STUDIA INFORMATICA 2011
Volume 32 Number 2B (97)

Marcin SZUMILAK
Politechnika Lubelska, Instytut Informatyki, Koło Naukowe Informatyki „Pentagon”
Kamil ŻYŁA
Politechnika Lubelska, Instytut Informatyki

STRATEGIE ZARZĄDZANIA KOPIAMI ZAPASOWYMI DANYCH I METODY PRZYWRACANIA DANYCH W PRZYPADKU AWARII WSPÓŁCZESNYCH BAZ DANYCH MYSQL

Streszczenie. Ważne jest, aby każdy administrator posiadał plan działania w przypadku awarii powodujących utratę całości lub części danych, bądź uszkodzenie bazy danych. Niniejszy artykuł stanowi analizę strategii zarządzania kopiami zapasowymi danych oraz metod ich przywracania w przypadku awarii. Autorzy zwrócili uwagę na wydajność poszczególnych rozwiązań, zapewnienie ciągłości usług oraz integralności danych.

Słowa kluczowe: MySQL, bezpieczeństwo danych, MyISAM, InnoDB, HEAP

STRATEGIES OF MANAGING DATA BACKUPS AND METHODS OF RECOVERING MODERN MYSQL DATABASES AFTER FAILURE

Summary. It is vital for database administrator to have a roadmap in case of database failure resulting in complete or partial data loses. This article provides analysis of backup management strategies which can be used to quickly restore affected services. Authors focused on the efficiency of particular solutions and ways of ensuring the integrity of data.

Keywords: MySQL, data safety, MyISAM, InnoDB, HEAP

1. Wprowadzenie

Odtwarzanie stanu bazy danych sprzed awarii jest szczególnie istotne w przypadku poważnych awarii połączonych z utratą danych. Wobec groźby ich wystąpienia istotne jest od-

powiednie zarządzanie kopiami zapasowymi, w tym ich sprawne wykonywanie i przywracanie.

Wykonywanie kopii zapasowych wymaga rozważenia sposobu ich tworzenia, częstotliwości oraz rozróżniania kopii. Najważniejszą kwestią jest łatwy dostęp w przypadku potrzeby jej przywrócenia, na co przekłada się klarowne oznaczenie poszczególnych kopii oraz sposób zapisu danych, np. czy w pliku zrzutu przechowywane są poszczególne tabele z bazy danych, czy też każda z tabel jest zapisana w oddzielnym pliku. Bardzo ważne jest prawidłowe oznaczenie daty oraz monitorowanie, czy kopia zapasowa została poprawnie wykonana. Wybór odpowiedniej metody zależy od potrzeb, więc nie ma uniwersalnej metody, która będzie równie dobra w każdej sytuacji.

Mądrze wykonywane i zarządzane kopie zapasowe ułatwią walkę z często spotykanymi zagrożeniami danych, takimi jak:

- utrata zasilania bez skutków w postaci uszkodzenia plików, skutkująca niezamknięciem tabel, pozostawieniem blokad bądź niezapisaniem danych do tabeli;
- awaria dysku, awaria systemu plików, skutkująca błędami odczytu części danych lub ich całkowitą utratą;
- wstrzyknięcie kodu, skutkujące zmianą lub usunięciem całości lub części danych.

2. Silniki składowania danych MyISAM, InnoDB i MEMORY

Jednymi z najczęściej stosowanych w bazach danych MySQL silników składowania danych są MyISAM, InnoDB i MEMORY (HEAP). Wybranie możliwie najlepszej metody zapewnienia bezpieczeństwa przechowywanych danych wymaga poznania ich słabych i mocnych stron. Tabela 1 zawiera zwięzłe zestawienie najważniejszych cech poszczególnych silników.

Tabela 1

Porównanie silników składowania danych

Właściwość	MyISAM	InnoDB	MEMORY
Maksymalna wielkość tabeli	256TB	64TB	Ograniczony przez RAM
Obsługa transakcji	Nie	Tak	Nie
Obsługa kluczy obcych	Nie	Tak	Nie
Kompresja	Tak	Tak	Tak
Poziom blokady	Tabela	Wiersz	Tabela
Indeksowanie na podstawie drzewa binarnego	Tak	Tak	Tak
Indeksowanie wyszukiwania pełnotekstowego	Tak	Nie	Nie
Urządzenie zapisu danych	Dysk	Dysk	RAM

MyISAM to domyślny silnik bazy danych MySQL i jednocześnie jeden z najszybszych silników składowania danych, niestety jego główną wadą jest brak wsparcia dla kluczy obcych. MyISAM wykorzystuje następujące typy plików:

- frm – pliki zawierające strukturę tabel (nazwy kolumn oraz typy danych),
- MYD – pliki zawierające dane,
- MYI – pliki zawierające informacje o indeksach.

InnoDB jest wolniejszy niż MyISAM, jednakże oferuje wsparcie dla kluczy obcych, transakcyjność przetwarzania oraz blokowanie na poziomie wiersza, pozwalające na wykonywanie wielu równoległych operacji zapisu. InnoDB wykorzystuje dwa typy plików:

- frm – pliki zawierające strukturę tabel,
- ibd – pliki zawierające dane i informacje o indeksach.

Ponadto, tabele posiadają odpowiednie wpisy w wewnętrznym słowniku danych InnoDB w przestrzeni tabel [7].

Ostatni z silników (MEMORY) przechowuje dane w pamięci operacyjnej hosta (struktura tabel znajduję się na dysku komputera w postaci plików frm), co pozwala na bardzo szybkie wykonywanie na nich operacji. Niestety głównymi wadami tego rozwiązania są nietrwałość przechowywanych danych, spowodowana ulotnością pamięci RAM (restart bądź awaria bazy danych MySQL spowoduje utratę danych), oraz ograniczenie ich rozmiaru przez dostępną pamięć operacyjną. Ponadto, HEAP nie obsługuje typów danych, takich jak TEXT oraz BLOB.

3. Metody tworzenia kopii zapasowych

Metody tworzenia kopii zapasowych można podzielić na dwie kategorie – fizyczne i logiczne. Pierwsza kategoria dotyczy plików bazy danych, druga dotyczy zawartości tychże plików. Metody fizyczne są zdecydowanie szybsze, ze względu na brak konieczności jakiegokolwiek przetwarzania składowanych danych. Ponadto, kopie zapasowe wykonane metodami fizycznymi zajmują mniej przestrzeni dyskowej niż kopie zapasowe wykonane metodami logicznymi, które są często składowane jako pliki tekstowe [5].

Wśród narzędzi wspomagających wykonywanie logicznych kopii zapasowych można wyróżnić:

1) polecenie SELECT INTO OUTFILE (LOAD DATA INFILE),
2) mysqldump,
3) logi binarne (dziennik zdarzeń).

Do fizycznych metod wykonywania kopii zapasowej zalicza się kopiowanie plików bazy danych. Można je zastosować do silników MyISAM oraz InnoDB. Ideą metody jest skopiowanie plików zawierających dane oraz ich strukturę do odpowiedniego katalogu. Należy przy tym zabezpieczyć dane przed utratą spójności, jeśli są kopiowane podczas pracy serwera bazy danych, przez zatrzymanie tego serwera lub zablokowanie dostępu do tabel znajdujących się w kopiowanych plikach.

SELECT INTO OUTFILE można zastosować do silników MyISAM, InnoDB oraz HEAP. Metoda polega na zapisaniu danych (kolumny tabeli są rozdzielone znakiem \t) do pliku utworzonego przez instrukcję języka SQL. W trakcie tworzenia kopii zapasowej eksportowana jest tylko zawartość tabel (bez ich definicji), a do jej przywrócenia można użyć instrukcji LOAD DATA INFILE. SELECT INTO OUTFILE jest domyślnie wyłączona, ze względu na problemy z bezpieczeństwem LOAD DATA INFILE. Mogą też pojawić się pewne problemy ze zgodnością metod porównywania łańcuchów znakowych oraz strony kodowej przywracanych danych, warto więc zawczasu zadbać o ich zgodność z definicjami tabel. Pomimo swoich wad, opisywana metoda zyskała sporą popularność ze względu na dobrą dokumentację oraz dużą wydajność w specyficznych zastosowaniach.

Program mysqldump może zostać użyty do wykonania kopii serwera baz danychs, bazy danych lub tabeli niezależnie od silnika składowania danych oraz do migracji pomiędzy różnymi wersjami serwera MySQL (możliwość aktywowania kompatybilności wstecznej). Jego idea polega na eksportowaniu danych razem z ich strukturą do pliku – skryptu SQL. Plik wynikowy zawiera strukturę bazy danych, dane z tabel oraz ustawienia kodowania. W systemach nietransakcyjnych mysqldump blokuje niezbędny zbiór tabel w celu zapewnienia spójności kopiowanych danych. W systemach transakcyjnych kopia zapasowa może być wykonywana jako nowa transakcja.

Logi Binarne (nazywane również: Binary Logs lub binlogs) mogą być zastosowane do silników MyISAM, InnoDB oraz HEAP oraz w trakcie replikacji danych pomiędzy nadrzędnym i podrzędnym serwerem MySQL. Są to pliki zawierające historię wszystkich operacji wykonanych przez serwer bazy danych, w tym tworzenie, usuwanie i modyfikowanie baz danych i tabel oraz instrukcje manipulujące danymi, takie jak update, replace, insert i delete. Tworzenie logów binarnych jest domyślnie wyłączone, jednakże można je włączyć w pliku konfiguracyjnym MySQL. Wyróżnia się dwa podstawowe formaty logów binarnych:

- statement-based – zawiera każde wykonane zapytanie, nie może być użyty do zapytań niedeterministycznych, zawierających wywołania funkcji lub klauzulę LIMIT.
- row-based – zawiera zmiany indywidualnych wierszy tabeli w postaci oddzielnych zapytań, co powoduje znaczne zwiększenie rozmiaru plików z logami.

Trzeci format – mieszany (mixwd), łączący w sobie cechy dwóch powyższych – zachowuje się jak statement-based, gdzie tylko to możliwe (dla zapytań deterministycznych), natomiast jak row-based w pozostałych przypadkach. Jego użycie jest rekomendowane, ponieważ w pewnych sytuacjach wykorzystanie któregoś z podstawowych formatów może spowodować, że dane odtworzone z dziennika zdarzeń będą się różnić od danych wprowadzonych do bazy pierwotnie. Logi mogą być przetwarzane narzędziem mysqlbinlog, a wszelkie zarejestrowane zmiany powtórzone i wykonane podobnie do instrukcji języka DML i DDL.

4. Środowisko i procedura testowania

Analiza i testowanie wydajności opisanych metod zostały wykonane na podstawie następującej platformy sprzętowej: procesor Pentium IV 2,26GHz (Prescott), 1536MB DDR RAM 266MHz, dyski 2 x Western Digital WD400 40GB IDE, 7200RPM, 2MB Cache.

Rozmiar bazy danych „stovenna" (212MB plik mysqldump), użytej do testów, został ograniczony przez ilość pamięci RAM. Baza danych składała się z tabeli o nazwie „selbat", zawierającej 1 000 000 wierszy, i strukturze przedstawionej w tabeli 2. Ze względu na silnik HEAP, wykluczono duże obiekty. Kolumna Id została ustawiona jako klucz główny z włączoną autoinkrementacją.

Tabela 2

Struktura testowej tabeli

Nazwa kolumny	Typ pola
Id	int(11)
www	varchar(500)
redirect	varchar(500)
data	datetime
system1	varchar(500)
system2	varchar(500)
system3	varchar(500)
system4	varchar(500)

Wszystkie testy przeprowadzono z użyciem drugiego takiego samego dysku, który służył do składowania kopii zapasowych. Autorzy użyli drugiego dysku, a nie zdalnych kopii zapasowych, ze względu na ewentualny wpływ szybkości połączenia sieciowego oraz innych opóźnień związanych z pracą modułów sieciowych systemu operacyjnego. Nie zmienia to jednak faktu, że wykonywanie kopii zdalnych jest zalecane w środowisku produkcyjnym.

System operacyjny FreeBSD został skonfigurowany tak, aby działały wyłącznie procesy systemowe. Usługi, takie jak syslog i cron, zostały wyłączone. Przed każdym kolejnym testem eliminowano pozostałości danych z pamięci cache urządzeń.

Czas trwania testowanych operacji został określony na podstawie daty systemowej, wyświetlonej przez polecenie powłoki na początku i na końcu każdej z prób. Jedynie czas wykonania poleceń wydanych w wierszu poleceń MySQL był zliczany automatycznie przez MySQL.

5. Analiza metod odtwarzania baz danych MySQL

Baza danych „stovenna" została dostosowana do każdego z silników składowania danych, a następnie używana przez wszystkie metody wykonywania kopii zapasowych opisane w rozdziale 2. W zależności od silnika, pliki zawierające testową tabelę miały inny rozmiar, co przedstawiono w tabeli 3.

Tabela 3

Rozmiar testowej tabeli w zależności od silnika składowania danych

Silnik	MyISAM	InnoDB	Heap/MEMORY
Rozmiar	197MB	258MB	~1200MB

Nie należy zapominać, że MySQL oferuje wiele mechanizmów zapewniania bezpieczeństwa danych. Niekiedy łatwiej i szybciej jest naprawić kilka tabel, niż przywracać je z kopii zapasowej. Jako przykład można podać reset serwera MySQL, np. z powodu przerwy w zasilaniu, po którym zazwyczaj nie zachodzi potrzeba przywrócenia bazy danych z kopii zapasowej. Warto więc najpierw sprawdzić, czy wykonanie polecenia REPAIR TABLE lub użycie programu mysqlcheck jest wystarczające.

5.1. Odtwarzanie tabel MyISAM

5.1.1. Kopiowanie plików

Silnik MyISAM pozwala na wykonanie fizycznej kopii zapasowej, polegającej na skopiowaniu odpowiednich plików. Procedura skopiowania testowej tabeli była następująca:

1. Zablokowanie tabeli „selbat".
2. Skopiowanie plików tabeli „selbat" na drugi dysk twardy.
3. Odblokowanie tabeli „selbat".

Wykonanie powyższych kroków trwało 8 sekund (rys. 1) i wiązało się z wydaniem następującego polecenia powłoki systemowej:

mysql --user='root' --password='mysecrectpassword' -B -e 'lock tables selbat;' && rsync -av /var/db/mysql/stovenna/selbat. /mnt/backup/stovenna/ && mysql --user='root' --password='mysecrectpassword' -B -e 'unlock tables;'*

Procedura odtworzenia tabeli na podstawie skopiowanych plików była następująca:

1. Skopiowanie plików z drugiego dysku twardego do katalogu bazy danych.
2. Odświeżenie tabel otwartych przez serwer bazy danych.
3. Odblokowanie odtworzonych tabel.

Wykonanie powyższej procedury również zajęło 8 sekund (rys. 1) i wiązało się z wydaniem następującego polecenia powłoki systemowej:

rsync -av /mnt/backup/stovenna/selbat. /var/db/mysql/stovenna/ && mysql --user='root' --password='mysecrectpassword' -B -e 'flush tables; unlock tables;'*

'flush tables; unlock tables;' zmusza serwer MySQL do zamknięcia wszystkich otwartych tabel, ponownego odczytania ich struktury i odblokowania zablokowanych tabel (czas wykonania tych czynności jest pomijalnie krótki). Podobny efekt dałoby się osiągnąć, resetując serwer bazy danych, jednakże jest to relatywnie czasochłonne oraz wiąże się z przerwaniem ciągłości usług.

W powyższym przypadku stan bazy danych przed odtworzeniem nie gra roli. Tabele mogą istnieć bądź nie, mogą być wypełnione danymi bądź puste. Ważne jest natomiast, aby kopiować pliki pomiędzy tymi samymi wersjami serwera bazy danych. W pewnych przypadkach nawet niewielka różnica wersji może wpłynąć na poprawność procesu odzyskiwania danych.

5.1.2. SELECT INTO OUTFILE (LOAD DATA INFILE)

Kopia zapasowa zawartości tabeli „selbat" może zostać wykonana przez wydanie następującego polecenia w konsoli MySQL:

*select * from stovenna.selbat into outfile '/mnt/backup/stovenna_backup'*

Wykonanie polecenia zajęło 24 sekundy (rys. 1), a jego wynikiem było utworzenie pliku stovenna_backup wewnątrz katalogu /mnt/backup.

Odtworzenie tak wykonanej kopii zapasowej wymaga istnienia bazy danych, pustej tabeli o odpowiedniej strukturze oraz wydania następującego polecenia w konsoli MySQL:

load data infile '/mnt/backup/stovenna_backup' into table stovenna.selbat

Powyższe polecenie umieściło dane z pliku /mnt/backup/stovenna_backup w pustej tabeli „selbat", co zajęło 35 sekund (rys. 1).

Podczas odtwarzania danych mogą wystąpić dwa typy problemów:

1. Niekompatybilność pól i typów – wynik próby umieszczenia danych w tabeli o niewłaściwej strukturze.
2. Niezgodność stron kodowych – wynik odtworzenia danych tekstowych bez odpowiedniej konwersji (jeśli była konieczna).

5.1.3. mysqldump

Wykonanie kopii zapasowej przy użyciu programu mysqldump wiąże się z wydaniem następującego polecenia powłoki systemowej:

mysqldump --opt –quote-names --user=root –password='mysecretpassword' stovenna > /mnt/backup/stovenna.sql

Utworzenie skryptu SQL (/mnt/backup/stovenna.sql), zawierającego dane oraz ich strukturę, zajęło 37 sekund (rys. 1).

Parametr --quote-names powoduje umieszczenie nazw w cudzysłowie, co pozwala na użycie nazw zabronionych w bazie danych. Proces odtwarzania danych nie zostaje przerwany w przypadku ich wystąpienia.

W celu odtworzenia danych z tak wykonanej kopii zapasowej, należy wydać następujące polecenie powłoki systemowej:

mysql --user='root' --password='mysecrectpassword' stovenna < /mnt/backup/stovenna.sql

Przed jego wykonaniem, które zajęło 57 sekund (rys. 1), utworzono pustą bazę danych o nazwie „stovenna".

Stan bazy danych, wymagany przed odtworzeniem danych, zależy od zawartości skryptu SQL. Jeśli nie zawiera instrukcji CREATE DATABASE, baza danych musi zostać utworzona ręcznie. Jeśli nie zawiera instrukcji CREATE TABLE, należy ręcznie utworzyć tabele.

W przypadku gdy strona kodowa nie została określona (nie użyto parametru --opt) i jednocześnie jest inna niż zdefiniowana jako domyślna dla bazy danych, pewne znaki mogą zostać zastąpione przez inne, np. umieszczenie danych w UTF8 w tabelach z kodowaniem latin2 spowoduje zastąpienie cudzysłowu przez znak zapytania.

5.1.4. Logi binarne

Logi binarne są używane do odtwarzania różnicowych kopii zapasowych. Czas wykonania kopii zapasowej nie został zmierzony, ponieważ logi binarne zawierają wszystkie instrukcje wykonane przez bazę danych, więc czas ich powstawania zależy od czasu wykonania tych instrukcji i czasu działania bazy danych od momentu zapisania pierwszej z nich.

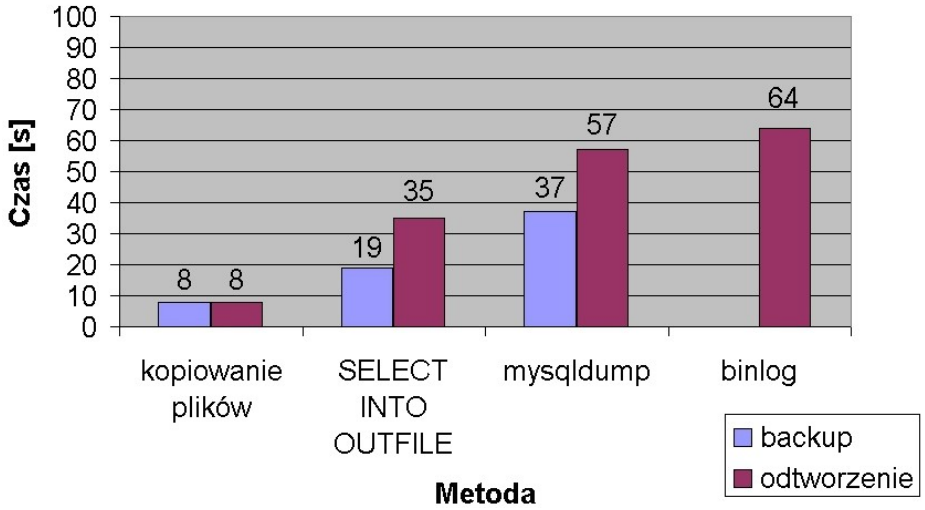
Odtworzenie bazy danych „stovenna" zajęło 64 sekundy (rys. 1). Baza danych nie istniała przed odtworzeniem, a logi binarne (typu mieszanego) zawierały instrukcje tworzące bazę danych i strukturę jej tabel. W celu odtworzenia danych wydano następujące polecenie powłoki systemowej:

mysqlbinlog --character-set=utf8 --database= stovenna /var/log/mysql/binlog | mysql --default-character-set=utf8 --user='root' --password='mysecrectpassword' stovenna

Pierwsza część polecenia odczytuje instrukcje zapisane w logu, a kolejna część wykonuje je.

Możliwe jest odtworzenie konkretnego stanu bazy danych, posługując się parametrami --start-position i --stop-position. W zależności od zapisanych instrukcji, baza danych może wymagać odpowiedniego przygotowania. Poniżej zaprezentowano przykładowe polecenie powłoki systemowej, odczytujące wybrany ciąg instrukcji z logu binarnego:

mysqlbinlog --character-set=utf8 --database= stovenna /var/log/mysql/binlog --start-position=966 --stop-position=221887782



Rys. 1. Czasy odtworzenia testowej tabeli dla silnika MyISAM
Fig. 1. MyISAM backup and recovery times

5.2. Odtwarzanie tabel InnoDB

5.2.1. Metody fizyczne

Skopiowanie plików bazy danych jest o wiele bardziej skomplikowane niż w przypadku silnika MyISAM. Procedura wykonania kopii zapasowej bazy danych „stovenna" , zawierającej jedną tabelę „selbat", była następująca:

1. Zatrzymanie serwera bazy danych.
2. Skopiowanie plików: Ibdata, /var/db/mysql/stovenna/selbat.ibd
   4) i /var/db/mysql/stovenna/selbat.frm.
3. Uruchomienie serwera bazy danych.

Kopiowanie plików zajęło 13 sekund (rys. 2), ponieważ ich rozmiar był większy niż w przypadku silnika MyISAM.

Odtworzenie tabeli z tak wykonanej kopii zapasowej wymagało:

1. Zatrzymania serwera bazy danych.
2. Skopiowania plików *.ibd i *.frm do katalogu /var/db/mysql/stovenna/.
3. Uruchomienia serwera bazy danych.

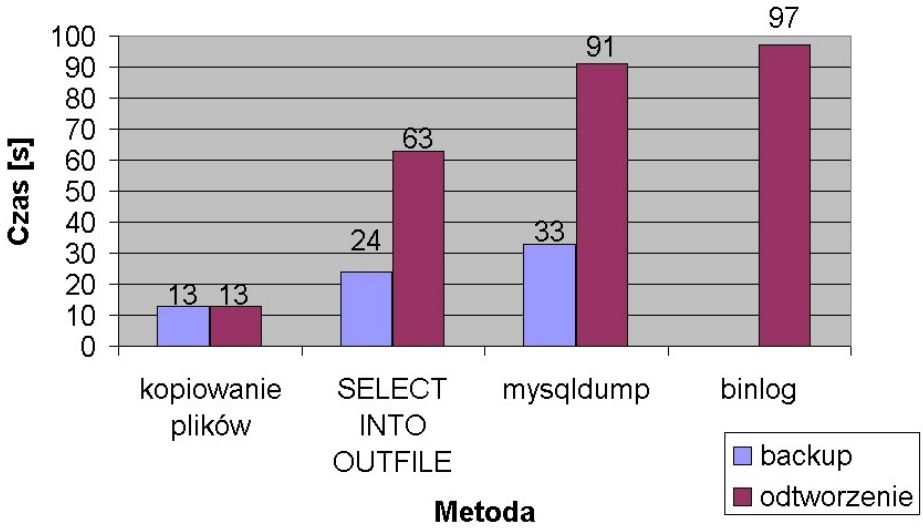
Przywrócenie plików również zajęło 13 sekund (rys. 2).

InnoDB zapisuje informacje o bazach danych i tabelach w kilku miejscach, w związku z tym należy zwrócić uwagę, czy pliki są przywracane dokładnie w te miejsca, z których zostały skopiowane. Nie jest również możliwa zmiana struktury danych oraz przenoszenie tabel do innych baz danych lub serwerów baz danych, gdzie baza danych, posiadająca tę samą strukturę co przenoszone dane, nie istniała.

Niekiedy zachodzi potrzeba odtworzenia tabel InnoDB na serwerze MySQL, gdzie nie istniała baza danych o odpowiedniej strukturze. Wtedy wszystkie pliki bazy danych, ibdata, lb_logfile0 i lb_logfile1 powinny zostać skopiowane (serwer nie może zawierać innych baz danych InnoDB). Po odtworzeniu bazy danych, zbędne tabele i bazy danych mogą zostać usunięte.

5.2.2. Metody logiczne

Procedury wykonywania logicznych kopii zapasowych różnią się, względem silnika MyISAM, jedynie czasami wykonania (rys. 2).



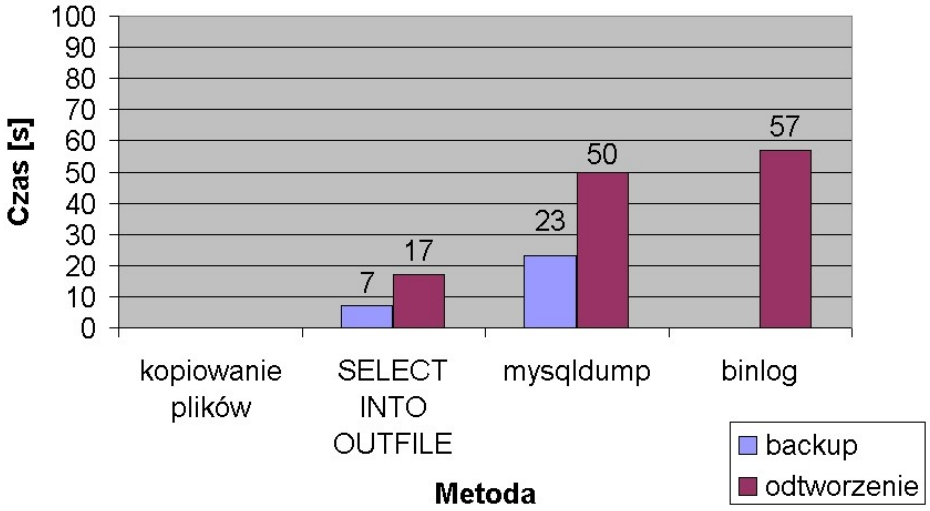
Rys. 2. Czasy odtworzenia testowej tabeli dla silnika InnoDB
Fig. 2. InnoDB backup and recovery times

5.3. Odtwarzanie tabel MEMORY

W przypadku silnika MEMORY nie jest możliwe wykonanie fizycznej kopii zapasowej przez skopiowanie plików z danymi, bo ich nie ma. Dane są składowane w pamięci RAM hosta, na dysku twardym znajduje się wyłącznie struktura tych danych.

Procedury wykonywania logicznych kopii zapasowych różnią się, względem pozostałych silników, jedynie czasami wykonania (rys. 3).

Ze względu na ulotność składowanych danych, nie ma procedur recovery dla silnika MEMORY.



Rys. 3. Czasy odtworzenia testowej tabeli dla silnika MEMORY
Fig. 3. MEMORY backup and recovery times

6. Podsumowanie

Każda z zaprezentowanych i zbadanych metod wykonywania oraz odtwarzania kopii zapasowej danych powinna być stosowana w specyficznych warunkach, aby uzyskać jej maksymalną wydajność.

Kopiowanie plików jest bardzo szybką metodą wykonywania kompleksowych kopii zapasowych. Pozwala błyskawicznie przywrócić dane wraz z ich logiczną organizacją. Niestety nie można polecić jej stosowania podczas codziennej pracy serwera MySQL, ponieważ wymaga zatrzymania serwera lub zablokowania tabel przeznaczonych do skopiowania.

Metoda SELECT INTO OUTFILE ma relatywnie wąski zakres zastosowań. Może zostać użyta do szybkiego przenoszenia dużych zbiorów danych z kilku tabel (najczęściej z baz danych opartych na silniku InnoDB) pomiędzy bazami danych lub serwerami MySQL. Niestety nie pozwala na zachowanie struktury danych, której kopia musi zostać wykonana inną metodą.

Logi binarne powinny być wykorzystywane do odtwarzania zmian, które zaszły w bazie danych od czasu wykonania ostatniej kopii zapasowej do awarii.

Mysqldump jest jednym z najwolniejszych rozwiązań, ale za to niezawodnym. W każdej sytuacji pracuje stabilnie i pozwala wykonać kopię zapasową danych razem z ich strukturą. Powinien być zastępowany przez inne metody jedynie w specyficznych sytuacjach, takich jak opisane powyżej.

Podsumowując, najlepszym kompromisem pomiędzy kłopotliwością i wydajnością odtwarzania danych w bazach danych MySQL jest:

1) kopiowanie plików i mysqldump – silnik MyISAM,
2) mysqldump – silnik InnoDB,
3) mysqldump – silnik HEAP.

BIBLIOGRAFIA

1. Dubois P.: MySQL Cookbook. O'Reilly, Sebastopol 2002.
2. Halling D. J., Zawodny J.: High Performance MySQL. O'Reilly, Sebastopol 2004.
3. MySQL AB.: MySQL® Administrator's Guide. Sams Publishing, Canada 2004.
4. Pachev A.: MySQL Enterprise Solutions. Wiley Publishing, Inc., Indianapolis 2003.
5. MySQL Backup and Recovery online documentation, http://dev.mysql.com/doc/refman/ /5.1/en/backup-and-recovery.html.
6. MySQL Storage Engines online documentation, http://dev.mysql.com/doc/refman-/5.1/en/storage-engines.html.
7. http://dev.mysql.com/doc/refman/5.1/en/innodb-table-and-index.html.

Recenzenci: Dr inż. Paweł Kasprowski
Dr inż. Bożena Małysiak-Mrozek

Wpłynęło do Redakcji 16 stycznia 2011 r.

Abstract

MySQL as a free of charge tool for data storage and management has gained high popularity, which resulted in a big number of its commercial applications. Storing and processing vast amounts of data requires ability to make quick data backup and to restore the database efficiency rapidly after a failure, even a hardware one.

This article provides an analysis of risks associated with storing and processing data in MySQL databases based on MyISAM, InnoDB and HEAP engines. Authors focused on the problems of: preserving data integrity in case of a failure, data backup management, data restoring and ensuring the continuity of services.

Article contains description of MyISAM, InnoDB and HEAP engines, which characteristics are presented in table 1. Each engine can be backed up using specific subset of following methods: copying files, SELECT INTO OUTFILE, mysqldump and binary logs. Analysis of

mentioned methods applications is extended by tests performed on one-table database containing 1 mln rows, which was adjusted to each engine. Test includes making physical and logical backup and restoring it. Measured times of execution are presented as charts in figures form 1 to 3.

Description of performed trials, good practices, potential threats and indication of the most suitable methods of backing up and restoring data for each engine makes this article particularly helpful in broadening the competences of administrators, who come across described problems every day.

Adres

Kamil ŻYŁA: Politechnika Lubelska, Instytut Informatyki, ul. Nadbystrzycka 36b, 20-618 Lublin, Polska, kamilz@cs.pollub.pl.