

Re: mysql performance on 4 * dualcore opteron

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/performance/2006-04/msg00014.html>

- *From:* Michael Vince <mv@xxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 20 Apr 2006 19:46:54 +1000
-

Michael Vince wrote:

I just ran a test on 6_stable (April 5th) on a Dell 2850 dual CPU (single core 3.60GHz) using the AMD64 build of FreeBSD and got similar speeds as you. Its interesting how Sven could have 8 cores with what appears to be less MySQL speed then just having a few cores. After enabling libthr it does jump by about 3,600 on a generic SMP kernel compile, I didn't try any more serious tweaks.

For those who are interested in exactly how I tested wheres what I did.

```
portupgrade -RN -m 'BUILD_OPTIMIZED=yes WITH_PROC_SCOPE_PTH=yes'  
/usr/ports/databases/mysql41-server  
portupgrade -RN /usr/ports/benchmarks/super-smack
```

```
super-smack -d mysql /usr/local/share/super-smack/select-key.smack 10 10000  
Query Barrel Report for client smacker1  
connect: max=4ms min=1ms avg= 2ms from 10 clients  
Query_type num_queries max_time min_time q_per_s  
select_index 200000 0 0 22061.88
```

With this below in my /etc/libmap.conf for libthr and a MySQL restart
/usr/local/etc/rc.d/mysql-server restart the numbers do jump.

```
[/usr/local/libexec/mysqld]  
libpthread.so.2 libthr.so.2  
libpthread.so libthr.so
```

```
super-smack -d mysql /usr/local/share/super-smack/select-key.smack 10 10000  
Query Barrel Report for client smacker1  
connect: max=238ms min=0ms avg= 117ms from 10 clients  
Query_type num_queries max_time min_time q_per_s  
select_index 200000 0 0 25601.49
```

Interestingly I just did a install of i386 FreeBSD 6.1RC1 and installed a PAE kernel (for 6gigs of ram) on this very same server (which had AMD64 FreeBSD on before hand) and run the exact same tests and its now a good deal slower!

Re: mysql performance on 4 * dualcore opteron

```
# super-smack -d mysql /usr/local/share/super-smack/select-key.smack 10 10000
Query Barrel Report for client smacker1
connect: max=3ms min=2ms avg= 2ms from 10 clients
Query_type num_queries max_time min_time q_per_s
select_index 200000 0 0 19234.02
```

And without libthr its even slower

```
# super-smack -d mysql /usr/local/share/super-smack/select-key.smack 10 10000
Query Barrel Report for client smacker1
connect: max=100ms min=22ms avg= 60ms from 10 clients
Query_type num_queries max_time min_time q_per_s
select_index 200000 0 0 16583.43
```

Does any one have any explanation of this?

Mike

I have also done benchmarking with libthr against Apache using 'ab' and found it can deliver an extra amount of megabytes/sec of data (I think it was about an extra 2000/requests sec) at the cost of giving the server from what I remember almost double the 'average load' according to 'top'

Given that if your machine has nothing else to do but deliver data purely from Apache then even libthr is more worth while for Apache as well.

Mike

Steven Hartland wrote:

Looking at this on a dual box here (waiting for the new MB for dual dual core)

All the time is spent processing super-smack and only 25% on mysqld.

Even dropping to 10 clients a large portion is take by the clients.

That said there is a lot that can be gained by using the tweaks out there

i.e. ULE + libthr + TSC + context_time.patch + cpu_acct_1.patch +
cpu_acct_2.patch

Adding these jumps from a baseline:

```
select_index 2000000 8 0 18624.60
```

to:

```
select_index 2000000 5 0 29942.10
```

The biggest increases coming from libthr (thanks DavidXu) and the ULE scheduler.

[log]

== 4BSD + libpthread + ACPI-Fast ==

```
super-smack -d mysql select-key.smack 100 10000
```

Re: mysql performance on 4 * dualcore opteron

Re: mysql performance on 4 * dualcore opteron

Query Barrel Report for client smacker1
connect: max=46ms min=6ms avg= 25ms from 100 clients Query_type
num_queries max_time min_time q_per_s
select_index 2000000 8 0 18624.60

super-smack -d mysql select-key.smack 10 100000
Query Barrel Report for client smacker1
connect: max=5ms min=0ms avg= 1ms from 10 clients Query_type
num_queries max_time min_time q_per_s
select_index 2000000 0 0 23983.87

== 4BSD + libthr + ACPI-Fast ==
super-smack -d mysql select-key.smack 100 10000
Query Barrel Report for client smacker1
connect: max=107ms min=2ms avg= 45ms from 100 clients Query_type
num_queries max_time min_time q_per_s
select_index 2000000 13 0 22413.39

super-smack -d mysql select-key.smack 10 100000
Query Barrel Report for client smacker1
connect: max=2ms min=1ms avg= 1ms from 10 clients Query_type
num_queries max_time min_time q_per_s
select_index 2000000 0 0 26841.07

== 4BSD + libthr + TSC ==
super-smack -d mysql select-key.smack 100 10000
Query Barrel Report for client smacker1
connect: max=46ms min=1ms avg= 21ms from 100 clients Query_type
num_queries max_time min_time q_per_s
select_index 2000000 11 0 23428.03

super-smack -d mysql select-key.smack 10 100000
Query Barrel Report for client smacker1
connect: max=2ms min=0ms avg= 1ms from 10 clients Query_type
num_queries max_time min_time q_per_s
select_index 2000000 0 0 26403.95

== ULE + libthr + TSC ==
super-smack -d mysql select-key.smack 100 10000
Query Barrel Report for client smacker1
connect: max=41ms min=0ms avg= 23ms from 100 clients Query_type
num_queries max_time min_time q_per_s
select_index 2000000 5 0 28581.18

super-smack -d mysql select-key.smack 10 100000
Query Barrel Report for client smacker1
connect: max=4ms min=0ms avg= 1ms from 10 clients Query_type
num_queries max_time min_time q_per_s
select_index 2000000 0 0 30128.44

== ULE + libthr + TSC + context_time.patch + cpu_acct_1.patch +

Re: mysql performance on 4 * dualcore opteron

```
cpu_acct_2.patch ==
super-smack -d mysql select-key.smack 100 10000
Query Barrel Report for client smacker1
connect: max=27ms min=0ms avg= 14ms from 100 clients Query_type
num_queries max_time min_time q_per_s
select_index 2000000 5 0 29942.10
```

```
super-smack -d mysql select-key.smack 10 100000
Query Barrel Report for client smacker1
connect: max=12ms min=0ms avg= 4ms from 10 clients Query_type
num_queries max_time min_time q_per_s
select_index 2000000 0 0 31057.52
```

```
== 4BSD + libthr + TSC + context_time.patch + cpu_acct_1.patch +
cpu_acct_2.patch ==
super-smack -d mysql select-key.smack 100 10000
Query Barrel Report for client smacker1
connect: max=54ms min=20ms avg= 38ms from 100 clients Query_type
num_queries max_time min_time q_per_s
select_index 2000000 9 0 24144.22
```

```
super-smack -d mysql select-key.smack 10 100000
Query Barrel Report for client smacker1
connect: max=2ms min=0ms avg= 1ms from 10 clients Query_type
num_queries max_time min_time q_per_s
select_index 2000000 0 0 27073.46
```

```
** update test **
super-smack -d mysql update-select.smack 10 100000
Query Barrel Report for client smacker
connect: max=3ms min=0ms avg= 0ms from 10 clients Query_type
num_queries max_time min_time q_per_s
select_index 1000000 1 0 6468.70
update_index 1000000 0 0 6468.70
[/log]
```

Machine:
Dual 244, 2Gb running FreeBSD 6.1-PRERELEASE (i386)
Package install of mysql 4.0
Port install of super-smack

Notes:
No detectable disk activity throughout the tests
ULE scheduler breaks the output from top with everything showing as
WCPU 0% in the 100 concurrency test and the numbers not adding up
at all in 10 concurrency test or showing 0%.
To get context_time.patch to work I needed the attached patch which
is basically two failed chunks of: kern/kern_exit.c moved to
kern/kern_thread.c

Steve

Re: mysql performance on 4 * dualcore opteron

----- Original Message ----- From: "Sven Petai" <hadara@xxxxxx>
To: <freebsd-performance@xxxxxxxxxxxx>
Sent: Tuesday, April 04, 2006 5:42 PM
Subject: mysql performance on 4 * dualcore opteron

hi

Before I begin, let me just say that I'm probably aware most of the threads about mysql performance in various fbsd lists over last couple of years, so please let's not concentrate on the usual points made over and over again like how filesystems are mounted under linux, how fast time() is or how various combinations of scheduler/threading library/compiler flags give you ~5-10% better performance. It's very unlikely that any of these reasons, or even all of them together can explain performance differences of 2-3 * so now a little bit of the background...

I usually use MySQL benchmark called super-smack as one of the benchmarks on all the new machines to get a general feeling of the servers performance.

I certainly agree that the default smack workloads are far too simple to say much about actual production performance, but still... better than nothing...

In general 2.4Ghz amd64 UP box (6.1 betaX) can do about 17400 q/s with select-smack+4bsd+thr combination and 4300 q/s with update-smack+4bsd+thr

on dualcore 2Ghz opteron (6.1 prerelease) the results are: 20000 q/s with select-smack+4bsd+thr and 4500 q/s with update-smack+4bsd+thr

performance for update-smack seems to be always 4XXX q/s, no matter how many CPUs the box has or what kind of raid controller/disks are used (i have tested on about 8 rather different machines). I have no idea if IO on all the servers I have tried really maxes out at this point or is there some bottleneck in UFS.

select-smack performance gains on dualcore are not quite as good as one might expect, but then again that dualcore box uses ECC memory which is probably somewhat slower because of the checksum calculations, and synchronisation has some overhead too... Anyway all in all I'm more or less happy with these results, even though linux will do about twice as much selects on the same hardware.

Today I had a chance to test 4 * 2Ghz dualcore opteron machine, so this machine has 8 cores in total and 8G of RAM.

Re: mysql performance on 4 * dualcore opteron

Now, on that server I get:

11000 q/s for select-smack+4bsd+thr combination (with KSE it's around 6000 q/s, ule+thr gives somewhere around 12000 q/s)

4100 q/s for update-smack+4bsd+thr

So the 8 core machine got almost 2* worse result for select than UP server.

After some tinkering I found out that renicing mysqld to -5 will make it push out 21000 q/s (4bsd, thr), so I suspect part of the problem is in the scheduling – probably super-smack with it's 100 processes gets just a lot more CPU time otherwise than mysql with it's 100 threads servicing them. But anyway even this result is still only about equal in performance to what I get from dualcore machine.

As I ran out of good (macro)tuning ideas at this point, and wanted to make sure higher scores are indeed achievable, I tried Linux on the same hardware.

Here are the results for same tests on Suse enterprise linux 9 (2.6.5-7.97-smp):

76857 q/s for select-smack

10050 q/s for update-smack

the mysql configuration was identical to the one I used under freebsd (my-huge). This Suse uses ReiserFS, but I have no idea about what kind of FS guarantees it provides, didn't see any sync/async stuff in the mount output.

I also repeated the tests on identical box that had Fedora installed (2.6.9-22-ELsmp) and used ext3'fs.

select-smack results were obviously almost the same as it doesn't touch the FS, update was about 8000 q/s.

I'm relatively sure that this kind of huge performance differences can't be explained by mere speed difference of time(), I haven't yet tested phk'd and roberts timer hacks, but at some point in time I rewrote mysql's timing code to completely avoid any calls to time() by keeping internal timestamp that was updated from TSC reg. value. It was certainly very ugly and imprecise, but worked well enough since mysql uses these code paths mainly for statistics and for setting various safeguard timeouts. Even with ~90% time() calls removed the performance still didn't get measurably better.

Of course it's possible that I fucked up somehow, so if someone has tested roberts and phk's changes then it would be certainly nice to hear about your results.

To make the long story short – does anyone have any good

Re: mysql performance on 4 * dualcore opteron

ideas about where might the bottleneck and how to debug it ?

PS

Here's some system/test information:

super-smack was used with concurrency of 100 and reqs. set to 10000

it was running on the same machine as the mysqld and connections were done over local socket.

timer: acpi-fast in all the cases

mysql: 4.1.18_2 from ports, table type is myisam

mysql configuration file:

<http://bsd.ee/~hadara/debug/mysql3/2way/my.cnf>

in general it's just my-huge.cnf from mysql distribution, with increased max_connections

kernel config is GENERIC-SMP (no it doesn't have WITNESS enabled)

== 4 * dualcore opteron ==:

vmstat 1, during select-smack test:

<http://bsd.ee/~hadara/debug/mysql3/8way/vmstat.txt>

dmesg:

<http://bsd.ee/~hadara/debug/mysql3/8way/dmesg.boot>

sysctl -a:

<http://bsd.ee/~hadara/debug/mysql3/8way/sysctl.txt>

== 1 * dualcore opteron ==:

vmstat 1, during select-smack test:

<http://bsd.ee/~hadara/debug/mysql3/2way/vmstat.txt>

dmesg:

<http://bsd.ee/~hadara/debug/mysql3/2way/dmesg.boot>

sysctl -a:

<http://bsd.ee/~hadara/debug/mysql3/2way/sysctl.txt>

freebsd-performance@xxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-performance>

To unsubscribe, send any mail to

"freebsd-performance-unsubscribe@xxxxxxxxxxx"

=====
This e.mail is private and confidential between Multiplay (UK) Ltd. and the person or entity to whom it is addressed. In the event of misdirection, the recipient is prohibited from using, copying, printing or otherwise disseminating it or any information contained in it.

In the event of misdirection, illegible or incomplete transmission please telephone (023) 8024 3137

or return the E.mail to postmaster@xxxxxxxxxxxxxxxxxxx

Re: mysql performance on 4 * dualcore opteron

freebsd-performance@xxxxxxxxxxx mailing list
<http://lists.freebsd.org/mailman/listinfo/freebsd-performance>
To unsubscribe, send any mail to
"freebsd-performance-unsubscribe@xxxxxxxxxxx"

freebsd-performance@xxxxxxxxxxx mailing list
<http://lists.freebsd.org/mailman/listinfo/freebsd-performance>
To unsubscribe, send any mail to "freebsd-performance-unsubscribe@xxxxxxxxxxx"

freebsd-performance@xxxxxxxxxxx mailing list
<http://lists.freebsd.org/mailman/listinfo/freebsd-performance>
To unsubscribe, send any mail to "freebsd-performance-unsubscribe@xxxxxxxxxxx"