PTP @ Scale - Learning from Meta’s Journey
PTP @ Scale
Learning from Meta’s Journey

Dotan Levi, NVIDIA
Ahmad Byagowi, Meta
Once Upon A Time (circa 2020)
Roadblocks (Deal Breakers)
Turn Disadvantages into Advantages
PTP in DCs just got easier
This is how things looked like when we started

**Limited Scale**
- We need to build a PTP clock to scale by the millions
- Back than (less than 2 years ago) the answer was BC
- The root (AKA PTP Grandmasters client scale is ~ 100s)

**Total Error Bound – out of the question** - Impractical with boundary clocks, PTP only measure one leg...

**Limitations on Grandmasters** [check out the next session]
- With a handful of vendors feature velocity is low
- Lack of high speed (QSFP) support

**Limited tooling**: provisioning, monitoring and benchmarking

**Limited PTP**: no unicast; no IPv6; PTP Profiles are tailored to Telecom specs

**Lack of user-space time APIs**
THE UNEXPECTED CHALLENGE

- Boundary clocks are a challenge for an existing Data center installation
- Impact on entire existing network...
- Seems like a deadend...
This is what we started with

How – Conceptual PTP Network Design – F16 Scenario

- One-step PTP using IPv6 Unicast only
  Up to 1GM per plane

- No H/W changes to spines, no software stack, just TC configuration of switches

PTP @ Network scope

- Only RSWs have >1PPM clock module, and a servo Linux PTP stack
  Fallover implemented at BC

PTP @ RSW to server scope

- 50 PPM, NIC clock, as usual
  Can use Linux PTP stack
Multiple Roadblocks

Lack of Features
Lack of Capabilities
Closed Source
Requires Switch OS changes
Change in Net Topology
Lack of insight on total error due to boundaries

All these Serious Roadblocks!
Hack discovery, Project is Dead?!
Turn a Disadvantage Into an Advantage
This is the Outcome

Open Time Server
opencompute.org/wiki/Time_Appliance_Project

Spines & Cores
Transparent Clocks

ToRs
Transparent Clocks

NICs
Ordinary Clocks

OPEN POSSIBILITIES.
This is the Outcome

**Rich Open-Source Software**
- PTP grandmaster; Provisioning; Monitoring; Benchmarking; Security

**OCP Marketplace offering** Time Servers and Time Cards

**Works at any scale**, proven at hyper-scaler (~100ks)

**Minimal changes** to existing deployments
- E2E Error bound calculation by the servers made possible
- Support for Transparent Clocks network clock tree
- Unicast? IPv6? High-speed (100/200/400GbE)? no problem
PTP Everywhere

- Time Cards go into all sort of appliances, vehicles
- Time Cards in the Linux kernel
- Support for Windows and VMWare
- SmartNICs getting the PHC closer
- Software APIs to PTP error bound and attributes
## TAP Workstreams

<table>
<thead>
<tr>
<th>Workstream</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1 Open Time Server</td>
<td>Development of an open time server</td>
</tr>
<tr>
<td>#2 Data Center PTP Profile</td>
<td>Development of a PTP Profile tailored for DCs</td>
</tr>
<tr>
<td>#3 Precision Time API</td>
<td>Time APIs for the user space</td>
</tr>
<tr>
<td>#4 Oscillators</td>
<td>Classification and measuring of oscillators</td>
</tr>
<tr>
<td>#5 PTP Servos</td>
<td>Design and implement Advanced PTP Servos</td>
</tr>
<tr>
<td>#6</td>
<td></td>
</tr>
</tbody>
</table>
Call to Action

- Challenge the Status Quo
- Open Source and Democratization builds a stronger ecosystem of solutions
- Does not exist? Is it possible? Let's build it!
- Turn a disadvantages into advantages
- [github.com/opencomputeproject/Time-Appliance-Project](https://github.com/opencomputeproject/Time-Appliance-Project)

Every other Wed @ 9am (Pacific)
Open Discussion