PŘENOSY DAT V HETEROGENNÍM PROSTŘEDÍ

Jaromír Ocelka

ÚVT MU, Botanická 68a, 602 00 Brno, ČR, ocelka@ics.muni.cz

Abstrakt
V heterogenním prostředí informačních systémů je nutné řešit problematiku přenášení dat mezi jednotlivými systémy. Je možné použít replikační mechanizmy databázových serverů, dávkové přenosy v textových souborech či vlastní mechanizmy uzpůsobené konkrétní situaci. Vznikem WWW prezentace Masarykovy univerzity v Brně v roce 1996, se svým odděleným datovým základem od personální a mzdové databáze, se začaly vyvíjet mechanizmy pro přenos dat, které se postupem času uplatňovaly a přizpůsobovaly dalším izolovaným datovým zdrojům univerzity. Článek uvádí jednotlivé metody použití přenosů dat z různorodých systémů a situacích na MU v Brně.

1. Úvod

V roce 1996 vznikla na Masarykově univerzitě v Brně myšlenka zrealizovat jednotnou celouniverzitní www prezentaci (podrobněji viz [1]). Do této doby měla každá fakulta svoji prezentaci bez jakéhokoliv univerzitního zastřešení. Hlavní myšlenkou jednotné prezentace bylo vytipovat kategorie informací, které jsou pro všechny fakulty a celouniverzitní pracoviště totožné, a tyto prezentovat. Fakulty si pak ve svých prezentacích budou pouze udržovat specializované informace. Hlavním zdrojem dat, který se začal od počátku prezentace používat, byla personální databáze (viz [2] – číselník pracovišť, zaměstnanci, …). Od počátku byl záměr nepsat stránky „ručně“, ale vytvářet je dynamicky na vyžádání (ať už on-line či předgenerováním – viz [4]) s tím, že podklady pro generování stránek bude poskytovat relační databáze. Z bezpečnostních důvodů (a z důvodu nevyzkoušených nových technologií) bylo stanoveno, že www server s dynamickými stránkami nemůže čerpat informace přímo z personální databáze (kde jsou také mzdové údaje), ale je nutné vybudovat oddělenou databázi, přičemž personální databáze by poskytovala data v pravidelných intervalech.

Pro řešeních tohoto úkolu nebyly zvoleny databázové replikační nástroje (mimo jiné také z důvodu jiného databázového výrobce – MS SQL & Informix), ale vlastní dávkové mechanizmy, které pro www prezentaci postačují jednosměrné. V průběhu let byly www stránky obohacovány o data i z jiných datových zdrojů, a to dalo impuls k postupnému budování automatizovaného systému importu dat.

2. Typy datových zdrojů

Kromě již zmiňované personální databáze, která je zástupcem relačních databází, lze také zařadit mezi datové zdroje soubor definující e-mail účty – na UNIX systémech například soubor /etc/passwd:

```
novak:x:219:200:Josef Novak, UVT MU:/home/novak:/bin/ksh
polak:x:237:200:Karel Polak, UVT MU:/home/polak:/bin/ksh
zlata:x:336:200:Jirina Zlata, UVT MU:/home/zlata:/bin/ksh
```

Obr. 1: Příklad textového souboru s oddělovači

Je to zástupce doposud nejpřenositelnějšího úložiště dat – textový soubor (dále TXT), kde na každém řádku je uložen právě jeden záznam. Nejjednodušší varianta je konstantní počet atributů (oddělených speciálním znakem – '|','':', …) na každém řádku.
Dalším typem je, od přelomu tisíciletí zažívající velký rozmach, XML. Oproti TXT má výhodu ve vnitřním variabilním formátu, umožňujícím lépe a přehledněji uchovat entity různých typu:

```
<SKOLA Skola="Lesnická univerzita">
  <REKTOR Titul_Pred="Prof." Jmeno="Jan" Prijmeni="Zeleny">
    <OBDOBI Od="1.11.1995" Do="31.10.1998"/>
  </REKTOR>
  <REKTOR Titul_Pred="Doc. Ing." Jmeno="Hana" Prijmeni="Modrá">
    <OBDOBI Od="1.11.1998" Do="31.10.2001"/>
    <OBDOBI Od="1.11.2001" Do="31.10.2004"/>
  </REKTOR>
</SKOLA>
```

Obr. 2: Příklad XML souboru

Z dalších zástupců můžeme jmenovat dbf (datové soubory z dBase a FoxPro), dokumenty, které produkují tabulkové procesory:

	A	B	C	D
1	zaměstnanec	telefon	e-mail	kancelář
2	Novák Josef	8721	novak@test.muni.cz	B42
3	Polák John	4278	polak@test.muni.cz	G38
4	Žlutá Oldřiška	4257	zluta@test.muni.cz	F87

Obr. 3: Příklad dat v tabulkovém procesoru

Kromě již výše uvedených zástupců strukturovaných zdrojů dat mohou být pochopitelně použity i informace uložené bez záměrné organizace. Příkladem může být www stránka či její specifikovaná část (aktuální teplota na stránkách meteorologického ústavu, články konference, …), Tyto však vyžadují nalézt v konkrétním datovém zdroji specifické „záchytné body", podle kterých lze s velkou mírou pravděpodobnosti nalézt požadovanou informaci i v budoucnu.

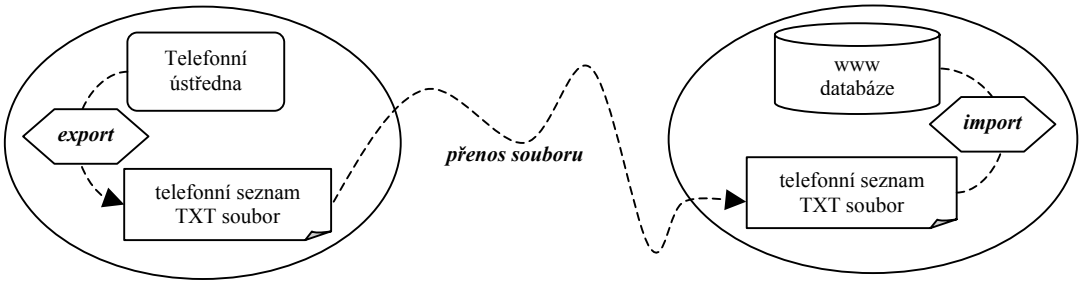
3. Přenosové mechanizmy

Aby bylo možné čerpat informace z daného datového zdroje, je také nutné zajistit přenos dat do cílového místa. V nejjednodušším případě to nejčastěji v minulosti býval přenos datového souboru pomocí diskety. S nástupem sítí je situace jednodušší a přenos lze zautomatizovat. Způsoby přenosu z datového zdroje do cílového místa lze rozdělit na dvě základní kategorie – přímé nebo nepřímé propojení.

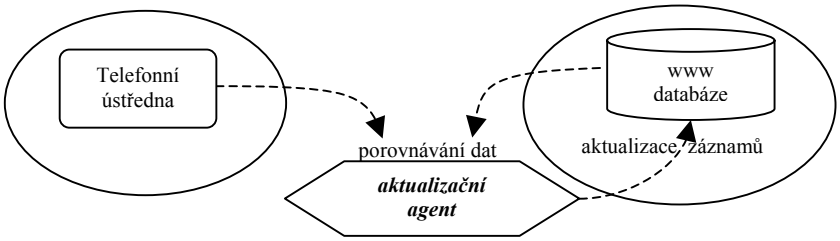
Přímé propojení datových zdrojů je silně závislé na jednotlivých datových zdrojích a jejich softwarových implementacích. Bez jejich znalosti nemůžeme dopředu určit, jak si budou datové zdroje svá data vyměňovat. V relačních databázích je možné použít replikační mechanizmy. Ne vždy je však žádoucí propojit jednotlivé zdroje on-line. Veřejné www prezentace jsou typickým důvodem, kdy je nutné data přenášet dávkově ze zdrojových databází, protože veřejnosti je lépe prezentovat data celistvá. V případě on-line replikací, například z personální databáze, kdy by personalistka převáděla deset zaměstnanců z jednoho pracoviště na druhé, by se tato změna postupně projevovala na www stránkách. Podobných případů by se jistě našlo více.

Pro řešení jednosměrné aktualizace dat v podpůrné databázi www prezentace MU z personální a studijní databáze byla s výhodou využita vlastnost Microsoft SQL databázového serveru, kdy lze zpřístupnit datové tabulky z externích relačních databází (i jiných výrobců) tak, jako kdyby byly uloženy přímo v této databázi. K aktualizaci dat z jiných systémů se pak přistoupilo jako k transformaci dat v rámci jednoho databázového systému.

Pokud *nelze* datové zdroje *propojit přímo* jejich vlastními mechanizmy, postupuje se většinou pomocí exportu a importu dat (viz obr. 4). Z původního datového zdroje se nejprve vytvoří datový soubor, který se přenese do cílového místa, a zde po načtení data proběhne příslušná zaktualizace. Samotný přenos souboru lze automatizovaně učinit například pomocí elektronické pošty, sdílení disků, protokolu ftp, http či zabezpečeného scp. Místo souboru určeného pro přenos dat lze také použít vlastní mechanizmy, kdy program hrající roli aktualizačního agenta (viz obr. 5) čte data ze zdroje a provádí aktualizace na cílové straně na základě srovnání dat na obou stranách.

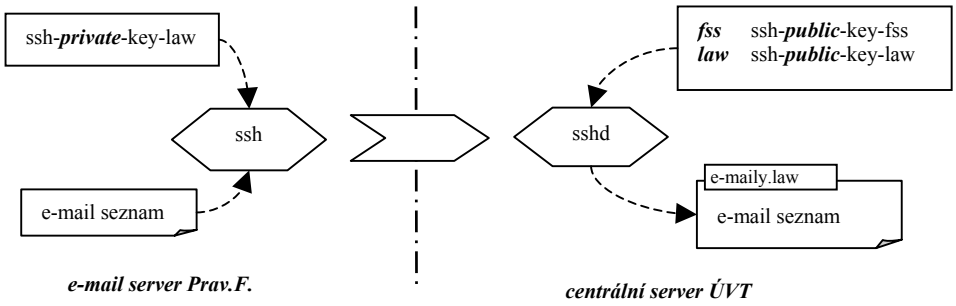


Obr. 4: Nepřímé propojení datových zdrojů



Obr. 5: Přímá aktualizace nepropojených datových zdrojů

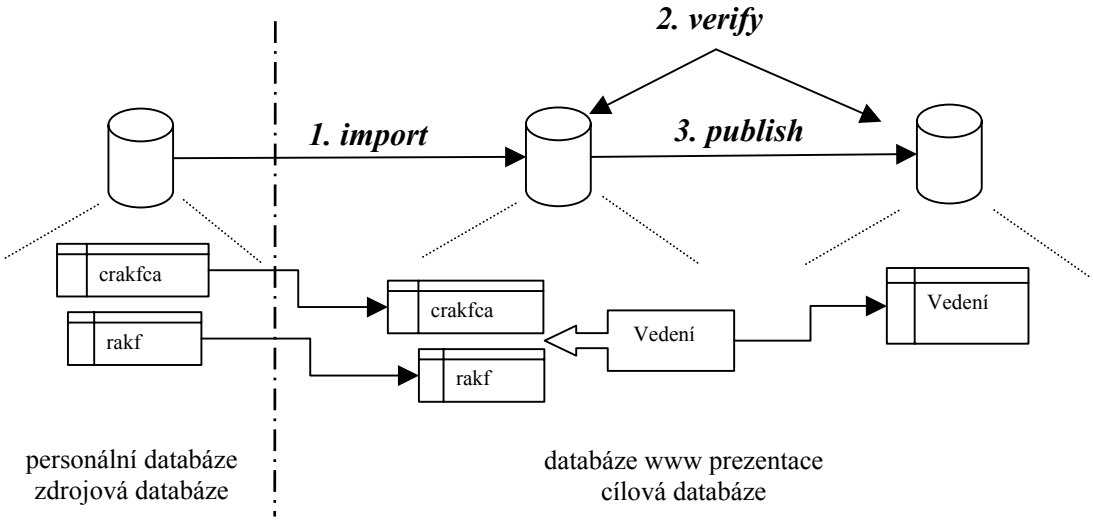
V případě www prezentace MU byl při řešení problematiky datových přenosů (viz [3]) z nezávislých fakultních e-mail serverů do podpůrné www databáze vytvořen specializovaný přenosový protokol založený na ssh. Připravený datový soubor je vypsán na standardní vstup příkazu ssh (varianta zabezpečeného příkazu telnet) s parametry určujícími jméno cílového serveru, jméno speciální účtu určeného pro příjem dat, jméno datového zdroje (pro rozlišení dat na cílovém serveru) a s privátním klíčem jehož veřejný protějšek je zaregistrovaný na cílovém serveru s určením, že přihlašujícímu s tímto klíčem se automaticky spustí ukládání standardního vstupu do souboru. Na MU byl zvolen pro přenos nezávislý prostředník a proto se odtud data přenášejí stejným principem, jen se místo čtení standardního vstupu zapisuje a místo zápisu na standardní vstup se data čtou. Tento mechanizmus umožňuje zdroji dat odeslat data kdykoliv, dle své časové preference, aniž by to jakkoliv zatížilo cílového odběratele dat, ten si data vyzvedne, až je to pro něj vhodné.



Obr. 6: Příklad přenosu dat pomocí ssh

4. Automatizace a kontrolní mechanizmy

V prvních letech www prezentace MU byly jednotlivé importy spouštěny manuálně a výsledek byl kontrolován správcem systému. Tento způsob však nebylo možné provádět příliš často, a proto byla zvolena perioda aktualizace jeden měsíc. Mechanizmy však byly vylepšovány a na základě zkušenosti byl prováděn jejich postupný převod k plné automatizaci. Zároveň však byl kladen důraz na možnou kontrolu v každém kroku přenosu – obzvláště při hledání případné chyby. Výsledný, v současné době používaný, mechanizmus se skládá ze tří fází: *import*, *verify* a *publish*.



Obr. 7: Fáze aktualizace dat

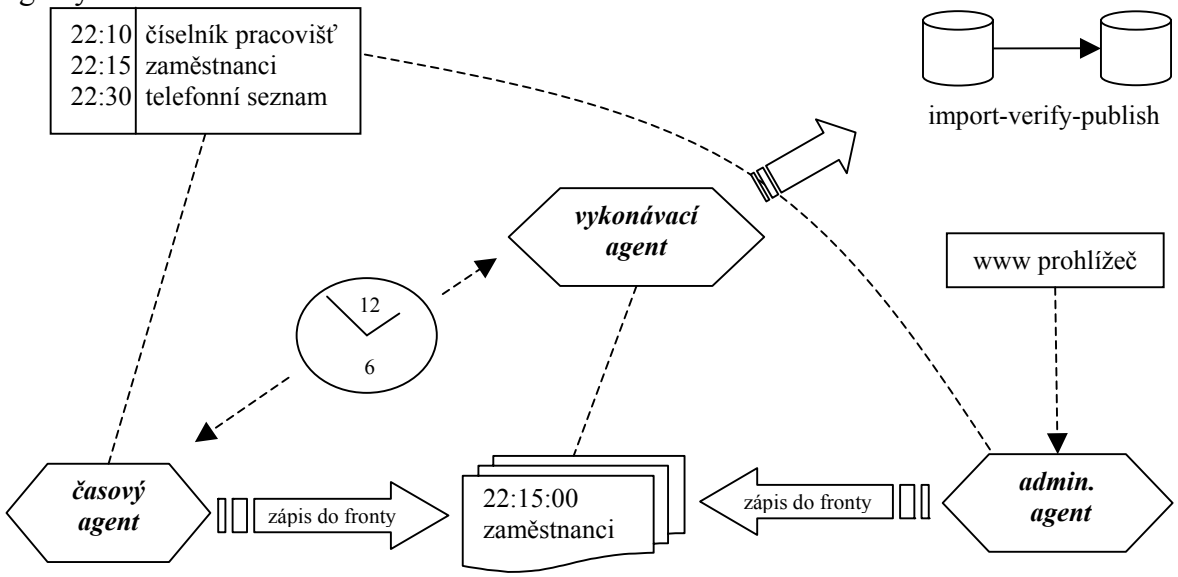
Fáze *import* zajišťuje převod z datového zdroje do databázového serveru www prezentace. Cílem však není databáze odkud se zveřejňují data na www, ale oddělená databáze (na stejném databázovém systému) výhradně určená pro importy. V této databázi se data načtou do identických struktur jako má poskytovatel dat. Důvodem, který vedl ke stejným strukturám, byla snaha o co nejprůhlednější komunikaci s poskytovatelem dat při případném dohledávání chyb. Nad těmito datovými strukturami byly pak vytvořeny databázové pohledy, které poskytovaly data ve stejné struktuře jako v databázi se zveřejněnými daty. Tyto pohledy nejsou nutné pro samotný mechanizmus přenosu (transformaci dat lze udělat vnitřním algoritmem převodního programu) a aktualizace dat, ale jejich primárním důvodem existence je opět možnost snadné kontroly. Pokud (v případě složitých transformací) nelze použít pohled, jsou data v posledním kroku fáze import převedena do tabulek majících stejné schéma jako v „ostré“ databázi. Tyto tabulky se však stále nachází v databázi určené pro import. Na obr. 7 je uveden příklad, kdy se tabulka řídících a akademických funkcí přenáší společně s číselníkem názvů funkcí do výsledné tabulky Vedení.

Ve fázi *verify* jsou prováděny různé kontroly. Nejčastější je kontrola na úbytek dat. Pokud například při aktualizaci obdrží odběratel dat o 10% méně pracovních poměrů oproti předchozímu stavu, jsou data zamítnuta. Dalším typem kontroly je přítomnost anglických překladů k českým textům. Není důvod, aby nemohlo být v personální databázi založené pracoviště, i když není znám jeho anglický název (na výpočet mezd to nemá vliv). Na www prezentaci, která je důsledně dodržovaná anglicky, však není tato absence přípustná. Pokud tedy fáze *verify* provede zamítnutí dat, informuje příslušné odpovědné pracovníky e-mailem a zastaví další zpracování získaných dat.

Poslední fáze *publish* provádí samotnou aktualizaci dat v cílové databázi prezentovaných dat. Toto lze opět obecně provést několika způsoby. Pokud se na data neváží v cílové databázi další informace, je možné v transakci provést „DELETE, INSERT from SELECT“ – tj. smazat ostrá data a provést jejich vložení ze stejné struktury v pomocné databázi. Pokud toto není možné, jsou data aktualizována pomocí databázového kurzoru.

5. Spouštění aktualizačních úloh

Tím, jak postupně přibývalo množství vystavovaných informací čerpaných z různých zdrojů, nabývala na důležitosti časová souhra jednotlivých prováděných importů. Samotné spouštění jednotlivých úloh v určitou dobu se ukázalo jako nedostatečné, protože nelze vždy odhadnou čas dané aktualizace a tím pádem se občas aktualizační úlohy časově překrývaly. Byl proto vybudován vlastní plánovací mechanizmus (viz obr. 8), který lze rozdělit na několik logických celků.



Obr. 8:Mechanizmus spouštění aktualizací dat

Každá aktualizační úloha má definovaný čas spuštění a periodu opakování. Protože však nemůže být přesně znám čas konce ukončení předchozí úlohy, je u každé úlohy také určen maximální čas prodlevy pro spuštění. *Časový agent* na základě této definice generuje v době pro spuštění příslušné úlohy požadavky na spouštění do fronty. Použitím jedné fronty se podařilo zabránit paralelnímu zpracování importu dat, aniž by se zbytečně obsadil velký časový interval snahou o „pro jistotu“ velké časové prodlevy mezi jednotlivými úlohami.

O spouštění jednotlivých sekvencí *import-verify-publish* se stará *vykonávací agent*. Pokud neprobíhá žádná úloha, čeká na čas první události ve frontě, kterou posléze spustí. Postupně vykonává jednotlivé fáze a v případě chyby v dané fázi ukončí vykovávání celé úlohy a ohlásí

chybu. Jak již bylo naznačeno výše, může občas příslušná fáze probíhat delší dobu (aktuálně přetížený zdrojový server, výrazný nárůst dat, …), a tím může dojít k výpadku následující aktualizační úlohy. Ke každé fázi dané úlohy je tedy definována maximální přípustná délka a pokud je překročena, je její vykonávání násilně přerušeno.

Občas může nastat situace, kdy je nutné provést aktualizaci dat na vyžádání mimo stanovené pořadí. Proto vznikl *administrátorský agent*, který formou formulářů v intranetu umožňuje vytvořit a vložit požadavek na spuštění konkrétní úlohy do fronty.

6. Závěr

WWW prezentace Masarykovy univerzity v Brně přijímá popsanými mechanizmy data celkem ze dvou velkých relačních databázi (studijní a personální) a z různých lokálních datových zdrojů fakult (převážně e-mail seznamy občas doplněné o čísla kanceláří, telefonů, …). Z těchto zdrojů každý den plně automaticky proběhne celkem 35 různých datových přenosů v celkové době trvání přibližně jedné hodiny. V průměru jeden až tři případy za měsíc nastává situace, kdy je na základě kontrolních mechanizmů potřeba požádat poskytovatele dat o opravu chyby či doplnění dat, aby vyhovovala požadavkům na vystavení. Vyvinuté mechanizmy tak slouží dobře nejenom www prezentaci, ale navíc pomáhají odhalovat chyby v datech zdrojových databází interních univerzitních systémů.

Literatura:

1. KOHOUTKOVÁ, Jana. Masarykova univerzita na internetových WWW stránkách. Zpravodaj ÚVT MU : bulletin pro zájemce o výpočetní techniku na Masarykově univerzitě. ISSN 1212-0901, 1997, roč. 7, č. 3, s. 10-13
2. VALACH, Zdeněk. Personální databáze zaměstnanců MU pro WWW. Zpravodaj ÚVT MU : bulletin pro zájemce o výpočetní techniku na Masarykově univerzitě. ISSN 1212-0901, 1997, roč. 8, č. 1, s. 10-13
3. PROCHÁZKOVÁ, Šárka, OCELKA, Jaromír. Internetová prezentace MU v Brně. In UNINFOS 2000. Zborník príspevkov. Nitra : SPU v Nitre, 2000. ISBN 80-7137-713-9, s. 186-189
4. OCELKA, Jaromír. WWW Server v přílivu uživatelů Internetu. In Tvorba softwaru 2003. Sborník příspěvků. Ostrava: Tanger s.r.o Ostrava, MARQ, 2003. ISBN 80-85988-83-6, s.154-160