňka*>, více viz následující příklad a diskuze problematiky voleb atributů v [Sit03].

- Atribut `shadowExpire` by měl být standardně kontrolován klienty dle RFC2307 (např. PAM moduly). Jeho hodnota by měla korespondovat s atributem `status`, který znají pouze *META Centru* specifické aplikace (tj. zejména hodnota v budoucnu by měla být pouze u stavu ACTIVE).

Sémantika hodnot atributu `status`, který je specifický pro *META Centrum* je definována v následující tabulce.

Hodnota atributu `status`	význam
ACTIVE	aktivní uživatel
DISABLED	pozastavený přístup
EXPIRED	uživatelské konto je již neplatné (záznam slouží pro vyhodnocení accountingu apod.)

4.2.1 Tvorba DN

DN uživatelských položek bude tvořeno pouze atributem `uid`, který odpovídá jednoznačnému uživatelskému jménu v rámci *META Centra*. Složené DN ze dvou atributů (znak plus v DN) je vlastností stále méně podporovanou klienty a přítomnost `cn` v DN není nezbytná.

4.2.2 Kořen LDAP stromu

Umístění informací o uživatelích je `ou=People,o=meta,c=cz`. Použití komponent `dc`, které je v současné době považováno za standard, je v našem případě problematické. Jedná se totiž o využití existující infrastruktury DNS a přímou vazbu na ni. A *META Centrum* nemá DNS doménu `meta.cz` (ta navíc existuje!). Možným řešením je použití `dc=meta,dc=cesnet,dc=cz`, což ovšem vyžaduje odpovídající rozhodnutí v oblasti DNS.

4.2.3 Čeština

Celý obsah adresářové služby *META Centra* je primárně česky, nicméně bez háčků a čárek. Změna tedy neznamená zavedení češtiny (ale ani vícejazyčnosti), ale zavedení české diakritiky.

Aktuální stav je kompromisem, který dovoluje vytvářet klienty plnohodnotně pracující s českou diakritikou a přitom je transparentní pro starší klienty. Podorobnější popis a diskuze této tématiky viz [Sit03].

Atributy, které mají textový obsah obsahující češtinu jsou uvedeny s několika *language tagy*.

- Atribut bez tagu je cesky (ASCII ekvivalent).

- Atribut s tagem `lang-cs` je česky – diakritika ve znakové sadě Unicode, kódování UTF-8.

- Atribut s tagem `lang-cs-ascii` je cesky (tj. ASCII ekvivalent).

Další poznámky:

- Při hledání (není-li explicitně uveden tag) se prohledávají hodnoty všech tagů daného atributu. Tj. položku najdete ať hledáte s češtinou či bez češtiny. Pokud chcete najít jen položky které vyhovují přesně včetně diakritiky, je nezbytné použít tag.
- Pro takovou konfiguraci platí, že nezařídíte-li explicitně použití atributu s tagem `lang-cs`, jsou použité výsledky ASCII. Při hledání s češtinou (pokud klient použije správně UTF-8) se příslušné položky najdou. Takto se např. chová MS Outlook Express (verze 4.5 a vyšší) zatímco Pine s UTF-8 nepracuje, nelze ho pro češtinu vůbec použít (nelze do hledaného řetězce napsat české znaky, protože budou hledány jako latin2).

4.2.4 Vzorek LDIF reprezentace uživatele

Vzorek LDIF reprezentace uživatele

```
dn: uid=sitera, ou=People, o=meta, c=cz
objectclass: inetOrgPerson
objectclass: eduPerson
objectclass: MetaPerson
objectclass: posixAccount
objectclass: shadowAccount
cn: Jiri Sitera
cn;lang-cs: Jiří Sitera
cn;lang-cs-ascii: Jiri Sitera
uid: sitera
sn: Sitera
sn;lang-cs: Sitera
sn;lang-cs-ascii: Sitera
givenName: Jiri
givenName;lang-cs: Jiří
givenName;lang-cs-ascii: Jiri
displayName: Jiri Sitera
displayName;lang-cs-ascii: Jiri Sitera
displayName;lang-cs: Jiří Sitera
telephoneNumber: 37763 2845
postalAddress: Univerzitni 22, 30614 Plzen
postalAddress;lang-cs: Univerzitní 22, 30614 Plzeň
postalAddress;lang-cs-ascii: Univerzitni 22, 30614 Plzen
organizationname: Zapadoceska univerzita v~Plzni
organizationname;lang-cs: Západočeská univerzita v~Plzni
organizationname;lang-cs-ascii: Zapadoceska univerzita v~Plzni
mail: sitera@civ.zcu.cz
```

```
eduPersonPrincipalName: sitera@META
homeCell: zcu.cz
homeDirectory: /afs/zcu.cz/users/s/sitera/home
loginShell: /bin/tcsh
status: ACTIVE
shadowExpires: -1
userPassword: {crypt}*
gidNumber;x-meta-ics.muni.cz: 1000
gidNumber;x-meta-ruk.cuni.cz: 100
gidNumber;x-meta-zcu.cz: 100
uidNumber;x-meta-ics.muni.cz: 23456
uidNumber;x-meta-ruk.cuni.cz: 1437
uidNumber;x-meta-zcu.cz: 1001
```

5 Uživatelské rozhraní

5.1 Přístup k datům

K informacím o zdrojích *META Centra* lze přistupovat z několika různých hledisek. Z hlediska dostupnosti lze k datům přistupovat přes MetaPortál resp. jeho aktivní část, nebo lze s daty manipulovat z příkazové řádky shellu prostřednictvím několika k tomu určených programů.

Pro činnosti, které se v systému provádějí pouze vyjímečně, nebo velmi zřídka je ponechán přístup přes SQL.

Z hlediska bezpečnosti lze přístup k datům rozdělit na přístup uživatelský a přístup administrátorský. Přístupem anonymním lze nazvat první přístup uživatele k MetaPortálu.

5.1.1 Anonymní přístup

Přístup osoby, která dosud nemusí být uživatelem, zaregistrovaným v *META Centru*. Takový uživatel nemá ještě přiděleno logname ani vstupní heslo, neudal žádné osobní údaje ani nepodal přihlášku, ale právě tak hodlá učinit.

5.1.2 Uživatelský přístup

Řadoví uživatelé mají přístup pouze k některým datům, především těm, která se vážou k jejich osobě a účtům. Ostatní data jsou přístupná jenom informativně, nebo nejsou dostupná vůbec.

Pro všechny řadové uživatele je v databázi vytvořen jediný společný databázový uživatel. Přístup řadových uživatelů k datům je zabezpečen autentizací proti danému kerberovskému realmu (META).

5.1.3 Administrátorský přístup

Administrátorský přístup umožňuje prohlížení i práci s většinou informací, uložených v databázi systému. Administrátorský přístup však neopravňuje uživatele ke změnám ve struktuře databáze. Každý uživatel-administrátor má vytvořena vlastního databázového uživatele, na nějž je přímo mapován principál v administrativním realmu ADMIN.META.

Jak bylo popsáno v kapitole 2.7.2, řada aplikačních dat je rozdělena do tzv. ACL. Každý uživatel-administrátor přísluší k jednomu či více těchto ACL a určitému administrátorovi je tak dovoleno manipulovat pouze s daty, patřícími k některému z ACL, k nimž je oprávněn.

5.2 Struktura www rozhraní

K aplikačním datům systému Perun lze především přistupovat přes *Aktivní část* MetaPortálu. Nabídkové menu vypadá po rozvinutí zhruba takto:

- ***Přihláška*** - anonymní přístup
  - *Přihláška* – vstup údajů nezbytných k podání přihlášky do $\mathcal{META}$ *Centra*.
  - *Kontrola stavu a storno přihlášky* – uživatel může zjistit v jakém stadiu zpracování se aktuálně nacházejí jeho přihlášky (*podaná*/*přijatá*/*vyřízená*), podle potřeby může přihlášku zrušit.

    Tento přístup je zabezpečen heslem zadaným v přihlášce, které je dočasně před vytvořením účtu uloženo v tabulce PASSWORDS.
  - *Kontaktní informace* – kontakt na osoby, které mají na starosti příjem papírových přihlášek a dalších souvisejících dokumentů a jejich odsouhlasení s elektronickou formou.
- ***Osobní administrativa*** – uživatelský přístup
  - *Osobní údaje* – kontrola a oprava osobních údajů, které byly vyplněny do přihlášky.
  - *Podané žádosti* – přehled přihlášek, podaných právě pracujícím uživatelem, zobrazení jejich stavu, možnost tisku.
  - *Výroční zpráva o prodloužení účtu* – žádost o prodloužení účtu na následující kalendářní rok a její zdůvodnění.
  - *Projekty* – práce s projekty je v současné verzi systému ve ztádiu rozpracovanosti. Uživatel může vytvořit nebo editovat nový projekt, jehož bude hlavním řešitelem, nebo se může přidat k některému již existujícímu projektu, eventuelně se ze zvoleného projektu vyjmout.
  - *Aktivace účtů* – uživatel může svůj účet na některém stroji učinit dočasně neaktivním, pokud se z nějakého důvodu rozhodne jej nepoužívat, nebo jej může opětovně aktivovat, pokud takový důvod již pominul.

- *Žádost o další účty* – uživatel může vybírat ze strojů a clusterů, na kterých dosud nemá zřízen účet.

- ***Administrativní část MetaPortálu***
  - *Přihlášky* – administrátorský přístup
    - *Přehled přihlášek* – umožňuje zadat uživatele a zobrazit přehled jeho přihlášek, eventuelně vytisknout vybranou přihlášku.
    - *Přijetí papírové přihlášky* – odsouhlasení údajů v přihlášce s tištěnou formou, potvrzení správnosti a úplnosti potřebných dokumentů. Provádí oprávněná osoba, nikoli nutně administrátor.
    - *Zpracování přihlášky MetaCentra* – skutečné přijetí přihlášky administrátorem, tj. vytvoření záznamu v informačních službách a kerberovi.
    - *Vytvoření účtů na základě přihlášky* – založení účtů na požadovaných strojích či clusterech, buďto na základě přihlášky, nebo na základě žádosti o další účty.
    - *Zpracování žádostí o prodloužení účtu* – administrátorské odsouhlasení prodloužení platnosti účtů na jednotlivých strojích na základě žádosti podané uživatelem.
  - ***Rozeslání hromadné zprávy*** – nástroj, který umožňuje rozeslat e-mailovou zprávu vybrané cílové skupině.
  - ***Perun*** – administrátorský přístup
    - *Seznam ovládaných strojů* – pro zobrazení a editaci všech strojů pod správou systému.
    - *Přístupy do databáze* – zobrazení a editace většiny zdrojů *META Centra*, uložených v aplikačních tabulkách.
    - *Historie změn databáze* – umožňuje dohledání takových změn ve významných databázových tabulkách, jež se ukládají do tabulky JOURNAL (viz 2.2 – 2.4).
    - *Přehled stavu služeb* – poskytuje informaci o stavu služeb na konkrétním zdroji, umožňuje provést okamžité vypropagování služby (tj. naplánování služby k okamžitému provedení – viz 2.3.3 – 2.4), nebo též změnit konfigurační parametry.
    - *Struktura databáze* – detailní struktura databáze, včetně vazeb (constraints, checks) a triggerů.

Poznámka: Prázdné kroužky v menu představují nápisy, které nelze prokliknout na spuštění programu či podmenu.

5.3 Frekventované činnosti

5.3.1 Jak se stát uživatelem *META Centra*

- **Uživatel** – bez jakékoliv autentizace vyplní přihlášku do *META Centra*, volba ***Aktivní část-Přihláška-Přihláška*** (viz 5.2). Uživatel vyplní do přihlášky běžné personální údaje a příslušnost k organizaci. Dále může zadat svoji příslušnost k některému z projektů, řešených v rámci *META Centra*. Uvede požadované logname, heslo pro první přihlášení, domovskou AFS buňku, požadovanou diskovou kapacitu a dobu expirace. Kromě toho může v rámci přihlášky požádat o zřízení vybraných účtů.

  Zadáním přihlášky se v systému Perun vytvoří nový uživatel, je zavedeno jeho logname a zaregistrovány jeho požadavky na přidělení účtů a dalších zdrojů. Rovněž je zařazen do tabulky kerberovských realmů, prozatím jen jako „placeholder“ tj. je mu rezervováno vybrané logname, aby nemohlo dojít k duplicitě.

  Prvotním heslem je potom chráněn přístup k jeho osobním informacím a údajům o stavu přihlášky až do jeho změny uživatelem.

- **Sekretářka**, nebo jiná pověřená osoba – provede přijetí přihlášky, které se děje již v administrátorské části MetaPortálu, volba ***Aktivní část-Administrativní část MetaPortálu-Přijetí papírové přihlášky***. Osoba, která přihlášku přijímá, se musí autentizovat svým uživatelským jménem a heslem pro přístup do databáze. Odsouhlasí-li přijetí přihlášky, přihláška se převede ze stavu *podaná* do stavu *přijatá*.

  Hlavním smyslem tohoto kroku je kontrola shody papírové, uživatelem a jeho nadřízeným podepsané verze přihlášky s její elektronickou formou.

- **Administrátor** dále provede:

  – zpracování přihlášky volbou ***Aktivní část-Administrativní část MetaPortálu-Zpracování přihlášky MetaCentra***. Když administrátor přihlášku potvrdí, vytvoří se pro nového uživatele domovský AFS volume, účet na clusteru „meta“, v tabulce kerberovských realmů se změní příznak „placeholder“ na „N“ (viz výše).

  – vytvoření účtů volbou ***Aktivní část-Administrativní část MetaPortálu-Vytvoření účtů na základě přihlášky***. Pokud administrátor vytvoří uživateli účty na všech požadovaných strojích či clusterech, přejde přihláška ze stavu *přijatá* do stavu *vyřízená*.

5.3.2 Osobní údaje

- Osobní údaje jsou vkládány při vyplnění přihlášky do *META Centra* (5.3.1).

- Volba ***Aktivní část-Osobní administrativa-Osobnní údaje*** umožňuje opravovat a doplňovat osobní údaje, pořízené při vyplnění přihlášky. Přístup je chráněn heslem uživatele v realmu „META“.

5.3.3 Účty – vytvoření, aktivace

- Základní sada účtů může být pořízena již současně s přihláškou do *META Centra* (viz 5.3.1).
- **Uživatel** si v případě potřeby může požádat o další účty volbou ***Aktivní část-Osobní údaje a účty-Žádost o další účty***. Zde se zobrazí stroje a clustery, na kterých přihlášený uživatel ještě nemá účet zřízen a z nich si může vybrat ty, které potřebuje.

  Úloha je volně přístupná, uživatel ovšem může žádat pouze o zřízení účtu sám pro sebe.
- Vlastní vytvoření účtů musí provést **administrátor** přes ***Aktivní část-Administrativní část MetaPortálu- Vytvoření účtů na základě přihlášky*** (viz 5.3.1).
- **Uživatel** může svůj účet na některém stroji učinit dočasně neaktivním, pokud se z nějakého důvodu rozhodne jej nepoužívat, nebo jej může opětovně aktivovat, pokud takový důvod již pominul. K tomuto účelu je určena volba ***Aktivní část-Osobní údaje a účty-Aktivace účtů***, která zajišťuje uživatelům přístup pouze k vlastním účtům.

5.4 Výroční zpráva

K účtům se váže také volba ***Aktivní část-Osobní údaje a účty- Výroční zpráva a prodloužení účtu***, která slouží k napsání výroční zprávy a zdůvodnění žádosti o prodloužení účtů na další kalendářní rok. Účty o jejichž prodloužení nikdo nepožádá, budou posléze zrušeny.

5.5 Administrátorské přístupy

5.5.1 Přístupy přes MetaPortál

Administrátorské prostředky, z nichž některé už byly zmíněny při práci s přihláškami a účty, jsou vesměs přístupné přes volbu ***Aktivní část-Administrativní část MetaPortálu***. Zde je stručný přehled těch, jejichž funkce nebyla ještě zmíněna:

- ***Rozeslání hromadné zprávy*** – umožňuje rozeslat e-mailovou zprávu vybrané cílové skupině. V rámci cílové skupiny umožňuje oslovit pouze aktivní uživatele či jen uživatele s prošlým účtem. Při rozesílání provádí optimalizaci pořadí adres za účelem urychlení atd.
- ***Perun-Seznam ovládaných strojů*** – tato volba umožňuje administrátorům prohlížet a upravovat informace o všech strojích pod správou systému Perun.
- ***Perun-Přístupy do databáze*** – jak se patrno z následujícího submenu, tato volba zprostředkuje administrátorský přístup k většině zdrojů *META Centra*:

- ***Uživatelé*** – výběr uživatele podle logname nebo příjmení. Výběr je nezbytný pro další tři položky submenu.
  - ***Uživatel-detail*** – zobrazení a editace personálních údajů, pořízených vesměs v procesu zadávání přihlášky. Zobrazuje také seznam UNIXových *uid*, které jsou prokliknutelné na detail *uid*.
  - ***Účty*** – účty vybraného uživatele. Volba umožňuje pomocí *Vybrat všechny* zobrazit seznam účtů pro hromadné rušení, nebo detail účtu na vybraném stroji nebo clusteru. V detailu lze pro vybraný účet použít volby:
    - *Zrušeni tohoto ÚČTU*
    - *Účty na vybraném HOST*
  - ***Přihlášky*** – zobrazení a editace detailu vybrané přihlášky. Dále též volba:
    - *Zrušení této přihlášky*

    Současně s detailem přihlášky se zobrazují související *požadavky na stroje a clustery* a rovněž *speciální požadavky*, pokud jsou.
- ***Hosts*** – výběr stroje, zobrazení a editace detailních informací o vybraném stroji. Z detailu jsou možné tyto volby:
  - *Zrušení tohoto HOST*
  - *Účty na tomto HOST* – zobrazení seznamu uživatelů s účtem na vybraném stroji, *logname* uživatele lze prokliknout k detailu tohoto účtu.
  - *Členství v clusterech* – zobrazuje seznam clusterů, v nichž je vybraný stroj členem. Jméno clusteru je opět prokliknutelné na detail dotyčného clusteru.
  - *Shelly* – seznam shellů používaných na tomto stroji
  - *Partitions* – zobrazení partitions na stroji a jejich velikostí.
- ***Clusters*** – výběr clusteru, zobrazení a editace detailních informací o clusteru. Možné volby:
  - *Zrušení tohoto CLUST*
  - *Účty na tomto CLUST* – seznam uživatelů, kteří mají účet na tomto clusteru. V seznamu je *logname* prokliknutelné na detail účtu vybraného uživatele na tomto clusteru.
  - *Stroje na tomto CLUST* – dává k dispozici seznam strojů z nichž se cluster skládá, jména strojů jsou prokliknutelná k detailní informaci o vybraném stroji.
  - *Členství v dalších clusterech* – pro clustery, které jsou členy vetších, složených clusterů je k vidění také informace o všech clusterech, jichž jsou členem. V seznamu je název clusteru prokliknutelný na svůj detail.
  - *Clustery na tomto CLUST* – clustery, které spojují několik menších clusterů v jeden celek, mají možnost zobrazit seznam těchto clusterů.
  - *Shelly* – seznam shellů používaných na tomto clusteru.

- ***Realms*** – výběr a editace informací o realmu. Volby z detailu:
  - *Zrušení tohoto REALM*
  - *Založení nového REALM*
  - *Logins in REALMS* – poskytuje seznam všech uživatelů v daném realmu. *Logname* je prokliknutelné na detail uživatele. Seznam lze použít k hromadnému rušení členství v realmu, obsahuje také volbu
    - *Přidání nového uživatele do tohoto REALMU*
  - *Stroje na tomto REALM* – zobrazuje seznam strojů patřících k vybranému realmu, jméno stroje v seznamu je prokliknutelné na detail stroje.
- ***Cells*** – výběr a editace údajů o buňce. Volby:
  - *Založení nové CELL*
  - *Uids na této CELL* – poskytuje seznam uživatelů příslušných k buňce a jejich UNIXových *uid*. *Logname* uživatele je prokliknutelné k detailu uživatele, *uid* lze prokliknout ke svému detailu, kde je též možnost
    - *Zrušení thoto UID*
  - *Volumy na této CELL* – volba, která zobrazuje seznam uživatelů, jim přidělených volumů a diskových kapacit. *Logname* je proklitnutelné na detail uživatele. Volume je prokliknutelný na detail *Volume* a jeho editaci. Tento detail zahrnuje také možnost
    - *Zrušení thoto VOLUME*
    - *Vytvoření nového VOLUME pro tohoto uživatele*, čímž se rozumí uživatel, jemuž patří právě zobrazený detail *Volume*.

  Z detailu se lze přes *Volumy na této Cell* dostat zpět na seznam uživatelů a jejich volumů.
- ***Filesystemy*** – výběr a editace informací o filesystemech. Možnosti:
  - *Zrušení tohoto FILESYSTEMU*
  - *Založení nového FILESYSTEMU*
  - *Účty na tomto FILESYSTEMU* – seznam uživatelů s účtem na tomto filesystemu. *Logname* je prokliknutelné k detailu účtu.
  - *Directoties a Quotas na tomto FILESYSTEMU* – zobrazuje seznam uživatelů, jejich *directories* a diskových kapacit. *Logname* je prokliknutelné na detail uživatele.
    Jméno *directory* lze prokliknout do detailu *directory*, kde je též možnost
    - *Zrušení této DIRECTORY*
    - *Založení nové DIRECTORY*

    Přes hodnotu *quoty* lze vstoupit do jejího detailu. Zde lze též použít
    - *Zrušení této QUOTY*
    - *Založení nové QUOTY*

Detaily *Directories* i *Quotas* mají možnost návratu do seznamu *Directoties a Quotas na tomto FILESYSTEMU.*

- ***Grupy*** – výběr a editace informací o grupách. Obsahuje také UNIXové *gids*. Možné volby:
  - *Zrušení této GRUPY*
  - *Založení nové GRUPY*
  - *Členové této GRUPY* – dává seznam členů zvolené grupy. *Logname* je prokliknutelné do detailu uživatele. Seznam dává možnost hromadného rušení členů z grupy a také
    - *Přidání nového člena do této GRUPY*
  - *Hosty a Clustery kde je tato GRUPA primární grupou* – zobrazí jejich seznam. Jména strojů a clusterů jsou prokliknutelná do detailu stroje či clusteru.
  - *Působnost GRUPY na Hostech a Clusterech* – seznam strojů a clusterů, kde je grupa definována. Jména strojů a clusterů jsou prokliknutelná do svých detailů.
- ***Projekty*** – zobrazení a editace informací o projektech zavedených v systému. Volby:
  - *Zrušení tohoto PROJEKTU*
  - *Založení nového PROJEKTU*
  - *Řešitelé tohoto PROJEKTU* – seznam řešitelů vybraného projektu. *Logname* je prokliknutelné do detailu uživatele. Umožňuje též hromadné rušení řešitelů a volbu
    - *Přidání nového řešitele do tohoto PROJEKTU*
  - *Volumy vyhrazené PROJEKTU* – zobrazí seznam vlastníků *volumů*, které jsou příslušné k vybranému projektu. *Logname* je prokliknutelné k detailu uživatele. Název *volume* lze prokliknout k detailu *volumů* (viz též *Cells*), stejně tak název buňky vede na detail *Cells*.
- ***Organizace*** – zobrazení a editace informací o organizacích, zavedených v systému. Kromě toho zahrnuje:
  - *Zrušení této ORGANIZACE.*
  - *Založení nové ORGANIZACE*
  - *Uživatelé v této ORGANIZACI* – zobrazení uživatelů, kteří patří pod tuto organizaci. *Logname* je prokliknutelné k detailu uživatele.
  - *Uživatelé z cizích ORGANIZACÍ* – zobrazuje seznam uživatelů, kteří patří do organizací, jež nejsou zavedeny v systému. *Logname* je rovněž prokliknutelné k detailu uživatele.
  - *Projekty v této ORGANIZACI* – dávají seznam projektů na kterých se vybraná organizace podílí. Název projektu lze prokliknout do detailu projektu.

- ***Perun-Historie změn databáze*** – umožňuje prohlížet tabulku JOURNAL, do které se ukládají průběžné změny v tabulkách zdrojů *META Centra* (viz 2.2 – 2.4). Takto lze zjistit, jaký typ změny byl proveden, které položky byly měněny a jakých nabývaly hodnot, kdo změny prováděl a také kdy je prováděl. Z důvodů větší frekvence určitých dotazů je toto zobrazení rozděleno na tři speciální případy:
  - *ACCOUNTS* – umožňuje prohlížet historii změn v tabulce ACCOUNTS, vybírat podle logname uživatele, který změnu provedl a stroje či clusteru, na kterém byl účet zřízen.
  - *REGISTRATIONS* – umožňuje prohlížet změny v tabulce REGISTRATIONS, vybírat podle čísla přihlášky, jména a příjmení nebo logname uživatele.
  - *Obecný výběr podle ROWID* – výběr libovolných změn podle rowid hledaného záznamu v původní tabulce.
- ***Perun-Přehled stavu služeb*** – barevná tabulka, zobrazující všechny stroje, clustery, buňky a realmy pod správou *META Centra*. Barvami jednotlivých políček jsou znázorněny nejzávažnější stavy v propagaci služeb na daný cíl. Vysvětlivky k významu barev jsou součástí zobrazení.

  Kliknutím na políčko s názvem cíle se přejde do detailu služeb, kde jsou zobrazeny stavy jednotlivých služeb, pro daný cíl nadefinovaných, čas, na který jsou naplánovány ke zpracování či čas poslední propagace, eventuelně chybová zpráva. Z této úlohy lze také spustit aktuální vypropagování zvolené služby.

  Přes název služby lze přejít do konfigurace dané služby, změnit aktuální parametry nebo přidat novou služby do konfigurační tabulky CONF_SERVICES (viz 2.3.3 – 2.4).
- ***Perun-Struktura databáze*** – strukturu aplikační části databáze si udržuje databázový stroj ve svých interních tabulkách. Tyto pak mohou být využity ke zobrazení této struktury a uchování komentářů a vazeb mezi aplikačními daty.
  - ***Tabulky*** – seznam aplikačních tabulek a jejich stručných popisů umožňuje přejít přes jméno tabulky do zobrazení detailu. Zde je k dispozici seznam a typ všech sloupců vybrané tabulky a jejich význam.

    Také jsou zobrazeny primární a unikátní klíče a seznam ostatních tabulek, které na tyto klíče ukazují, rovněž tak cizí klíče, seznam sloupců, které je tvoří a prokliknutelné odkazy na tabulky, do nichž cizí klíče ukazují.

    Rovněž seznam a tvar databázových checků je zobrazen na detailu vybrané tabulky.
  - ***Tabulky-údržba komentářů*** – nástroj pro vkládání a údržbu komentářů, které popisují určení jednotlivých tabulek a význam jejich sloupců.

5.5.2 Přístupy z příkazové řádky

Pro komplexnější úlohy, které jsou v činnosti administrátorů poměrně frekventované, byly vytvořeny skripty, které lze spustit z příkazové řádky shellu. Spouští se příkazem **perun *jméno skriptu***, kde skript **perun** zajistí kerberovské lístky, nastavení prostředí, cest a další náležitosti. Každý ze skriptů má „Help" na volbě `-?` , která zobrazí možné parametry a jejich použití. Seznam skriptů a jejich určení je následující :

- ***account*** – pro vytvoření, opravu i zrušení účtu na zadaném stroji či clusteru. Zahrnuje vytvoření nebo opravu:

  - záznamu o účtu

  je-li zadáno, vytvoří dále:

  - uživatele do grupy
  - domovský adresář
  - scratch adresáře

  Při rušení jsou tyto záznamy, které se vážou k účtu zrušeny.

  Standardní vytvoření a změny účtů se provádějí přes webové rozhraní v procesu podání a schvalování přihlášky a žádosti o účty (viz 5.3.1 a 5.3.3).

- ***user*** – pro založení, opravu a zrušení uživatele. Zahrnuje vytvoření nebo opravu:

  - záznamu o uživateli
  - uložení logname
  - vytvoření skutečného nebo „placeholder" (podle zadání) záznamu do tabulky kerberovských realmů

  dále, je-li zadáno:

  - vytvoření záznamu o *uid*
  - zaznamenání počátečního hesla

  Při zrušení uživatele jsou tyto svázané záznamy zrušeny.

  Rovněž vytvoření nového uživatele je standardně spojeno s podáním přihlášky do *META Centra* (viz 5.3.1).

- ***new-host*** – pro vytvoření, opravu a zrušení stroje. Zahrnuje vytvoření či opravu:

  - záznamu o stroji

  je-li zadáno, vytvoření:

  - záznamu o primární grupě na stroji

  - záznamu o působnosti všech zadaných grup na stroji
  - přidání stroje do clusteru
  - záznamů o shellech používaných na tomto stroji
  - záznamů do konfigurační tabulky pro služby
  - vytvoření záznamu o filesystemu

  Při zrušení stroje jsou tyto svázané záznamy zrušeny.

  V současné verzi systému je toto jediný způsob jak regulerně přidat nový stroj pod správu systému.

- ***gm*** – pro přidání, opravu a zrušení uživatele v grupě. Zahrnuje vytvoření nebo opravu záznamu o členství v grupě.

  Alternativně lze použít webové rozhraní, volbu *Perun–Přístupy do databáze–Grupy–Členové této GRUPY–Přidání nového člena do této GRUPY.*

- ***volume*** – pro vytvoření, opravu a zrušení volumu. Zahrnuje vytvoření eventuelně update záznamu o volume a AFS-mountpointu. Při zrušení jsou oba tyto záznamy smazány.

  Přes webové rozhraní lze pro práci s volumy použít volbu *Perun–Přístupy do databáze–Cells–Volumy vyhrazené Cell* a prokliknout sloupec se jménem příslušného volume, nebo použít volbu *Vytvoření nového VOLUME.*

- ***quotaup*** – pro změnu velikosti přidělené diskové kapacity.

  Tutéž změnu můžeme zadat přes webové rozhraní, použijeme-li stejnou volbu jako pro práci s volumy.

6 Závěr

V současné době je systém nasazen v plně produkčním prostředí *META Centra*, Superpočítačového centra ÚVT MU, a v Laboratoři pokročilých síťových technologií FI MU. Jsou spravovány účty více než 200 aktivních uživatelů na superpočítačích SGI, téměř 200 uzlech PC clusterů a desítkách pracovních stanic. V roce 2003 byla realizována i podpora pro správu účtů na testbedu projektu EU GridLab[3].

Systém je aktivně udržován, bezprostřední plány na rozvoj zahrnují notifikační službu, která bude informovat jednak uživatele o změnách týkajících se jejich účtů, přihlášek apod., a dále administrátory o nutnosti provedení určitých akcí, upozorňovat na opakující se chybové stavy apod.

[3] `http://www.gridlab.org/`

Dlouhodobé plány směřují jednak k rozšíření působnosti systému na správu konfigurací, které se netýkají jen uživatelských účtů, a dále k přesnější definici obecnějšího abstraktního modelu, který dovolí preciznější sledování požadavků a realizace změn, replikaci částí systému a odolnost proti výpadkům apod.

Literatura

[RGL88] Mark A. Rosenstein, Daniel E. Geer, and Peter J. Levine. The athena service management system (Moira). In *Proc. Winter Usenix*, 1988.

[RKM99] M. Ruda, A. Křenek, and L. Matyska. Infrastruktura metacentra. *Zpravodaj ÚVT MU*, 10(2), 1999. `http://www.ics.muni.cz/bulletin/issues/vol10num02/krenek/krenek.html`.

[Sit01] Jiří Sitera. Využití adresářových služeb (ldap) v projektu metacentrum. Technical report, Technická zpráva CESNET 2/2001, 2001.

[Sit03] Jiří Sitera. Aktuální stav ldapu metacentra. Technical report, Technická zpráva CESNET 29/2003, 2003.