Benchmarking and monitoring tools

Section one:
Benchmarking

Presented by,
MySQL & O'Reilly Media, Inc.

Benchmarking tools and the like

- mysqlslap
- sql-bench
- supersmack
- Apache Bench (combined with some sample PHP scripts)
- MySQL's benchmark() function
- MyBench
- WAST
- JMeter

MySQL Conference & Expo

Presented by,
MySQL & O'Reilly Media, Inc.

sql-bench

WHAT IS IT ?

SQL-BENCH is a collection of test scripts written in Perl that are included in a MySQL install. In order to run them you just execute the run-all-tests script in the mysql directory

In addition to could run individual test.

SQL-BENCH

- PROS

  Ubiquitous

  long history of use
- CONS

  Single thread

  Perl

  not always real-life test cases (create 10,000 tables ?)

SQL-BENCH

- More Details...

script	Aim
test-alter-table	Test of alter table
test-big-tables	Test of extreme tables
test-create	This test is for testing how long it takes to create tables, make a count(*) from them and finally drop the tables.
test-select	Test of selecting on keys that consist of many parts
test-ATIS	Creation of 29 tables and a lot of selects on them.
test-connect	This test is for testing the speed of connections and sending data to the client.
test-insert	Test of creating a simple table and inserting $record_count records in it, $opt_loop_count rows in order,$opt_loop_count rows in reverse order and $opt_loop_count rows in random order. (opt_loop_count is 100000)
test-wisconsin	This is a port of the PostgreSQL version of this benchmark (create and insert table from Wisconsin directory)

SQL-BENCH

- ISSUES
  - In 5.1 it may be source distro's only.
  - Perl troubles in windows
- URL

  http://dev.mysql.com/doc/refman/5.0/en/mysql-benchmarks.html

SuperSmack

- Details

  Originally developed by Sasha Pachev, Modified extensively and promoted by Jeremy Zawodny in his famous book, “High Performance MySQL”MySQL

  WHAT IS IT ?

  Super Smack is a benchmarking, stress testing, and load generation tool for MySQL. Like the apache bench (ab) tool that ships with Apache, super smack helps to give you a handle on how well your server will perform.

  Compiled easily on my mac so should be portable.

  Flexible

SuperSmack

- Usage example..

```
super-smack -d mysql select-key.smack 50 1000
sudo /usr/local/bin/super-smack -d mysql select-key.smack 50 1000
Password:
Query Barrel Report for client smacker1
connect: max=51ms  min=33ms avg= 40ms from 50 clients
Query_type      num_queries     max_time        min_time        q_per_s
select_index    100000  10      0       5505.19

   Time: 0
  State: statistics
   Info: select * from http_auth where username = 'omthglrulsxi82440'
*************************** 5. row ***************************
```

SuperSmack

- mysql> show full processlist \G

```
     Id: 575
   User: test
   Host: localhost
     db: test
Command: Query
   Time: 0
  State: statistics
   Info: select * from http_auth where username = 'tkmpjihfegrq77160'
*************************** 3. row ***************************
     Id: 576
   User: test
   Host: localhost
     db: test
Command: Query
   Time: 0
  State: statistics
   Info: select * from http_auth where username = 'cmcswrhugbsn1200'
*************************** 4. row ***************************
```

SuperSmack

- PROS

  Launches concurrent connections so we are testing something more realistic than one thread creating a bazilliion queries
- CONS

  ??

SuperSmack

- Discussion

  SuperSmack is well documented and a rather flexible benchmarking tool.. check it out.

Apache Bench

- Details

  Apache Bench is a webserver benchmark tool that is provided with the apache webserver. Some RPM based installs of apache may not have it, if it is unavailable a quick source compile of apache will give you an executable to use.

Apache Bench

- Good News

  Benchmarks your whole web application, if you have a web application
- Bad News

  Benchmarks your whole web application, if you have a web application.

Apache Bench

- **URL**
  http://httpd.apache.org/docs/2.2/programs/ab.html

Apache Bench

- Example

  ab -n 100 -c 10 http://localhost/index.html

  Benchmarking localhost (be patient).....done
  Server Software: Apache/1.3.33
  Server Hostname: localhost
  Server Port: 80
  Document Path: /index.html
  Document Length: 1456 bytes

AB example continued..

Concurrency Level: 10
Time taken for tests: 0.662 seconds
Complete requests: 100
Failed requests: 0
Broken pipe errors: 0
Total transferred: 188000 bytes
HTML transferred: 145600 bytes
Requests per second: 151.06 [#/sec] (mean)
Time per request: 66.20 [ms] (mean)
Time per request: 6.62 [ms] (mean, across all concurrent requests)
Transfer rate: 283.99 [Kbytes/sec] received

AB example continued

Percentage of the requests served within a certain time (ms)

50%	15
66%	15
75%	15
80%	15
90%	16
95%	477
98%	483
99%	485
100%	485 (last request)

BENCHMARK()

- MySQL has a built in function, benchmark()
- Typically used to benchmark the time it takes to execute a function , with the use of built in functions we can expand the use of benchmark()

Benchmark()

- Examples

```
mysql> select benchmark(10000, sha1(md5(rand())))\G
*************************** 1. row ***************************
benchmark(10000, sha1(md5(rand()))): 0
1 row in set (0.08 sec)
```

Benchmark()

- Usage

  benchmark(Number of times to run, function_name)\G
- URL

  http://dev.mysql.com/doc/refman/5.0/en/information-functions.html#function_benchmark

Benchmark()

- PROS

  Very Simple to use and test some basic possible bottlenecks. Is “x” a bottleneck ? If it is a function then benchmark() it.
- CONS

  Only benchmarks functions

Benchmark()

- Creative Uses

  You can create your own functions that make database calls and test those using benchmark.

  This is not without risks, meaning you might benchmark something besides your target. Regardless here is a quick example

Benchmark()

- Create a function..

```
CREATE function `rand_city`()
returns int
BEGIN
select floor(rand() * 4000), id into @rand, @id from city
where id=@id limit 1;
return @id;
END $$
```

Benchmark()

- Benchmark the function

```
mysql> select benchmark(1000, rand_city()) \G
*************************** 1. row
   ***************************
benchmark(1000, rand_city()): 0
1 row in set (1.93 sec)
```

Benchmark()

- Benchmark recap
  - Always available
  - Easy to use
  - Simple, perhaps too simple.
  - “creative” benchmarks using your own functions might not emulate real Database access.
  - Single thread.

myBench

- MyBench

  MyBench is a very simple benchmarking tool.

  written in PHP

  Runs a simple benchmark..

myBench

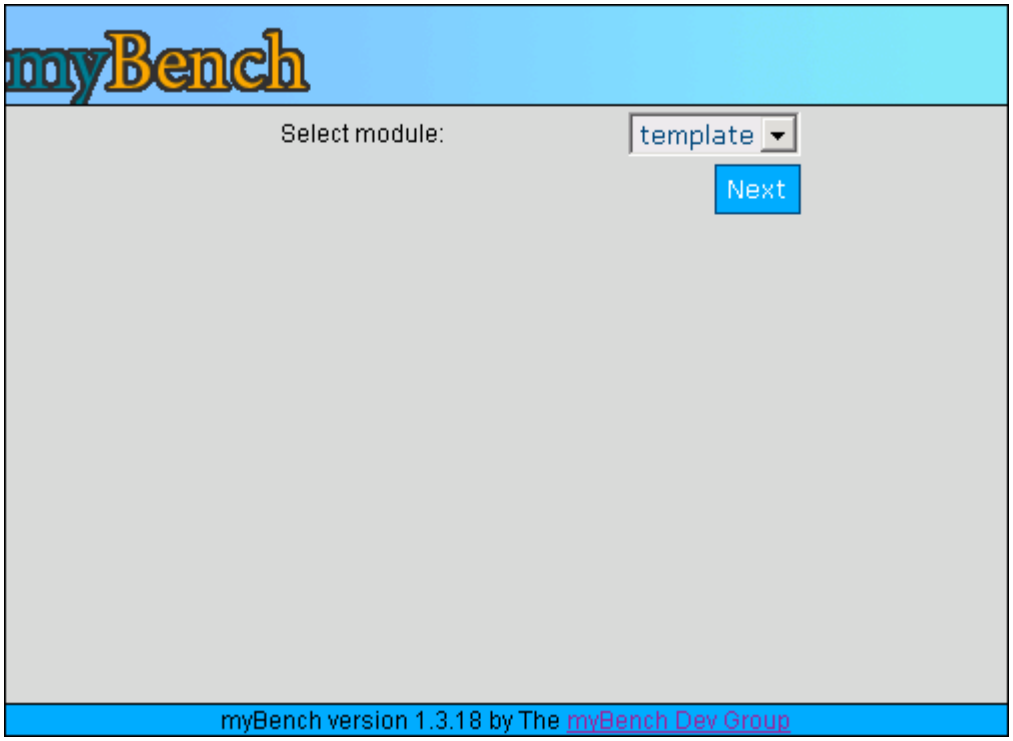
- Is it useful ??

  Somewhat, it is very simple. Good starting point to write your own.

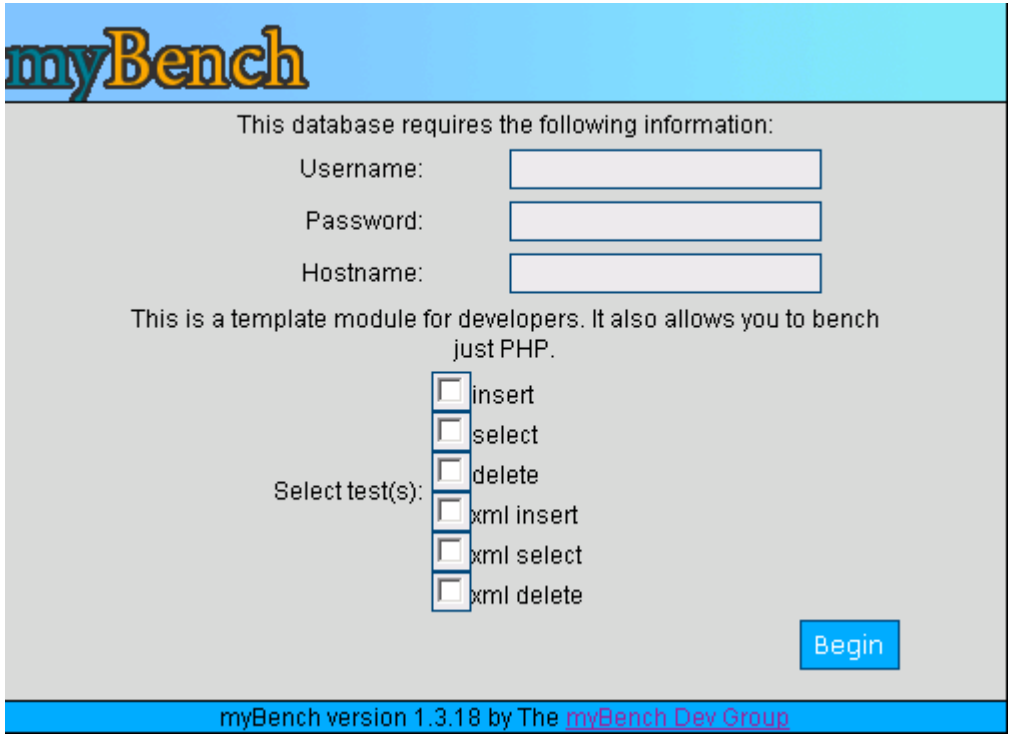
myBench

- Url
  http://mybench.sourceforge.net/

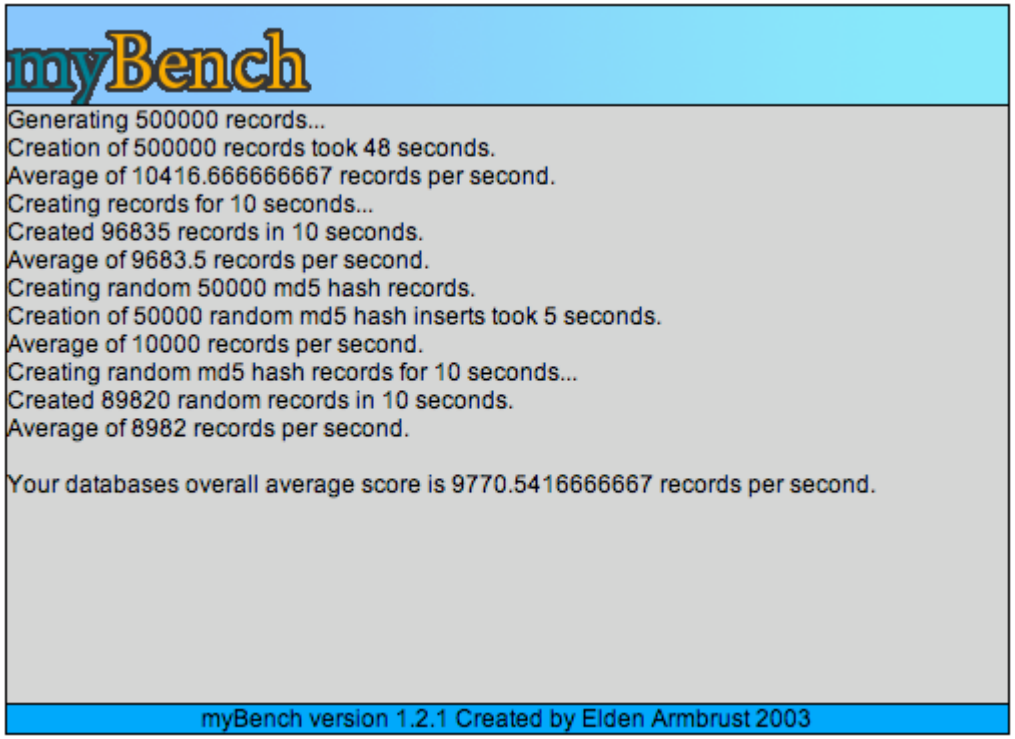
myBench


myBench
Select module:
template
Next
myBench version 1.3.18 by The myBench Dev Group

myBench


myBench
This database requires the following information:
Username:
Password:
Hostname:
This is a template module for developers. It also allows you to bench just PHP.
Select test(s):
insert
select
delete
xml insert
xml select
xml delete
Begin
myBench version 1.3.18 by The myBench Dev Group

myBENCH


myBench
Generating 500000 records...
Creation of 500000 records took 48 seconds.
Average of 10416.666666667 records per second.
Creating records for 10 seconds...
Created 96835 records in 10 seconds.
Average of 9683.5 records per second.
Creating random 50000 md5 hash records.
Creation of 50000 random md5 hash inserts took 5 seconds.
Average of 10000 records per second.
Creating random md5 hash records for 10 seconds...
Created 89820 random records in 10 seconds.
Average of 8982 records per second.
Your databases overall average score is 9770.5416666667 records per second.
myBench version 1.2.1 Created by Elden Armbrust 2003

WAST

- Web Application Stress Tool
  Developed by Microsoft
  Free Download
  Windows only

WAST

- Pros

  Flexible, more advanced in some ways than Apache Bench.

  Easy to use GUI (duh)

  Replay example click trail from browsing, build your own multiple sessions.

WAST

- CONS

  Windows only, surely you have access to a windows machine though.

WAST

- URL

  Google WAST and you will find the download

  Good tutorial..

  http://west-wind.com/presentations/webstress/webstress.htm

WAST

- Recap

  I have very little personal experience with WAST but I really like what I saw. It seems to be a well thought out well configured and easy to use tool.

Jmeter

- Java application developed for testing web applications.
- URL

  http://jakarta.apache.org/jmeter/
- Docs

  http://jakarta.apache.org/jmeter/usermanual/index.html

  The documentation for Jmeter is worth a read regardless of if you decide to use the tool. You will find some good information there.

Jmeter

- Jmeter is extremely configurable, some would say it is perhaps too complex. The documentation is good however and jmeter is certainly flexible and configurable enough to make a good benchmarking tool and even perhaps cross into the monitoring section of this talk.

Jmeter

- Configuration

  Benchmarking an application in Jmeter begins by setting up a test plan.

  Once we have the test plan which is little more than a name and some comments we add the “thread group”

  The “thread group” is where we configure the concurrency. We select the number of concurrent connections and define the iteration of the test

Jmeter

- Configuration continued

  Once we have configured the concurrency and the looping we add the http request. We can specify controls for cookie settings, post variables, authentication. You have plenty of control and flexibility in Jmeter.

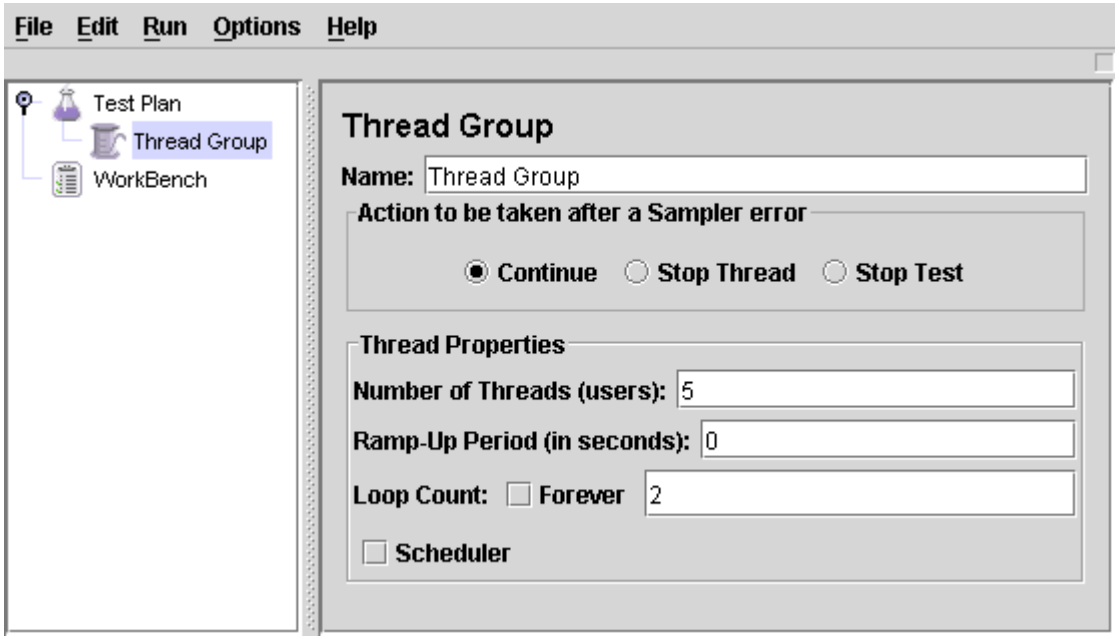
Jmeter

- Viewing the Results

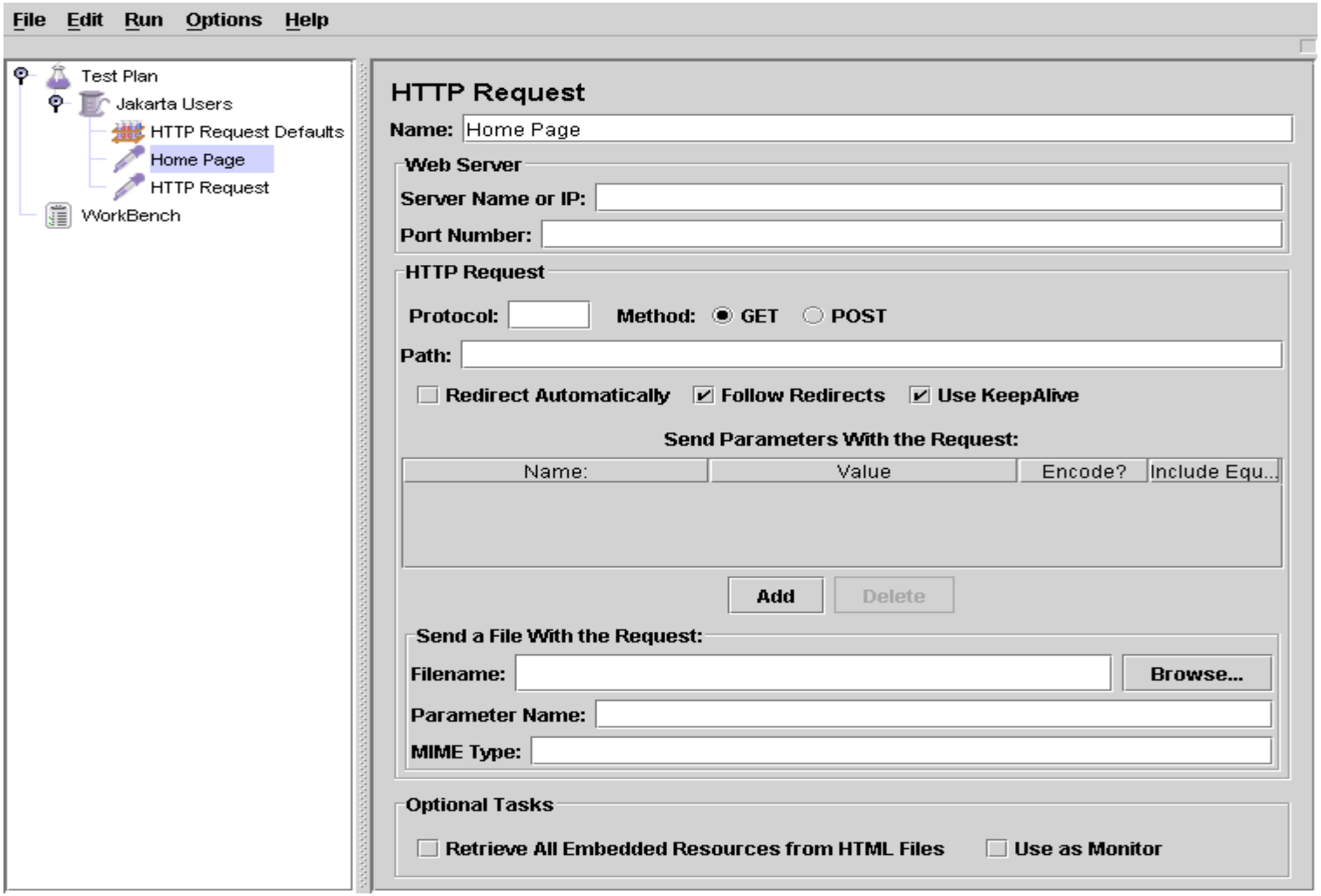
  Displaying the results from the Jmeter test involves adding a “Listener”

  The most straightforward “listener” is a table. but monitors, graphs and other options are available.

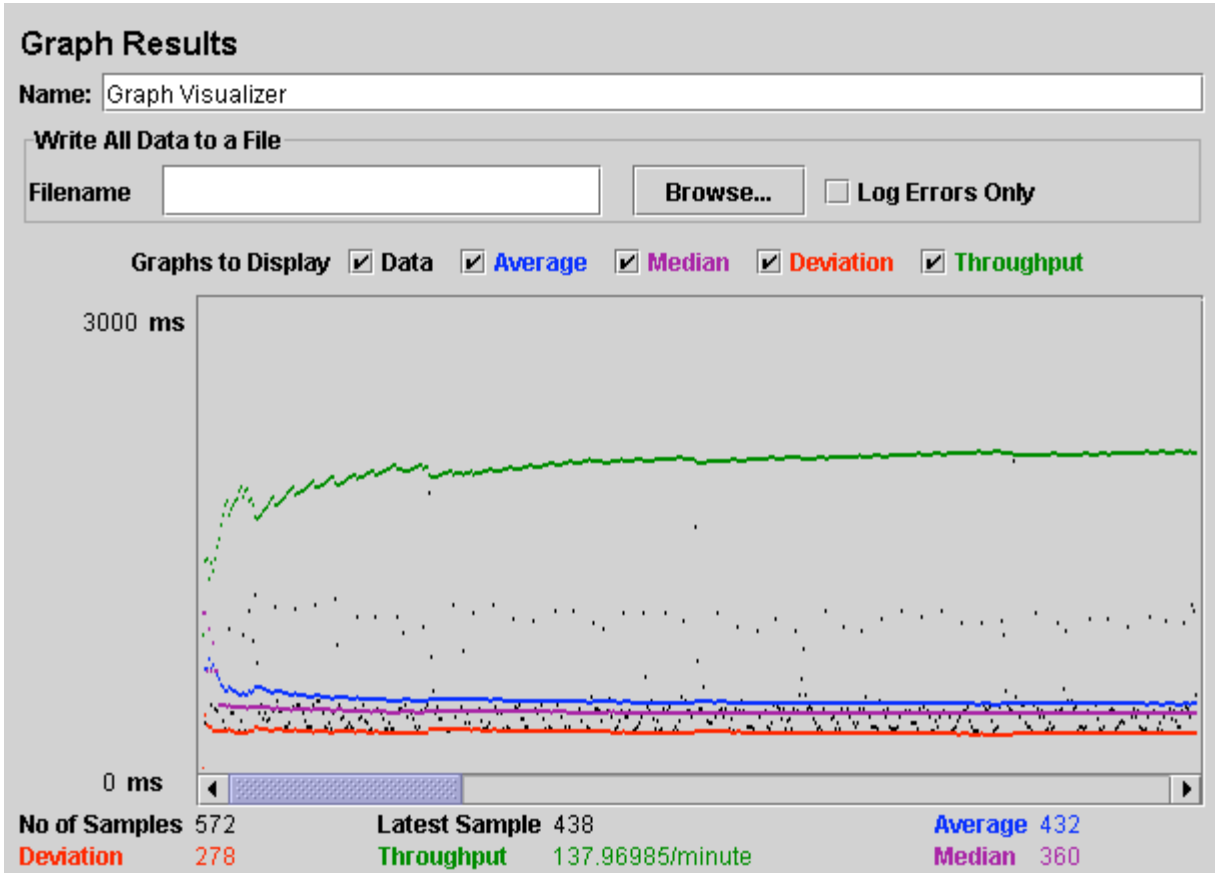
Jmeter


File Edit Run Options Help
Test Plan
Thread Group
WorkBench
Thread Group
Name: Thread Group
Action to be taken after a Sampler error
Continue
Stop Thread
Stop Test
Thread Properties
Number of Threads (users): 5
Ramp-Up Period (in seconds): 0
Loop Count: Forever 2
Scheduler

Jmeter


File Edit Run Options Help
Test Plan
Jakarta Users
HTTP Request Defaults
Home Page
HTTP Request
WorkBench
HTTP Request
Name: Home Page
Web Server
Server Name or IP:
Port Number:
HTTP Request
Protocol:
Method: GET POST
Path:
Redirect Automatically
Follow Redirects
Use KeepAlive
Send Parameters With the Request:
Name:
Value
Encode?
Include Equ...
Add
Delete
Send a File With the Request:
Filename:
Browse...
Parameter Name:
MIME Type:
Optional Tasks
Retrieve All Embedded Resources from HTML Files
Use as Monitor

Jmeter


Graph Results
Name: Graph Visualizer
Write All Data to a File
Filename
Browse...
Log Errors Only
Graphs to Display
Data
Average
Median
Deviation
Throughput
3000 ms
0 ms
No of Samples 572
Latest Sample 438
Average 432
Deviation 278
Throughput 137.96985/minute
Median 360

Jmeter

- Review
  Java
  A little bit complex
  Good docs
  Nice graphs
  Flexible

mysqlslap

- MySQL developed benchmark tool

  Shipping with the 5.1 release we have mysqlslap. In my opinion it is sort of like Apache Bench but better.

  Simple syntax, flexible easy to use.

mysqlslap

- URL

http://dev.mysql.com/doc/refman/5.1/en/mysqlslap.html

mysqlslap

- Feature rich..... getting richer all the time

  Every time I look at the docs I see new features. This is good so take a look at the docs. Here are the basics...

mysqlslap

- Examples

shell>mysqlslap --user=root --auto-generate-sql

Benchmark

Average number of seconds to run all queries: 0.006 seconds

Minimum number of seconds to run all queries: 0.006 seconds

Maximum number of seconds to run all queries: 0.006 seconds

Number of clients running queries: 1

Average number of queries per client: 0

The –auto-generate-sql switch creates a table, executes an INSERT query and saves dummy data to it, executes a SELECT query to retrieve the dummy data, and then drops the table. To see the queries executed add a -v to the command.

mysqlslap

- More Examples

  The previous example wasa single thread, not often a realistic test of 'live' database activity.

  shell> mysqlslap --user=john --auto-generate-sql --concurrency=100

The –concurrency runs the test with 100 connections

mysqlslap

- More Examples

  Repeat Repeat... one of the rules of benchmarking is repeat the test and average the results..

  shell> mysqlslap --user=root --auto-generate-sql --concurrency=100 --iterations=5

  Now the test runs 5 times

mysqlslap

- More Examples

  --number-of-queries=10000

  Run 10,000 queries total divide total by number of connections to get q's per connect

  --query=”select name from city where id=12”

  Specify the query to run or a file to run queries from

mysqlslap

- More Examples

  --engine=myisam

  Specify the engine for autogenerated SQL

  --query=”select name from city where id=12”

  --concurrency=1,2,4,8,20,100

  Run same test increasing concurrency for each run.

mysqlslap

- Mysqlslap seems likely to be the most flexible tool for benchmarking the database. Easy to use yet full featured. Download it and give it a try
- Issues..

  Last I checked it did not work on windows.