TSC-I2: A Lightweight Implementation for Precision-Augmented Timekeeping

Xun Luo, University of Illinois at Chicago, xluo@cs.uic.edu
Jeff W. Boote, Internet2, <boote@internet2.edu>

The quality of timekeeping is critical for many network protocols and measurement tools. Software packages, such as UDT[1], thrulay[2] and owamp[3], make control decisions and network statistics by checking the timestamps on the sent/received packets, which are normally labeled according to the system clock. Although when synchronized with NTP [4], system clock could achieve satisfying accuracy, its precision is still a problem at microsecond level. For example, most Linux kernels update their system time counters at 10-millisecond interval, and get the time in-between by interpolating TSC register. As interpolation parameters are obtained at start-up and TSC register subject to frequency wander, the interpolated time's precision is questionable.

TSC-I2 (TSC-Internet2) is devised to address the precision problem mentioned above. The basic idea is to make TSC rate calibration a continuous process, thus the accuracy of interpolation parameters could be ensured, which in turn results in satisfying clock precision. TSC-I2 maintains a soft clock of its own, compares this clock to system clock periodically. During each comparison, it synchronizes itself with the system clock, and adjusts the interpolation rate based on the offset and rate errors regarding to system clock. Whenever the accuracy of the soft clock is ensured, TSC-I2 uses this clock to report time to the library user; otherwise, the system clock value is reported. The advantage of this design is that system clock is enhanced rather than substituted.

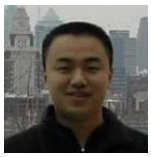
The clock discipline algorithm is enlightened by NTP. A state-machine-controlled PLL (Phase Lock Loop) traps the rate-induced phase difference between TSC-clock and system clock. Rate wander is captured within one loop delay, and corrected in 3 to 4 following loops. To avoid incorrect recognition of noise as rate-induced error, two filters -- a popcorn filter and a spike-detector -- are used. There are two usage modes: DAEMON and CLIENT. In DAEMON mode, a standalone daemon takes charge of timekeeping, serving one or more clients. In CLIENT mode, the library creates a thread running within the hosting process. Thus it minimizes the application's external dependency. There are also clear distinctions between TSC-I2 algorithms and its NTP counterparts, mainly due to the different natures of referencing sources. TSC-I2 is referencing to single, local, low-delay source, while NTP is referencing to multiple, networked and high-dispersion sources. Readers who are interested in TSC-I2 internals can visit the project website, where more details are illustrated.

TSC-I2 is fully implemented in C, around 2000 lines of code, fairly lightweight. It has been published under Open Source License at http://tsc-xluo.sourceforge.net. Present release is 0.08, which includes a user-mode daemon tsci2d, a C library libtsci2 and a set of utility tools as tsci2demo and tsci2measure. TSC-I2 currently supports IA32, AMD64 and Power PC architectures, as well as Linux, FreeBSD, Mac OSX and Microsoft Windows operating systems.

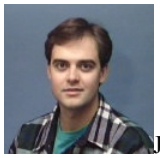
References

[1] http://udt.sourceforge.net/.
[2] http://www.internet2.edu/~shalunov/thrulay/.
[3] http://e2epi.internet2.edu/owamp/.
[4] http://www.ntp.org/

About The Authors:



Xun Luo is currently a Ph.D. candidate in Computer Science department, University of Illinois at Chicago. He is the author of TSC-I2. Xun has a Master's degree from University of Electronic Science & Technology of China and used to work at Global Software Group, Motorola China.



Jeff Boote is a Network Software Engineer of Internet2 and mentor of this project. He is the author for the One-Way Active Measurement Protocol (OWAMP), a tool for one-way latency measurement, which is being written as the sample implementation for a protocol draft going through IETF. Before coming to Internet2 in March of 2002, Jeff worked at the National Center for Atmospheric Research (NCAR).