



Databázový server MiniSQL

Pavel Janík ml.
<http://www.janik.cz>



Operační systém Linux se již pomalu stává stálicí i na databázovém trhu, a tak již není žádný problém provozovat téměř libovolný databázový server na Linuxu. Můžete si vyzkoušet například [Oracle](#), [Informix](#), [Sybase](#) a další.

Jenže... Kde je tedy problém? Co když náš počítač nepatří zrovna mezi ty nejžhavější novinky a jeho procesor a paměťové moduly (např. Intel 486/8MB) by mu mohlo závidět i leckteré muzeum počítačové historie? Jaké máme možnosti? Oracle či Informix použít nemůžeme, protože tyto databázové systémy mají někdy problémy i na podstatněji výkonnějších strojích (ostatně tyto databáze jsou vytvořeny pro enterprise prostředí, kde také hrají prim). Zbývají nám systémy jako [PostgreSQL](#), [MySQL](#) nebo [MiniSQL](#). Servery PostgreSQL i MySQL jsou velmi výkonné, ale i přesto velmi pomalé na naší hypotetické konfiguraci. Podívejme se tedy na server MiniSQL (v minulosti též nazývaný mSQL nebo Minerva).

MiniSQL je komerčním produktem společnosti [Hughes Technologies Pty, Ltd](#). Licenční podmínky jsou však benevolentní pro nekomerční subjekty, které mohou MiniSQL kopírovat, šířit, či dokonce modifikovat bez jakýchkoli omezení. Poslední verze s číslem 2.0.11 byla uvolněna 23. srpna roku 1999.

Instalace

Instalace serveru MiniSQL je ve světě Linuxu poměrně nestandardní. Je sice využito standardních prostředků poskytovaných většinou linuxových distribucí (`autoconf`), ale velice specifickým způsobem. Ale i přesto je instalace poměrně jednoduchá a pokud zvolíte již připravenou binární distribuci ve formátu RPM (např. ze serveru společnosti [RedHat](#)), je naprosto bezproblémová. Máme-li k dispozici pouze zdrojové texty serveru MiniSQL, musíme je nejdříve rozbalit:

```
tar xzf msql-2.0.11.tar.gz
```

Rozbalené zdrojové texty jsou nyní v adresáři `msql-2.0.11` a mají níže popsanou strukturu. V hlavním adresáři jsou soubory `BUGS` (informace o tom, jak ohlašovat chyby v serveru `msql`), `INSTALL` (popis instalace serveru), `MSQL_BOOK` (o MiniSQL vyšla i kniha :-), `Makefile` (standardní soubor pro `make`), `README`, `README.sco`, `RELEASE_NOTES` (informace o aktuální verzi serveru MiniSQL). Vedle těchto souborů jsou v distribuci MiniSQL i adresáře `demos` (obsahuje demo aplikaci záložky), `doc` (obsahuje kompletní dokumentaci ve formátu HTML a PostScript), `misc`, `scripts` a `src`. Poslední jmenovaný adresář obsahuje kompletní zdrojové texty serveru a ještě se o něm zmíníme.

A nyní k vlastní instalaci. Jak jsem již výše zmínil, není instalace vyřešena zrovna ideálním způsobem, a proto aplikujeme velmi jednoduchý patch, který instalaci poněkud zjednoduší:

```
cd msql-2.0.11; patch -p1 <<EOM
diff -urN msql-2.0.11.orig/src/conf/site.mm.in msql-2.0.11/src/conf/site.mm.in
--- msql-2.0.11.orig/src/conf/site.mm.in      Thu Aug 19 05:33:09 1999
+++ msql-2.0.11/src/conf/site.mm.in          Mon Jul 10 19:57:38 2000
@@ -4,7 +4,7 @@
```

```
COMPILER= @CC@
-INST_DIR= /usr/local/Hughes
+INST_DIR= @prefix@
HAVE_DYNAMIC= -DHAVE_DYNAMIC
CC_ONLY_FLAGS=
LINK_ONLY_FLAGS= @LINUX_DYNAMIC@
```

```
EOM
```

MiniSQL je možno přeložit na velkém množství platform a operačních systémů, a tak obsahuje podporu pro více platform, jež je implementována na adresářovém principu. Každá architektura bude mít svůj adresář, který vytvoříme pomocí příkazu

```
make target
```

Výstup, který bychom mohli obdržet, může vypadat například takto:

```
Making target directory for Linux-2.3.20-i686

Building directory tree.
```

```
Adding common
Adding conf
Adding lang-common
Adding lite
Adding makedepend
Adding makegen
Adding msq1
Adding regexp
Adding tests
Adding tests/rtest.src
Adding w3-msql
Adding w3-msql/tests
```

Adding sym-links

Build of target directory for Linux-2.3.20-i686 complete

Nyní bude vytvořen adresář `targets` a v něm další adresář pro naši architekturu. V našem případě bude mít jméno `Linux-2.3.20-i686`. V tomto adresáři již budeme mít připraveny zdrojové texty k překladu, a proto se do něj přepneme:

```
cd targets/Linux-2.3.20-i686
```

Poté spustíme konfigurační utilitku využívající služeb `autoconf` k rozpoznání vlastností cílového prostředí. Předpokládáme, že budeme instalovat server do adresáře `~/MiniSQL`:

```
./setup --prefix=~/MiniSQL
```

Výstup tohoto příkazu je velmi podrobný a ukazuje průběh testování jednotlivých vlastností našeho systému:

```
Starting configuration of mSQL 2.
```

```
creating cache ./config.cache
checking for gcc... gcc
checking whether the C compiler (gcc ) works... yes
checking whether the C compiler (gcc ) is a cross-compiler... no
checking whether we are using GNU C... yes
checking whether gcc accepts -g... yes
checking return type of signal handlers... void
checking for ranlib... ranlib
checking for bison... bison -y
checking for sysconf... yes
checking for strdup... yes
checking for rindex... yes
checking for random... yes
checking for bcopy... yes
checking for getdtablesize... yes
checking for strsignal... yes
checking for ftruncate... yes
checking for setrlimit... yes
checking for madvise... no
checking for msync... yes
checking for strncasecmp... yes
checking for mktemp... yes
checking for snprintf... yes
checking for strtoul... yes
checking for strptime... yes
checking for strerror... yes
checking for -lsocket... no
```

```

checking for -lnsl... yes
checking for -lgen... no
checking for -lx... no
checking for -ldl... yes
checking how to run the C preprocessor... gcc -E
checking for sys/sockio.h... no
checking for dlfcn.h... yes
checking for sys/dir.h... yes
checking for dirent.h... yes
checking for netinet/in_sysm.h... yes
checking for select.h... no
checking for sys/select.h... yes
checking for sys/un.h... yes
checking for strings.h... yes
checking for sys/timeb.h... yes
updating cache ./config.cache
creating ./config.status
creating site.mm
creating makegen/makegen.cf
creating common/config.h

checking your directory stuff. Using dirent.h and struct dirent
checking your mmap(). Your mmap() is fine.
checking your msync(). Using 3 args to msync.
checking for sys_errlist. You're fine.
checking for u_int. You're fine.
checking for int32_t and friends. You're fine.
checking for ssize_t. You're fine.
checking for a working getrlimit. You're fine.
checking for day light saving info. Yup, we'll use it.
checking for HP-UX ranlib. Nope, it's OK to use ranlib.
checking for Linux. Yup, it's Linux. Adding -rdynamic to the link flags.
Also forcing msync for Linux.

```

Ready to build mSQL.

You may wish to check "site.mm" although the defaults should be fine. When you're ready, type "make all" to build the software

Nyní již jenom zkontrolujeme obsah souboru `site.mm` a můžeme spustit vlastní překlad a instalaci:

```
make all && make install
```

Mezi tím, co nám doběhne překlad, si musíme rozmyslet, pod jakým uživatelem náš SQL server poběží — pro testování můžeme spouštět SQL server pod právy uživatele, který server nainstaloval, ale pro rutinní provoz bude lépe vytvořit zvláštního uživatele (např. `msql`) sloužícího pouze pro běh SQL serveru. Překlad serveru je velmi rychlý a může být hotov i během deseti sekund (pokud máte rychlý hardware :-).

Konfigurace

Server MiniSQL je možno jednoduše konfigurovat. Veškeré konfigurační informace jsou uloženy v souboru `~/MiniSQL/msql.conf`. Konfigurační soubor je rozdělen do sekcí `general`, `system` a `w3-mysql`. Sekce `general` může vypadat například takto:

```

[general]

Inst_Dir = /home/pavel/MiniSQL
DB_Dir = %I/msqldb
mSQL_User = pavel

```

```

Admin_User = pavel
Pid_File = %I/msql2d.pid
TCP_Port = 1114
UNIX_Port = %I/msql2.sock

```

Hodnota `Inst_Dir` ukazuje na adresář, který jsme zvolili při překladu. Pod tímto adresářem jsou schovány ostatní zajímavé soubory a adresáře. Proměnná `DB_Dir` ukazuje na adresář, kde jsou uloženy datové soubory serveru. Pokud je ve jméně tohoto nebo i ostatních adresářů uvedeno `%I`, je tato dvojice znaků nahrazena obsahem proměnné `Inst_Dir`.

Proměnná `mSQL_User` obsahuje jméno uživatele, pod kterým poběží databázový server jako takový (viz výše). `Admin_User` je uživatel, který bude mít povoleno provádět privilegované příkazy (vytváření resp. rušení databází, případně zastavení či reload databázového serveru). Proměnná `Pid_File` obsahuje jméno souboru, v němž bude v každý okamžik běhu databázového serveru MiniSQL uloženo identifikační číslo procesu hlavního démona `msql2d`. Proměnná `TCP_Port` obsahuje číslo TCP portu, na kterém bude server MiniSQL čekat na příchozí spojení (pokud bude spojení ze vzdáleného serveru povoleno). Poslední proměnnou je `UNIX_Port`. Obsahem této proměnné je jméno socketu, který bude použit pro komunikaci s databázovým serverem na lokálním počítači.

Sekce `system` má stejnou strukturu, ale odlišné proměnné:

```

[system]

Msync_Timer = 30
Host_Lookup = True
Read_Only = False
Remote_Access = True
Local_Access = True
Query_Log = True
Query_Log_File = %I/query.log

```

Proměnná `Msync_Timer` obsahuje počet sekund, po kterých bude synchronizován obsah databáze na disku s obsahem databáze v paměti databázového serveru. Proměnná `Host_Lookup` má podobnou funkci jako hodnota `KNOWN` v `tcp_wrapperu`. Pokud je tato proměnná nastavena na hodnotu `True` a klient, který se připojuje přes TCP port k lokálnímu databázovému serveru nemá v pořádku záznam v reverzní doméně, má bohužel smůlu a nepodaří se mu k databázovému serveru připojit. Pokud je tato hodnota nastavena na `False`, připojit se může i ten, kdo tento záznam v pořádku nemá. Význam proměnné `Read_Only` je zbytečně vysvětlovat. Pokud je tato proměnná nastavena na `True`, je server spuštěn v režimu pouze pro čtení a tedy žádný SQL příkaz nemůže změnit obsah databáze.

Další dvě proměnné umožňují specifikovat režim serveru — server může poskytovat služby pouze lokálním klientům (`Remote_Access = False`, `Local_Access = True`), pouze vzdáleným klientům komunikujícím po TCP (`Remote_Access = True`, `Local_Access = False`), nebo úplně všem klientům (`Remote_Access = True`, `Local_Access = True`). Pro nasazení na webový server je vhodné použít první popsanou metodu.

Poslední dvě volby slouží především pro odlaďování aplikací, kdy je možno zaznamenávat veškeré přijaté SQL dotazy do souboru v následujícím formátu:

```

11-Oct-1999 22:09:45 pavel UNIX_SOCKET pokus 95
INSERT INTO tabulka VALUES (1, 'sloupek1')

```

Formát je zřejmý — v prvním řádku je uvedeno datum a čas položení SQL dotazu z druhého řádku. Dále je zde uveden uživatel (`pavel`), databáze (`pokus`) a typ spojení (`UNIX_SOCKET`). Pro reálné nasazení doporučuji tyto volby ponechat vypnuté, neboť poněkud snižují odezvu serveru.

Startujeme

Nyní již tedy známe vše potřebné pro provozování vlastního serveru, ale stále ještě nevíme, jak vlastní server spustit. Je to jednoduché — při instalaci databázového serveru byl vytvořen i adresář `bin/`, který obsahuje několik důležitých programů. Nejdůležitějším z nich je program `msql2d`, což je vlastní server. Po jeho spuštění si můžeme ukázat funkci dalších programů z tohoto adresáře:

```

~/MiniSQL/bin/msql2d

```

Po spuštění server vypíše základní informace:

```
Mini SQL Version 2.0.11
Copyright (c) 1993-94 David J. Hughes
Copyright (c) 1995-99 Hughes Technologies Pty Ltd.
All rights reserved.
```

```
Loading configuration from '/home/pavel/MiniSQL/msql.conf'.
Server process reconfigured to accept 200 connections.
Server running as user 'msql'.
Server mode is Read/Write.
```

```
Warning : No ACL file. Using global read/write access.
```

Nyní je již vše nachystáno pro naši práci — server běží a je připraven naslouchat příkazům svého pána — databázového administrátora, který má k dispozici silný nástroj. Je jím program `msqladmin`, který umožňuje komfortní práci s databázovým serverem. Podporuje několik parametrů, pomocí kterých je možno např. vytvořit databázi (`msqladmin create`), zrušit databázi (`msqladmin drop`), kopírovat, či přesunout databázi (`msqladmin copy/move`). Samozřejmě je možné databázový server restartovat (`msqladmin reload`) či ukončit (`msqladmin shutdown`). Zajímavou vlastností je také možnost sledovat aktuální stav serveru samotného pomocí příkazu `msqladmin stats`:

```
Server Statistics
```

```
-----
```

```
Mini SQL Version 2.0.11
Copyright (c) 1993-94 David J. Hughes
Copyright (c) 1995-99 Hughes Technologies Pty Ltd.
All rights reserved.
```

```
Config file      : /home/pavel/MiniSQL/msql.conf
Max connections  : 200
Cur connections : 1
Running as user  : msql
Server uptime    : 0 days, 0 hours, 8 mins, 36 secs
Connection count : 7
Query count      : 0
```

```
Connection table :
```

Sock	Username	Hostname	Database	Connect	Idle	Queries
5	pavel	UNIX Sock	pokus	0H 0M	0	1

Dalšími zajímavými příkazy pro administrátora jsou `msqlimport` a `msqlexport`, které umožňují jednoduchý přenos dat mezi textovou a databázovou podobou. A ještě zajímavějším příkazem je `msqldump` ukládající kompletní databázi do textového formátu, pomocí kterého je možné okamžitě databázi obnovit. To je vhodné zvláště pro zálohování kompletního databázového serveru.

Zajímavým nástrojem je také program `relshow`, který nám pomůže při zjišťování obsahu databázového serveru — může nám prozradit seznam databází na serveru (`relshow`) nebo obsah databáze, tj. seznam tabulek (`relshow database`) či sloupců v tabulce (`relshow database table`).

Přístupová práva

Téměř samozřejmou vlastností databázových serverů moderní doby je možnost správy přístupových práv až na úroveň jednotlivých sloupců v tabulkách. Databázový server MiniSQL je bohužel v této oblasti poměrně pozadu a umožňuje pouze kontrolovat přístupová práva na úrovni jednotlivých databází. Vše se opět děje na úrovni editace souboru `msql.acl` (oproti standardním SQL příkazům `GRANT` a `REVOKE`), který je v naší instalaci umístěn v adresáři `~/MiniSQL`:

```

database=pokus
read=nobody
write=pavel
host=*
access=local,remote

```

Pokud soubor `msql.acl` vypadá jako výše uvedený příklad, tak uživatel `pavel` může databázi `pokus` měnit a uživatel `nobody` z ní může pouze číst. Přístup je povolen jak z lokálního, tak ze vzdáleného počítače. Podrobnější dokumentaci k přístupovým právním naleznete v manuálu k serveru.

SQL příkazy

Server MiniSQL není plnohodnotným SQL serverem a to je nutno při jeho nasazení brát v úvahu. Samozřejmě jsou podporovány jednoduché příkazy `CREATE` (`TABLE`, `[UNIQUE] INDEX`, `SEQUENCE`), `DROP`, `INSERT`, `DELETE`, `UPDATE`, `SELECT` (`WHERE`, `ORDER BY`, `AND`, `OR`). Nejsou podporovány agregátní funkce jako např.:

```
SELECT MAX(Plat) FROM Zamestnanci;
```

a vnořené příkazy `SELECT`. Zajímavou se může zdát podpora standardních operátorů `LIKE` a také vlastní implementace operátorů `RLIKE` (kompletní regulární výraz) či `SLIKE` (fonetické porovnávání).

Příkazy samotné je možno zadávat po spojení s databázovým serverem. Spojení je možno navázat pomocí programu `msql`, který je klientskou částí MiniSQL.

Systémové proměnné

Zajímavou vlastností serveru MiniSQL jsou také systémové proměnné — tedy jakési virtuální sloupce v tabulkách. Systémové proměnné poskytují informace nezávislé na vlastní struktuře a významu dat. Mezi implementované systémové proměnné patří `_rowid` (jednoznačný identifikátor řádku v tabulce), `_timestamp` (časová známka poslední modifikace řádku tabulky), `_sysdate` a `_systime` (aktuální datum a čas) a další:

```
mSQL > select _systime from pokusna_tabulka\g
```

```
Query OK. 1 row(s) modified or retrieved.
```

```

+-----+
| _systime |
+-----+
| 18:05:41 |
+-----+

```

```
mSQL >
```

Celkové hodnocení

MiniSQL je velice rychlým a jednoduchým databázovým serverem. Jeho rychlost je ovšem vyvážena nekompletní podporou jazyka SQL a tím i složitější prací. Jednoduchá administrace je zase protipólem příliš hrubému rozvrstvení přístupových práv (pouze pro databáze, přičemž lepší servery mohou přístupová práva specifikovat až na úrovni sloupců tabulek).

Před jeho nasazením je třeba pečlivě zvážit všechna pro a proti. Rychlost versus nízký komfort při administraci, jednoduchost versus kompatibilita. Ale i o tomto je databázový svět — databázových serverů je bezpočet, jen si vybrat ten nevhodnější.

O autorovi

Autor je nezávislým konzultantem v oboru informačních technologií, specializuje se na Linux, unixové operační systémy a programování Open Source projektů.