GPS based time synchronization of PC hardware

Antti Gröhn
antti.grohn@netlab.hut.fi
Research goal and background

• Enable accurate network traffic measurements when analysis is based on packet arrival or departure time at different network nodes.

• On fast network 20 us inter-computer clock offset can make measurement results totally misleading.
  – According to time stamps packets seem to be received before even being sent.

• I have a dream:
  – inter-computer clock offset should be less than 2 us
Part 1  (April 2004 – October 2004, 7 months)

- Synchronization was done with NTP (Network Time Protocol) utility.
- Time reference: Trimble Acutime GPS receiver
  - NMEA / TSIP ASCII timecode output
  - 1 PPS (Pulse-per-second) output
    » Pulse rising edge indicates second transition
    » accuracy to UTC: 50 ns (1 sigma)
- Both signals are interfaced through computer’s serial port.
Research timetable

Part 2  (since October 2004, 4 months)

- System clock upkeeping is done using self-made PCI counter card and customized Linux kernel.
- Just make the clock run correctly
  - less need for actual synchronization
- Time reference: Trimble Thunderbolt GPS receiver
  - TSIP ASCII timecode output
  - 10 MHz output
  - 1 PPS output
    » Pulse rising edge indicates second transition
    » accuracy to UTC: 20 ns (1 sigma)
Part 1

- Motherboard’s clock oscillator can easily have frequency error of 10 ppm. (= 10 us per second or 864 ms per day)
- System clock is updated 100 times per second.  
  - on each timer interrupt (IRQ 0) 10 ms is added to system clock’s current time.
- NTP software changes the virtual frequency of system clock by changing the time increment on each timer interrupt.
- Stear the clock towards GPS and beat it

REALLY REALLY HARD
Synchronization system

- Antenna
- Trimble Acutime GPS (Stratum 0)
- ASCII timecode (RS232)
- PPS signal divider
- PPS signal (RS232)
- PPS signal (BNC / TTL)
- RS422

- Primary NTP server (Stratum 1)
- Secondary NTP servers (Stratum 2)
- Ethernet switch
Pulse processing delay measurement setup

- Trimble Acutime GPS
- PPS signal divider
- Pulse input
- Echo output
- NTP server
- Monitoring computer + NI PCI-6602 high resolution PCI interval counter
- Pulse detection and echo out
- Interrupt
Measured serial port echo delay and clock offset reported by NTP utility

Average echo delay is between 8 – 20 us on different computers
Upper and lower bound for individual clock offset

![Graph showing upper and lower bound for individual clock offset]

- Maximum offset (Intel Celeron 300 MHz) [us]
- Maximum offset (Intel Celeron 466 MHz) [us]
- Minimum offset (Intel Celeron 300 MHz) [us]
- Minimum offset (Intel Celeron 466 MHz) [us]
Upper bound for inter-computer clock offset

Realistic offset is 10 us less because both computers are likely to deviate the amount of echo delay from absolute time.
Loading the CPU with sequential C compiler processes

- **Process starts:**
  - Clock slows down

- **NTP utility can compensate effects of CPU load**

- **Process ends:**
  - Applied correction is too large and clock jumps ahead
Conclusions

- Inter-computer clock offset < 46 us (99%)
- Major error contributors:
  - Processing delay on serial port (avg 10 – 20 us)
  - Temperature fluctuations can cause periodical clock offset as high as ± 20 us.
  - CPU load variations can cause over 50 us instantaneous clock offset.
- NICE, BUT NOT ENOUGH!!!
  - What time is it, sir?
    » well it’s 12:15:36.455567 ± 20 us
- Let’s go even further and push the GPS system to it’s limits.
Part 2

- Motherboard’s timer interrupts determine only the time instant when system clock is updated.
  - Actual time increment between two successive timer interrupts is determined using counter card.
- Several error factors should be minimized:
  - PLL vs Temperature fluctuations
  - Dynamic time increment vs CPU load variations
- One might be attempted to modify motherboard directly but more general solution is desirable.
Counter card (SynPCI™)

10 MHz INPUT

Phase-difference detector

100 MHz VCO

Loop filter

Divide-by-10

24 bit counter

24 bit counter

PPS latch memory

Time interval latch memory

8 bit PPS counter

PCI Control Logic

Lattice 28 pin SPLD

Optoisolation

8 MSB

24 LSB

32 bit PCI bus
Counter card (SynPCI™)

- Count resolution: 10 ns
- Maximum count: 167 ms
- Mapped on I/O address space (0x0000 – 0xFFFF)
  - Address is configured with on-board jumpers
- For example: configured address 0x300
  - I/O read from 0x300 ->
    » poll active counter value
  - I/O read from 0x304 ->
    » read counter value + 8 bit PPS counter value and change active counter
  - I/O read from 0x308 ->
    » read PPS latch value
System clock second transition vs PPS pulse rising edge
Loading the CPU with sequential C compiler processes

System Clock Second Transition vs. PPS Pulse rising edge

Start of compiler process
Conclusion

• NICE AND ALMOST ENOUGH!!!
• Minor clock adjustments are still needed
• Time increments should be filtered to enhance system clock stability.
• Reading the system clock accurately is yet another problem.
  – If time reading procedure is delayed, the time it returns can be several microseconds wrong.
• Possible Applications
  – Test networks, time critical network services and general timing.
Any Questions?

Research must go on!

Please keep us in mind when giving away your money!