select, purchase and set-up server for octoi hub colocation

07/22/2022 09:58 AM - laforge

<table>
<thead>
<tr>
<th>Status:</th>
<th>Resolved</th>
<th>Start date:</th>
<th>07/22/2022</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>High</td>
<td>Due date:</td>
<td></td>
</tr>
<tr>
<td>Assignee:</td>
<td>laforge</td>
<td>% Done:</td>
<td>100%</td>
</tr>
<tr>
<td>Category:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

History

#1 - 08/10/2022 07:57 AM - laforge
- Priority changed from Normal to High

#2 - 08/11/2022 01:18 PM - laforge
- Status changed from New to In Progress
- % Done changed from 0 to 20

currently discussing the specs with the supplier. Will likely be a 2U machine with lots of CPU (16core EPYC) and 4x full-height PCIe slots (for TE820 or the like).

RAM / SSD I'd go for rather "minimal" config (in terms of an EYPC server, 64GB / 1TB) for now, can always be upgraded later if we should have need.

The point of the "lots of CPU" is to accomodate the un-optimized processing of many virtual E1 lines.

#3 - 08/11/2022 03:35 PM - laforge
Ok, specs are rather settled: AMD EPYC 7313P, Supermicro H12SSW-iN, 64 GB ECC DDR4-3200, 1TB NVMe

#4 - 08/13/2022 10:58 AM - laforge
- Checklist item [ ] purchase added
  - Checklist item [ ] OS installation added
  - Checklist item [ ] VM with SRV-IO like DIVO added
  - Checklist item [ ] install 2x TE820 added
  - Checklist item [ ] build + install timing cable added
  - Checklist item [ ] test timing cable added
- % Done changed from 20 to 0

server should arrive August 24th/25th and can subsequently be installed/configured.

I've also ordered IDC connectors + ribbon cables for both legacy + differential timing ports of the TE820 cards.

#5 - 08/25/2022 02:03 PM - laforge
- Checklist item [x] purchase set to Done

#6 - 08/25/2022 02:12 PM - laforge
laforge wrote in #note-4:

    I've also ordered IDC connectors + ribbon cables for both legacy + differential timing ports of the TE820 cards.

ordered the parts for this 2 weeks ago by digikey; should be at sysmocom office but somehow whoever unpacked it neither put them on my desk or otherwise notified me where they are stored :/

The server has meanwhile arrived. Physical installation of the 2x TE820 cards worked; 2 more empty slots for more cards in the future, if needed. No timing cable so far due to the unknown location of the ordered parts.
laforge wrote in #note-4:

I've also ordered IDC connectors + ribbon cables for both legacy + differential timing ports of the TE820 cards.

ordered the parts for this 2 weeks ago by digikey; should be at sysmocom office but somehow whoever unpacked it neither put them on my desk or otherwise notified me where they are stored :/

i think i put those in a extra box and thought labeled it. not 100% sure where it put it, but i will find it and put in on your desk.

laforge wrote in #note-6:

- Checklist item [x] OS installation set to Done
- Checklist item [x] install 2x TE820 set to Done

- % Done changed from 0 to 30

  - debian11 'host' OS with kvm-qemu installed
  - TE820 cards are recognized and showing up in lxpci.

Next is the VM setup with SRV-IO, need to copy what I did on DIVO.

- Checklist item [x] VM with SRV-IO like DIVO set to Done
- Checklist item [x] build + install timing cable set to Done

- % Done changed from 30 to 80

  - on the software side, osmo-e1d, trunkdev, yate, etc. setup has been created, systemd services should start everything at boot.
  - next up is testing the external E1 connections to PM3, AS54xx, ice1usb, ... including the timing cable (look if clocks drift between both TE820)

- Checklist item [ ] modprobe.d/kvm.conf:options kvm halt_poll_ns=0 added

- Status changed from In Progress to Stalled
- % Done changed from 80 to 90

the timing cables (differential or legacy) both don't work as expected.

from a bug report to digium support:

The setup is as follows:

- 0000:02:01.0 is the "master" card; it receives the reference clock on span 1
- 0000:02:02.0 is the "slave" card, it is expected to receive the clock from the "master" card

the expectation is that the bit-clock of the Tx signal generated by each span of each port to be synced with the received reference clock.

The kernel log shows correctly:

```bash
[1141.576130] wct4xxp 0000:02:01.0: TE8XXP: Span 1 configured for CCS/HDB3/CRC4
[1141.576297] wct4xxp 0000:02:01.0: SPAN 1: Primary Sync Source
[1141.577141] wct4xxp 0000:02:01.0: SCLK is master to timing cable
```

```bash
[1141.576130] wct4xxp 0000:02:02.0: TE8XXP: Span 2 configured for CCS/HDB3/CRC4
[1141.576297] wct4xxp 0000:02:02.0: SPAN 2: Primary Sync Source
[1141.577141] wct4xxp 0000:02:02.0: SCLK is master to timing cable
```
I'm using multi-channel oscilloscopes to look at the freq/phase relationship of the

The "master" card is correctly recovering the clock from it's "master" port, and I can see correct clock/phase
on the 7 other ports of the master card.

However, when plotting the E1 Tx signal of one of the ports on the "master" card vs. one of the ports on the "slave" card, then I can see very burry behavior:
* for something like 2s, the two signals are locked at a stable frequency + phase
* for the next 2s, the signals are experiencing significant frequency differences, before the process restarts (2s stable / 2s drift / 2s stable / ...)

How can I achieve proper sync between the two cards?

The dmesg output for the digium cards is as follows:

```
[ 2.085119] wct4xxp 0000:02:01.0: Found a Wildcard: Wildcard TE820 (5th Gen) (SN: DM21161300021)
[ 2.107918] wct4xxp 0000:02:01.0: firmware: direct-loading firmware dahdi-fw-oct6114-256.bin
[ 2.107921] VPM450: echo cancellation for 256 channels
[ 7.689054] dahdi: Detected time shift.
[ 9.972360] wct4xxp 0000:02:01.0: VPM450: hardware DTMF disabled.
[ 9.972363] wct4xxp 0000:02:01.0: VPM450: Present and operational servicing 8 span(s)
[ 9.972645] PCI Interrupt Link [GSIE] enabled at IRQ 20
[ 9.973309] wct4xxp 0000:02:02.0: 5th gen card with initial latency of 2 and 1 ms per IRQ
[ 9.973323] wct4xxp 0000:02:02.0: Firmware Version: 1.76
[ 9.973434] wct4xxp 0000:02:02.0: firmware: direct-loading firmware dahdi-fw-te820.bin
[ 9.979089] wct4xxp 0000:02:02.0: Found a Wildcard: Wildcard TE820 (5th Gen) (SN: DM21161300047)
[10.001607] wct4xxp 0000:02:02.0: firmware: direct-loading firmware dahdi-fw-oct6114-256.bin
[10.001609] VPM450: echo cancellation for 256 channels
```

9/21/2022
the system.conf looks like this:

span=1,1,0,ccs,hdb3,crc4
bchan=1-15,17-31
dchan=16

span=2,0,0,ccs,hdb3,crc4
bchan=32-46,48-62
dchan=47

span=3,0,0,ccs,hdb3,crc4
bchan=63-77,79-93
dchan=78

span=4,0,0,ccs,hdb3,crc4
bchan=94-108,110-124
dchan=109

span=5,0,0,ccs,hdb3,crc4
bchan=125-139,141-155
dchan=140

span=6,0,0,ccs,hdb3,crc4
bchan=156-170,172-186
dchan=171

span=7,0,0,ccs,hdb3,crc4
bchan=187-201,203-217
dchan=202

span=8,0,0,ccs,hdb3,crc4
bchan=218-232,234-248
dchan=233

span=9,0,0,ccs,hdb3,crc4
bchan=249-263,265-279
dchan=264

span=10,0,0,ccs,hdb3,crc4
bchan=280-294,296-310
dchan=295

span=11,0,0,ccs,hdb3,crc4
bchan=311-325,327-341
dchan=326

span=12,0,0,ccs,hdb3,crc4
bchan=342-356,358-372
dchan=357

span=13,0,0,ccs,hdb3,crc4
bchan=373-387,389-403
dchan=388

span=14,0,0,ccs,hdb3,crc4
bchan=404-418,420-434
dchan=419

span=15,0,0,ccs,hdb3,crc4
bchan=435-449,451-465
dchan=450
Here you can see the scope plot of comparing the Tx signal of a port from master (yellow) with a port from slave (magenta): https://people.osmocom.org/laforge/photos/te820_timingcable_problem.mp4

As can be seen, for some very brief periods the signals are perfectly locked, but then all mayhem breaks loose, only to recover for a brief moment later.

I've meanwhile also tried with a legacy timing cable instead of the differential timing cable: Exactly the same behavior. Without timing cable, dahdi_cfg "hangs" and there are various errors in dmesg, and the signals transmitted by either card are continuously running off their own clock, so I'm rather confident that the cables do work.

It's the sync mechanism that is not working as expected.

For reference, when I remove timing cables and related config and use an E1 loopback cable between one of the "master" ports to the "slave" card ports and use that as sync source, the lock between any of the master card spans and slave card spans is perfect, see https://people.osmocom.org/laforge/photos/te820_notimingcable_loopback.mp4

That is the kind of sync lock one would expect with the timing cable.

Waiting for their response on this now.

#14 - 09/15/2022 03:23 PM - laforge
- Status changed from Stalled to Resolved
- % Done changed from 90 to 100

laforge wrote in #note-13:

the timing cables (differential or legacy) both don't work as expected.

Many experiments later:

I think I have uncovered that first problem. However, a second problem follows. Details below.

So it seems that those strange bursty clock changes I have observed and reported only occur as long as the "slave" card doesn't yet have any of its spans in GREEN/OK state. Once any span is GREEN, the bursty clock changes disappear on all of its spans, even those that do not have GREEN/OK state.

This is quite strange, and I do not know that kind of behavior from any other equipment with high E1 port density, like Cisco equipment for example. The transmitted bitclock is always rock-solid, even when the link is not aligned, and particularly completely independent of whatever happens on the other links.

So basically a remote receiver must not use the bitclock provided by a Digium card until/unless its link or any other link is fully aligned.

So while that mystery is resolved, it now uncovers a second problem:

Even when the span[s] of the "slave" are in GREEN/OK state, their bitclock still slowly drifts compared to the bitclock of the "master"

As far as I can see from crude manual estimates on the scope screen, it is drifting roughly 250ns every 20 or so seconds, i.e. 12.5 ppb.

This will of course lead to cycle slips due to overruns/underruns eventually and in my opinion defeats the point of synchronizing multiple cards.

As before, when feeding the clock via an external E1 loopback cable from one of the spans of the "master" to a span of the "slave", this problem goes away. At that point, the clocks are completely locked, with no drift whatsoever visible on the scope screen.

So whatever the timing cable is doing in the background (it's sadly not documented for users to understand), the type of sync/lock achieved is much inferior to the type of sync/lock that is achieved when sacrificing one span on each card and plugging in a loopback cable.

Is this a known/expected problem? It almost looks like the bitclock of the slaves is not really sourced from the timing cable. It looks like the hardware is designed to just "sample/measure" the clock from the timing cable to adjust its local clock to roughly match that received from the timing cable. Something like a sample or rounding error in that process then translates to the observed slow clock drift.

In other words, the timing cable doesn't provide a real frequency "lock" like a proper PLL would provide.

Unless there is something that can still be fixed in the gateware/firmware or software, it looks like the 8-port cards can really only be used as 7-port cards when the requirement is to drive all ports off one reference clock.
The latter is exactly what I'm now doing in the setup, see AVSt_Server. Port 8 of the master card is used as clock output to port 8 of the slave card where it is used as input. The timing cable is physically present/connected, but not used (no related kernel module parameter)