

Projektowanie systemów rozproszonych

ĆWICZENIE Równoważenie obciążeń

Dariusz Dwornikowski
9 kwietnia 2011

1. Wstęp

Celem ćwiczenia jest zapoznanie się z koncepcją równoważenia obciążeń serwerów WWW. Środowiskiem dla ćwiczeń będzie system operacyjny GNU/Linux, serwer WWW Nginx oraz aplikacje WWW.

2. Przygotowanie środowiska

1. Włącz system operacyjny GNU/Linux.
2. W katalogu domowym stwórz katalog **LB** oraz wejdź do niego.
3. Ze strony <http://www.cs.put.poznan.pl/ddwornikowski/wsnhid/psr/> ściągnij pliki: **nginx.tar.gz** oraz **webapp.jar**.
4. Rozpakuj plik **nginx.tar.gz** komendą **tar xvf nginx.tar.gz**. W efekcie stworzony zostanie katalog **LB/nginx**. Wejdź do niego i wykonaj następujące polecenia:
 - (a) **./configure --prefix=\$HOME/LB --without-http_rewrite_module --without-http_cache**
 - (b) **make**
 - (c) **make install**
5. Jeśli wszystko przebiegło pomyślnie, w katalogu **LB/** powinny pojawić się katalogi **conf/** oraz **sbin/**.

3. Ćwiczenie

3.1. Scenariusz pierwszy

Scenariusz pierwszy obejmuje konfigurację serwera równoważenia obciążeń nginx oraz jednej instancji serwera aplikacyjnego. Serwer aplikacji uruchamiany jest poprzez wywołanie polecenia **java -jar webapp.jar** z dwoma argumentami: **port** oraz **nazwa**. Przykładowo: **java -jar webapp.jar 8080 Aplikacja1** uruchomi serwer na porcie **8080 localhost** oraz przydzieli nazwę **Aplikacja1**, widoczną po wejściu przeglądarką na adres **http://localhost:8080/**. Pozwoli to rozróżnić aplikacje w sytuacji, gdy uruchomionych będzie więcej serwerów aplikacyjnych.

1. Wejdź do katalogu **conf/**, otwórz plik **nginx.conf** do edycji.
2. Znajdź sekcję **server { ... }** i zmień port z 80 na 8000.

```
server {  
    listen 8000;  
    server_name localhost;  
    ...  
}
```

3. Nad sekcją **server { ... }** umieść następujący wpis definiujący serwery aplikacji, na które kierowane będą połączenia. Scenariusz pierwszy wykorzystuje tylko jeden serwer aplikacji.

```
upstream lb {  
    server localhost:8080;  
}
```

4. W sekcji `server { ... }` znajdź sekcję `location { ... }` i zmień ją odpowiednio jak w przykładzie poniżej, aby zdefiniować, że wszystkie żądania na `/` będą kierowane do `lb`, zdefiniowanego wcześniej.

```
location / {  
    proxy_pass http://lb;  
}
```

5. Uruchom serwer aplikacji na odpowiednim porcie: `java -jar webapp.jar 8080 Aplikacja1`, wyłączyć można go kombinacją klawiszy: `CTRL+C`. Uruchom serwer nginx z katalogu `sbin/` poleceniem `./nginx`. Serwer można wyłączyć wykonując komendę `./nginx` z parametrem `-s stop`.
6. Sprawdź w przeglądarce działanie systemu, kierując się na adres: `http://localhost:8080`.

3.2. Scenariusz drugi

Scenariusz drugi przewiduje uruchomienie dwóch instancji serwerów aplikacji oraz równoważenie ich obciążeń za pomocą serwera nginx.

1. Otwórz plik konfiguracyjny `nginx.conf`.
2. W sekcji `upstream { ... }` dodaj kolejny serwer aplikacji.

```
upstream lb {  
    server localhost:8080;  
    server localhost:8081;  
}
```

3. Zapisz plik konfiguracyjny i przeładuj serwer nginx.
4. W osobnym terminalu włącz dodatkową instancję serwera aplikacji na porcie `8081` z nazwą `Aplikacja2`.
5. Sprawdź w przeglądarce działanie środowiska. Jaki jest algorytm równoważenia obciążeń?
6. Wyłącz jeden z serwerów i sprawdź co się dzieje.

3.3. Scenariusz trzeci

Scenariusz trzeci dotyczy równoważenia obciążeń w przypadku, gdy serwery aplikacyjne nie są jednakowej wydajności. Zakładamy, że serwer pierwszy jest mocniejszy od drugiego, zatem większość żądań powinna zostać kierowana na niego. W serwerze nginx takim zachowaniem można sterować dzięki parametrowi `weight`.

```
upstream lb {  
    server localhost:8080 weight=3;  
    server localhost:8081;  
}
```

Jak teraz wygląda algorytm kierowania żądań?

3.4. Scenariusz czwarty

Dodaj kolejne serwer aplikacyjny do serwera nginx oraz uruchom go na odpowiednich portach z nazwami **Aplikacja1**, **Aplikacja2**, **Aplikacja3**. Pamiętaj, że porty nie mogą ze sobą konfliktować.

```
upstream lb {  
    server localhost:8080;  
    server localhost:8081;  
    server localhost:8082;  
}
```

Dodaj dyrektywę **ip_hash** i przeładuj serwer. Jak zmienił się sposób dystrybucji żądań ?

```
upstream lb {  
    ip_hash;  
    server localhost:8080;  
    server localhost:8081;  
    server localhost:8082;  
}
```

Zmień sekcję **upstream { ... }** w następujący sposób. Jak rozdzielane są żądania w przypadku, gdy działają wszystkie serwery oraz w przypadku, gdy działa jedynie trzeci ?

```
upstream lb {  
    server localhost:8080;  
    server localhost:8081;  
    server localhost:8082 backup;  
}
```