Globally Synchronized Time via Datacenter Networks

Prof. Hakim Weatherspoon, Cornell University
Joint with Ki Suh Lee, Han Wang, and Vishal Shrivastav

Cornell/Princeton Center on Network Programming Retreat
CornellTech, NYC
June 20, 2016
How can we scalable synchronize clocks with high precision?

Scalable – Entire datacenter
High precision – *bounded precision*; e.g. no clock differs by more than hundreds of nanoseconds

Capability essential for network and distributed applications
Networks – One-way delay, consistent updates, etc
Distributed systems – consensus, snapshots, etc
Problem: Clock synchronization is non-trivial

- Precision: difference between any two clocks
- Typical clock offset synchronization
  - RTT
  - Offset
Problem: Clock synchronization is non-trivial

- Precision: difference between any two clocks
- Typical clock offset synchronization
  - RTT = \( t_3 - t_0 - (t_2 - t_1) \)
  - Offset = \( \frac{t_1 + t_2}{2} - \frac{t_0 + t_3}{2} \)
Problem: Clock synchronization is non-trivial

- Precision: difference between any two clocks
- Problems affecting precision
  - Oscillator skew (i.e. frequency of clocks differ)
  - Reading remote clocks: timestamps, network stack, network jitter
  - Resynchronization frequency
Problem: Clock synchronization is non-trivial

• Precision: difference between any two clocks
• Synchronization Protocols

<table>
<thead>
<tr>
<th></th>
<th>Precision</th>
<th>Scalability</th>
<th>Overhead</th>
<th>Extra Hardware</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTP</td>
<td>us</td>
<td>Good</td>
<td>Moderate</td>
<td>None</td>
</tr>
<tr>
<td>PTP</td>
<td>sub-us</td>
<td>Good</td>
<td>Moderate</td>
<td>PTP-enabled devices</td>
</tr>
<tr>
<td>GPS</td>
<td>ns</td>
<td>Bad</td>
<td>None</td>
<td>Timing signal receivers, cables</td>
</tr>
</tbody>
</table>

Protocol precision dependent upon configuration, implementation, and/or network conditions
Solution: Use the PHY to synchronize clocks

- Protocol in the PHY
  - Each physically link is already synchronized!
  - No protocol stack overhead
  - No network overhead
  - Scalable: peer-to-peer and decentralized

Bounded precision, scales with network, No network traffic
Datacenter Time Protocol (DTP) [SIGCOMM 2016]

Precise and bounded synchronization

- 4 oscillator ticks (25ns) bounded peer-wise synchronization
- 150ns precision synchronization for an entire datacenter
- No clock differs by more than 150ns
- Free – No network traffic: Use the PHY!
Outline

- Introduction
- Design
- Evaluation
- Discussion
- Conclusion
DTP

• 10 Gigabit Ethernet
  – Idle Characters (/I/) and Control blocks (/E/)
  – Standard requires at least 12 idle characters /I/ between pckts
    • i.e. At least one 64-bit Control Block /E/ between pckts
  – Idle characters / control blocks sent even if no packets to send
  – DTP overwrites idle characters (control blk) to send protocol msgs

DTP does not effect standard at all
DTP

• DTP synchronizes 106-bit clock
• synchronizes low 53-bits between every pckt

• Properties of DTP
  – Bounded Precision
    • Bounded by 4 oscillator ticks, 25.6ns (oscillator tick is 6.4ns)
      – Oscillator tick at receiver, sender, and across clock domains at sender
    • Network precision bounded by $4TD$
      – where $T$ is oscillator period and $D$ is network diameter in hops
  – Requires NIC and switch modifications (like PTP-enabled devices)
DTP

Topics discussed in paper

• Precision in userspace applications
• Handling failure
• Different standards: 1GbE, 40GbE, 100GbE, 400GbE, etc
• External synchronization (i.e. synchronizing to true time)
• Incremental deployment
Outline

• Introduction
• Design
• Evaluation
• Discussion
• Conclusion
Evaluation

- Compare measured precision of DTP and PTP
  - Measurement and observation period was two days
- PTP: Compare precision between Timeserver and Servers
  - Mellanox NIC (hardware), IBM G8264 Switch, Timekeeper server
- DTP: Compare precision between leaf servers and switches
  - Terasic DE5 FPGA-based development Net board
PTP – Idle Network (No Network Traffic)

Clocks differ by a few hundred nanoseconds
Clocks differ by a few hundred nanoseconds
PTP – Medium Loaded Network (4Gbps Traffic)

Clocks differ by tens of *micro*seconds
PTP – Medium Loaded (4Gbps Network Traffic)

Clocks differ by tens of microseconds
PTP – Heavily Loaded Network (9Gbps Traffic)

Clocks differ by *hundreds of microseconds*
PTP – Heavily Loaded Network (9Gbps Traffic)

Clocks differ by *hundreds of microseconds*
DTP – Heavily Loaded Network (9Gbps Traffic)

Clocks *never* differed by more than 4 clock ticks, 25ns

*Bounded Precision*
DTP – Heavily Loaded Network (9Gbps Traffic)

Clocks never differed by more than 4 clock ticks, 25ns

**Bounded Precision**
Outline

- Introduction
- Design
- Evaluation
- Discussion
- Conclusion
Broader Context

SW programming the network

- Dataplane programming (e.g. P4) via P4FPGA [http://p4fpga.org]

- Physical Layer programming via SoNIC [NSDI 2013]
Outline

• Introduction
• Design
• Evaluation
• Discussion
• Conclusion
Conclusion
DTP provides

• Bounded precision: 4 oscillator ticks (25ns)
• Scalability: 150ns for entire datacenter
• Free – No network traffic: Use the PHY!
• Enables distributed and networked applications

Contact: hweather@cs.cornell.edu
http://www.cs.cornell.edu/~hweather
Project website: http://sonic.cs.cornell.edu
Group website http://fireless.cs.cornell.edu
Broader Context and My Contributions

Cloud Networking
- P4FPGA/SoNIC in SIGCOMM 2016, NSDI 2013/2014, and IMC 2014
- Bifocals in IMC 2010 and DSN 2010
- Maelstrom in ToN 2011 and NSDI 2008
- Wireless DC in ANCS 2012 (best paper) and NetSlice in ANCS 2012
- Chaired Tudor Marian’s PhD 2010 (now at Google)

Cloud Computation & Vendor Lock-in
- Plug into the Supercloud in IEEE Internet Computing-2013
- Supercloud/Xen-Blanket in EuroSys-2012 and HotCloud-2011
- Overdriver in VEE-2011
- Chaired Dan William’s PhD 2012 (now at IBM)

Cloud Storage
- Gecko in FAST 2013 / HotStorage 2012
- RACS in SOCC-2010
- SMFS in FAST 2009
- Chaired Lakshmi Ganesh’s PhD 2011 (now at Facebook)
Thank you