PRACE NAUKOWE Akademii im. Jana Długosza w Częstochowie
Technika, Informatyka, Inżynieria Bezpieczeństwa 2016, t. IV, s. 49–57

http://dx.doi.org/10.16926/tiib.2016.04.04

Szymon Berski[1], Radosław Szatan
[1] *Politechnika Częstochowska*
al. Armii Krajowej 19, 42-200 Czestochowa
e-mail: berski@wip.pcz.pl

ZASTOSOWANIE KLASTRA WYSOKIEJ DOSTĘPNOŚCI W BEZPIECZEŃSTWIE RELACYJNYCH BAZ DANYCH

Streszczenie. W pracy zaproponowano mechanizm zapewniania bezpieczeństwa danych poddanych zagrożeniom wynikającym z awarii systemu komputerowego lub sprzętu komputerowego. Przedstawione w pracy rozwiązanie polega na uruchomieniu klastra wysokiej dostępności dla bazy danych środków trwałych Instytutu Przeróbki Plastycznej i Inżynierii Bezpieczeństwa opartego o narzędzie do replikacji danych oraz narzędzie do kontroli dostępności poszczególnych węzłów klastra. W pracy zastosowano DRBD - narzędzie do replikacji urządzeń blokowych, które działa jako moduł jądra służący do mirrorowania systemów plików przez sieć [6]. DRBD umożliwia ochronę danych komputerowych poprzez zapis lustrzany w pamięciach dyskowych. Polega to na dublowaniu zapisu danych, w wyniku czego dwa urządzenia przechowują taką samą informację. Jako narzędzia kontroli dostępności poszczególnych węzłów klastra w pracy użyto z kolei systemu Heartbeat, który sprawdza operacje wejścia-wyjścia w celu zapewnienia spójności danych, a następnie przenosi zasoby na alternatywny węzeł [5].
Słowa kluczowe: bezpieczeństwo danych, replikacja danych, bazy danych.

APPLICATION OF HIGH AVAILABILITY CLUSTER FOR SECURITY OF RELATIONAL DATABASES

Abstract. In the work the security system of data subjected to risks arising from the failure of a computer system or computer equipment was presented. The analyzed solution is based on creation of high-availability cluster for database of fixed assets in Institute of Metal Forming and Safety Engineering. The proposed security system is based on DRBD software, replicated storage solution mirroring the content of hard disks, partitions, logical volumes etc. between hosts connected via Ethernet [6]. DRBD enables replication of selected services like MySQL server or http server to ensure mirroring of data managed by these servers. The Heartbeat service was used in the analyzed solution

as a tool to control availability of the several nodes of the cluster to check the input–output operations and decides when to transfer the selected resources to satellite node [5].
Keywords: data security, data replication, databases.

Cel pracy

Celem pracy jest zwiększenie bezpieczeństwa danych przechowywanych i zarządzanych przez relacyjny system zarządzania bazami danych (SZBD) MySQL® [1]. Zaproponowany model do badań to klaster zbudowany z dwóch serwerów: nadrzędnego i podrzędnego, oba serwery pracują w oparciu o system operacyjny Linux Gentoo [7]. Serwer nadrzędny realizuje funkcje systemu zaporowego oraz dostarcza działających usług, natomiast serwer podrzędny testuje funkcje serwera nadrzędnego i nieprzerwanie komunikuje się z nim poprzez protokół Heartbeat [1]. Na serwerze nadrzędnym zainstalowano i skonfigurowano system zarządzania bazą danych MySQL oraz, korzystając z narzędzi importu danych, utworzono i skonfigurowano bazę danych środków trwałych. Baza zawierająca niezbędne dane dotyczące środków trwałych oraz osób, które mają nad poszczególnymi środkami pieczę została zaimportowana z kopii bazy działającej w Instytucie Przeróbki Plastycznej i Inżynierii Bezpieczeństwa na serwerze z systemem operacyjnym Debian i SZBD MySQL. W przypadku, kiedy serwer nadrzędny ulegnie awarii, system automatycznie przejdzie w stan naprawy, przekazując funkcje systemu zaporowego serwerowi podrzędnemu, i wyłączy serwer nadrzędny. Do poprawnego działania zaproponowanego rozwiązania skonfigurowano usługi DRBD oraz Heartbeat [5, 6]. Weryfikacja poprawności działania zaproponowanego rozwiązania zostanie przeprowadzona poprzez zamodelowanie awarii serwera nadrzędnego i przejęcia funkcji gromadzenia i przetwarzania danych przez serwer podrzędny. Dane, które będą podlegać zabezpieczeniu to dane dotyczące środków trwałych, związanych z nimi dokumentów finansowych oraz wrażliwe dane osobowe.

Elementy systemu zapewniania zwiększonego bezpieczeństwa bazy danych

Proponowany system zapewniania zwiększonego bezpieczeństwa składa się z klastra dwóch serwerów pracujących pod kontrolą systemu operacyjnego Gentoo oraz narzędzi takich jak: MySQL, DRBD oraz Heartbeat.

Urządzenie DRBD oraz podstawowe parametry konfiguracyjne

DRBD (*Distributed Replicated Block Device*) jest to softwarowa wersja macierzy RAID1, w analizowanym przypadku dla dwóch komputerów połączonych siecią IP, realizowana jako moduł jądra systemu Linux [4]. Do badań wykorzystano dwie maszyny z zainstalowanym systemem operacyjnym Linux Gentoo. Na dysku utworzono urządzenie DRBD. W procesie konfiguracji ustalono podstawowe parametry usługi DRBD:

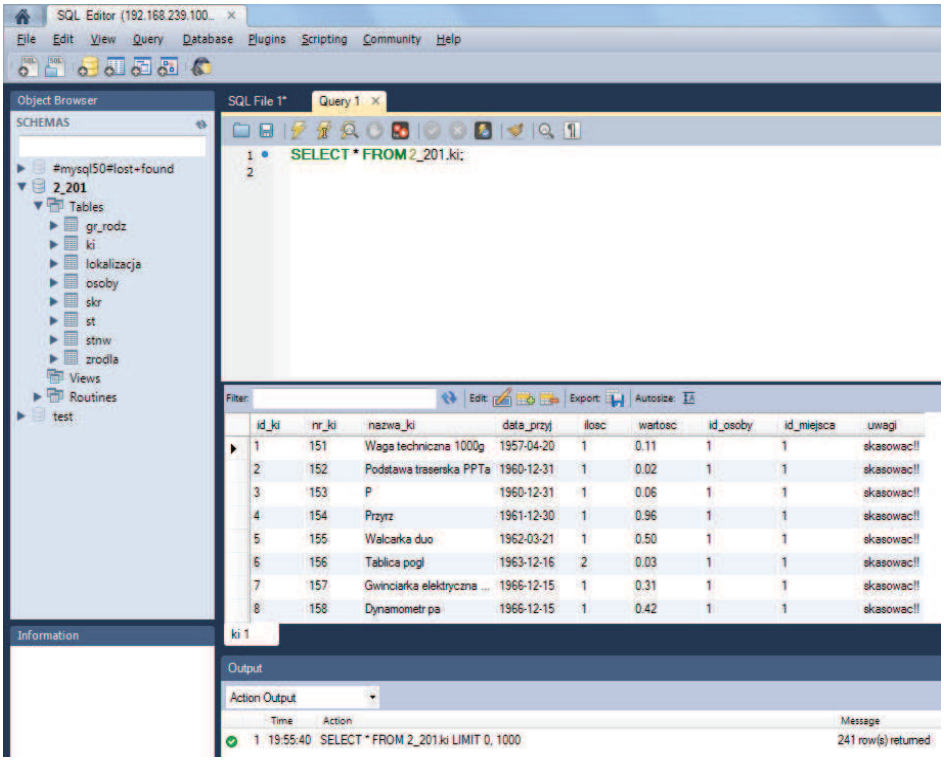
```
Vhandlers {pri-on-incon-degr "echo 'DRBD: primary requested but inconsistent!' | wall; /etc/init.d/heartbeat stop"; #"halt -f";pri-lost-after-sb "echo 'DRBD: primary requested but lost!' | wall; /etc/init.d/heartbeat stop"; #"halt -f";}
//definicja działania w przypadku niespójności danych lub utraty komunikacji//
startup {wfc-timeout 0; # 2 minutes degr-wfc-timeout 120; # 2 minutes.}
//określenie czasu oczekiwania na połączenie z węzłami//
syncer {rate 10M; # 10 MByte/s = 80 Mbit/s al-extents 257;} //definicja dopuszczalnej prędkości synchronizacji//
net {cram-hmac-alg "sha256"; shared-secret "Haslo"; after-sb-0pri discard-younger-primary; after-sb-1pri consensus; after-sb-2pri disconnect; rr-conflict disconnect;} //uwierzytelnianie
parametrów węzła ze wspólnego hasła poprzez skrót sha256//
disk {on-io-error pass_on;}} // Postępowanie na wypadek błędów I/O na dysku//
resource "drbd0" {device /dev/drbd0; disk /dev/sdb1; meta-disk internal; on serwer1 { address 10.0.0.1:7788; } on serwer2 { address 10.0.0.2:7788;}}
//Definicja urządzenia DRBD0 wraz z fizycznymi parametrami oraz adresami serwerów posiadających odpowiednie urządzenia DRBD//
```

Po skonfigurowaniu ustawień usługi DRBD na maszynach: ***serwer1*** i ***serwer2***, w razie awarii lub konieczności wyłączenia maszyny ***serwer1***, na maszynie ***serwer2*** istnieje gotowa do użycia kopia partycji/systemu plików.

Konfiguracja i import bazy danych

W celu uruchomienia bazy danych środków trwałych IMiAPPP o nazwie: ***2_201*** zainstalowano SZDB MySQL w systemie operacyjnym Gentoo. Dokonano podstawowej konfiguracji SZBD MySQL tj, założono uruchamianie usługi mysqld wraz ze startem systemu operacyjnego, określono położenie plików konfiguracyjnych (/etc/mysql/my.cnf), udostępniono nasłuchiwanie na wszystkich adresach określono podstawowych użytkowników oraz ich uprawnienia i sposób kontroli dostępu. Przy użyciu podstawowego narzędzia, jakim jest program MySQL, utworzono bazę testową o nazwie: ***gentoo.***

Do procesu importu bazy danych wykorzystano narzędzie MySQL Workbench. Następnie zweryfikowano poprawność zaimplementowania wszystkich danych w nowej bazie. Na rys. 1 przestawiono fragment danych z tabeli ***karoteka ilościowa*** importowanej bazy.



Rys. 1. Struktura oraz fragment danych zaimportowanej bazy ***2_201***

Wszystkie istotne dane zostały poprawnie wprowadzone do nowej bazy, wystąpił jedynie problem z importem czcionek z polskimi literami diakrytyzowanymi. Problem został usunięty w następnej operacji importu, w której ujednolicono stronę kodową czcionek (utf8).

Konfiguracja Heartbeat

Klaster wysokiej dostępności wymaga dodatkowego połączenia między węzłami zwanego Heartbeat. Połączenie to służy węzłom do wzajemnej weryfikacji czy pracują poprawnie. W badanym przypadku usługa Heartbeat wykorzystuje interfejs sieciowy *ethernet* [2, 3]. Wybrane parametry konfiguracji przedstawiono na poniższych listingach z komentarzami:

```
# What interfaces to heartbeat over?#udp eth1bcast eth1 //Definicja trybu pracy na określonym interfejsie sieciowym, broadcast na eth1.//
# keepalive: how many seconds between heartbeats keepalive 2000ms //Określenie interwału w ms, co ile sprawdzany jest stan węzłów.//
```

```
# Time in seconds before issuing a "late heartbeat" warning in the logs.
warntime 10 //Liczba sekund, po której logowana jest informacja o opóźnieniu odpowiedzi.//
# Node is pronounced dead after 15 seconds. deadtime 15 # With some configurations, the network takes some time to start working after a reboot.# This is a separate "deadtime" to handle that case. It should be at least twice the normal deadtime.initdead 30 //Czas, po którym węzeł jest uznawany za wyłączony.//
# Mandatory. Hostname of machine in cluster as described by uname -n. node serwer1 node serwer2 //Określenie nazw hostów/serwerów tworzących klaster heartbeat.//
# When auto_failback is set to on once the master comes back online, it will take # everything back from the slave. auto_failback off //Ustawienie, które stwierdza czy po awarii master ma zostać przywrócony poprzedniemu masterowi czy następuje zmiana ról.//
# Some default uid, gid info, This is required for ipfail apiauth default uid=nobody gid=nobody apiauth ipfail uid=nobody apiauth ping gid=nobody uid=nobody //Określenie nazw użytkownika oraz identyfikatorów grup.//
# This is to fail over if the outbound network connection goes down. respawn nobody /usr/lib/heartbeat/ipfail //Definicja tekstu określającego prawidłowe połączenie z siecią - awaria jest nie tylko wyłączeniem węzła, ale także utratą przez niego komunikacji ze światem.//
#debugfile /var/log/ha-debug # File to write other messages to #logfile /var/log/ha-log # Facility to use for syslog()/logger #logfacility local0 //Określenie poziomu logowania działań serwera wraz ze ścieżkami do plików dziennika.//
```

Bardzo istotnym elementem jest określenie, jakie usługi oraz zbiory danych mają być poddane replikacji. W badanym przypadku są to: system bazodanowy MySQL oraz serwer http:

```
serwer1 192.168.239.100 drbddisk::drbd0
Filesystem::/dev/drbd0::/var/hiavse/apache::ext4 apache2
serwer1 drbddisk::drbd1 Filesystem::/dev/drbd1::/var/hiavse/mysql::ext4 mysql
```

Analiza pracy usługi DRBD w badanym systemie

1. Zaproponowany model do badań to klaster zbudowany z dwóch serwerów o nazwach ***„serwer1”*** – nadrzędny, i „***serwer2***” - podrzędny. Zbudowany klaster pełni rolę wysoko dostępnej platformy, co oznacza, że jest odporny na awarię. W przypadku, kiedy ***serwer1*** zlokalizuje awarię, system automatycznie przejdzie w stan naprawy, przekazując funkcje systemu zaporowego ***serwerowi2*** i wyłączy ***serwer1***.

Uruchamiając polecenie na ***serwerze1***: # `rc-status` można uzyskać informację z listy usług, jakie zostały uruchomione, czyli: *DRBD*, *Heartbeat*, *MySQL* i *Apache2*, oraz w jakim trybie działają. W tym wypadku jest to "*runlevel: default*", co oznacza, że jest to serwer nadrzędny ponieważ na serwerze2 usługi *MySQL* i *Apache2* nie są uruchomione. Istotną częścią rozwiązania jest poprawna konfiguracja i kontrola działania usług sieciowych. Informacje o sieci wyświetlane są po wpisaniu komendy: # `ip a s`. Usługa umożliwia uzyskanie adresów wszystkich interfejsów sieciowych. "*Eth0*" i "*Eth1*" oraz identyfikację ich podstawowych parametrów takich jak adresy rozgłoszeniowe czy MAC.

Wyświetlenie statusu urządzeń *DRBD* możliwe jest po wpisaniu komendy: # `cat /proc/drbd` Po uruchomieniu obu maszyn rozpoczyna się proces synchronizacji danych z obu serwerów.

```
serwer1   # cat /proc/drbd
version: 8.3.11 (api:88/proto:86-96)
built-in
 0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
    ns:9 nr:19 dw:25 dr:678 al:3 bm:4 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
 1: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
    ns:384 nr:416 dw:808 dr:4601 al:6 bm:7 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

Rys. 2. Status urządzenia DRBD w ***serwerze1***

Z danych przedstawionych na rys. 2 można zaobserwować, że ***serwer1*** jest serwerem nadrzędnym i został on poprawnie zsynchronizowany z ***serwerem2***, a z danych przedstawionych na rys. 3, że ustanowione jest połączenie ***serwera2*** podrzędnego z ***serwerem1***.

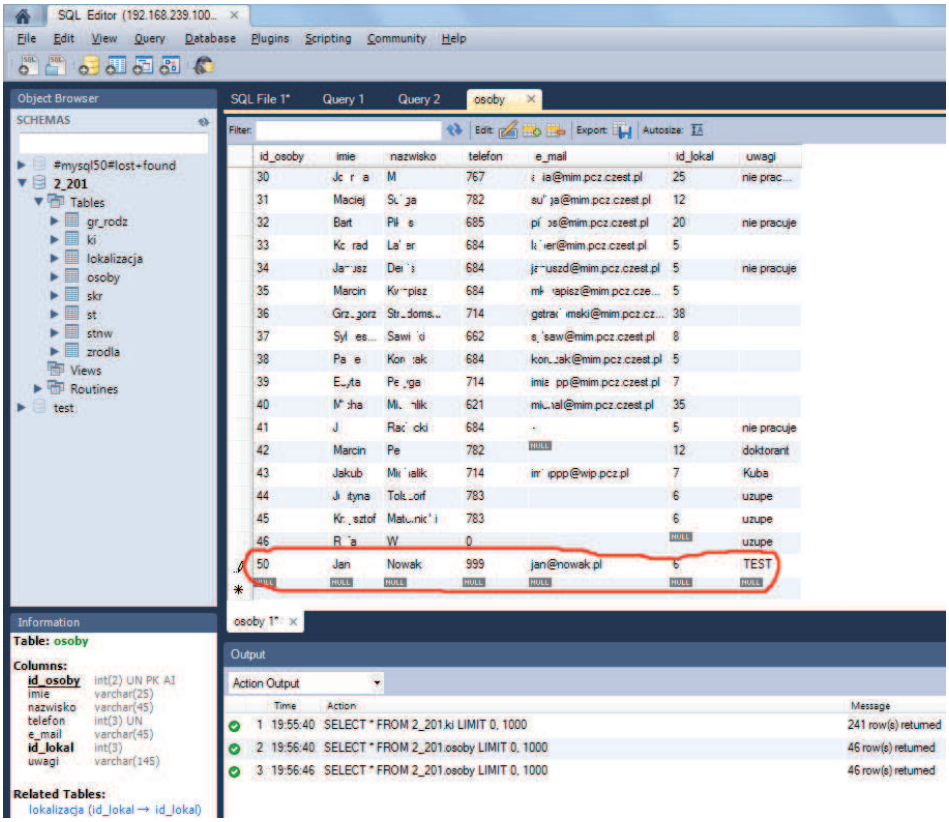
```
serwer2   # cat /proc/drbd
version: 8.3.11 (api:88/proto:86-96)
built-in
 0: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r-----
    ns:0 nr:9 dw:9 dr:0 al:0 bm:1 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
 1: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r-----
    ns:0 nr:384 dw:384 dr:0 al:0 bm:4 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

Rys. 3. Status urządzenia DRBD w ***serwerze2***

Synchronizacja danych następuje również w momencie wystąpienia awarii oraz wtedy, gdy naprawiony węzeł powraca do klastra. Węzły wykrywają się nawzajem i węzeł, który do tej pory odpowiedzialny był za zapewnianie dostępu do usług pozostanie, jako kontynuujący swoją pracę host, który dołączy do działającego klastra i stanie się zasobem funkcjonalnie niepotrzebnym do czasu wystąpienia kolejnej awarii. Po wykonaniu synchronizacji, system *DRBD* będzie gotowy do działania.

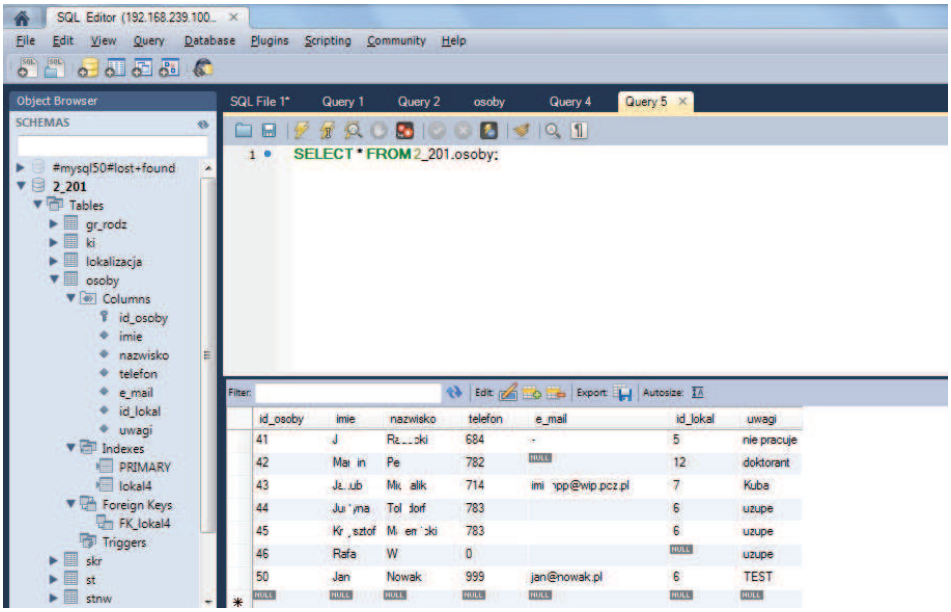
Modelowanie awarii jednego z serwerów

Przed awarią systemu do bazy danych zostały dodane nowe dane przestawione na rys. 4.



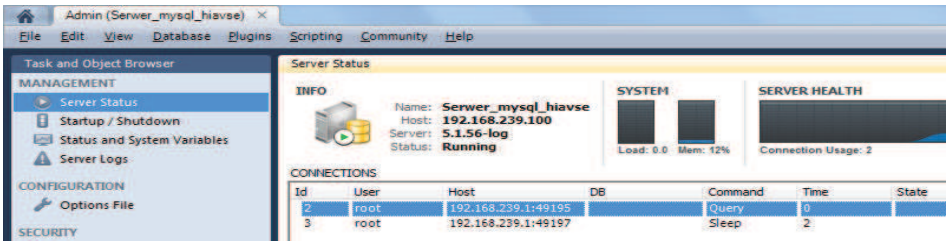
Rys. 4. Uzupełniona o dodatkowe dane testowe baza danych - kartoteka ilościowa

Poprzez wpisanie komendy # poweroff zamodelowano awarię serwera nadrzędnego. Po sprawdzeniu statusu ***serwera2*** można stwierdzić, że nastąpiło przejęcie funkcji serwera nadrzędnego, na którym została uruchomiona usługa MySQL oraz Apache, które pozwalają na dalsze zarządzanie spójną bazą oraz jej konfigurację. Na serwerze tym nie trzeba przeprowadzać tworzenia systemu plików. Proces replikacji podczas operacji synchronizacji przenosi wszystkie dokonywane dotychczas zmiany w głównym węźle (***serwer1***) na węzeł zapasowy (***serwer2***), co zostało przedstawione na rys. 5.



Rys. 5. Aktualna baza danych po awarii

Wszystkie aktualnie działające usługi na poziome systemu plików zostały przeniesione na serwer podrzędny i baza danych MySQL mogła być użytkowana bez ograniczeń (rys. 5). Usługa *Heartbeat* umożliwiła przełączenie wspólnego adresu IP pomiędzy serwerami klastra. Każdy z serwerów na interfejsie dostępnym dla klientów posiada zdefiniowany adres IP (np. 192.168.0.1 i 192.168.0.2). Serwer, który aktualnie świadczy usługi (***serwer1*** - nadrzędny) dodatkowo na tym samym interfejsie ma zdefiniowany drugi adres IP (192.168.239.100), z którym łączą się klienci. W sytuacji, gdy ***serwer1*** uległ awarii, usługa Heartbeat umożliwiła przypisanie do interfejsu sieciowego ***serwera2*** drugiego adresu IP (192.168.239.100), na który zgłaszają się klienci (rys. 6) oraz uruchomienie niezbędnych usług sieciowych i serwera baz danych.



Rys. 6. Ustanowione połączenia klientów do serwera baz danych po awarii

Podsumowanie i wnioski

Na podstawie przeprowadzonych badań można sformułować następujące stwierdzenia i wnioski:

- Stworzono klaster wysokiej dostępności oraz dokonano replikacji przygotowanej bazy danych z ***serwera1*** na ***serwer2*** w warunkach awarii serwera głównego.
- Analiza ujawniła, że podczas awarii wszystkie dane z analizowanej bazy zostały poprawnie przeniesione (replikowane) na serwer zapasowy.
- Dzięki wykorzystaniu mechanizmu DRBD, pracując na niskim poziomie systemu, możliwe jest wykorzystanie go dla różnego typu usług nawet tych, które nie posiadają wprost obsługi serwerów zapasowych.
- Przedstawione rozwiązanie wykorzystuje otwarte oprogramowanie w środowisku Linux, dzięki czemu może być łatwo i szybko wdrażane bez ponoszenia niepotrzebnych kosztów związanych z zakupem specjalistycznego oprogramowania.
- Zbudowany przy pomocy DRBD system umożliwia zwiększenie bezpieczeństwa danych, może być stosowany jako jedno z narzędzi podnoszących niezawodność i dostępność platformy klastrowej, aczkolwiek w tego typu rozwiązaniach zalecane jest również korzystanie z redundancji na poziomie bazy danych.

Literatura

[1] Dudek W., *Bazy Danych MySQL. Teoria i praktyka*, Wydawnictwo Helion, 2006.
[2] Rochkind M.J., *Programowanie w systemie Unix dla zaawansowanych*, WNT, Warszawa, 2007.
[3] http://artemis.wszib.edu.pl/~rgorecki/swn.stara_strona/hb_appendix/node2.html (data dostępu: 15.05.2016).
[4] http://jakilinux.org/linux/debian/wirtualizacja-czyli-serwerownia-modnie-i-wygodnie/ (data dostępu: 15.05.2012).
[5] https://www.drbd.org/en/doc/users-guide-83/ch-heartbeat (data dostępu: 01.06.2016).
[6] http://www.drbd.org/en/doc/users-guide-90/about (data dostępu: 02.06.2016).
[7] https://www.gentoo.org/get-started/about/ (data dostępu: 02.06.2016).