

Re: blocking non blocking

Source:

<http://www.tech-archive.net/Archive/Development/microsoft.public.win32.programmer.networks/2006-07/msg00098>

- *From:* "Arkady Frenkel" <arkadyf@xxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 6 Jul 2006 09:40:42 +0200
-

The author mean case where you have nothing to do before all data arrive , so you need to wait any case , but if you can/have to do with data coming (even parts) async sockets preferable , OTOH if you need to do some work not connected to incoming data on the same thread , async sockets have to be used.

Arkady

"Daniel" <DanielV@xxxxxxxxxxxxxxxxxxxx> wrote in message news:ekvP5zIoGHA.1332@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Ok now i am very confused...here is why i think non blocking is less efficient, a quote from TCP/IP sockets in c# Practical guide for programmers:

"like polling, non blocking sockets typically involve some busy waiting and are not very efficient"

So you see why i am confused? I am getting conflicting opinions :(Can anyone explain and put me straight?

"Alexander Nickolov" <agnickolov@xxxxxxxx> wrote in message news:ekoQv3snGHA.1664@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

Why do you think non-blocking is less efficient? There's no real difference for a single client (occasional three system calls instead of one), but for multiple clients you dramatically cut on the number of threads needed thus significantly improving performance with non-blocking or overlapped model. (BTW, overlapped is the better deal of the two...)

I have some vague recollection from NT4 days that an NT4 server of those days (one CPU) crashed with under 1000 threads. Just to put this into perspective, even if you can manage somehow a server with 50K threads, it'll be prohibitively more expensive than an equivalent server using a non-blocking or overlapped I/O servicing 50K clients. Dual core won't cut it a by a long shot! (Perhaps 16 or 32 dual cores...)

Re: blocking non blocking

--

=====
Alexander Nickolov
Microsoft MVP [VC], MCS D
email: agnickolov@xxxxxxx
MVP VC FAQ: <http://www.mvps.org/vcfaq>
=====

"Daniel" <DanielV@xxxxxxxxxxxxxxxx> wrote in message
news:OCXwCIHnGHA.2256@xxxxxxxxxxxxxxxxxxxxxxxx

I have currently designed it using asynchronous threads.
From what i have
read and learnt about blocking and non blocking, it seems
non blocking is
less efficient than blocking but i have to beware of deadlock?

I am not using select and wait. I spawn a thread per client
which stays
active for as long as the client is connected. So theoretically i
could
have 50k threads running at any time. My server would
definitely be dual
core, so 25k threads per cpu to handle, and i may even use 2
multi core
servers and load balancing.

Does that sound like it could handle it? Is my current set up
ok for how
scalable a solution i require?

"Alexander Nickolov" <agnickolov@xxxxxxx> wrote in
message
news:OdqRGOHnGHA.4240@xxxxxxxxxxxxxxxxxxxxxxxx

50K open sockets definitely calls for IOCP
and a thread pool!
It may also call for multi-CPU or
multi-core machine I might add...
The other model that can deal with lots of
sockets is asynchronous
socket I/O, but it doesn't scale as much. With
select and wait
functions you are limited to 64 sockets per
thread. Blocking
sockets can't go beyond a handful of sockets
or risk major
latency introduced at your server...

--

Re: blocking non blocking

=====

Alexander Nickolov
Microsoft MVP [VC], MCSD
email: agnickolov@xxxxxxx
MVP VC FAQ: <http://www.mvps.org/vcfaq>

=====

"Daniel" <DanielV@xxxxxxxxxxxxxxxxxxxx>
wrote in message
news:uowZ029mGHA.4620@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

I may need to handle up to
50,000 clients at any given
time. Is that
how scalable you are
talking, is blocking ok for
this kind of size? My
data size is tiny though, at
largest 8Kb – 16Kb roughly
at any given
time.

"Alexander Nickolov"
<agnickolov@xxxxxxx>
wrote in message
news:%23%23kqq46mGHA.2252@xxxxxxxxxxxxxxxxxxxxxxxxxxxx

ioctlsocket
with
FIONBIO
or one of
the select
API calls
that
do it for you
(WSAEventSelect
and
WSAAsyncSelect).

The
advantage is
you don't
need two
threads per
socket per
client so
your server
can actually
scale to

Re: blocking non blocking

more than a
handful
of clients.
The
disadvantage
is the
somewhat
harder logic
needed
to
implement
non-blocking
communication.
However,
for a truly
high
performance
scalable
server you
want
overlapped
I/O and
a
completion
port used
with a
thread pool,
which is
significantly
harder to
implement...

--

=====
Alexander
Nickolov
Microsoft
MVP [VC],
MCSD
email:
agnickolov@xxxxxxx
MVP VC
FAQ:
<http://www.mvps.org/vcfaq>
=====

"Daniel"
<DanielV@xxxxxxxxxxxxxxxxxxxx>
wrote in
message
news:%23yE41I6mGHA.3376@xxxxxxxxxxxxxxxxxxxxxxxx

Re: blocking non blocking

Hey
guys

My
socket
connection
is
blocking=true,
and
i
have
multiple
clients
connecting
to
my
server.

The
more
clients
i
have
the
more
problems
i
have.

How
do
i
set
the
socket
to
non
blocking?
what
are
the
disadvantages
of
this?
What
are
the
advantages
of
this?

Re: blocking non blocking

Thanks