

Benchmark MySQL Performance On FreeBSD and Linux

Leo Huang

NetEase.com, Inc.

E-mail: leo.huang.gd@gmail.com

1 Introduction

It seems to be the time to evaluate the MySQL performance on FreeBSD and Linux with the releasing of FreeBSD 6.0 and MySQL 5.0.16.

The following OS were used for our comparison evaluating:

- **FreeBSD 4.11**
- **FreeBSD 6.0**
- **Debian 3.1(kernel: 2.6.8)**

On each OS, we tested two versions of MySQL:

- **MySQL 4.1.15**
- **MySQL 5.0.16**

The evaluating computer is DELL PowerEdge 1650. Its hardware configuration is followed:

- **CPU: 2 * Intel Pentium III 1.33GHz 512KB Level 2 Cache(smp)**
- **Memory: 1024MB ECC SDRAM**
- **HD: SEAGATE ST336706LC (36GB Ultra160 SCSI 10000RPM)**
- **NIC : Intel(R) PRO/1000 Network Connection**

2 Evaluating Environment Setup

2.1 Install OS

2.1.1 Install FreeBSD 4.11

After the default installing FreeBSD 4.11, I recompiled the kernels with adding the following options:

```
maxusers      0
```

<i>options</i>	<i>SMP</i>	<i># Symmetric MultiProcessor Kernel</i>
<i>options</i>	<i>APIC_IO</i>	<i># Symmetric (APIC) I/O</i>
<i>options</i>	<i>MAXDSIZ="(1024*1024*1024)"</i>	
<i>Options</i>	<i>NMBCLUSTERS=65536</i>	

2.1.2 Install and Configuration FreeBSD 6.0

After the default installing FreeBSD 6.0, we recompiled the kernels with adding the following options:

<i>options</i>	<i>SMP</i>	<i># Symmetric Multiprocessor Kernel</i>
<i>options</i>	<i>MAXDSIZ="(1024*1024*1024)"</i>	

2.1.3 Install Debian 3.1

After the default installing Debian 3.1, we installed the kernel package: kernel-image-2.6.8-2-686-smp.

2.2 Install MySQL

2.2.1 Install MySQL on FreeBSD 4.11

We compiled our own MySQLs from MySQL tarball with linuxthreads On FreeBSD. The URL of MySQL tarballs are followed:

- MySQL 4.1.15 Tarball:
<http://dev.mysql.com/get/Downloads/MySQL-5.0/mysql-5.0.16.tar.gz/from/pick>
- MySQL 5.0.16 Tarball:
<http://dev.mysql.com/get/Downloads/MySQL-4.1/mysql-4.1.15.tar.gz/from/pick>

So, there are two MySQLs installed On FreeBSD 4.11

- **MySQL 4.1.15 On FreeBSD 4.11 Using linuxthreads (M4-F4-L)**
linuxthreads version: 2.2.3
GCC version: 2.95.4
- **MySQL 5.0.16 On FreeBSD 4.11 Using linuxthreads (M5-F4-L)**
linuxthreads version: 2.2.3
GCC version: 2.95.4

Their configuration parameters:

```
./configure --prefix=/usr/local/mysql --without-debug --enable-thread-safe-client
--enable-asmbler --with-mysqld-ldflags=-all-static --with-client-ldflags=-all-static
--with-extra-charsets=complex
--with-named-thread-libs='-DHAVE_GLIBC2_STYLE_GETHOSTBYNAME_R
-D_THREAD_SAFE -I /usr/local/include/pthread/linuxthreads -L/usr/local/lib -llthread -llgcc_r'
CFLAGS='-pipe -march=i686 -fno-omit-frame-pointer -O3 -DHAVE_BROKEN_REALPATH
-D__USE_UNIX98 -D_REENTRANT -D_THREAD_SAFE
-I/usr/local/include/pthread/linuxthreads' CXXFLAGS='-pipe -march=i686
-fno-omit-frame-pointer -O3 -felide-constructors -fno-rtti -fno-exceptions
-DHAVE_BROKEN_REALPATH -D__USE_UNIX98 -D_REENTRANT -D_THREAD_SAFE
-I/usr/local/include/pthread/linuxthreads' --with-big-tables --with-low-memory --with-innodb
```

2.2.1 Install MySQL on FreeBSD 6.0

MySQL can be compiled with many threading library: libpthread, linuxthreads, and libthr etc On FreeBSD 6.0. At first, we used the standard libpthread threading library. But we find that MySQL would crash when it was under heavy load. Daniel lo had reported this bug. You can get it from <http://bugs.mysql.com/bug.php?id=12251>. We decided to use linuxthreads and libthr. The linuxthreads version is 2.2.3. The GCC version is 2.3.4.

So, there are four MySQLs installed On FreeBSD 6.0

- **MySQL 4.1.15 On FreeBSD 6.0 Using linuxthreads (M4-F6-L)**
- **MySQL 5.0.16 On FreeBSD 6.0 Using linuxthreads (M5-F6-L)**

Note: Their configuration parameters are same as above.

- **MySQL 4.1.15 On FreeBSD 6.0 Using libthr (M4-F6-T)**
- **MySQL 5.0.16 On FreeBSD 6.0 Using libthr (M5-F6-T)**

The two above MySQL's configuration parameters are:

```
./configure --prefix=/usr/local/mysql --without-libedit --without-readline --without-debug
--enable-thread-safe-client --enable-asmbler --with-client-ldflags=-all-static
--with-extra-charsets=complex --with-named-thread-libs='-lpthread -D_THREAD_SAFE'
CFLAGS='-DHAVE_BROKEN_REALPATH -pipe -march=i686 -fno-omit-frame-pointer -O3
-D_THREAD_SAFE' CXXFLAGS='-pipe -march=i686 -fno-omit-frame-pointer -O3
-felide-constructors -fno-rtti -fno-exceptions -D_THREAD_SAFE' --with-big-tables
--with-low-memory --with-innodb
```

Then we built the file /etc/libmap.conf as:

<i>libc_r.so.5</i>	<i>libthr.so.2</i>
<i>libc_r.so.6</i>	<i>libthr.so.2</i>
<i>libthr.so.1</i>	<i>libthr.so.2</i>
<i>libpthread.so.1</i>	<i>libthr.so.2</i>
<i>libpthread.so.2</i>	<i>libthr.so.2</i>

2.2.3 Install MySQL on Debian 3.1

We used MySQL AB's binary distribution on Debian 3.1. And two MySQL were installed on Debian 3.1.

- **MySQL 4.1.15 On linux2.6.8 Using glibc2.3 (M4-L2.6-G2.3).** You can download it from <http://dev.mysql.com/get/Downloads/MySQL-4.1/mysql-standard-4.1.15-pc-linux-gnu-i686-glibc23.tar.gz/from/pick>
- **MySQL5.0.16 On linux2.6.8 Using glibc2.3 (M5-L2.6-G2.3).** You can download it from <http://dev.mysql.com/get/Downloads/MySQL-5.0/mysql-standard-5.0.16-linux-i686-glibc23.tar.gz/from/pick>

2.3 Configuration MySQL

We used my-huge.cnf as the basic MySQL configuration file. The additional options are following:

```
[mysqld]

key_buffer = 16M

# Try number of CPU's*2 for thread_concurrency
thread_concurrency = 4
max_connections = 1500
innodb_buffer_pool_size = 384M
innodb_additional_mem_pool_size = 20M
# Set ..log_file_size to 25 % of buffer pool size
innodb_log_file_size = 100M
innodb_log_buffer_size = 8M
innodb_flush_log_at_trx_commit = 1
innodb_lock_wait_timeout = 50
```

3 Benchmark Tools

We used Super Smack remotely (over TCP) to benchmark the MySQL performance. You can find it from <http://vegan.net/tony/supersmack/>.

4 Benchmark Database

Our benchmark database includes three tables: Member with 10 millions rows, Record with 1

million rows, and Result with 1 million rows. The statement of create tables are followed:

```
CREATE TABLE `Member` (  
  `aid` int(11) NOT NULL auto_increment,  
  `name` char(20) NOT NULL default "",  
  `passwd` char(16) NOT NULL default "",  
  `count` int(11) NOT NULL default '0',  
  `line` int(11) NOT NULL default '0',  
  `bill` int(11) NOT NULL default '2',  
  PRIMARY KEY (`aid`),  
  UNIQUE KEY `name` (`name`),  
  KEY `count` (`count`),  
  KEY `line` (`line`)  
) TYPE=InnoDB
```

```
CREATE TABLE `Record` (  
  `num` int(11) default NULL,  
  `host` int(11) NOT NULL default '0',  
  `aid` int(11) NOT NULL default '0',  
  `uid` int(11) NOT NULL default '0',  
  `sn` int(11) default NULL,  
  `count` int(10) unsigned default NULL,  
  `grp` int(11) default NULL,  
  `id` int(10) unsigned default NULL,  
  KEY `gr` (`grp`,`num`),  
  KEY `d` (`id`,`uid`)  
) TYPE=InnoDB
```

```
CREATE TABLE `Result` (  
  `id` int(11) NOT NULL auto_increment,  
  `uid` int(11) NOT NULL default '0',  
  `aid` int(11) NOT NULL default '0',  
  `count` int(11) NOT NULL default '0',  
  `time1` datetime NOT NULL default '0000-00-00 00:00:00',  
  `time2` datetime NOT NULL default '0000-00-00 00:00:00',  
  `total` int(11) default '0',  
  `countnum` int(11) default '0',  
  `num` int(10) unsigned NOT NULL default '0',  
  `times` int(11) default '0',  
  `point` int(11) default '0',  
  `ret` int(11) default '0',  
  PRIMARY KEY (`id`),  
  KEY `aid` (`aid`),  
  KEY `sid` (`ret`,`uid`)  
) TYPE=InnoDB
```

5 Evaluate MySQL

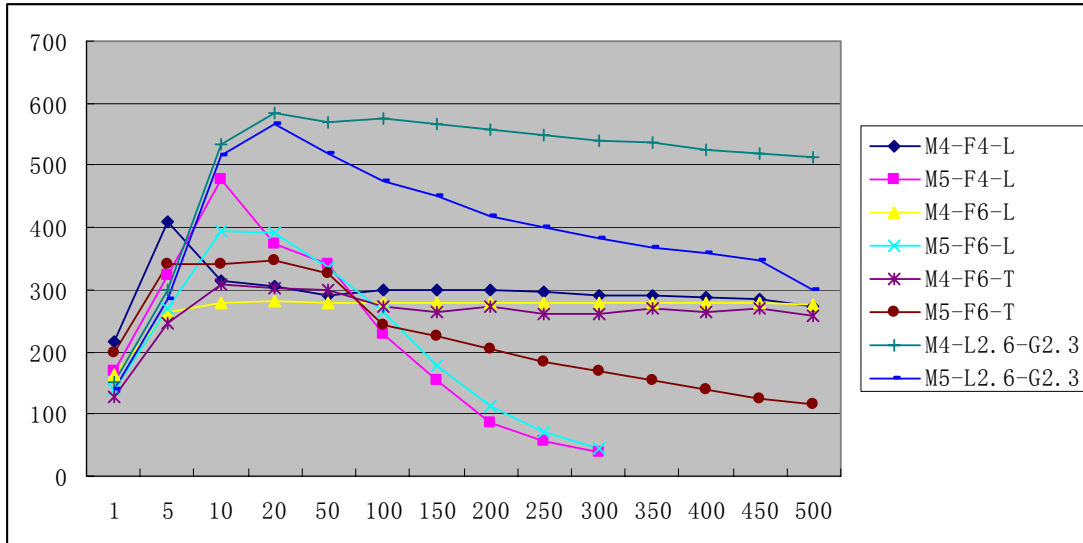
5.1 Benchmark MySQL Read Performance

We used Super Smack with setting of concurrent clients from 1 to 500. Each client executed the following select statement 1,000 times. We run this test 5 times, and averaged out the results.

select * from Member where aid=?;

The test results:

Client(s)	SELECT (queries per second)						
	1	5	10	20	50	100	150
M4-F4-L	216.264	409.522	313.254	306.7	290.594	300.848	298.902
M5-F4-L	170.058	323.16	477.446	372.566	339.972	227.59	152.85
M4-F6-L	164.312	264.014	278.518	282.086	278.774	279.702	279.768
M5-F6-L	138.452	269.988	394.386	391.1	333.834	264.302	177.408
M4-F6-T	127.532	246.678	308.684	302.258	299.73	273.992	263.85
M5-F6-T	199.006	342.458	341.26	348.496	325.02	243.772	226.098
M4-L2.6-G2.3	150.346	298.896	532.994	583.086	569.238	576.338	566.152
M5-L2.6-G2.3	140.874	283.878	516.414	567.25	519.364	475.932	451.25
Client(s)	200	250	300	350	400	450	500
M4-F4-L	298.372	295.694	291.804	289.356	287.588	283.786	273.736
M5-F4-L	84.964	55.88	37.502				
M4-F6-L	277.81	277.98	277.508	278.146	277.838	278.242	276.894
M5-F6-L	111.754	71.452	43.248				
M4-F6-T	273.038	261.792	261.202	269.052	264.376	271.11	258.178
M5-F6-T	205.888	184.448	169.568	154.178	139.282	124.466	115.778
M4-L2.6-G2.3	556.558	549.946	538.802	535.866	526.118	519.828	512.788
M5-L2.6-G2.3	419.354	399.936	381.212	368.348	359.238	346.448	298.148



Note: when the concurrent clients were 300, the performance of M5-F4-L and M5-F6-L were very low. So we didn't test further.

The test results show:

- The read performance of MySQL on Linux 2.6 is best. And their performance keeps high under heavy load.
- On FreeBSD 6.0, using libthr is better than linuxthreads.
- The read performance of MySQL 5.0.16 is poorer than MySQL 4.1.15, especially using linuxthreads on FreeBSD

5.2 Benchmark MySQL Write Performance

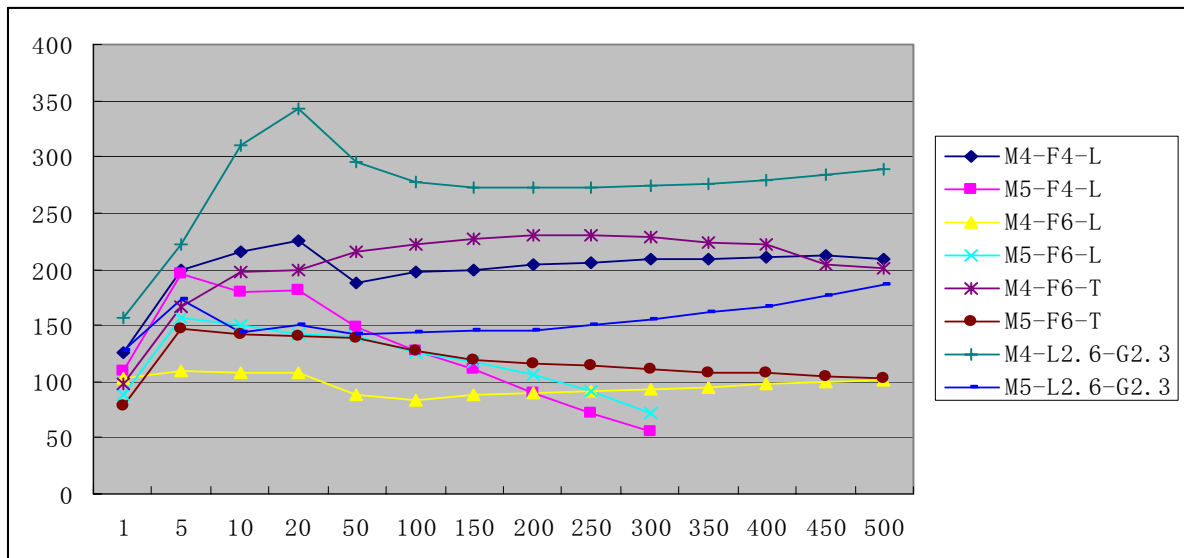
We used Super Smack with setting of concurrent clients from 1 to 500. Each client executed the following update statement 1,000 times. We run this test 2 times, and averaged out the results.

update Member set bill=1 where aid=?;

The test results:

Client(s)	UPDATE (queries per second)						
	1	5	10	20	50	100	150
M4-F4-L	125.125	199.625	216.135	225.375	187.57	197.92	199.96
M5-F4-L	109.915	196.365	180.395	181.695	148.76	127.385	110.225
M4-F6-L	103.26	109.775	107.585	107.53	88.015	82.915	88.455
M5-F6-L	87.715	155.95	149.96	141.695	141.025	125.71	118.345
M4-F6-T	97.88	167.065	196.91	199.56	215.215	221.64	227.045
M5-F6-T	78.34	147.215	141.975	140.475	138.105	127.03	119.075
M4-L2.6-G2.3	156.08	222.33	310.765	342.9	295.915	277.77	272.31
M5-L2.6-G2.3	127.015	173.285	144.255	149.75	142.565	144.335	144.65
Client(s)	200	250	300	350	400	450	500

M4-F4-L	203.725	205.375	209.06	209.445	210.95	212.075	208.285
M5-F4-L	90.245	72.405	54.965				
M4-F6-L	89.81	91.505	93.745	95.155	97.215	98.865	101.33
M5-F6-L	105.48	91.15	72.455				
M4-F6-T	230.795	230.555	228.66	224.44	222.19	203.79	200.64
M5-F6-T	115.8	113.745	111.41	108.32	108.005	105.29	103.5
M4-L2.6-G2.3	272.01	273.34	274.345	276.155	279.695	283.435	288.425
M5-L2.6-G2.3	145.81	150.48	154.695	160.955	167.31	176.45	186.86



Note: when the concurrent clients were 300, the performance of M5-F4-L and M5-F6-L were very low. So we didn't test further.

The test results show:

- The write performance of MySQL 4.1.15 on Linux 2.6 is best again.
- On FreeBSD 6.0, using libthr is quite good.
- The write performance of MySQL 5.0.16 is poorer than MySQL 4.1.15.

5.3 Benchmark MySQL OLTP Performance

We used Super Smack with setting of concurrent clients from 1 to 500. Each client executed the following transaction 100 times. We run this test 2 times, and averaged out the results.

```

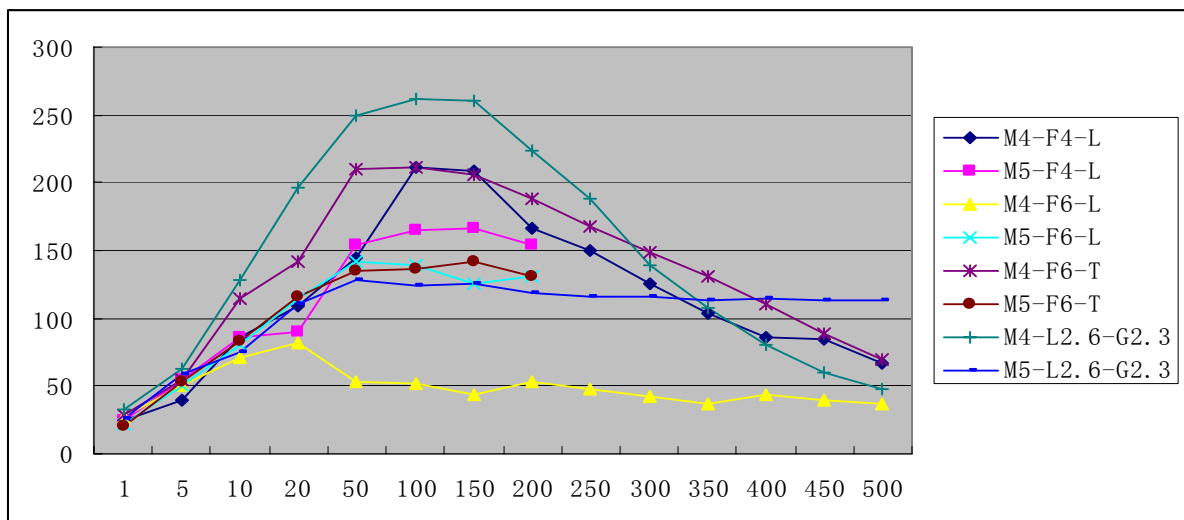
BEGIN;
select * from Record where id='$word' for update;
update Member set bill = 1 where aid='$wd';
insert into Result(uid,aid,count,time1,time2,total,countnum,
num,times,point, ret)
values(1,2,3,'2005-09-09 09:09:09','2005-10-10 10:10:10',
6,7,8,9,10,11);
delete from Record where id='$word';

```


COMMIT;

The test results:

		OLTP (tran per second)						
Client(s)		1	5	10	20	50	100	150
M4-F4-L		24.58	39.18	85.995	109.04	145.185	211.095	209.245
M5-F4-L		24.68	54.745	85.63	90.055	154.3	164.765	166.135
M4-F6-L		27.845	50.32	70.3	81.57	52.915	51.315	43.195
M5-F6-L		21.57	50.005	80.76	115.12	141.16	139.305	125.025
M4-F6-T		29.195	55.05	114.99	141.345	210.61	211.005	205.84
M5-F6-T		20.58	52.66	83.715	115.87	135	135.945	142.195
M4-L2.6-G2.3		32.895	63.22	127.685	195.785	249.185	262.12	260.545
M5-L2.6-G2.3		25.345	58.4	74.585	110.46	128.155	123.505	124.835
Client(s)		200	250	300	350	400	450	500
M4-F4-L		165.88	149.39	126.04	103.375	86.035	83.985	66.8
M5-F4-L		153.5						
M4-F6-L		53.2	47.165	42.245	37.26	44.12	39.555	37.39
M5-F6-L		131.25						
M4-F6-T		188.375	167.545	148.615	130.24	110.15	88.155	69.635
M5-F6-T		130.67						
M4-L2.6-G2.3		223.39	188.185	139.765	107.215	79.855	60.08	47.51
M5-L2.6-G2.3		118.045	116.275	115.49	113.68	114.11	113.565	113.49



Note: When we used 250 clients to test on M5-F4-L, M5-F6-L and M5-F6-T, the test fails because of deadlock.

The test results show:

- The OLTP performance of MySQL 4.1.15 on Linux 2.6 is best.
- On FreeBSD 6.0, using libthr is the better choice.
- The OLTP performance of MySQL 5.0.16 is worse than MySQL 4.1.16. On FreeBSD,

MySQL 5.0.16 will easy to cause deadlock under heavy load.

6 Conclusions

- The performance of MySQL 4.1.15 is better than MySQL 5.0.16 in our evaluating. Our suggestion is using MySQL 4.1.x now if you don't need the new functions of MySQL 5.0.16.
- The performance of MySQL on Linux 2.6 is better than FreeBSD.
- Using libthr on FreeBSD 6.0 is a very good choice. We don't advice anybody to use libpthread.