

Detailed deadtime measurements

Deadtime
measurements:

very
cumbersome:

Oscilloscope

Before

--acq-meas-dt

Nodes

DT (μ s)

42.8k	r	r41-49	MASTER	2, 1, 13; 13
--X--		x061-41	0F1B1	2, 2, 4; 21
--X--		x061-23	0F13C	2, 3, 5; 5
--X--		x061-99	0F110C	2, 2, 3; 21
--X--		x061-10	0F111C	2, 2, 3; 21
--X--		x061-77	0F112C	2, 2, 4; 23
--X--		x061-91	0F113C	2, 2, 3; 20
--X--		x061-10	0F11D1	2, 2, 3; 21
--X--		x061-10	0F11D2	2, 2, 3; 21
--X--		x061-10	0F11D3	2, 2, 3; 21
--X--		x061-10	0F11D4	2, 2, 3; 21
--X--		x061-40	gsi.de	EB 71.5M

Why?

To reduce global deadtime,
the slowest node(s) must be
identified and **characterized**

Detailed DT info:

- Live
- Per readout node

After

Forgiving time sorter

Any event-builder
(hard dead-time domain)
crash

⇒ (most often) ⇒

Restart of
all systems

Before

ts-disable=20s

Failing

```
21:18:46 Waited 10 seconds for timesorter data.  
21:18:56 Waited 20 seconds for timesorter data.  
21:18:56 No data - disabling timesorter source.  
21:20:02 [trig: 5, eventno: 23609, size 72]  
Got data - re-enabling timesorter source.
```

Repaired

Why?

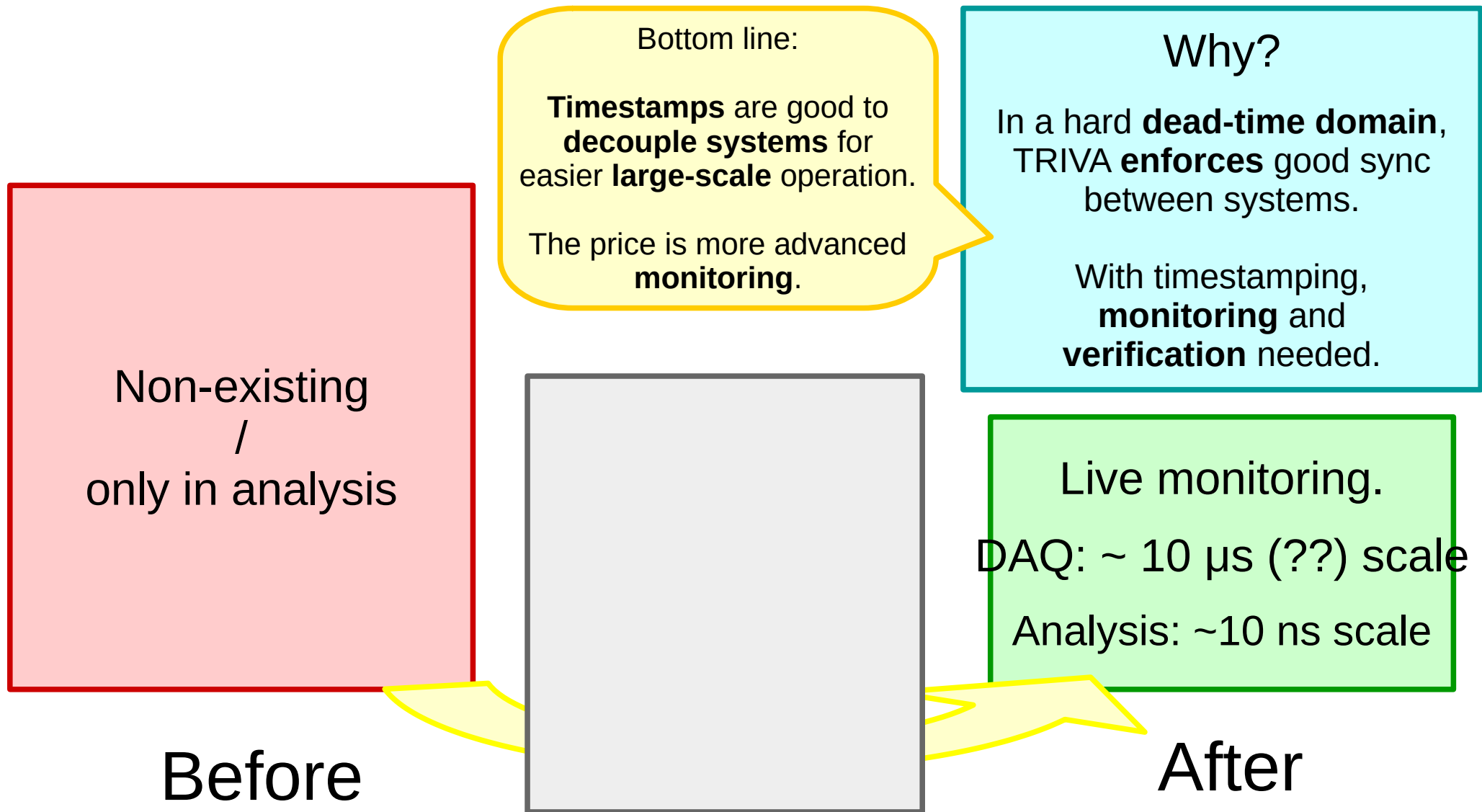
Maximize DAQ uptime!

Especially during setup:
Functional systems not punished.

Only offending
system(s) affected.

After

Live timestamp monitoring



Common readout library: NurdLib

Many
f_user.c

Modified per:

- Experiment
- Node

Recompile,
recompile...

Before

```
# Name and crate ID
CRATE("VME0")
{
  # Use multi-event readout
  multi_event = true
  # Type + address
  GSI_TRIDI(0x02000000) {}
  CAEN_V775(0x00010000)
  {
    common_start = true
    time_range = 1200 ns
  }
}
```

Why?

Make readout **code** manageable.
Run the **same code** everywhere.
Test the **same code**.
Fix bugs only **once**.

Configuration files.

After

No network timeouts

Unclean restart
or **crash**



Wait for
TCP timeouts

Before

EADDRINUSE:
Error creating socket

Why?

Faster restarts.

DAQ not running?
everyone waits...

Almost never.

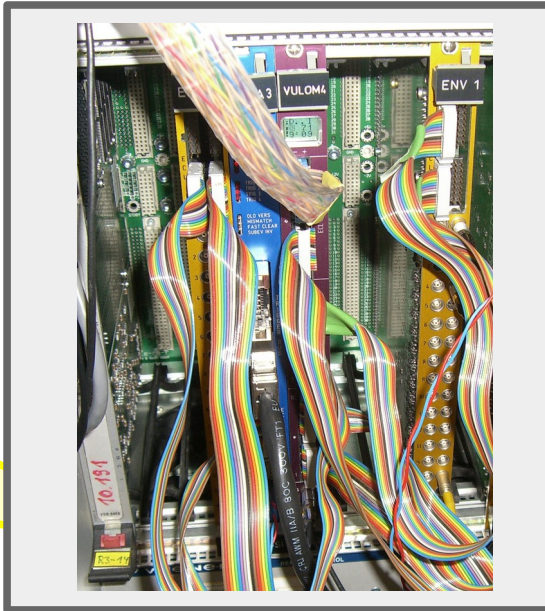
(Fix: Sacrificial TCP port.)

After

Trigger logic

Deadtime locking
+
NIM logic maze

Before



Why?

No leaking triggers.
Full remote control.

Full-featured
configurable logic
Swiss knife.

After

Unpacking for DAQsters

Handwritten parsing
of binary data...

(Sometimes in
FORTRAN...)

Before

```
SUPER_TDC(slot)
{
  UINT32 value;
}

SUBEVENT(ONE_CRATE)
{
  tdc1 = SUPER_TDC(slot=5);
  tdc2 = SUPER_TDC(slot=6);
}

EVENT
{
  crate1 = ONE_CRATE(type=5);
}
```

Why?

Check every single bit
when unpacking.

lex & yacc-driven:
Event specification
files
+
UCESB

After

Shell-scripted operations

Manual operations:

- a) Limited integrated shells, without variables...
- b) Colourful GUIs, without scripting

Before

```
#!/bin/bash
set -e

./vulom_free.bash

nurdlib/bin/m_read_meb.drasi \
--buf=size=1Gi \
--max-ev-size=0x100000 \
--max-ev-interval=50s \
--label=MDPP \
--server=drasi,dest=host \
--subev=control=1\\
    type=10,subtype=1 \
--server=stream:8001
```

Why?

Make DAQ operation manageable
(also at 03:30 in the night)

Simple bash scripts
per system:

For setup / start / stop

Easy integration with GIT.

After