

[기술자문 컨설팅]  
OSDL DBT-1

한국소프트웨어진흥원  
공개SW기술지원센터

<Revision 정보>

일자	VERSION	변경내역	작성자
2007.03.28	0.1	초기 작성	손승일

## 목 차

1. 문서 개요 .....	4
가. 문서의 목적 .....	4
나. 본 문서의 사용방법 .....	4
2. ODSL DBT-1 .....	5
가. 개요 .....	5
나. DBT-1 설치 .....	5
다. 환경 설정 .....	7
라. DBT-1을 이용한 측정 .....	11
3. 결론 .....	15
그림 1 .....	13

## 1. 문서 개요

본 문서는 공개 DBMS인 MySQL, PostgreSQL BMT 툴인 DBT-1 설치 및 사용법에 대해서 기술하며, DBT-1 사용으로 DBMS 성능을 확인하기 위한 기술 자문 컨설팅 참고자료로 활용을 위해 제작되었다.

이문서는 MySQL을 대상으로 작성되었다.

### 가. 문서의 목적

다음과 같은 세부적인 목적을 달성하기 위하여 작성되었다.

- DBT-1 소개
- DBT-1 설치 및 사용법
- 측정 결과 분석

### 나. 본 문서의 사용방법

다음과 같은 방법으로 사용할 수 있다.

- DBT-1의 설치 및 사용방법에 대해서 확인한다.
- DBT-1을 통한 측정 결과 분석방법을 확인한다.

## 2. ODSL DBT-1

### 가. 개요

OSDL DBT-1은 온라인 서점에서 물건을 검색 또는 구매하는 웹 사용자의 행동을 시뮬레이션하는 웹 트랜잭션 벤치마크 도구이다. DBT-1 시험은 성능 병목지점을 찾기 위해서 TPC-W 벤치마크 워크로드 특성을 사용하고 있다.

DBT-1은 eu(emulate user)수를 변경 설정하면서 반복수행하여 상호작용에 대한 평균 응답 시간, BT/sec을 측정하여 논리적인 값과 실제 측정값과의 비교를 통해 병목지점을 확인할 수 있다.

### 나. DBT-1 설치

우선 DBT-1을 수행하기 위해서는 MySQL과 MyODBC, iODBC가 미리 설치 되어있어야 한다.

#### 1) MySQL 설정

MySQL 설정 파일인 /etc/my.cnf에 아래 사항을 추가한다.

```
max_connections=120
```

#### 2) MyODBC 설정

ODBC를 통해서 MySQL에 연결하기 위해 user ID mysql로 로그인 한 후 /etc/odbc.ini 파일을 다음과 같이 추가한다.

```
[localhost:DBT1]
Driver = /usr/lib/libmyodbc3.so
Description = OSDL-DBT-1
Host = localhost
user = dbt
password = dbt
database = DBT1
server = localhost
port = 3306
socket = /var/lib/mysql/mysql.sock
```

#### 3) user 생성

root user로 로그인하고 dbt를 위한 user를 생성한다.

```
userID=dbt group=dbt password=dbt
```

```
# groupadd dbt
# useradd -g dbt -d /home/dbt -s /bin/bash dbt
# passwd dbt
Changing password for user dbt.
New UNIX password:
BAD PASSWORD: it's WAY too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

#### 4) sudo를 위한 설정

로그 정보 내에 root 권한을 요구하는 명령어들이 있으므로, /etc/sudoers 파일을 수정하여 DBT-1을 실행하는 user가 sudo 명령어를 사용할 수 있도록 한다.

root user로 /etc/sudoers에 다음을 추가한다. 추가 후에 쓰기권한을 삭제한다.

```
%dbt ALL=(ALL)NOPASSWD:ALL
```

#### 5) user 환경 설정

dbt user로 로그인하여 다음과 같은 내용들을 ~/.bashrc 파일에 추가한다.

```
export PATH=/usr/bin:$PATH
export LD_LIBRARY_PATH=/usr/lib:/usr/lib/mysql
export ODBCINI=/etc/odbc.ini
export SID1=DBT1
```

#### 6) ssh를 위한 설정

DBT-1은 원격 환경에서 시작할 수 있게 구성되어 있으며, 보안을 위해 ssh를 사용하므로 ssh-keygen과 함께 key를 생성해야 한다. dbt user로 로그인 하여 다음과 같은 작업을 한다.

```
$ssh-keygen -t rsa -N ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/dbt/.ssh/id_rsa):
Created directory '/home/dbt/.ssh'
Your identification has been saved in /home/dbt/.ssh/id_rsa.
Your public key has been saved in /home/dbt/.ssh/id_rsa.pub.
The key fingerprint is:
ac:16:14:89:01:ee:31:1c:da:1d:b3:5b:84:b8:2d:68 dbt@localhost.localdomain
$cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys
$chmod 600 ~/.ssh/authorized_keys
```

## 7) DBT-1 설치

아래 사이트로부터 MySQL(ODBC)를 위한 DBT-1 소스를 다운로드 받는다.

```
http://www.ipa.go.jp/software/open/forum/development/download/051115/dbt1-v2.1-MySQL-ODBC-1.0.tar.gz
```

dbt user로 로그인하여 user의 home 디렉토리에서 압축을 해제한다.

```
$tar xzvf dbt1-v2.1-MySQL-ODBC-1.0.tar.gz
```

DBT-1 컴파일

```
$cd ~/dbt1-v2.1  
$make
```

## 다. 환경 설정

### 1) 디렉토리 생성

dbt user로 로그인 하여 데이터를 위한 디렉토리를 생성한다.

```
$mkdir ~/data  
$chmod a+w ~/data
```

### 2) 데이터 생성

dbt user로 로그인 하여 다음과 같은 명령어로 시험을 위한 기본 데이터(text files)를 생성한다.

```
$cd ~/dbt1-v2.1/datagen  
$./datagen -d SAPDB -i 10000 -u 1000 -p /home/dbt/data -T i -T c -T  
a -T d -T o
```

\* 옵션들

- d : 데이터베이스 유형(SAPDB 또는 PGSQL)
- i : 생성된 item의 총 개수
- u : 생성된 가상 user의 수
- p : 데이터가 생성되는 데이터 디렉토리 (전체경로를 기술한다. 기술된 디렉토리에 데이터가 생성된다.)
- T : 생성된 데이터를 위한 테이블명
  - i -- item
  - c -- customer

```
a -- author  
d -- address  
o -- order
```

생성된 데이터 파일은 /tmp 디렉토리에 symbolic 링크로 만들어진다. /tmp 디렉토리에 있는 symbolic 링크는 데이터를 로딩할 때 사용된다.

### 3) build\_Db.sh 수정

dbt user로 로그인하여 데이터베이스를 생성하기 전에 생성 속도 향상을 위해 ~/dbt1-v2.1/scripts/mysql/build\_db.sh를 아래와 같이 수정한다.

```
$vi ~/dbt1-v2.1/scripts/mysql/build_db.sh  
  
date  
echo "create tables"  
./create_tables.sh  
echo  
  
date  
echo "load db"  
./load_db.sh  
echo  
  
date  
echo "create foreign key"  
./create_fk.sh  
echo  
  
date  
echo "create indexes"  
./create_indexes.sh  
echo
```

부분을 아래와 같이 순서를 변경한다.

```
date  
echo "create tables"  
./create_tables.sh  
echo  
  
date  
echo "create indexes"  
./create_indexes.sh  
echo
```

```
date
echo "create foreign key"
./create_fk.sh
echo

date
echo "load db"
./load_db.sh
echo
```

~/dbt1-v2.1/scripts/mysql/load\_db.sql 스크립트의 위 부분에

```
SET FOREIGN_KEY_CHECKS=0;
```

아래 부분에

```
SET FOREIGN_KEY_CHECKS=1;
```

추가한다.

#### 4) 데이터베이스 생성

MySQL 서버가 실행된 중에 아래 명령을 수행하여 DBT1 데이터베이스를 생성한다.(수행시간이 2 ~ 4시간 걸림)

```
$cd ~/dbt1-v2.1/scripts/mysql
$./build_db.sh
```

반복하여 수행하여야 하므로 생성을 완료한 후에 /var/lib/mysql 디렉토리를 압축하여 백업해 둔다.

#### 5) 파라미터 구성

DBT-1 구성 파일(~/dbt1-v2.1/scripts/stats/dbt1.config)을 수정한다.

- localhost를 호스트명으로 수정한다.
- 실제로 측정되는 가상 user 수에 대한 eu(emulate user의 총 수)와 run\_duration in seconds 부분만 수정하여 측정한다.

```
[database]
#hostname instance username password
localhost:DBT1:dbt:dbt

[cache]
#hostname port dbconnections items appCache_executable_directory
localhost:9999:5:10000:/home/dbt/dbt1-v2.1/cache

[appServer]
#hostname      - server name executing appServer
localhost
#port          - access port number to appServer
9992
#dbconnection  - DBMS connection number
100
#transaction_queue_size - dbdriver communication buffer
1000
#transaction_array_size - each processing buffer
1000
#items         - generating item number
10000
#appServer executable directory -appServer executable directory
/home/dbt/dbt1-v2.1/appServer

[dbdriver]
#hostname      - host name executing dbdriver
localhost
#items         - generation item number
10000
#customers     - generation customer number
2880000
#eu            - total emulate user number
1000
#eu/min        - user access rate(user/min)
100
#mean think_time - user operation interval
7.2
#run_duration in seconds - execution time of each user
4200
```

```
#dbdriver executable directory - dbdriver executable directory
/home/dbt/dbt1-v2.1/dbdriver
```

## 6) 파라미터 구성의 주의점

DBT-1이 만들어 졌을 때는 하드웨어, OS와 DBMS의 각 환경이 그렇게 높은 성능을 가진다고 가정되지 않았다. 그래서 실제 구동 환경에서는 몇 가지 제약이 있으며 수행 환경 설정 시에 주의가 필요하다.

- customers

데이터 생성시에 user 수에 2880을 곱한 값을 설정한다.

- eu

eu에 설정된 수만큼 쓰레드가 생성된다. eu는 4000 까지 가능하며 더 많은 로드를 필요로 한다면 mean think\_time을 변경함으로써 가능해 진다.

- eu/min

사용자 접속율(user/min). RampUpTime에 설정된 값만큼 연결을 시도한다. 기본값은 100이다.

예) eu=4000, eu/time=100인 경우에, RampUpTime은 40분이 걸린다고 계산되는데 이 경우, 측정시간을 충분히 가질 수 없기 때문에 값을 증가시켜야 한다. RampUpTime 중에 접속에러가 발생한다면 값을 감소시킨다.

- 평균 think\_time

사용자 클릭 동작 간격을 설정하는 것으로 기본 값은 7.2초이다. 하드웨어 성능이 좋을 경우 eu값이 4000을 넘을 경우가 있다면, think\_time을 3.6초 또는 1.8초로 설정한다.

think\_time이 3.6 또는 1.8초까지 설정되었다면, eu는 두 배 또는 네 배가되는 것과 같다.

- run\_duration in seconds

각 user의 실행 시간, 기본 값이 특별히 지정되지 않았더라도, RampUpTime을 고려하고 결정한다.

예) eu=2000, eu/min=100인 경우  $2000/100 + 10 * 60 = 1800$ 초

## 라. DBT-1을 이용한 측정

### 1) 데이터베이스 시동

MySQL 서버를 시작한다.

### 2) DBT-1 실행

다음 명령어로 DBT-1을 실행한다.

```
$cd ~/dbt1-v2.1/scripts/stats
$./run_dbt1.sh/home/dbt/U1000
```

run\_dbt1.sh의 argument로 미리 만들어진 디렉토리를 만들 필요는 없다.

실행 중 오류가 발생하면 ~/dbt1-v2.1/appServer/error.log를 확인한다.

실행 결과는 argument로 입력한 디렉토리(/home/dbt/U1000)에 BT라는 파일로 생성되며, 마지막 라인의 "total errors"가 0이면, 측정이 성공했음을 의미한다.

BT 파일의 예는 다음과 같다.

Interaction	%%	Avg. Response Time (s)
Admin Confirm	0.10	0.109
Admin Request	0.11	0.061
Best Sellers	4.98	0.075
Buy Confirm	1.22	0.121
Buy Request	2.59	0.145
Customer Registration	2.95	0.000
Home	16.70	0.073
New Products	5.02	0.070
Order Display	0.66	0.138
Order Inquiry	0.74	0.055
Product Detail	16.87	0.062
Search Request	19.80	0.000
Search Results	16.80	0.078
Shopping Cart	11.44	0.084
206.4 bogotransactions per second		
10.1 minute duration		
total bogotransactions 125096		
total errors 0		

### 3) 결과 분석

측정한 결과 중 bogotransactions per second 값이 DBT-1 구성 파일 (~/dbt1-v2.1/scripts/stats/dbt1.config)에 설정된 eu 값을 think\_time 값으로 나눈 결과와 유사한 경우 현재의 시스템에서 충분히 처리 가능 하다는 것을 나타낸다.

한 번의 측정으로 얻어진 결과는 설정한 eu에 해당하는 one point이고, 연속적으로 eu 수를 변화시켜 측정한 bogotransactions per second 값이 eu 값을 think\_time 값으로 나눈 결과와 많은 차이가 발생하는 지점이 확인되는데 이때 해당하는 eu가 병목지점에 해당한다.

측정 결과를 차트로 나타내면 아래와 같이 측정치 BT/s가 꺾이는 부분이 병목지점에 해당한다.

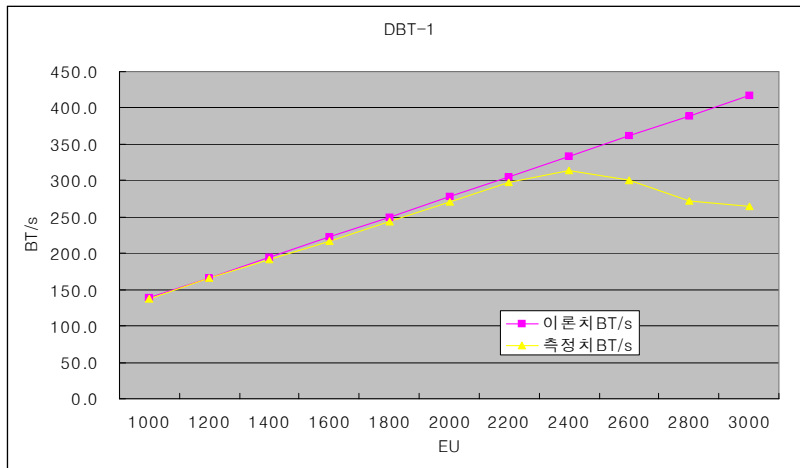


그림 1. DBT-1 측정결과

#### 4) 연속 수행을 위한 script

DBT-1은 병목지점을 확인하기 위해서 eu값을 변경하며 반복수행을 하여야 하는데 반복시마다 기존에 수행한 프로세스를 죽이고, DB를 백업하고, MySQL을 restart 시키는 작업들을 수행하여야 하는데 이를 간편하게 수행하기 위해 아래와 같은 script를 사용한다.

```
#!/bin/sh
killps() {
    echo "kill $1"
    for procs in $(ps ax | grep $1 | grep -v grep | awk '{print $1}')
    do
        sudo kill -KILL $procs
    done
}
allkiller() {
    sleep 10
    killps appCache
    killps appServer
    killps dbdriver
    killps db_stats
    sleep 120
}
```

```
reset_mysql() {
#   sudo /usr/local/mysql/bin/mysqladmin shutdown
    sudo /sbin/service mysqld stop
    sleep 120
    cd /var/lib
    sudo tar -xvf mysql.tar
    sleep 120
    cd /home/dbt/dbt1-v2.1/scripts/stats
}
rundbt () {
    sudo /sbin/service mysqld start
    sleep 120
    cp $1 dbt1.config
    ./run_dbt1.sh $2
    sleep 10
    mv run.sar.* $2
    cp /tmp/*.log $2
    allkiller
    reset_mysql
}

cd /home/dbt/dbt1-v2.1/scripts/stats
allkiller
rundbt T900 /home/dbt/T900
allkiller

rundbt T1100 /home/dbt/T1100
allkiller

rundbt T1300 /home/dbt/T1300
allkiller

rundbt T1500 /home/dbt/T1500
allkiller
```

이 script를 사용하기 위해서 DBT-1 구성 파일 (~dbt1-v2.1/scripts/stats/dbt1.config)을 변경된 eu값이 적용된 파일들로 새로 생성해야 한다. 예를 들면 eu 값이 900이면 변경한 파일을 T900 정도로 생성하고, 반복 수행할 대상만큼 파일을 생성한 후 위 script의 rundbt T900 /home/dbt/T900 부분들을 해당하는 파일명으로 변경하고, 생성된 파일 수만큼 rundbt ... 구문을 적용한다.

### 3. 결론

OSDL DBT-1의 설치 및 사용방법 그리고 측정결과를 통한 병목지점 확인 방법에 대해서 알아보았다.

DBT-1을 통해 현 DBMS 시스템의 성능을 진단해보고, 요구 수준에 합당한 성능을 실현하기 위한 DBMS 파라미터 튜닝 및 하드웨어 증설을 결정하는 데에 DBT-1을 사용할 수 있다.