EUROPEAN
SPALLATION
SOURCE

EPICS PVA Connections and Threads

Some more PVXS marketing

PRESENTED BY SKY BREWER, JOÃO PAULO MARTINS, STEPHANNE ARMANET, LUCAS MAGALHAES, ABRAHAM WOLK, MATEUSZ NABYWANIEC, SIMON ROSE, ANDERS LINDH OLSSON

2025-09-20

MCR Chaos

Beam commissioning begins with confused faces

OPIs cannot connect to PVs

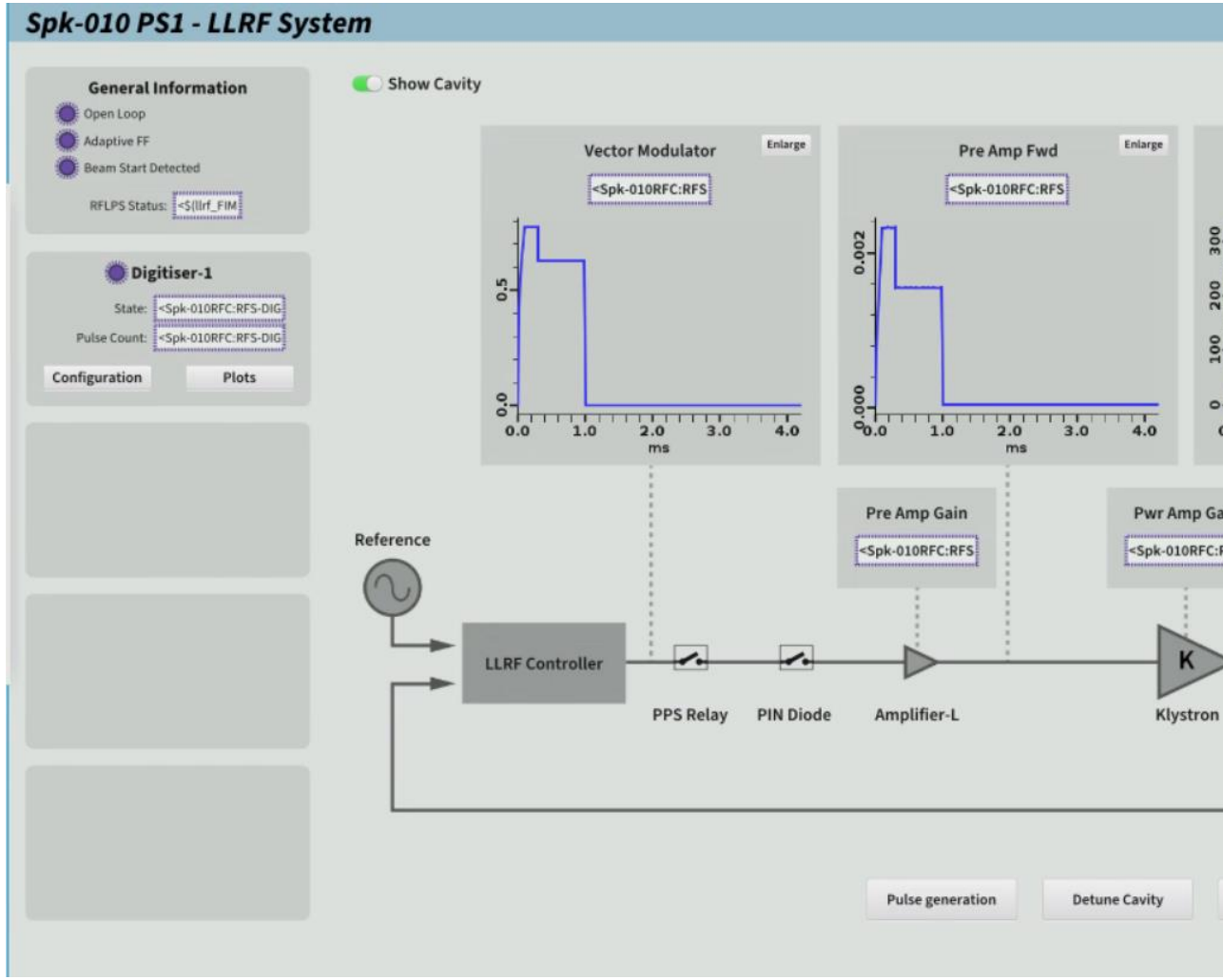
Beam commissioning begins with high and low level debugging

Our Infrastructure engineers go down to investigate, they initially assumed a network issue as many PVs were not getting connection.

OPIs already connected could continue monitoring them.

Using pvget and pvmonitor appeared to be working.

One System Owner suggests only looking at LLRF IOC



Temporary fixes

24 hour deadline to provide a solution

Reproduction is produced the next day by repeatedly opening and closing filter connections with the IOC at speed.

All decimation channel filters are removed from OPIs

In the short term, hotfix release of e3 for the LLRF IOC has the pva library replaced with pvxs, as this is tested to not have the problem.

Debugging the LLRF IOC

IOC crashes

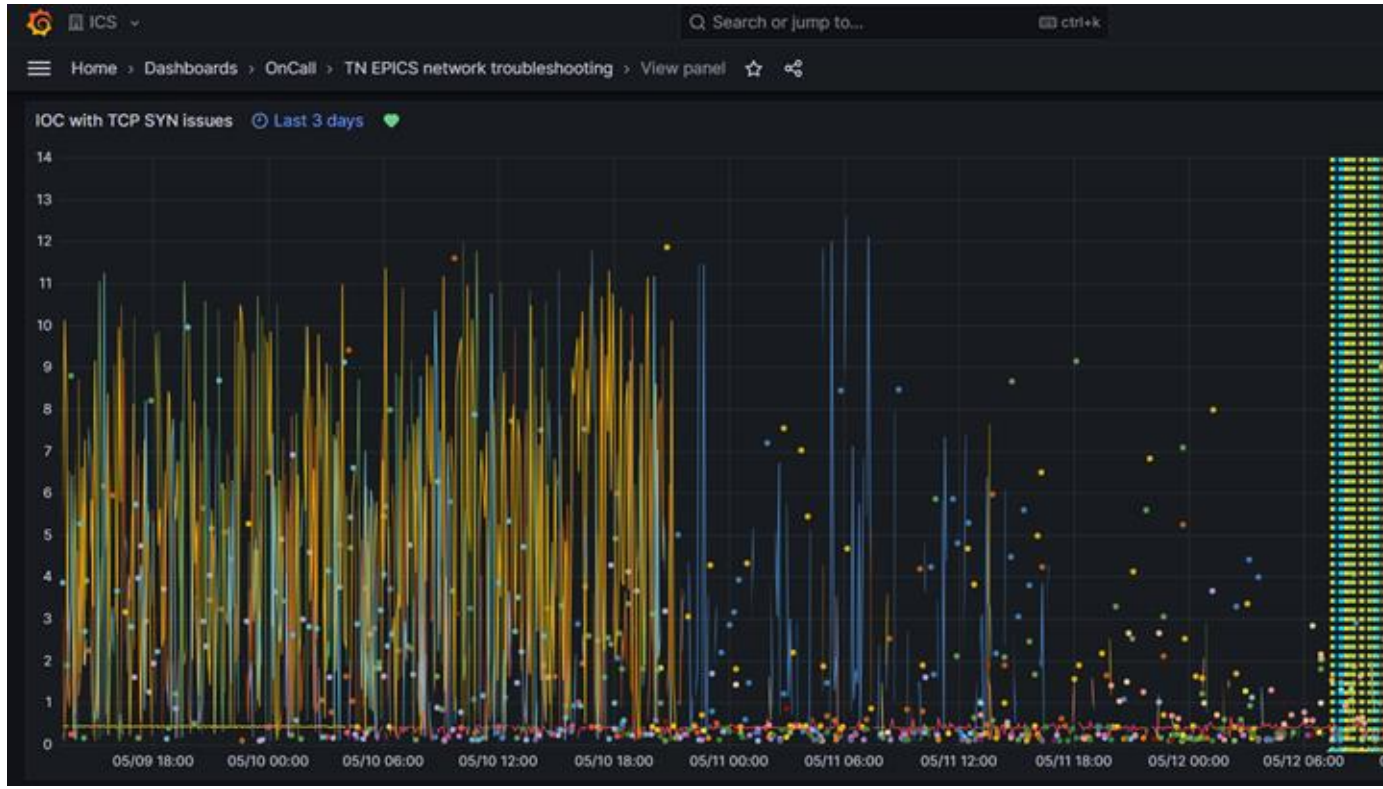
```
A call to 'assert(status ==
epicsMutexLockOK)'^M


    by thread 'PDB-event' failed in
../freeList/freeListLib.c line 169.^M
```

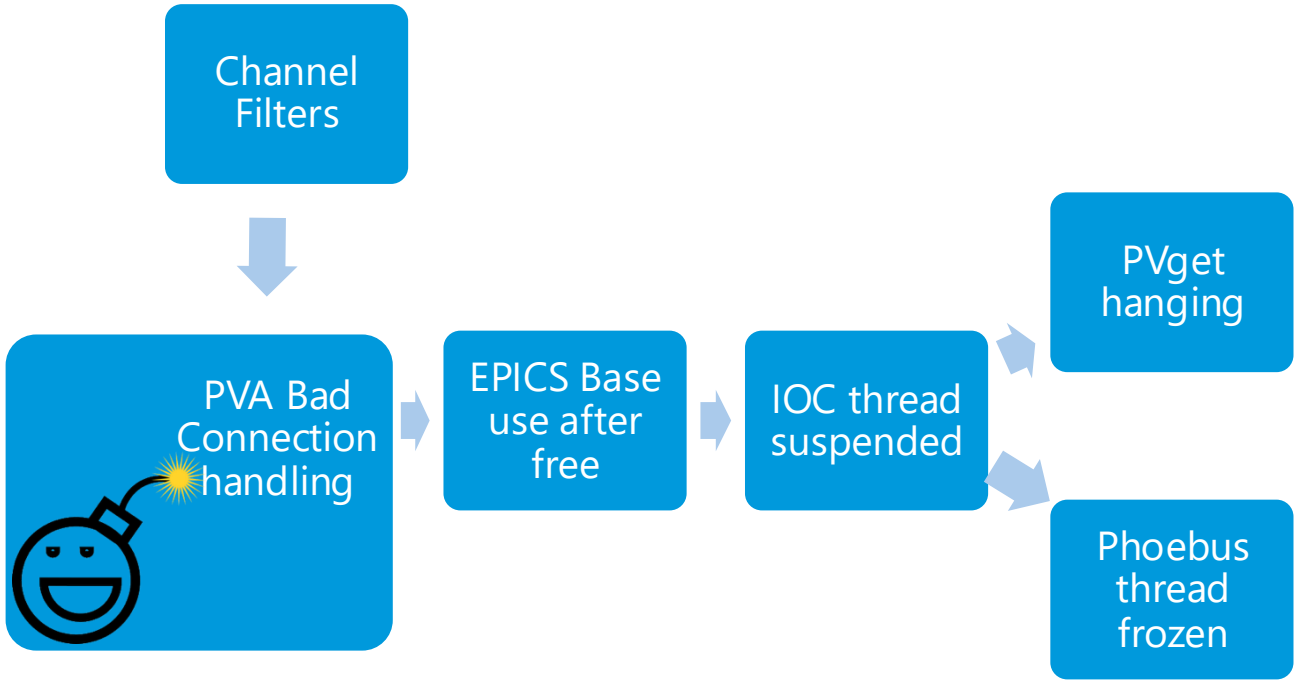
On the client

- UDP broadcast was reaching the IOC and the reply was reaching the WS

- the WS initiated a TCP SYN but the IOC never sent a TCP SYN+ACK



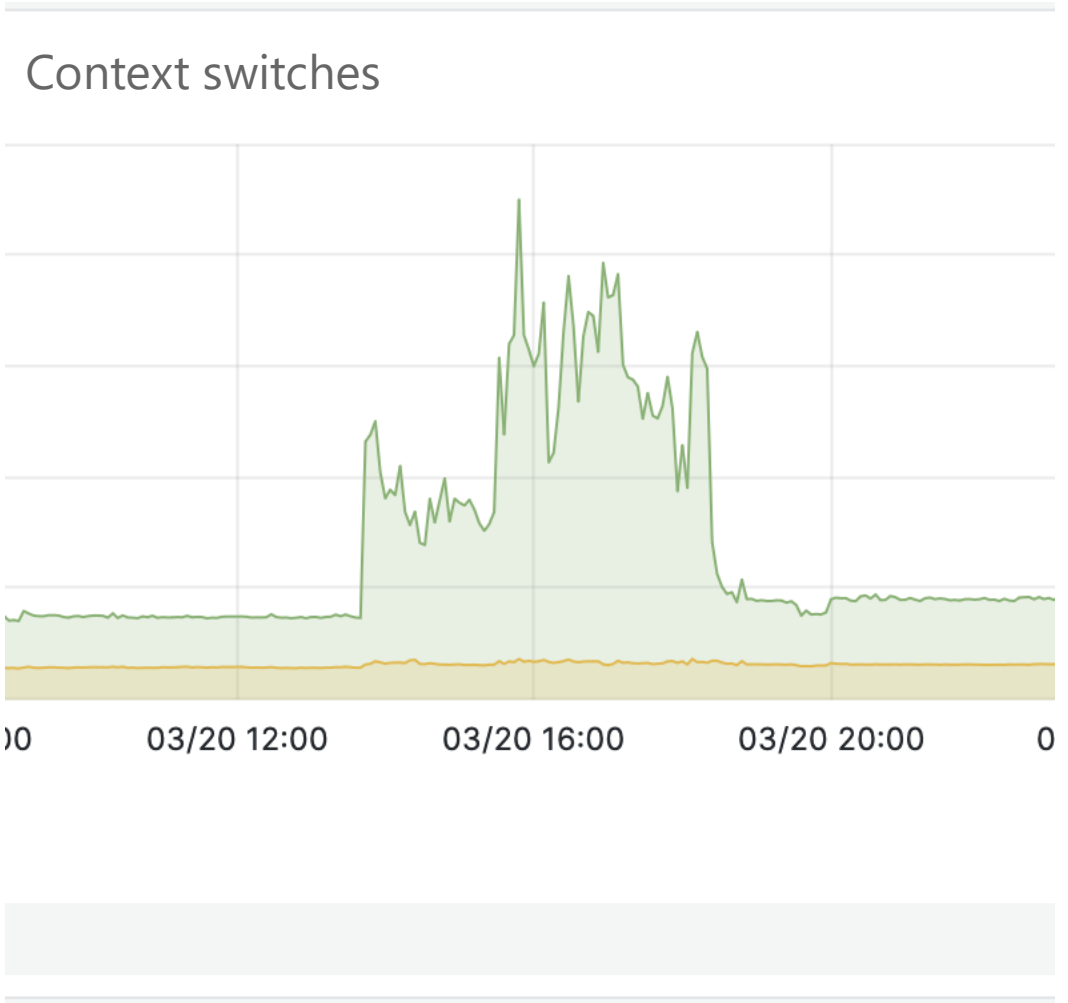
A perfect storm



Solutions Ideas in phoebus

TCP connection same thread for all PVs

- New thread per creation of a connection?
  - Heavy
- New Virtual thread per connection?
  - Requires migrating all of phoebus to Java 21
- Reduce the timeout?
  - Set up to be configurable



Solutions

Working together

EPICS

CS|studio

Discussion on a fix for one filter in epics-base

PVXS determined as the main solution

Investigation of using Virtual Threads

Investigation of thread management for new TCP Connections with existing thread handlers

Everything seems to be working fine...

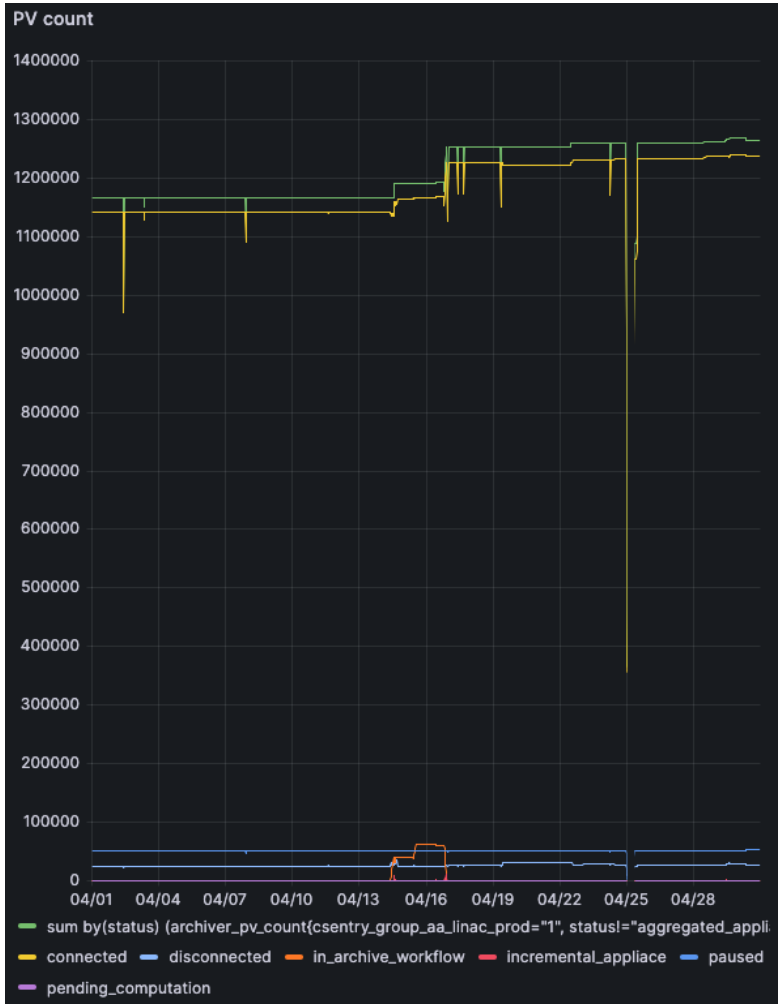
Then rolls another Friday afternoon...

Archiver blows up

Reports of PVs being “connected” in the archiver but not archiving?

What happened recently? 90 000 PVs were added to the Archiver using pvAccess.

Restarting affected IOCs doesn’t fix the issue.

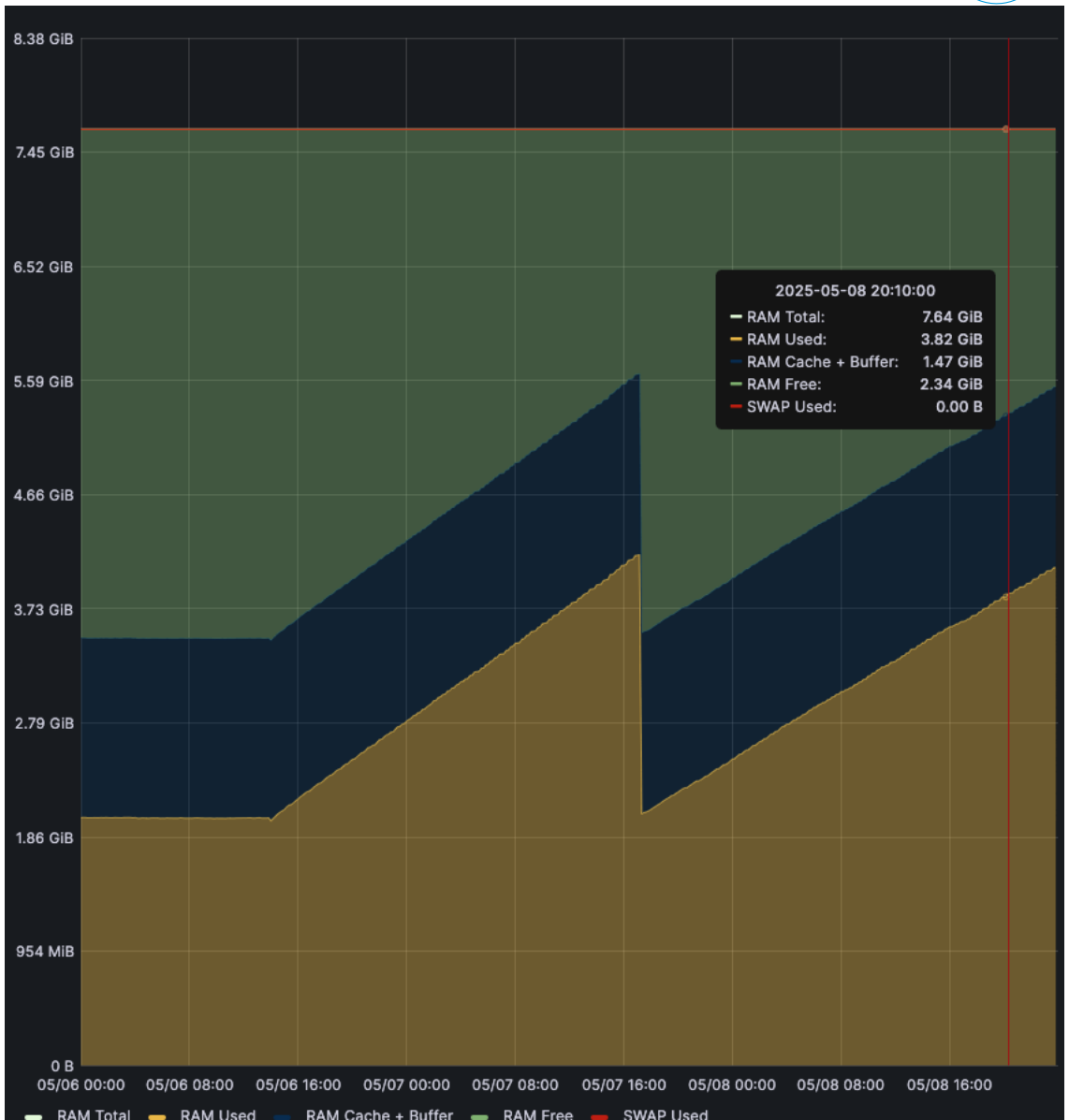


The rabbit hole

The Archiver Appliance engines had run out of memory.

Leading to logs of OutOfMemoryError and a drop of TCP connections.

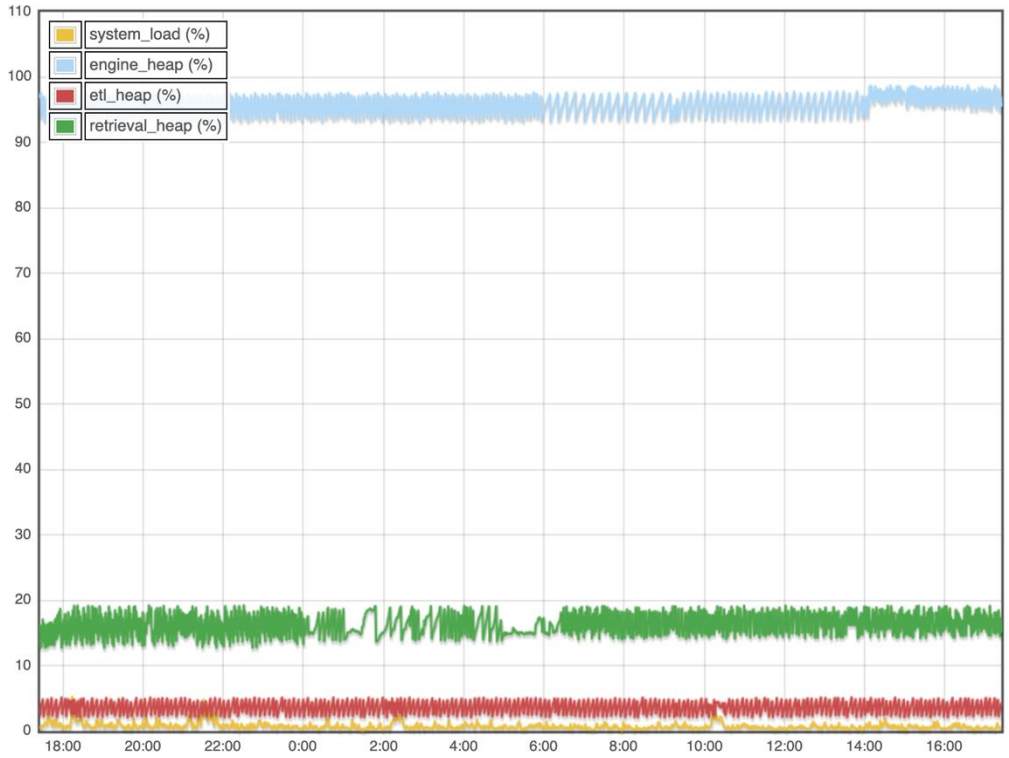
PVXS IOCs unaffected, but PVA IOCs creating zombie threads increasing IOC memory usage



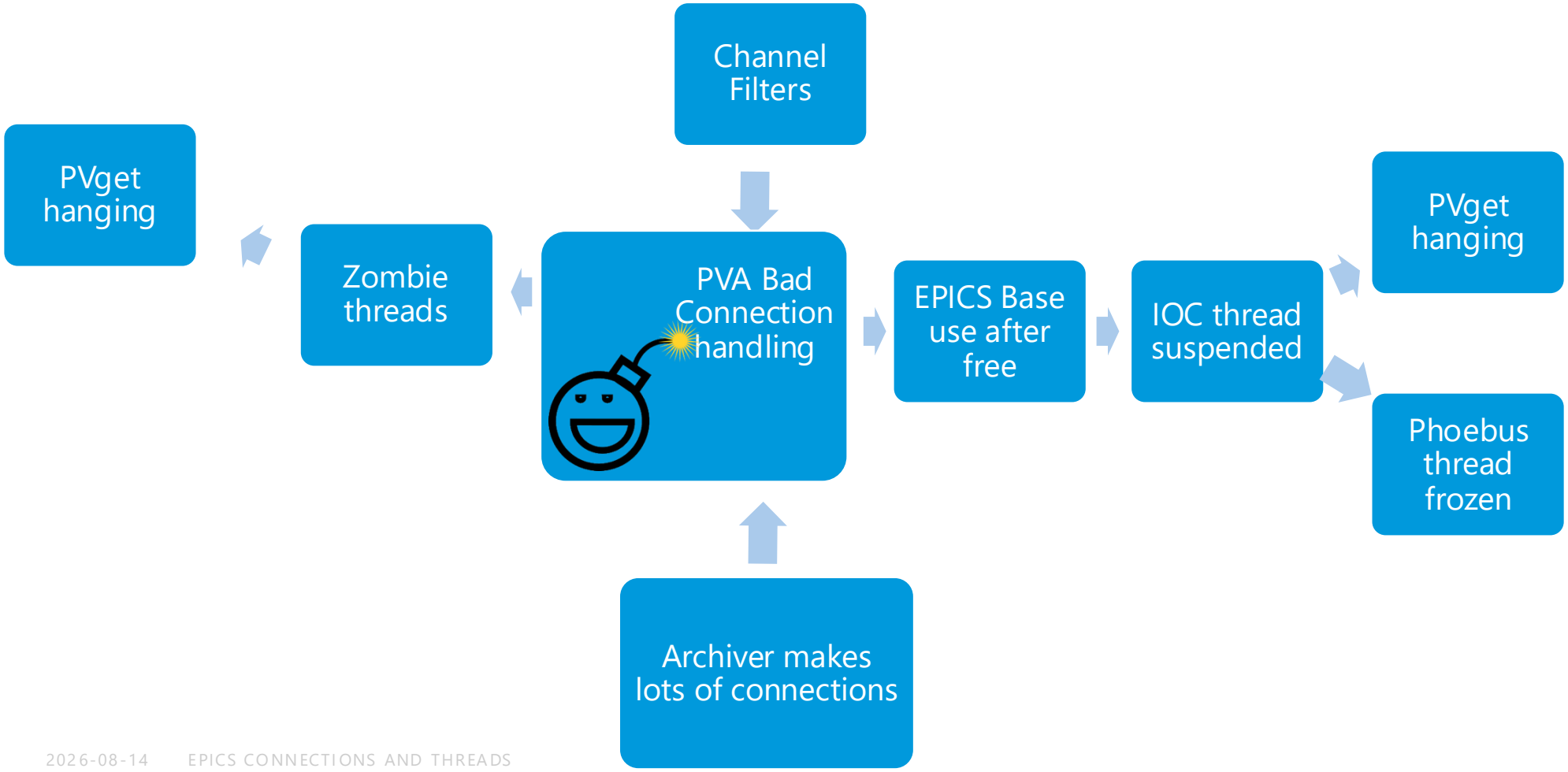
Short term fixes

Archiver Appliances restarted with more memory

Some PVs swapped to Channel Access in the Archiver



A perfect storm continued



In Progress Solutions

Archiver Appliance — ARCHIVE MILLIONS OF PVS

Latest Archiver Appliance (above 2.0.9) with new core-pva library found to be considerably more memory efficient.

i.e. 2GB memory engine handles 8000 vs 45 000 pva PVs.

e3

Next e3 release (ESS built version of epics) will use pvxs (QSRV 2) to replace the PVA stack for all IOCs at ESS.

cs|studio

Fix in phoebus core-pva library is finalized

Lessons Learned

PVXS rocks! Thanks Michael

Don’t run java applications at over 100% memory usage…

Core-pva also rocks! Thanks Kay

Scale produces problems everywhere you don’t expect

Multi team task forces enable quick solutions and debugging

Most of all, thank you all

ESS Infra:

Stephane Armanet

ESS Software:

Abraham Wolk

Simon Rose

Anders Lindh Olsson

ESS ICS-HWI:

João Paulo Martins

Lucas Magalhães

Mateusz Nabywaniec

Karl Vestin

Outside of ESS

Michael Davidsaver

Kay Kasemir

And I'm sure more.

Collaboration and Community works.

We have four implementations of pva

Lots of people willing to come in to help and debug

Questions