Web10G: TCP Extended Statistics

A collaboration of PSC and NCSA

JET meeting, 1/24/12

National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign

What is Web10G ?

- Instrumentation of the Linux kernel to add TCP Extended Statistics, as defined in RFC 4898.
  - Extensive per-connection metrics.
- Client tools for exploration of instruments.
- Minimal overhead added to kernel; most kernel code provided in loadable kernel module(s).

Prehistory

- Web10G follows up on the Web100 project, 2000--2003, with the main goals of updating the original kernel patch and moving the kernel API 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 all 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.
- RFC4898 became a draft standard shortly there after.

Why Web10G ?

- Web100 has limitations:
  - proc interface not suitable for mainline inclusion.
  - proc interface does not scale.
  - Kernel instrument set (KIS) does not match RFC 4898.

What does Web10G provide

- A minimal kernel patch defining the instrument set.
- A loadable kernel module managing the netlink communication.
- Conformance with RFC 4898.
- Client side tools.

Why should I adopt Web10G ?

- Web100 will never be adopted by the mainline kernel due to the proc interface.
  - Mainline inclusion of the instrument set will promote development.
- The Web10G kernel API is unobtrusive and extensible.
- The Web10G userspace API is lightweight.
- The project will be actively updating for current kernels.

Web10G kernel API

- The kernel API uses the netlink/genetlink framework present in the kernel.
- 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.

Web10G kernel API

- The kernel API uses the netlink/genetlink framework present in the kernel.
- 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

- To recall: netlink is a wire-format communications channel commonly used for kernel/userspace communication.
- 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

- In particular, Web10G uses this to define a “WEB10G” family for all communication.
- 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.

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; may be worth releasing.

Userspace API

- Implementation is not quite complete, but the core idea is similar to SNMP semantics: get, set, trap, etc.
- This is elegant enough to (hopefully) allow an easy transition of code currently written to use Web100.
- This has been a circumspect development away from the Web100 legacy code; some first-gen transitional patches have already been released.

(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 netlink framework.

Development schedule

- The core kernel API and userspace API are present in the demo provided for the JET meeting.
- The demo will be available on the Web10G web site for reference, as we ready for the release of alpha code in the coming weeks.
- Porting of NPAD and other tools will begin in the spring of this year.

Applications

- Various network diagnostic tools have been built using the Web100 framework: NPAD (PSC), NDT (I2), Mlab (Google).
- The intention is to update these tools to use Web10G: members of I2 have expressed interest in working on NDT; PSC will address NPAD.

Open questions

- Scalability: equipment on order for extensive testing of myriad bulk transfers; want to test upper bound on number of connections; lower bound on frequency of reads.
- Update to 3.2 kernel may occasion adjustment of instrument implementation.