Solaris OS Specific

Planned Features

Pluggable Congestion Control for TCP and SCTP

A proposed OpenSolaris projectblocked URL foresees the implementation of pluggable congestion control for both TCP and SCTP. HS-TCP and several other congestion control algorithms for OpenSolaris. This includes implementation of the HighSpeed, CUBIC, Westwood+, and Vegas congestion control algorithms, as well as `ipadm` subcommands and socket options to get and set congestion control parameters.

On 15 December 2009, Artem Kachitchkine posted an initial draft of a work-in-progress design specificationblocked URL for this feature has been announcedblocked URL on the OpenSolaris networking-discuss forum. According to this proposal, the API for setting the congestion control mechanism for a specific TCP socket will be compatible with Linux: There will be a `TCP_CONGESTION` socket option to set and retrieve a socket's congestion control algorithm, as well as a `TCP_INFO` socket option to retrieve various kinds of information about the current congestion control state.

The entire plugabble-congestion control mechanism will be implemented for SCTP in addition to TCP. For example, there will also be an `SCTP_CONGESTION` socket option. Note that congestion control in SCTP is somewhat trickier than in TCP, because a single SCTP socket can have multiple underlying paths through SCTP's "multi-homing" feature. Congestion control state must be kept separately for each path (address pair). This also means that there is no direct SCTP equivalent to `TCP_INFO`. The current proposal adds a *subset* of TCP_INFO's information to the result of the existing `SCTP_GET_PEER_ADDR_INFO` option for `getsockopt()`.

The internal structure of the code will be somewhat different to what is in the Linux kernel. In particular, the general TCP code will only make calls to the algorithm-specific congestion control modules, not vice versa. The proposed Solaris mechanism also contains `ipadm` properties that can be used to set the default congestion control algorithm either globally or for a specific zone. The proposal also suggests "observability" features; for example, `pfiles` output should include the congestion algorithm used for a socket, and there are new `kstat` statistics that count certain congestion-control events.

Useful Features

TCP Multidata Transmit (MDT, aka LSO)

Solaris 10, and Solaris 9 with patches, supports **TCP Multidata Transmit** (MDT), which is Sun's name for (software-only) Large Send Offload (LSO). In Solaris 10, this is enabled by default, but in Solaris 9 (with the required patches for MDT support), the kernel and driver have to be reconfigured to be able to use MDT. See the following pointers for more information from `docs.sun.com`:

- *TCP Multidata Transmit for Solaris 9*, http://docs.sun.com/app/docs/doc/817-0493/6mg9pruab?a=viewblocked URL
- *TCP Multidata Transmit for Solaris 10*, http://docs.sun.com/app/docs/doc/817-0547/6mgbdbsmn?a=view#whatsnew-updates-98blocked URL

Solaris 10 "FireEngine"

The TCP/IP stack in Solaris 10 has been largely rewritten from previous versions, mostly to improve performance. In particular, it supports Interrupt Coalescence, integrates TCP and IP more closely in the kernel, and provides multiprocessing enhancements to distribute work more efficiently over multiple processors. Ongoing work includes UDP/IP integration for better performance of UDP applications, and a new driver architecture that can make use of flow classification capabilities in modern network adapters.

Solaris 10: New Network Device Driver Architecture

Solaris 10 introduces GLDv3 (project "Nemo")blocked URL, a new driver architecture that generally improves performance, and adds support for several performance features. Some, but not all, Ethernet device drivers were ported over to the new architecture and benefit from those improvements. Notably, the `bge` driver was ported early, and the new "Neptune" adapters ("multithreaded" dual-port 10GE and four-port GigE with on-board connection de-multiplexing hardware) used it from the start.

Darren Reed has posted a small C programblocked URL that lists the active acceleration features for a given interface. Here's some sample output:

```
$ sudo ./ifcapability
lo0 inet
bge0 inet +HCKSUM(version=1 +full +ipv4hdr) +ZEROCOPY(version=1 flags=0x1) +POLL
lo0 inet6
bge0 inet6 +HCKSUM(version=1 +full +ipv4hdr) +ZEROCOPY(version=1 flags=0x1)
```

Displaying and setting link parameters with `dladm`

Another OpenSolaris project called *Brussels* unifies many aspects of network driver configuration under the `dladm` command. For example, link MTUs (for "Jumbo Frames") can be configured using

```
dladm set-linkprop -p mtu=9000 web1
```

The command can also be used to look at current physical parameters of interfaces:

```
$ sudo dladm show-phys
LINK         MEDIA                STATE      SPEED DUPLEX   DEVICE
bge0         Ethernet             up         1000 full      bge0
```

Note that Brussels is still being integrated into Solaris. Driver support was added since SXCE (Solaris Express Community Edition) build 83 for some types of adapters. Eventually this should be integrated into regular Solaris releases.

Setting TCP buffers

```
# To increase the maximum tcp window
# Rule-of-thumb: max_buf = 2 x cwnd_max (congestion window)
ndd -set /dev/tcp tcp_max_buf 4194304
ndd -set /dev/tcp tcp_cwnd_max 2097152

# To increase the DEFAULT tcp window size
ndd -set /dev/tcp tcp_xmit_hiwat 65536
ndd -set /dev/tcp tcp_recv_hiwat 65536
```

Pitfall when using asymmetric send and receive buffers

The documented default behaviour (tunable TCP parameterblocked URL `tcp_wscale_always` = 0) of Solaris is to include the TCP window scaling option in an initial SYN packet when either the send or the receive buffer is larger than 64KiB. From the `tcp(7P)`blocked URL man page:

```
For all applications, use ndd(1M) to modify the  confi-
guration      parameter      tcp_wscale_always.      If
tcp_wscale_always is set to 1, the window scale  option
will  always be set when connecting to a remote system.
If tcp_wscale_always is 0, the window scale option will
be set only if the user has requested a send or receive
window  larger  than  64K.   The   default   value   of
tcp_wscale_always is 0.
```

However, Solaris 8, 9 and 10 do not take the send window into account. This results in an unexpected behaviour for a bulk transfer from node A to node B when the bandwidth-delay product is larger than 64KiB and

- A's receive buffer (`tcp_recv_hiwat`) < 64KiB
- A's send buffer (`tcp_xmit_hiwat`) > 64KiB
- B's receive buffer > 64KiB

A will not advertize the window scaling option and B will not do so either according to RFC 1323blocked URL. As a consequence, throughput will be limited by a congestion window of 64KiB.

As a workaround, the window scaling option can be forcibly advertised by setting

```
# ndd -set /dev/tcp tcp_wscale_always 1
```

A bug report has been filed with Sun Microsystems.

References

- *Solaris Tunable Parameters Reference Manualblocked URL*, in particular the chapter on *TCP Tunable Parameters*blocked URL, Solaris 10 System Administrator Collection, June 2006
- *TCP Tuning Guide - Solaris*, http://www-didc.lbl.gov/TCP-tuning/Solaris.htmlblocked URL
- *Solaris - Tuning your TCP/IP stack*, http://www.sean.de/Solaris/soltune.htmlblocked URL
- *FireEngine - A New Networking Architecture for the Solaris Operating System,* White Paper, S. Tripathi, November 2004, PDFblocked URL
- *Solaris Networking*, Presentation, E. Nordmark, October 2005, PDFblocked URL
- *Solaris OS Network Performance*, BigAdmin System Administration Portal, http://www.sun.com/bigadmin/content/networkperf/blocked URL
- *Solaris NIC speed and duplex settings*, B. Hutchinson, April 2006. Web page describing how to set speed and duplex parameters on some common Ethernet interfaces under Solaris. Includes a Bourne Shell script that outputs current settings. http://www.brandonhutchinson.com/Solaris_NIC_speed_and_duplex_settings.htmlblocked URL
- *OpenSolaris project Nemo*, http://opensolaris.org/os/project/nemo/blocked URL
- *Solaris Networking - The Magic Revealed (Part II)* blog entry, Sunay Tripathi, November 2005, http://blogs.sun.com/sunay/entry/the_solaris_networking_the_magicblocked URL
- *Siwiki (Solaris Internals Wiki): Networks/Tuning Network Performanceblocked URL* - includes lots of useful information, in particular for recent hardware such as Niagara (T1000/T2000) and Niagara 2 systems (T5120/T5220) and newer GigE/10GE adapters, mainly PCI Express ones.
- *OpenSolaris Project Brussels: A Uniform Interface for Driver Administration Through the dladm Command*, May 2008, BigAdmin System Administration Portal, http://www.sun.com/bigadmin/sundocs/articles/nicddladmconf.jspblocked URL

-- ChrisWelti - 11 Oct 2005, added section for setting default and maximum TCP buffers
-- AlexGall - 26 Aug 2005, added section on pitfall with asymmetric buffers
-- SimonLeinen - 27 Jan 2005 - 16 Dec 2009