Web10G: TCP Extended Statistics

A collaboration of PSC and NCSA

Joint Techs Summer 2012
July 18, 2012

Chris Rapier rapier@psc.edu

The 30 Second Sound Bite

- Web10G is an efficient, scalable, and easy to implement solution bringing TCP information to the user.
- It facilitates diagnosis making it easy to find and fix problems saving time and resources.
- It will be easy to build new tools and applications allowing developers to explore new possibilities.

Origins of Web10G

- Web10G builds on the Web100 project, 2000 - 2003, with the main goals of updating the original kernel patch and moving the kernel ABI from /proc-based to Netlink-based communication.
- Web100 was a collaboration of NCAR/PSC/NCSA.

Why Web100?

- If there is a network performance problem, why not ask TCP about the problem? Expose more of TCP's hidden machinery and "look under the hood”.
- This work supported a second goal: to provide receiver-side auto-tuning in the Linux kernel, which was accepted into mainstream during the project.
  - Later adopted by Microsoft (Vista), Apple (OSX 10.4), and *BSD.
- RFC4898 (Mathis, Heffner, Raghunarayan) became a draft standard shortly there after.

Why Not Web100 ?

- Web100 has limitations:
  - proc interface is being deprecated in the mainline.
  - proc interface does not scale.
  - Kernel instrument set (KIS) predates RFC 4898.
  - The KIS only supports RENO.
- Web100 is no longer supported.
  - No forward ports to new kernels.
- Web100 will never be part of mainline Linux.

What is Web10G?

- Kernel: Instrumentation of the Linux kernel to add TCP Extended Statistics, as defined in RFC 4898.
  - 123 instruments to fully describe TCP connections.
  - RENO available. CUBIC, BIC, and H-TCP very soon.
- ABI: Netlink based communication with KIS to bring metrics to userspace.
- API: An easy to use API using 4 calls to read and write the Kernel Instrument Set (KIS).

Why should I adopt Web10G ?

- The Web10G kernel ABI is more efficient and extensible.
  - Suitable for production servers.
  - Web100 limited to 30K connections. Web10G should handle millions.
- The Web10G userspace API is easier to use.
  - Only 4 calls. No read/write groups. SNMP like.
- Web10G is actively updated to match current kernels.
  - And will continue to do so
- Lastly, Web100 is dead.

Web10G Kernel ABI

- The kernel ABI uses the Netlink/Genetlink framework present in the kernel.
- Brings the KIS data out of the kernel to the user.
- Two modes of communication have currently been tested: a streamlined read of all instruments, and a controlled read/write with ability to specify a subset of instruments.

Generic Netlink

- Netlink is a wire-format communications channel commonly used for kernel/userspace communication.
    - Actively supported replacement to /proc
    - Brings kernel data to the user using a socket model.
- Generic Netlink is a response to the increasing popularity of Netlink, and the resulting concern that Netlink family numbers would soon be exhausted.
- The Generic Netlink family was added as a Netlink multiplexer.

Genetlink

- Genetlink is a conservative extension of Netlink, in that once a family is created, communication proceeds similarly to that of Netlink.
- Well supported in the Linux kernel.
- Used by Web10G to construct a flexible kernel API.
- In particular, Web10G uses this to define a "WEB10G" family for all communication.
- Simple to use: Open a socket, read/write the instruments.

tcp_estats_diag

- Alternatively, we have also built a very lightweight module dependent on “inet_diag” which is a kernel module resident in the mainline source.
- Relies only on netlink proper; < 350 lines of code.
- Not as flexible; does only an atomic read of all instruments.
- Originally for testing; a rough cut was included in a release earlier this year as an example of the adaptability to other interfaces.
- Once the KIS is in place you can use multiple methods to access it.

Userspace API

- Implementation is at an advanced alpha, but the core idea is similar to SNMP semantics: get, set, trap, etc.
- This is elegant enough to allow an easy transition of code currently written to use Web100.
  - A porting document will be available shortly on web10g.org
- Four calls allow you to do everything
  - TCPE_CMD_LIST_CONNS (list the connections you own)
  - TCPE_CMD_READ_CONN (read all or a subset of the variables)
  - TCPE_CMD_WRITE_VAR (write a variable)*
  - TCPE_CMD_READ_ALL (read all variables for all connections)*

* not implemented just yet. Should be in two weeks.

(ge)Netlink Userspace Libraries

- Web10G currently uses libmnl, a "minimal Netlink library", written by Pablo Ayuso; we may move to libnl. The translation is not difficult, however, they each have their strengths.
- Both are actively developed, and LGPL'ed.
- In addition, Andrew Adams of PSC has done some work on a low level userspace MNetlink framework.

Development schedule

- The core kernel API and userspace API are available on the Web10G.org website along with a reference demo client.
  - The latest version, as demonstrated in the workshop, will be available shortly after this talk.
  - Working on QT based clients similar to the Web100 tools.
- Currently at an advanced alpha and hope to move to beta by late summer/early fall.
- Porting of NPAD and other tools will begin later this year. Likely after the beta release.
  - We are looking for people interested in helping with this task or other development ideas.

Applications and Ideas

- Working to update Web100 tools like NDT, NPAD, and MLAB to Web10G: members of I2 have expressed interest in working on NDT; PSC will work on NPAD.
- Currently exploring applications in other spaces; extended diagnostics, non-diagnostic, end user, etc.
  - realtime tcptrace
  - web10g enabled HPN-SSH
  - Iperf reporting OOPs, CWND/RWIN, RXMTs
  - Dynamic tuning of flows to meet provider needs

Open questions

- Scalability: equipment in place for extensive testing of myriad bulk transfers; upper bounds on concurrent connections; lower bounds on read frequency.
- Other congestion control algorithms. Currently Web10G supports RENO's congestion control. BIC, CUBIC, and H-TCP will be available very soon. What else?
- Maintaining parity with kernel. 3.2 required some reworking, in part, to deal with the use of PRR. As kernels change the stack an ongoing effort will be required to keep up.
- VMs may end up being problematic.

Keeping Up To Date

- Website: web10g.org
- Mailing List:

  https://lists.psc.edu/mailman/listinfo/web10g-users