leoleovich Fix broken OTS assembly picture link ... 20 days ago History

..	
LICENSE	3 months ago
README.md	20 days ago

README.md

*To save spec as **PDF** - select text below (including images) -> right click -> Print -> Save as PDF*

Open Time Server

Spec revision № 1.0

The Open Time Server (OTS) is an Open, Scalable and Validated reference architecture that can be deployed in Data Centers or in an edge environments.
This spec can be accessed using http://www.opentimeserver.com

Table of Contents

List of images

List Of Images	Description
Figure 1	Open Time Server System Diagram
Figure 2	Open Time Server Concept
Figure 3	Open Time Server Assembly

Abbreviations

Abbreviation	Description
AC	Atomic Clock
COTS	Commodity off-the-shelf
DC	Datacenter

Abbreviation	Description
GNSS	Global Navigation Satellite System
GTM	Go-To-Market
HW	Hardware
NIC	Network Interface Card
NTP	Network Time Protocol
OCP	Open Compute Project
OCXO	Oven-Controlled Oscillator
OTS	Open Time Server (Previously known as an **Open Grandmaster** or a **Leader**)
PHC	PTP Hardware Clock
PTM	Precision Time Measurements
PTP	Precision Time Protocol
SW	Software
TAP	Time Appliance Project
TCXO	Temperature-compensated Oscillator
ToD	Time of Day
TS	Timestamp
XO	Oscillator

Table 1. Abbreviations

General

OCP TAP is targeting to ease the addition of Time-sync as a service to the datacenter. The Project targets are to define the service requirements, deployment, and design of an open reference design.

The time-sync service is relying on a synchronization technology, for now, we are adopting PTP (IEEE 1588) with some addition to that and NTP.

PTP architecture is scalable and defines the time source from an entity called the **Time Server clock** (or stratum 1 in NTP terms). The Open Time Server is distributing time to the entire network and usually gets its timing from an external source, (GNSS signal).

The current state-of-the-art Open Time Server implementations suffer from a few drawbacks that we wish to accommodate:

- They are HW appliances that usually target different GTM than a DC
- They expose none standard and inconsistent Interfaces and SW feature-sets
- Development cycles and the effort needed to add new features are long and expensive
- It doesn't rely on open-source software
- The accuracy/stability grades aren't in line with DC requirements

High-Level Architecture

In general, the OTS is divided into 3 HW components:

1. COTS server
2. Commodity NIC
3. Time Card

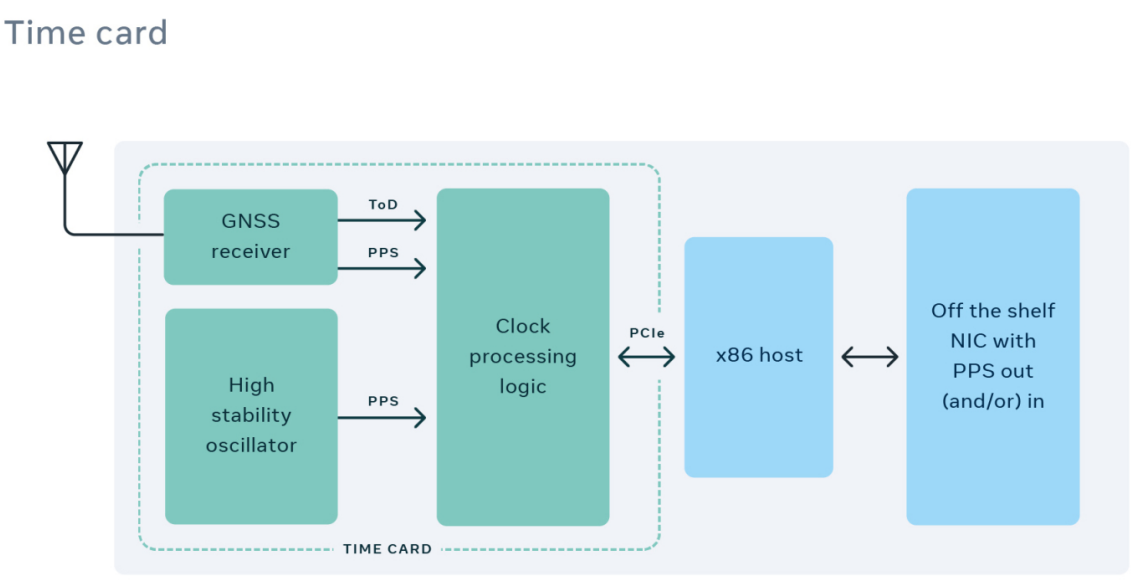

Figure 1. Open Time Server System Diagram

The philosophy behind this fragmentation is very clear, and each decision, modification that will be made, must look-out to this philosophy:

- COTS servers keep their "value for money" due to huge market requirements. They are usually updated with the latest OS version, security patches, and newer technology, faster than HW appliances.
- Modern Commodity NICs already support HW timestamp, lead the market with Ethernet and PCIe latest Speeds and Feeds. Modern NIC also supports a wide range of OS versions and comes with a great software ecosystem. NIC + COTS server will allow the OTS to run a full software (and even open source one) PTP and NTP stack.
- Timecard will be the smallest (conceptually) possible HW board, which will provide the GNSS signal input and stable frequency input. Isolating these functions in a timecard will allow OTS to choose the proper timecard for their needs (accuracy, stability, cost, etc) and remain with the same SW, interface, and architecture.

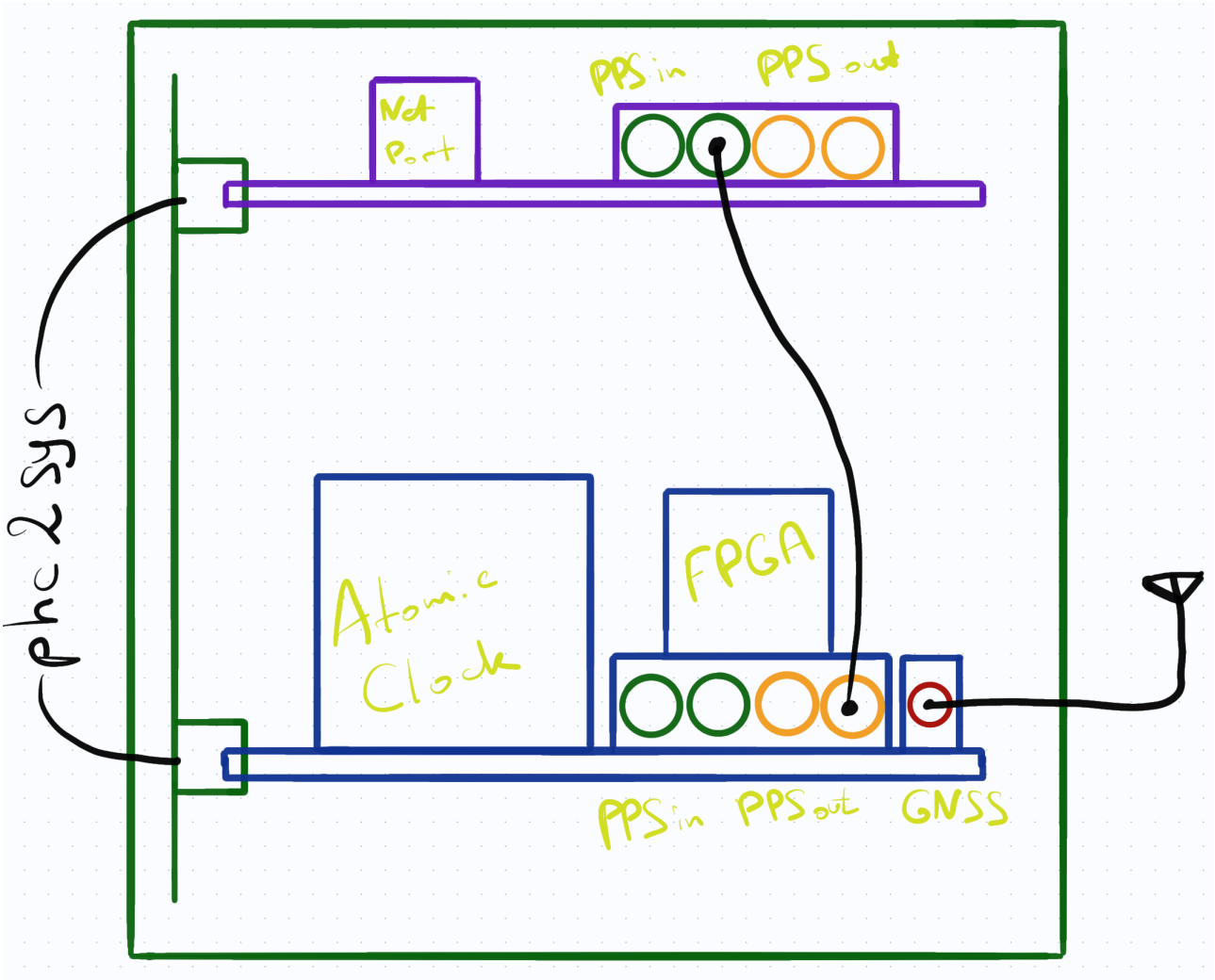

Figure 2. Open Time Server Concept

General Idea is the Time Card is connected via PCIe to the server and provides Time Of Day (TOD) via `/dev/ptpX` interface.
Using this interface `phc2sys` continuously synchronizes PHC on the network card from the atomic clock on the Time Card. This provides precision < 1us.
For the extremely high precision 1PPS output of the Time Card will be connected to the 1PPS input of the NIC, providing <100ns precision.

Responsibilities and Requirements

COTS Server

- Run commodity OS
- PCIe as an interconnect
- [optional] PTM Support

Network Interface Card

- Hardware timestamps
- [optional] PPS in/out
- [optional] PTM Support
- [optional] Time of day tunnel from timecard to SW

Time Card

- Holdover
- GNSS in
- PPS in/out
- Leap second awareness
- Time of day
- [optional] PTM Support

Detailed Architecture

Real life assembly components can be

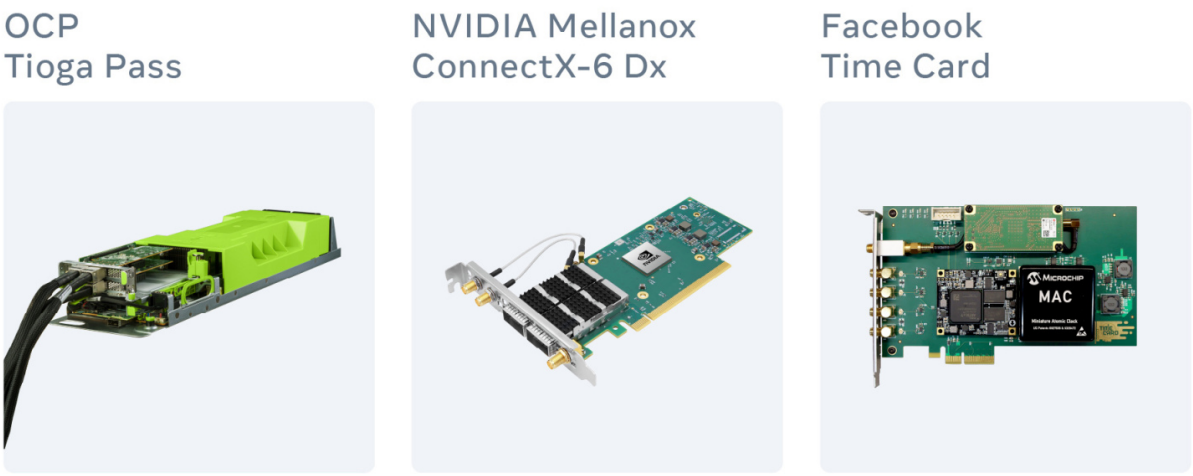

Figure 3. Open Time Server Assembly

COTS Server

Hardware

Most of the general purpose hardware can be used.
vt-d CPU support is required.
If multi-CPU system is chosen - we recommend connecting the Time Card and the Network Card to the same PCIe lane.
Synchronization between multiple CPUs will add an extra offset.

We've tested and proved setup is working with the following platform spec:

- HPE ProLiant DL380 Gen10
- OCP Tioga Pass
- SuperMicro 6019U-TRT - single or dual CPU
- SuperMicro 5019GP-TT - single socket with maximum throughput
- Intel NUC 9 Pro

Software

Please detailed software description document

- Linux operating system with the ocp_ptp driver (included in Linux kernel 5.12 and newer). Driver may require vt-d CPU flag enabled in BIOS
- NTP server - Chrony/NTPd reading `/dev/ptpX` of the Time Card

- PTP server - ptp4u or ptp4l reading `/dev/ptpX` of the NIC
  - phc2sys to copy clock values from the Time Card to the NIC

NIC

Most of the general purpose hardware can be used.
For the improved precision of NTP or PTP there can be extra requirements below.

Form-Factor

- Standard PCIe Stand-up HHHL Half-Height, Half-Length -or- OCP NIC 3.0
- Single Slot - Passive Cooling Solution
- Support for Standard PCIe Tall and Short brackets -or- OCP NIC 3.0 ejector types

PCIe Interface

- PCIe Gen3.0/Gen4.0 X n lanes on Gold-fingers, where n = at least 8

Network Ports

- Single or Dual-port Ethernet

Hardware timestamps

NIC should timestamp all ingress packets.
Non PTP packets can be batch and have a common TS in the SW descriptor, as long as they are not distant more than TBD nanosecond.
NIC should timestamp all PPP egress packets.

- PHC
- PTM
- 1PPS input
- [optional] 10MHz input which can be used as frequency input to the TSU unit
- [optional] Multi-host support

PPS out

- PPS Out Rise/Fall Time < 5 nano Sec
- PPS Out Delay < 400 pico Sec
- PPS Out Jitter < 250 femto Sec

- PPS Out Impedance = 50 Ohm
- PPS Out frequency 1Hz - 10MHz

PPS In

- PPS In Delay < 400 pico Sec
- PPS In Jitter < 250 femto Sec
- PPS In Impedance = 50 Ohm
- PPS In frequency 1Hz - 10MHz

Examples:

- NVIDIA ConnectX-6 Dx

Time Card

Please see Time Card details architecture document or simply visit www.timingcard.com.

General Idea is this card will be connected via PCIe to the server and provide Time Of Day (TOD) via `/dev/ptpX` interface. Using this interface ptp4l will continuously synchronize PHC on the network card from the atomic clock on the Time Card. This provides precision < 1us.

For the extremely high precision 1PPS output of the Time Card will be connected to the 1PPS input of the NIC, providing <100ns precision.

License

OCP encourages participants to share their proposals, specifications and designs with the community. This is to promote openness and encourage continuous and open feedback. It is important to remember that by providing feedback for any such documents, whether in written or verbal form, that the contributor or the contributor's organization grants OCP and its members irrevocable right to use this feedback for any purpose without any further obligation.