Acquisition Services Improvements and Fixes

Implemented since December 2022

Marcio Donadio / Control Systems Eng, / TID-CDS-ACS

04/25/2023

SLAC NATIONAL ACCELERATOR LABORATORY

Stanford University | U.S. DEPARTMENT OF ENERGY

Outline

Introduction

CATERS 158662 and 160141

BSA pointer trap

ATCA crate overview

Reason for crashes and implemented fixes

Issues solved since December 2022

Implementation of new features

Implemented since December 2022

Stress test with fault buffers

Focus of the tests

Exercising fault buffers

The dimension of the load

Exercising BSA user buffers

Stress test with BLD

General View

A Few Problems

Focus of the tests

Stress test with BSAS

General View

Calculation of the number of rows

Focus of the tests

External network load with camonitor

Python script to handle camonitor

Results

First scenario

Adding one IOC at a time

BLD has high load

Curious side effect used as a tool

Testing IOCs with a single ATCA crate

Comparison between 2 ATCA vs 1

BSAS can use a lot of resources, too

camonitor challenged lcls-srv01

Known issues

BPMs still has problem

BSAS no protect. for excess of rows

Conclusions and recommendations

IOCs are now stable

BLD plan not feasible w/ curr. CPUs

One CPU with to 2 ATCAs = fragility

Testing in production is a nightmare

1

Introduction

Introduction

Comments

~~Acknowledgements~~

~~Show current deployment: CPUs can have 1 ATCA, but can have 2. Mention 14.~~

~~Show EPICS network from the CPU.~~

~~Mention that conn with ATCA is I/O.~~

~~Describe that in the crate there's BPM, MPS, BLEN, etc.~~

~~FPGAs are fast and delivers a lot of data (analogy with hoses).~~

~~Interconnect (Ethernet between CPU and ATCA).~~

Consistency for nomenclature (sometimes it is CPU, sometimes LinuxRT).

Recommend a study on lcls-srv0x. Maybe it will become a bottleneck.

Section 2 is too technical. Focus on CATERS and high-level with cartoons. Send these slides to the backup.

Tests in amcCarrierTprTst in the past didn't catch problems.

Test in production is a requirement.

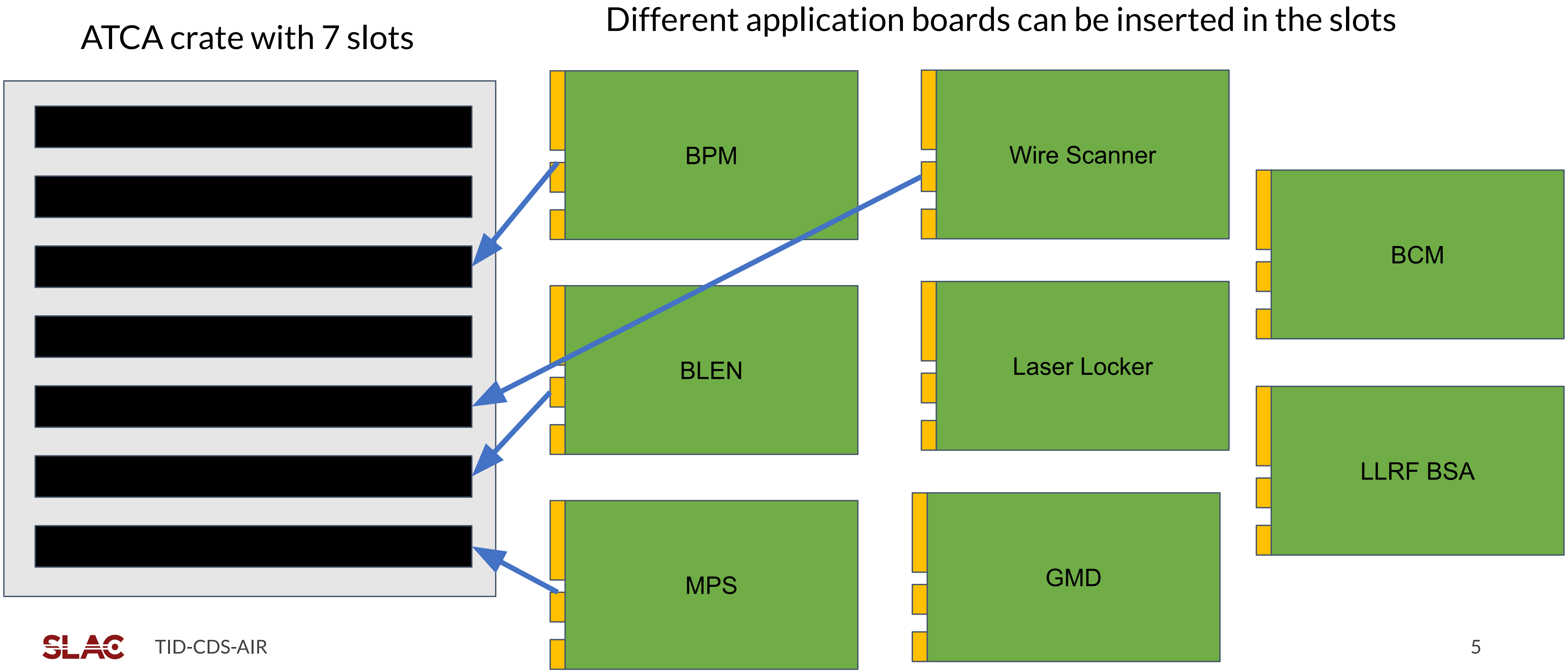
Not use the word nightmare.

Mention functional test.

~~Don't combine new releases with bug fixes.~~

Introduction

Current architecture

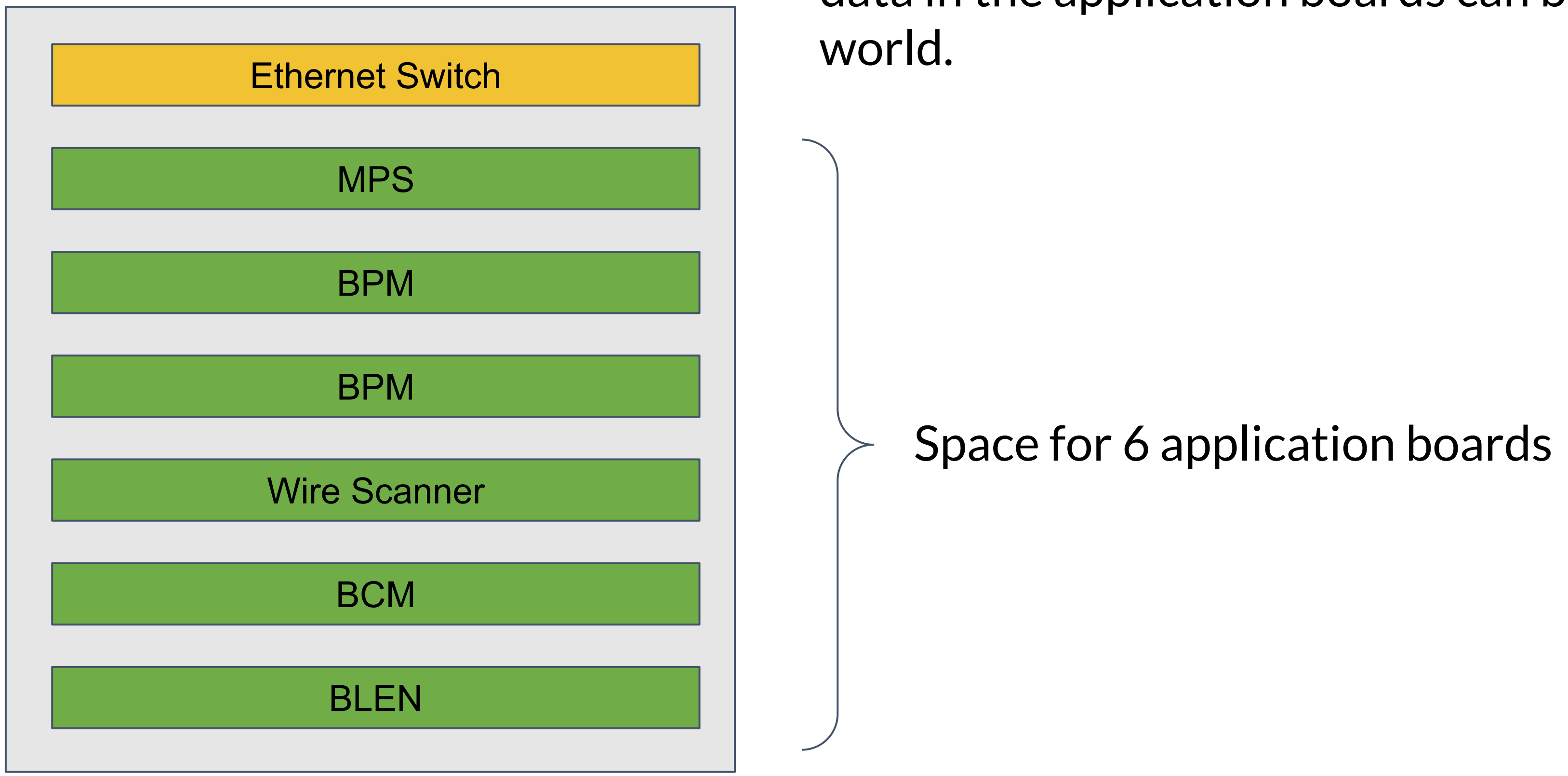


Introduction

Current architecture

ATCA crate with 7 slots

The first slot is special and receives an Ethernet switch so the data in the application boards can be sent out to the external world.



Ethernet Switch

MPS

BPM

BPM

Wire Scanner

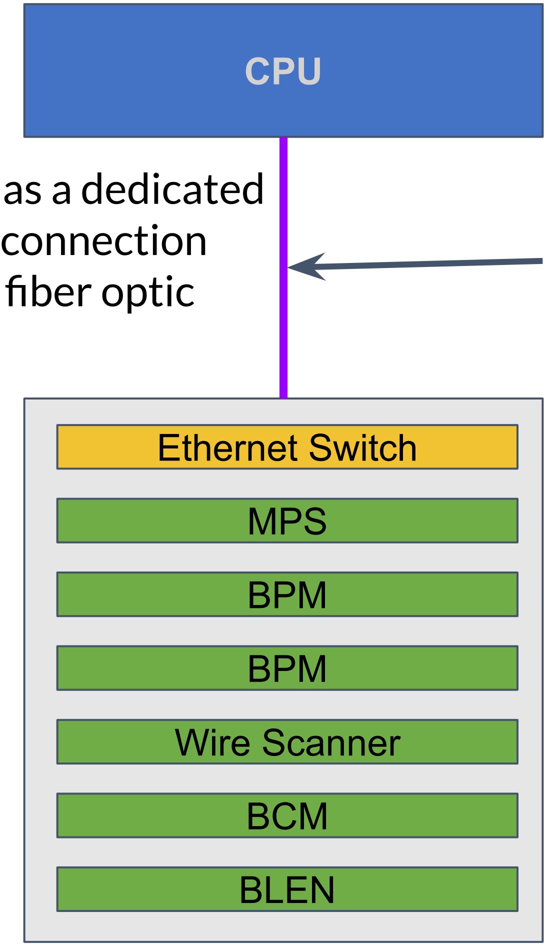
BCM

BLEN

Space for 6 application boards

Introduction

Current architecture



A CPU has a dedicated 10 Gb/s connection through fiber optic cable

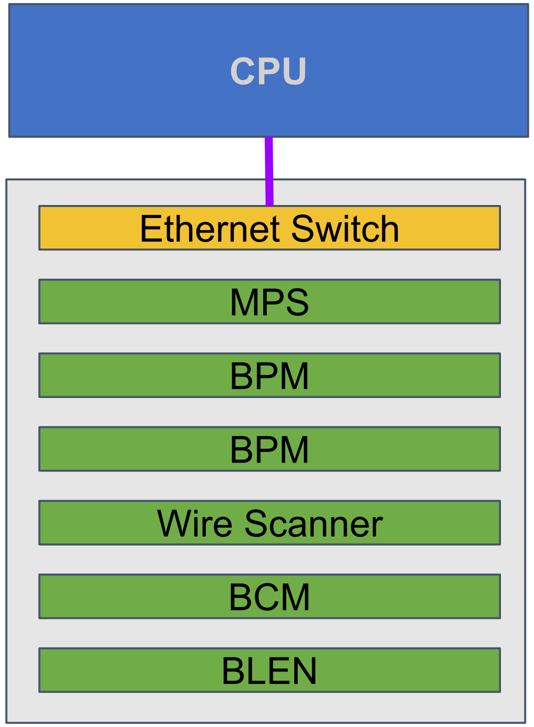
I/O Ethernet Interconnect

Introduction

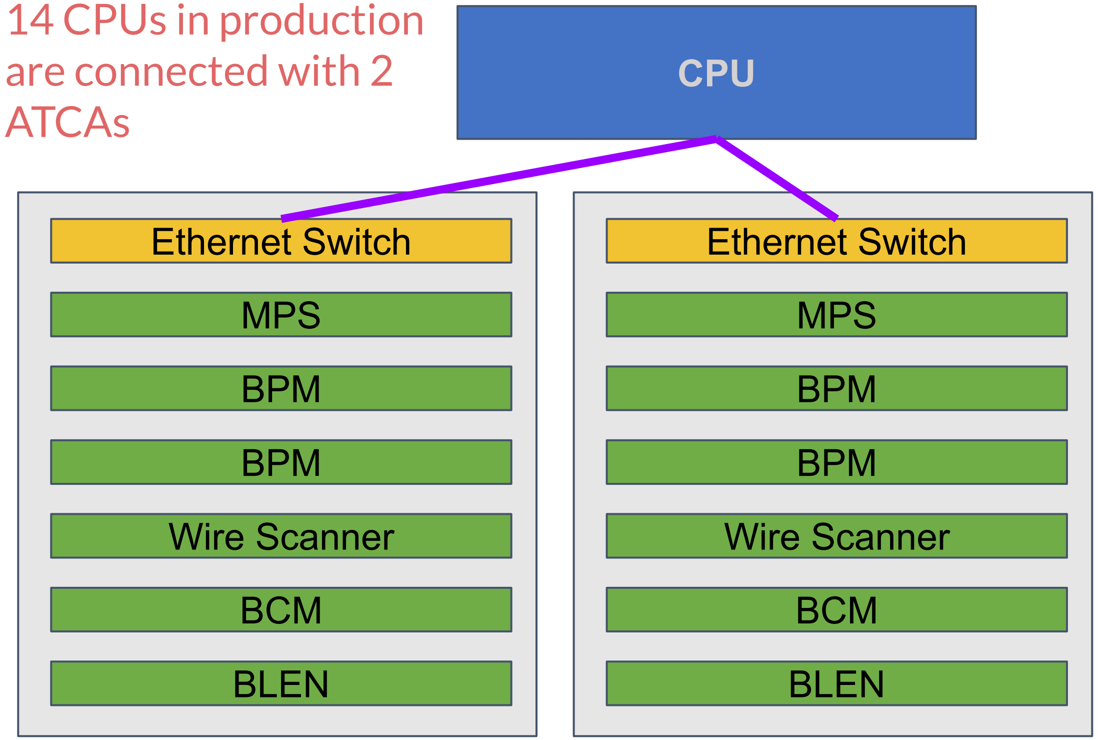
Current architecture

Different valid ways of connecting CPUs and ATCAs

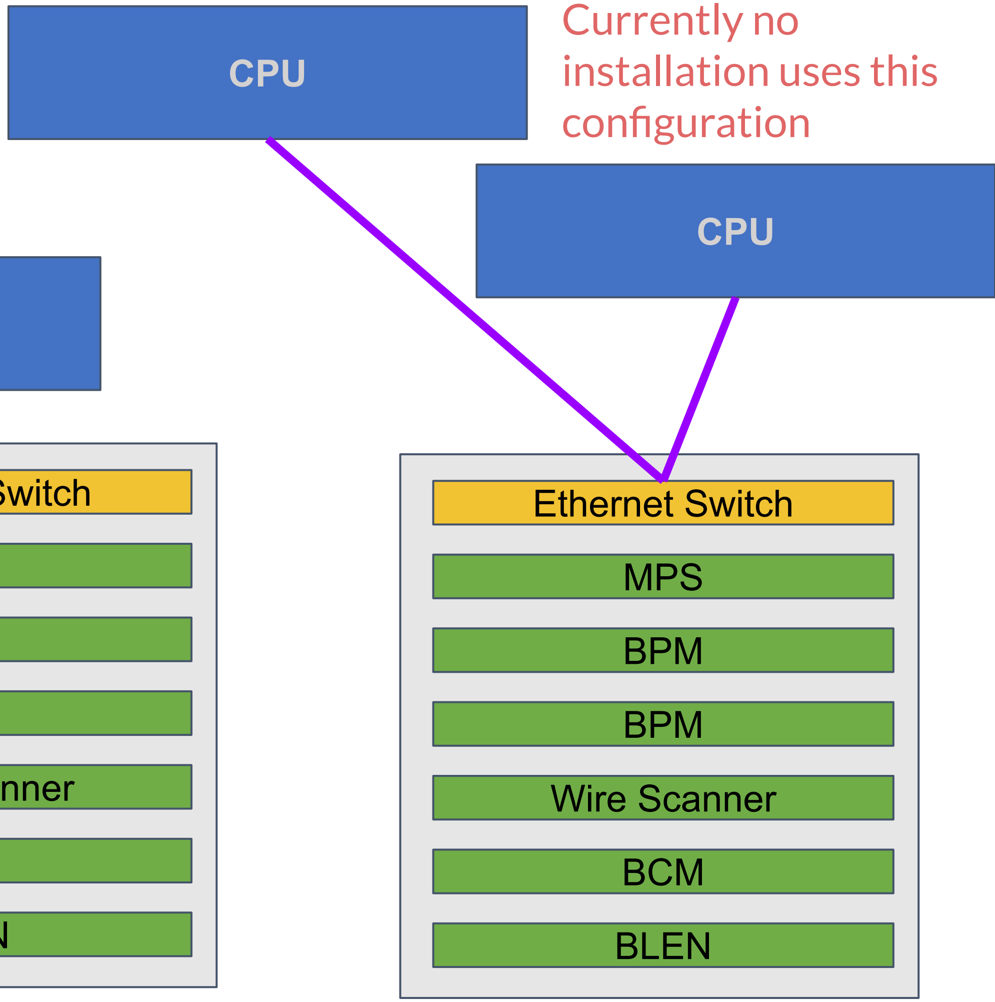
Almost all installations use one CPU with one ATCA



14 CPUs in production are connected with 2 ATCAs

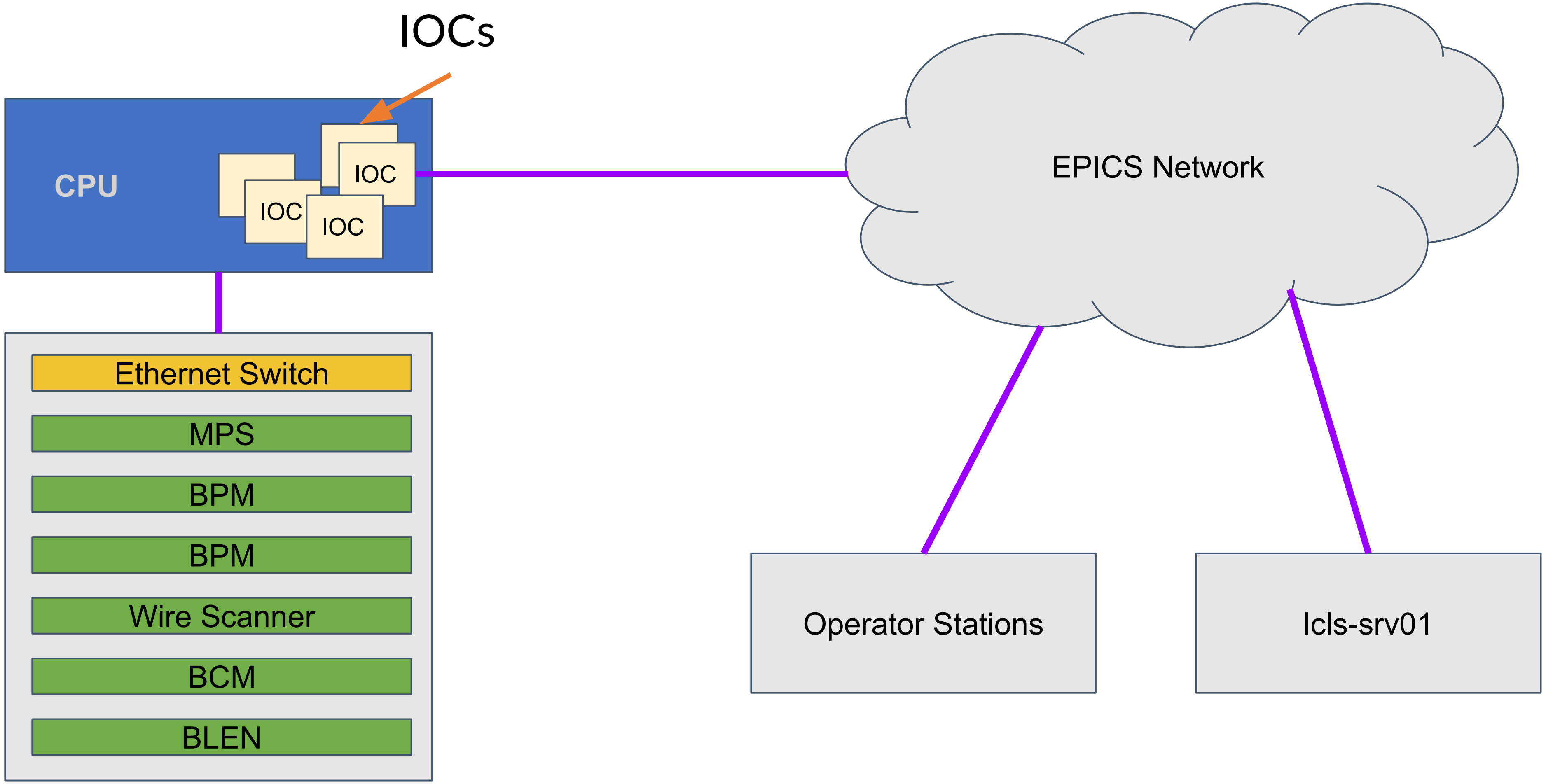


Currently no installation uses this configuration



Introduction

Current architecture

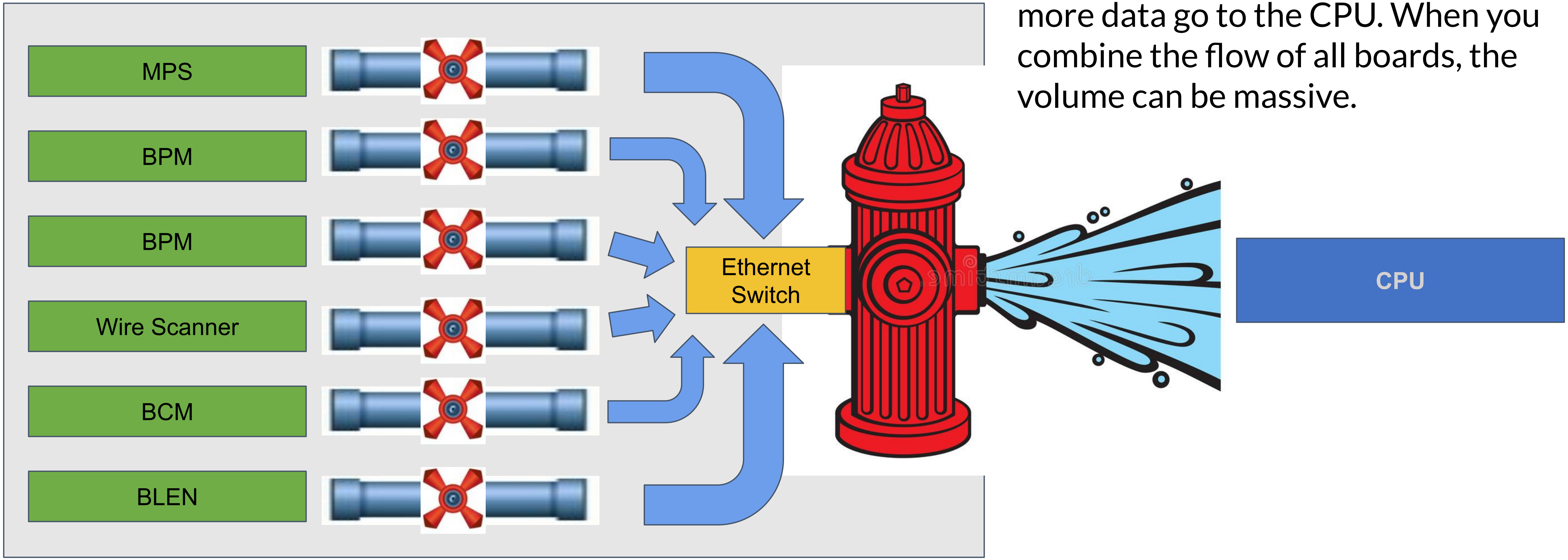


Introduction

Volume of data can be large

Data is provided by an FPGA in each board. FPGAs are fast!

Many triggers and configurations can open the "data faucet", making more data go to the CPU. When you combine the flow of all boards, the volume can be massive.



Introduction

Volume of data can be large

The software was not holding the amount of data when the "faucets" were opened too much.

These, plus a few bugs in software and firmware were causing the IOCs to crash frequently in production.

CATERS:

- 157205
- 158662
- 160141

Introduction

Acknowledgements

Kukhee Kim
Matt Weaver
Marcio Donadio
Larry Ruckman
Carolina Mattison
Sonya Hoobler
Jeremy Mock
Namrata Balakrishnan
Kyle Leleux
Ryan McClanahan
Leonid Sapozhnikov
Dawood Alnajjar
Michael Skoufis
Thuy Vu
Binh Nguyen
Minh Vu
Michael Zelazny
Ernest Williams
Kristi Luchini
Shawn Alverson
Timothy Maxwell

Thank you for these 21 people that helped one way or another to improve the Acquisition Services system.

2

Implemented fixes

Implemented fixes

Issues solved since December 2022

Bug fixes in firmware and software would make them to read the wrong memory region to retrieve data.

Better control of how much data should come out from each "data pipe". Keeping the analogy, we locked the water valve, not allowing it to open too much.

Better distribution of tasks among the CPU cores. Now multiple workers take care of the needed tasks to read data from the ATCA.

Made more efficient use of memory. We reduced ~3.6 GB of memory usage per IOC.

3

Implementation of new features

Implementation of new features

Rethinking the strategy

Additionally to the serious problems that we were observing in production, which needed months to fix, we had a list of new features to implement.

The implementation of the new features was made at the same time as the problems were being diagnosed and fixed.

A better approach would have been to bring the system to stability and, only later, to work on the new features:

- New features can distract the focus on the diagnostic because it is the same person who implements both.
- It may be difficult to tell if a test with negative results was due to a previous problem or a new one created with the new feature implementation.

Implementation of new features

Implemented since December 2022

BSA buffer allocation changes: more system buffers, less user buffers

- 29 user buffers.
- 15 system buffers.
- Changes in timing/bsa API and bsaDriver, with database template changes.

Merging BSSS timing filtering control with BSA

- Previously BSSS supported fewer timing filters than BSA. Now there's a perfect match between BSSS and BSA.
- Firmware had to instantiate two BSSS modules to support the same number of filters as BSA.
- Implementation of an abstraction in bsaDriver to hide the two BSSS modules.

Implementation of new features

Implemented since December 2022

Make the NORD field in the BSA HST PV and Fault Buffer PV work with camonitor so we can use NORD to check up the progress of BSA and Fault buffer

- The waveform device support in the asyn port driver doesn't support the CA monitor event for the NORD field.
- This complicates checking up the progress or completion of the BSA buffer and fault buffer.
- Since we've switched to using an own device support from the asyn port driver, we implement the CA monitor event for the NORD.

SC_BSA_ENABLE PV

- IOC engineer can turn on and off the poll thread for SC BSA.
- It can be used to turn off SC BSA mode when operating with NC beam.
- We don't believe that turning the polling off is significant because it just reads 8-bytes every 100 ms. Nevertheless, the feature is available for those interested.

Implementation of new features

Implemented since December 2022

SC_BSA_STATUS and SC_BSA_FLTCNT PVs

- The bsaDriver has a tiny state machine to handle the BSA fault and the polling cycle. SC_BSA_STATUS PV provides the state information FAULT and NORMAL with severity Major and No Alarm.
- The SC_BSA_FLTCNT provides the count value of state transition from Normal to Fault. These PVs can help monitoring BSA status for large number of IOCs.

README was completely refactored

- Concepts of what is each acquisition service are described.
- Each acquisition service has its own README now.
- Example of an st.cmd to instantiate all acquisition services.
- Please, read them!

Implementation of new features

Implemented since December 2022

Massive change in PV names after a long work with the naming Czars.

4

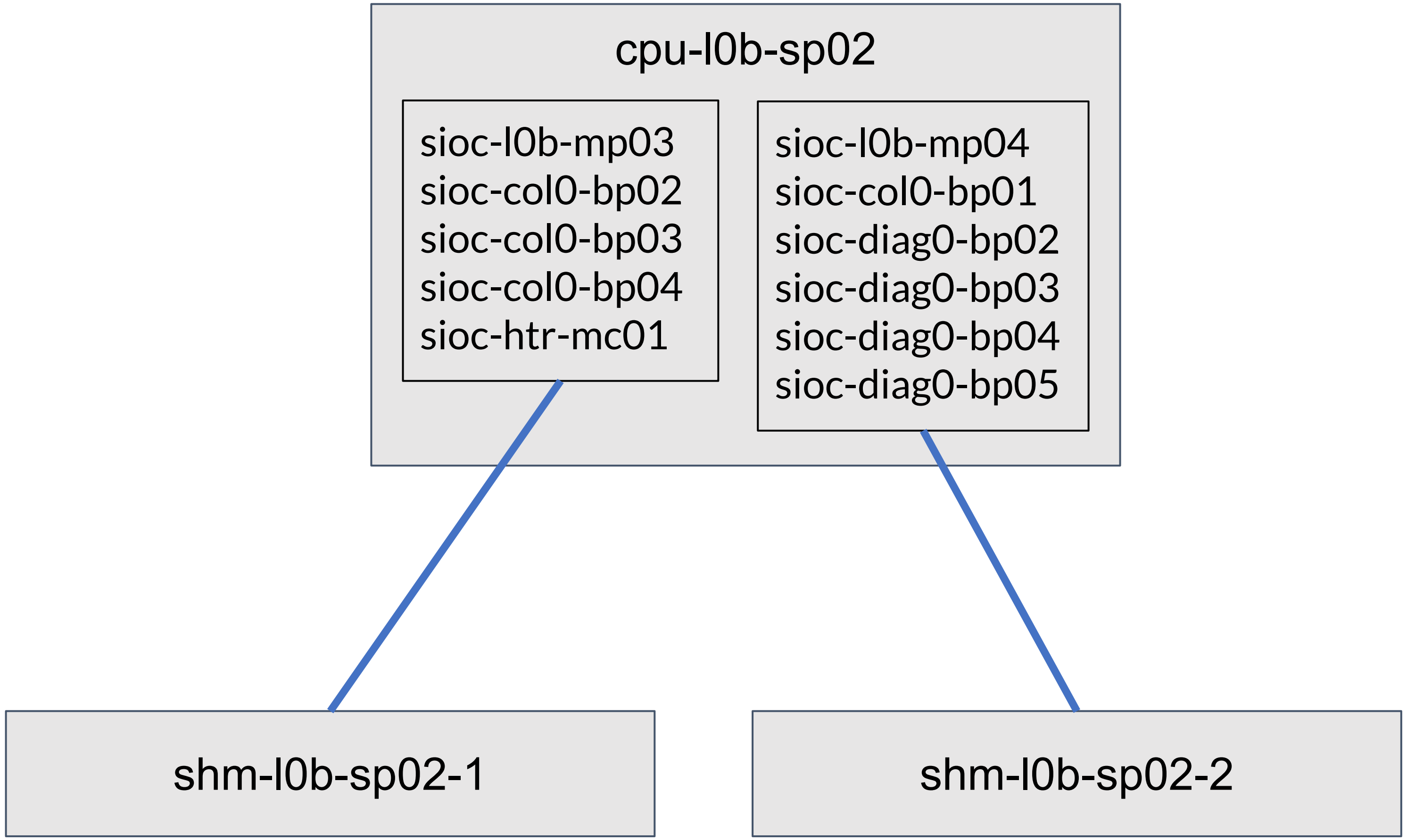
Stress test with fault buffers

Stress test with fault buffers

Focus of the tests

11 IOCs running in cpu-l0b-sp02 (production area):

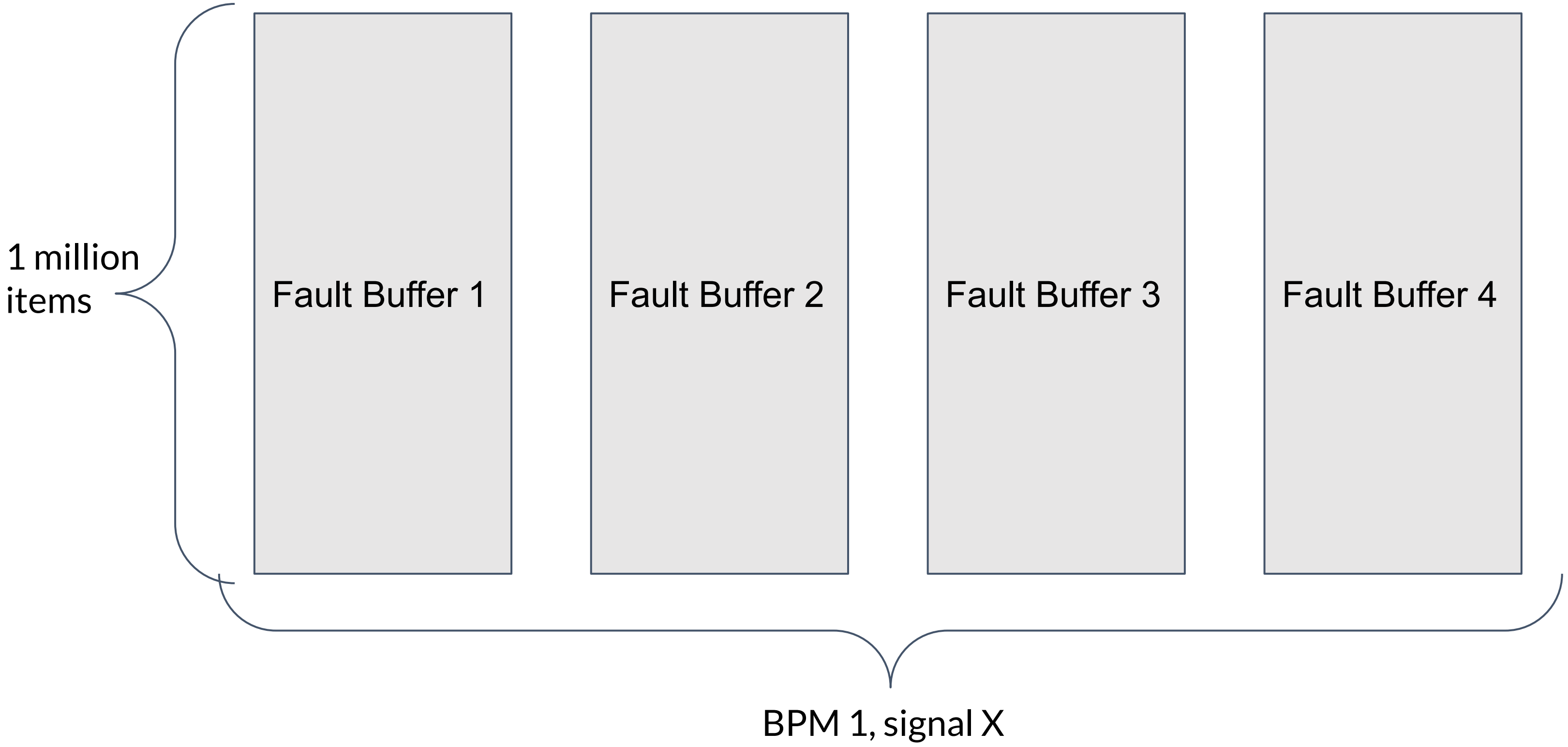
- 2 MPS
- 8 BPMs
- 1 Wire Scanner



Stress test with fault buffers

Exercising fault buffers

Each signal has 4 fault buffers, with 1 million items each.

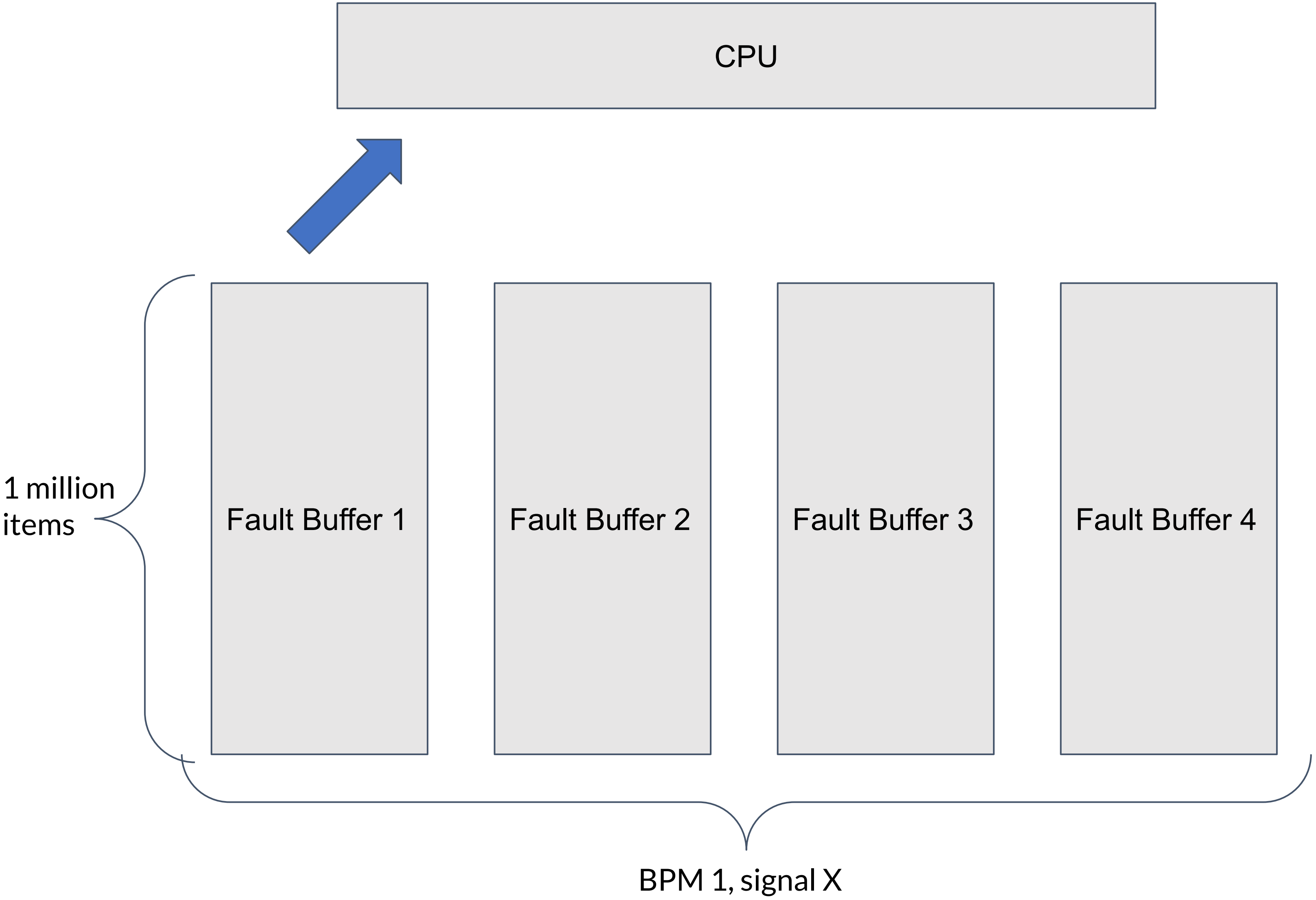


Stress test with fault buffers

Exercising fault buffers

Each signal has 4 fault buffers, with 1 million items each.

When a fault happens, one of the buffers fill and is transferred to the CPU.



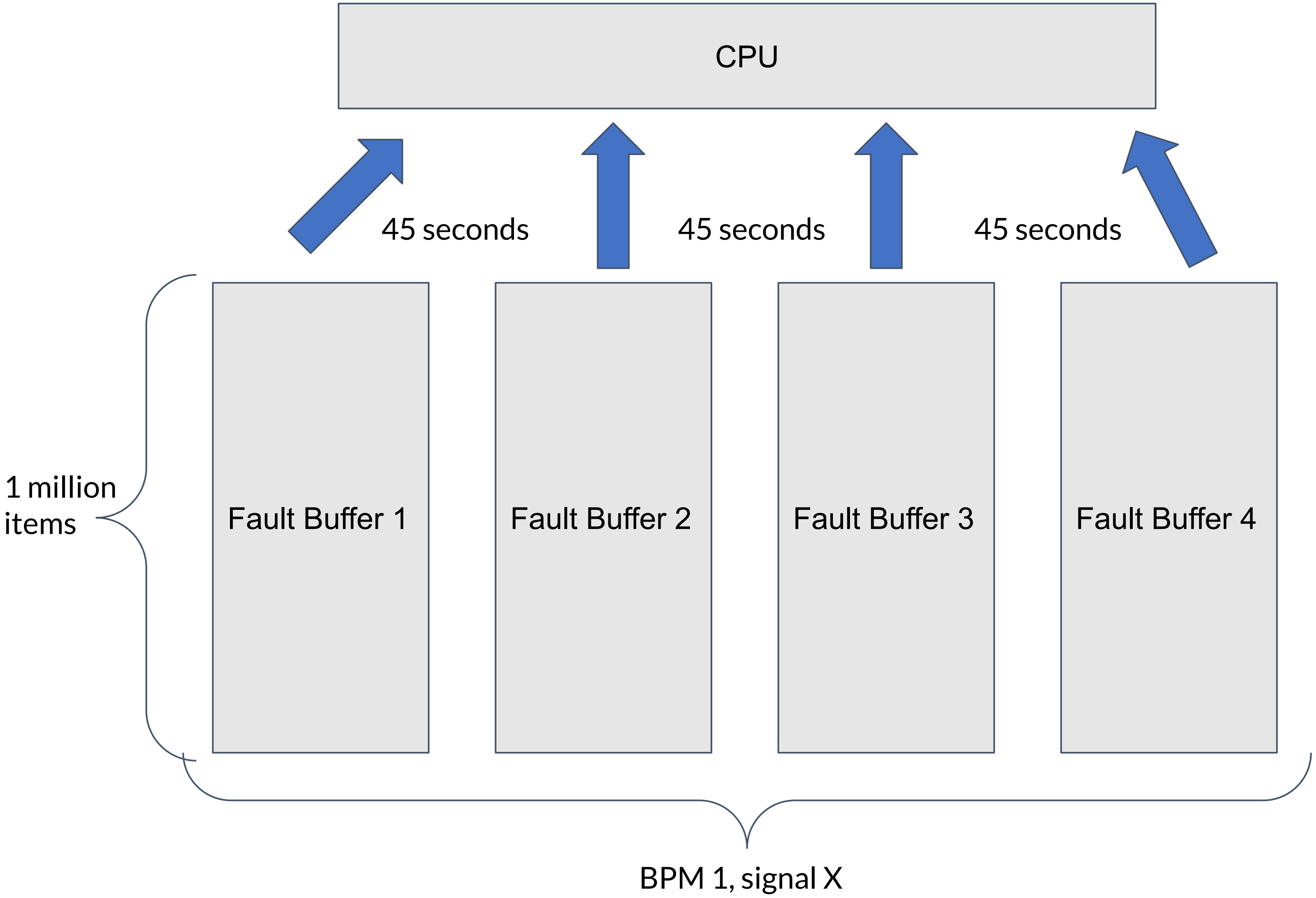
Stress test with fault buffers

Exercising fault buffers

Each signal has 4 fault buffers, with 1 million items each.

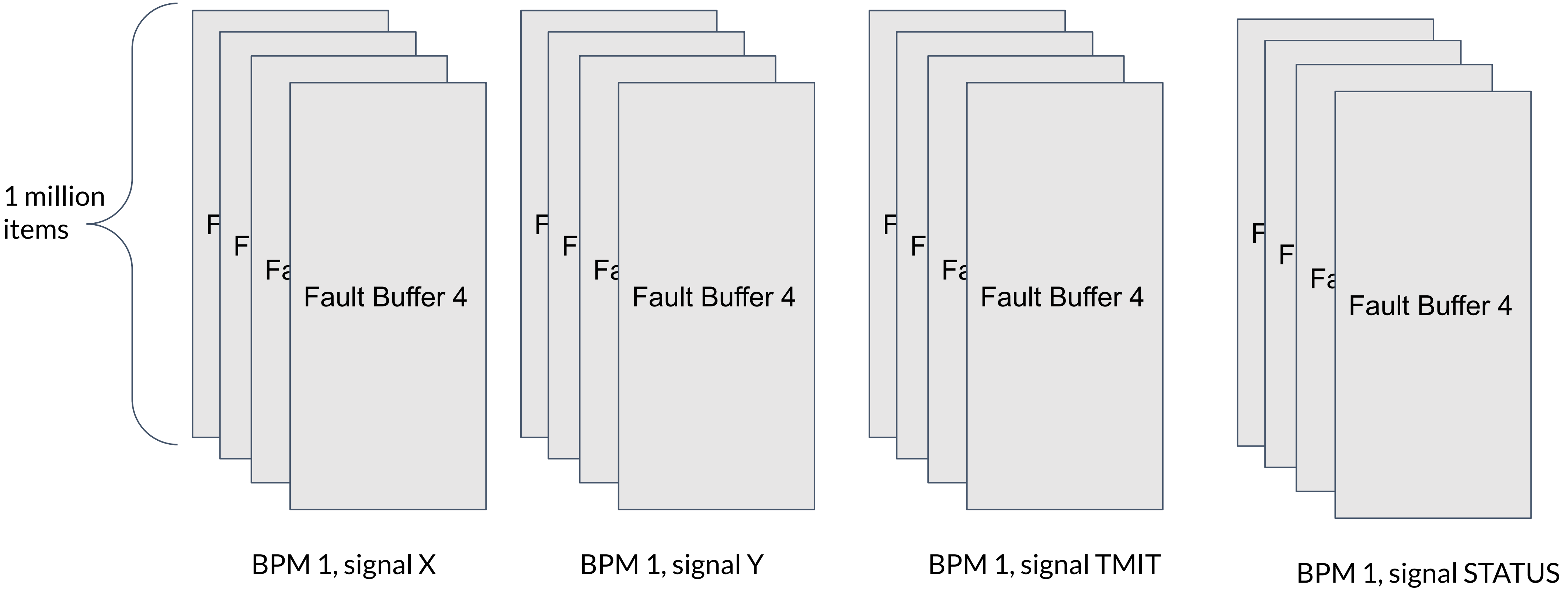
When a fault happens, one of the buffers fill and is transferred to the CPU.

A script was created to force a fault every 45 seconds and, when the 4 buffers are full, clean them.



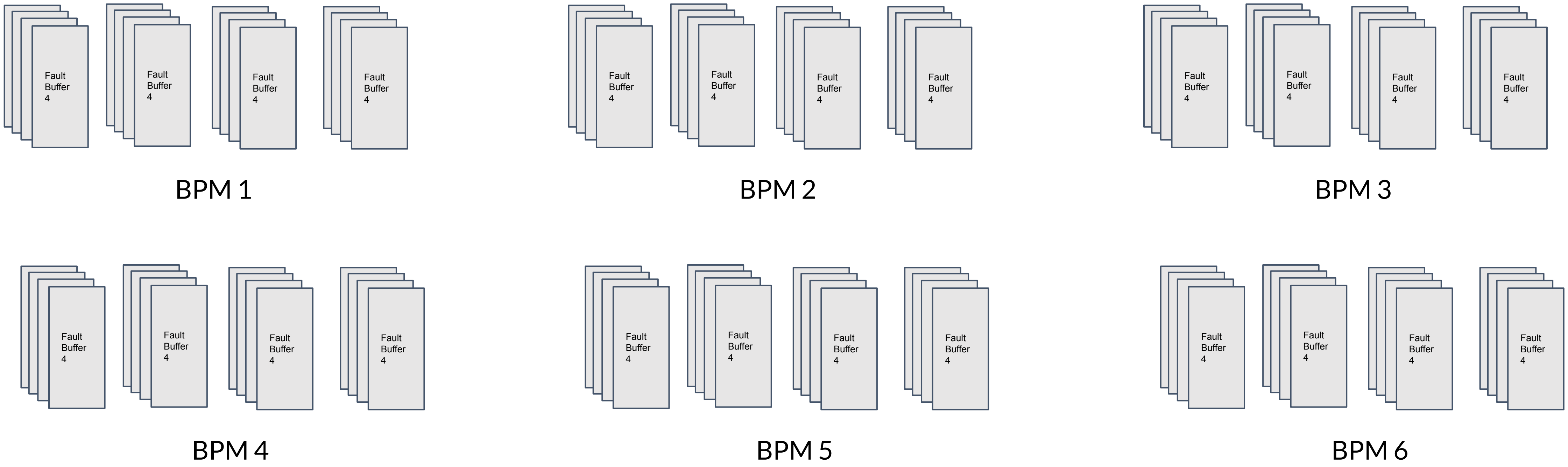
Stress test with fault buffers

The dimension of the load



Stress test with fault buffers

The dimension of the load



Fault Buffer 4 Fault Buffer 4 Fault Buffer 4 Fault Buffer 4

BPM 1

Fault Buffer 4 Fault Buffer 4 Fault Buffer 4 Fault Buffer 4

BPM 2

Fault Buffer 4 Fault Buffer 4 Fault Buffer 4 Fault Buffer 4

BPM 3

Fault Buffer 4 Fault Buffer 4 Fault Buffer 4 Fault Buffer 4

BPM 4

Fault Buffer 4 Fault Buffer 4 Fault Buffer 4 Fault Buffer 4

BPM 5

Fault Buffer 4 Fault Buffer 4 Fault Buffer 4 Fault Buffer 4

BPM 6

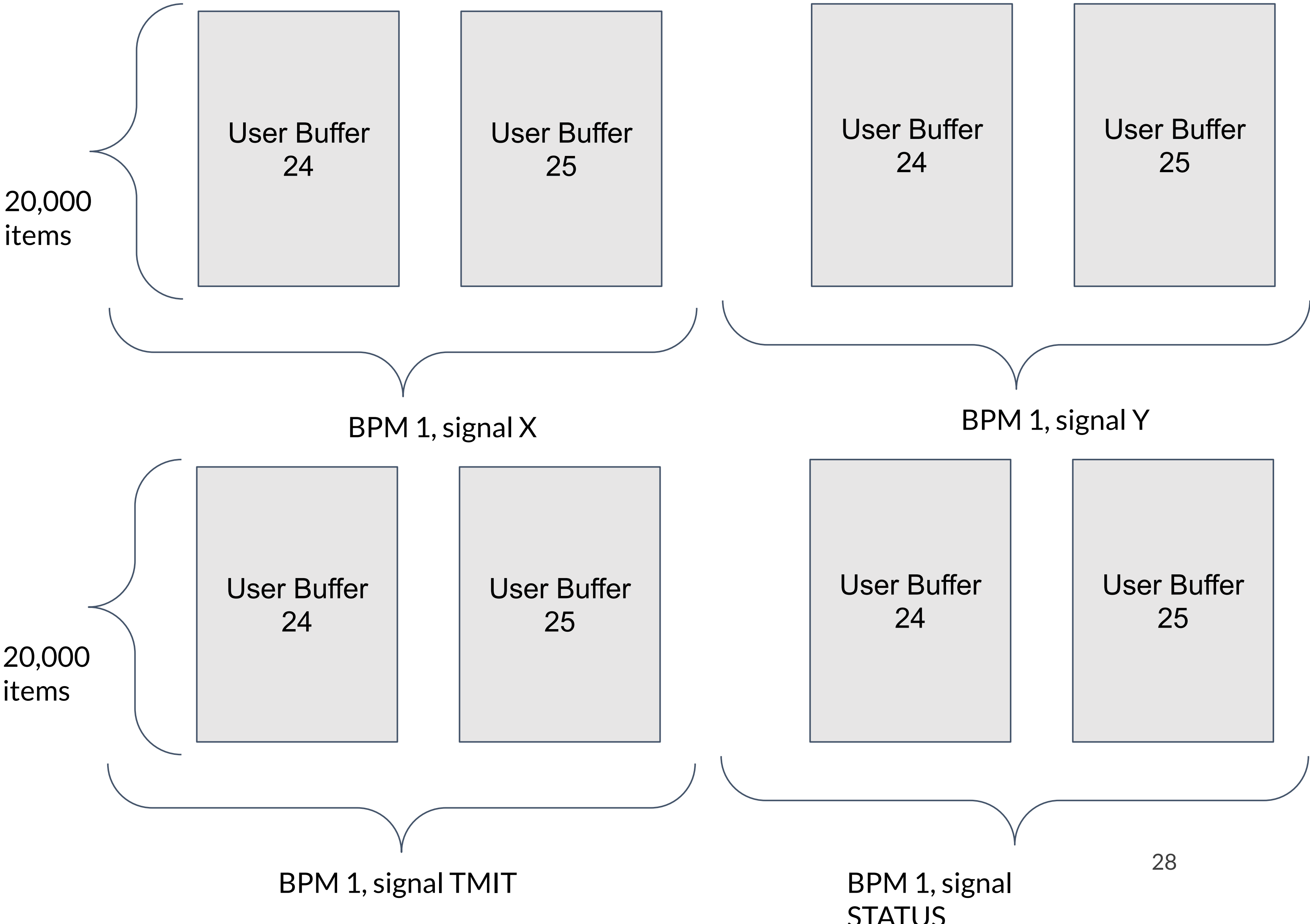
etc.

Stress test with fault buffers

Exercising BSA user buffers

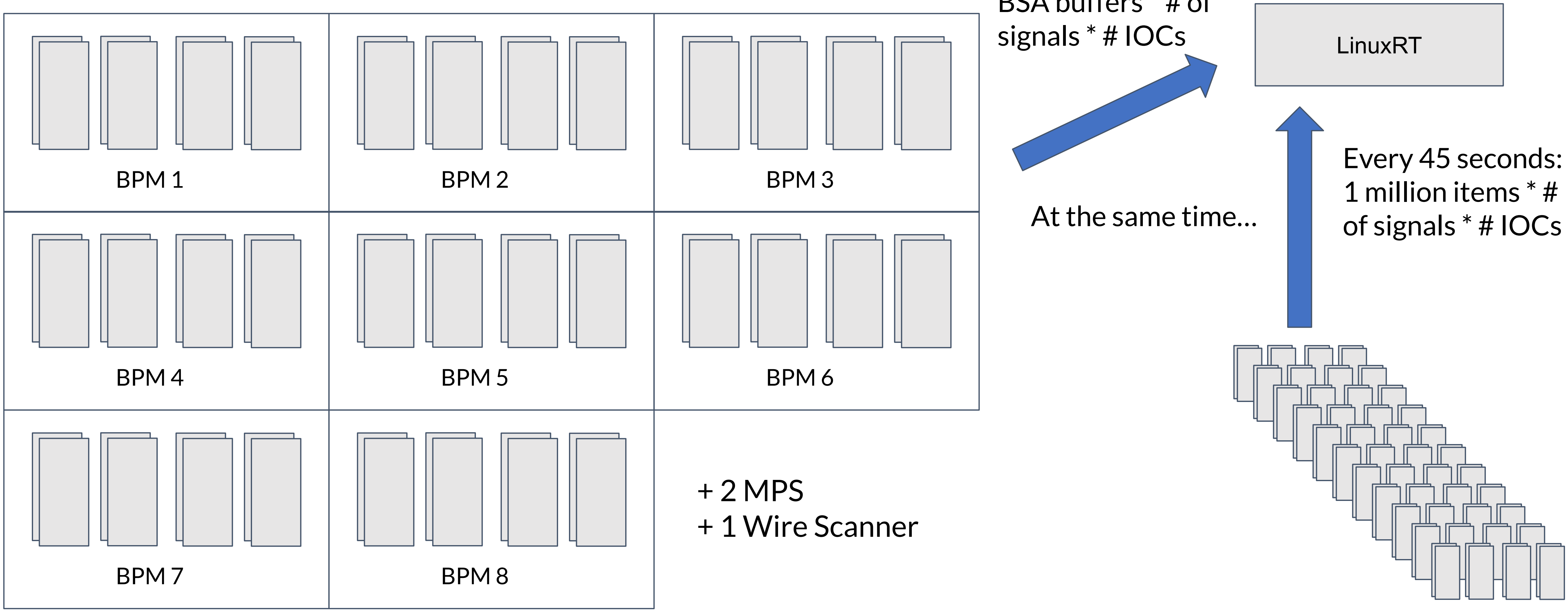
Not happy with flooding the system with fault buffers, we decided to configure 2 BSA user buffers:

- 1 MHz fixed, no destination, no averaging.
- Use all 20,000 positions available to each waveform array.
- Start an acquisition every 0.5 seconds, while the software is still reading the fault buffers.



Stress test with fault buffers

The dimension of the load



Every 0.5 seconds: 20,000 items * 2 BSA buffers * # of signals * # IOCs

LinuxRT

At the same time…

Every 45 seconds: 1 million items * # of signals * # IOCs

BPM 1

BPM 2

BPM 3

BPM 4

BPM 5

BPM 6

BPM 7

BPM 8

+ 2 MPS
+ 1 Wire Scanner

5

Stress test with BLD

Stress test with BLD

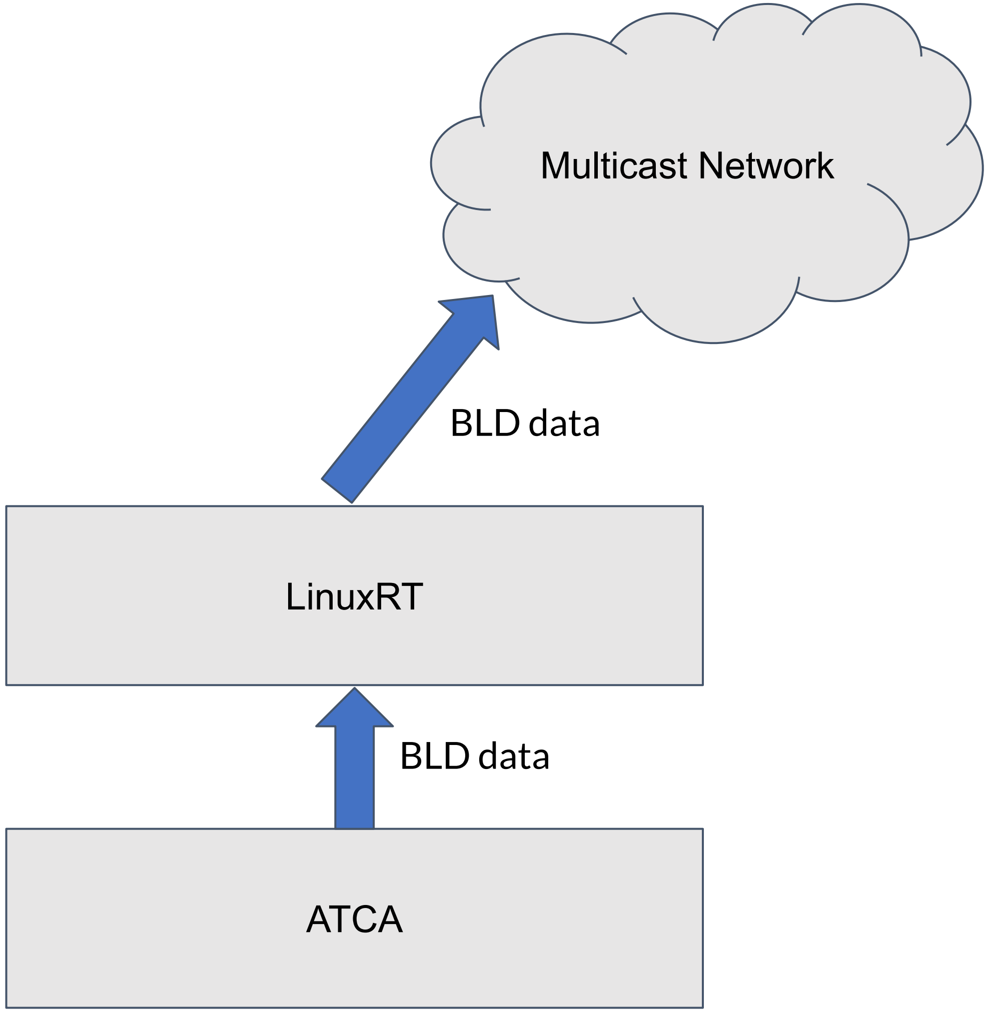
General View

BLD doesn't show data through PVs.

The IOC sends the data directly to the network and a client at the other side can subscribe to receive the data.

BLD must run at full rate as a requirement. In other words, bunch by bunch data needs to go to the network.

This requirement is already challenging enough for a stress test, so no additional scenario was tested beyond the default.

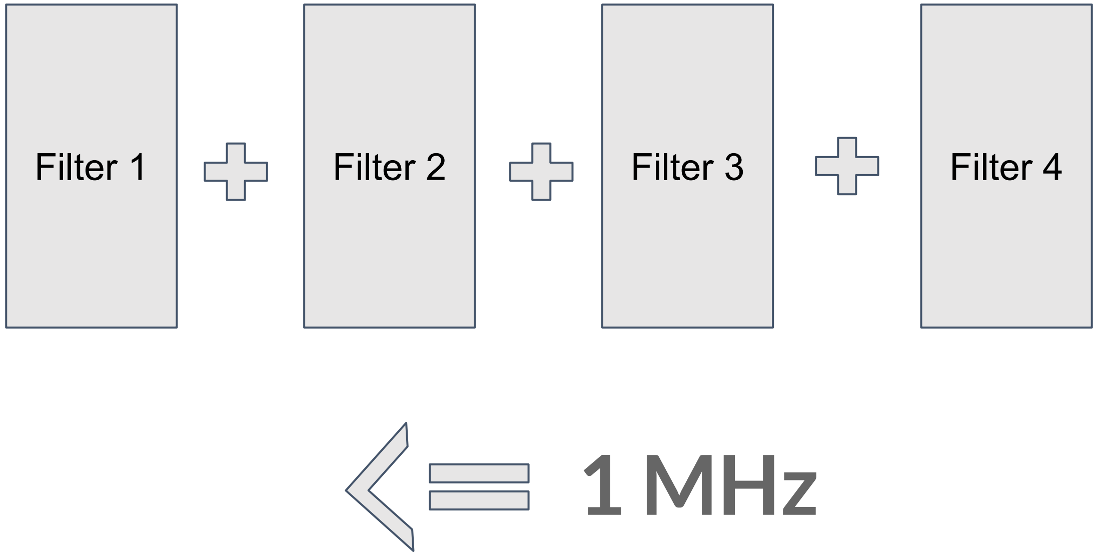


Stress test with BLD

General View

BLD provides 4 different timing filters.

The current requirements asks for mutual exclusive destinations in the filters, so we would never have more than 1 MHz worth of data to transport from ATCA to CPU to network, per IOC.



Currently these are the filter definitions:

- Fixed 1 MHz, destination SXR.
- Fixed 1 MHz, destination HXR.
- Not used.
- Not used.

Stress test with BLD

A Few Problems

The firmware, by default, assumes that the link between LinuxRT and ATCA supports jumbo frames.

- The firmware can be configured to send smaller packets.
- Doing so, the amount of transactions increased a lot and the system got too busy.
- When configuring jumbo frames, the system could handle more data, so I decided to keep it.

Couldn't send data outside the CPU because the network wasn't configured for jumbo frames.

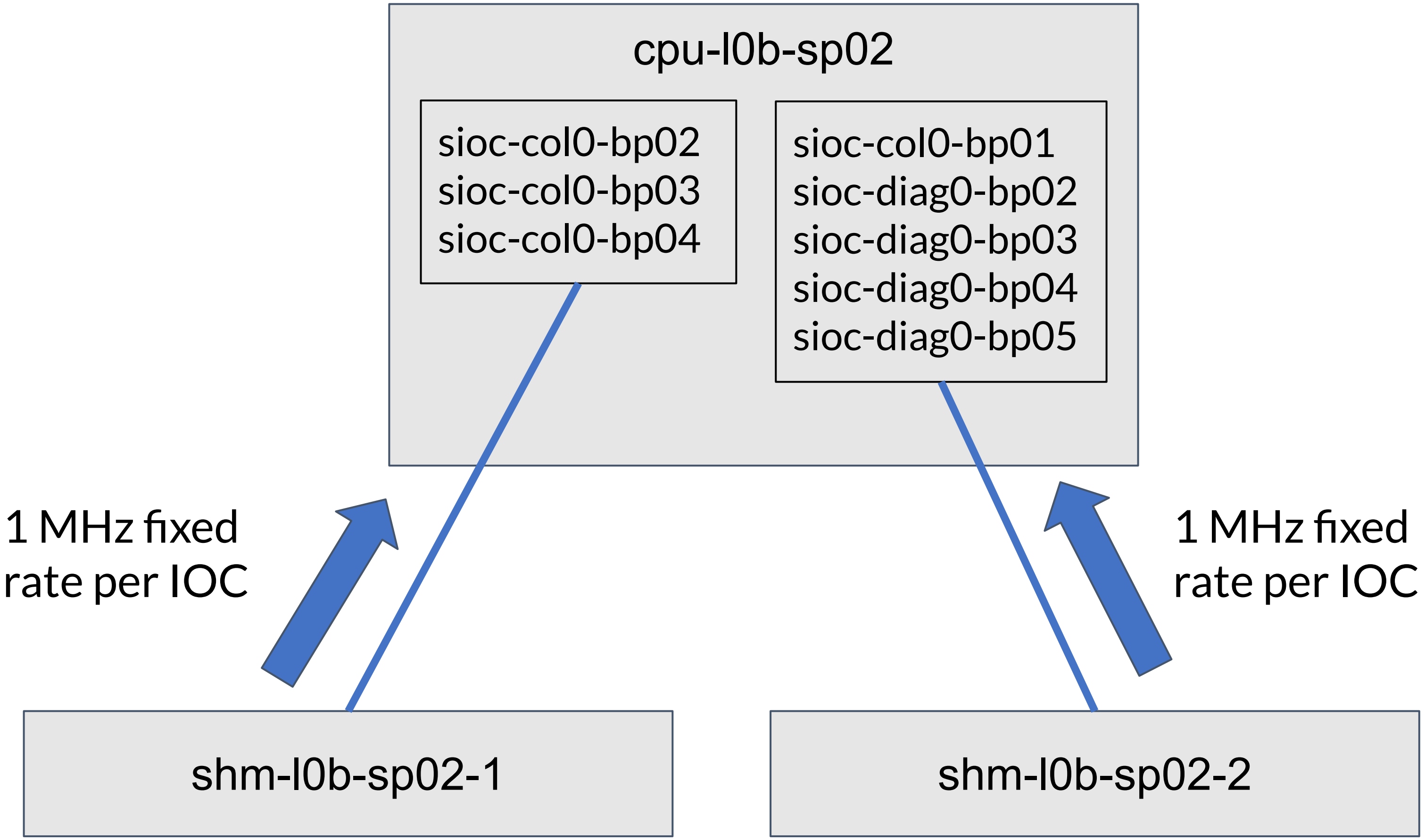
- The same way that data comes from the FPGA in jumbo frame size, the data must go to the outside world with approximately the same size.
- We used the loopback interface, instead: 127.0.0.1.

Stress test with BLD

Focus of the tests

8 BPM IOCs running in cpu-l0b-sp02 (production area).

20 bytes per bunch in case of the BPMs.



6

Stress test with BSAS

Stress test with BSAS

General View

BSAS provides 4 tables per carrier board. Each table is related to one destination:

- DIAG0
- BSYD
- HXR
- SXR

The number of columns of each table is proportional to the number of signals exposed.

The number of rows depend on a configuration (explained on the next slide).

secondsPastEpoch	nanoseconds	pulseId	X.CNT	X.VAL	X.AVG	X.RMS	X.MIN	X.MAX	Y.CNT	Y.VAL	etc.

Stress test with BSAS

Calculation of the number of rows

Two frequencies can be configured to determine how many rows the tables will have:

- Row Advance: the frequency at which a new row is created.
- Table Reset: the frequency at which the current table is thrown away and new data comes in.

If you are creating new rows at 10 kHz (Row Advance) and the table is resetting at 1 Hz, this will result in a table with 10,000 rows.

number of rows = Row Advance rate / Table Reset rate

These frequencies are configured in the TPG and there are no timing filters available for general usage, like BSA. The 4 available filters are determined by a committee.

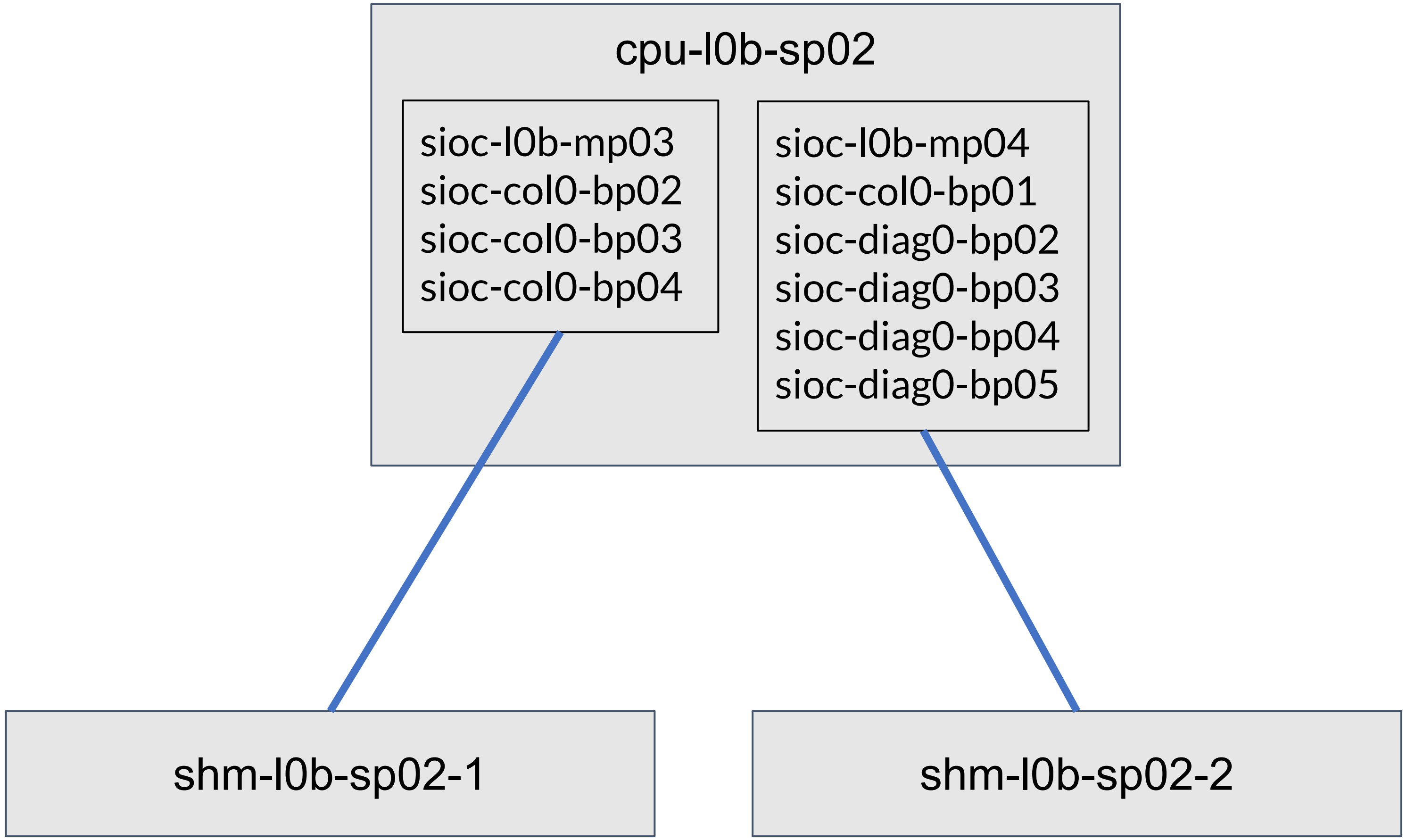
Stress test with fault buffers

Focus of the tests

10 IOCs running in cpu-l0b-sp02 (production area):

- 2 MPS
- 8 BPMs

1000 rows per table, 4 tables per IOC, once per second.



7

External network load with camonitor

External network load with camonitor

Python script to handle camonitor

A Python script was created to open thousands of camonitors

- Approximately 2400 waveform PVs with 20,000 elements each.
- Approximately 600 waveform PVs with 1 million elements each.

8

Results

Results

First scenario

Fault buffer script running.

BSAS with 1,000 rows per table updated every second.

camonitor script running.

BPM receiving debug streams at 100 Hz.

No BLD active.

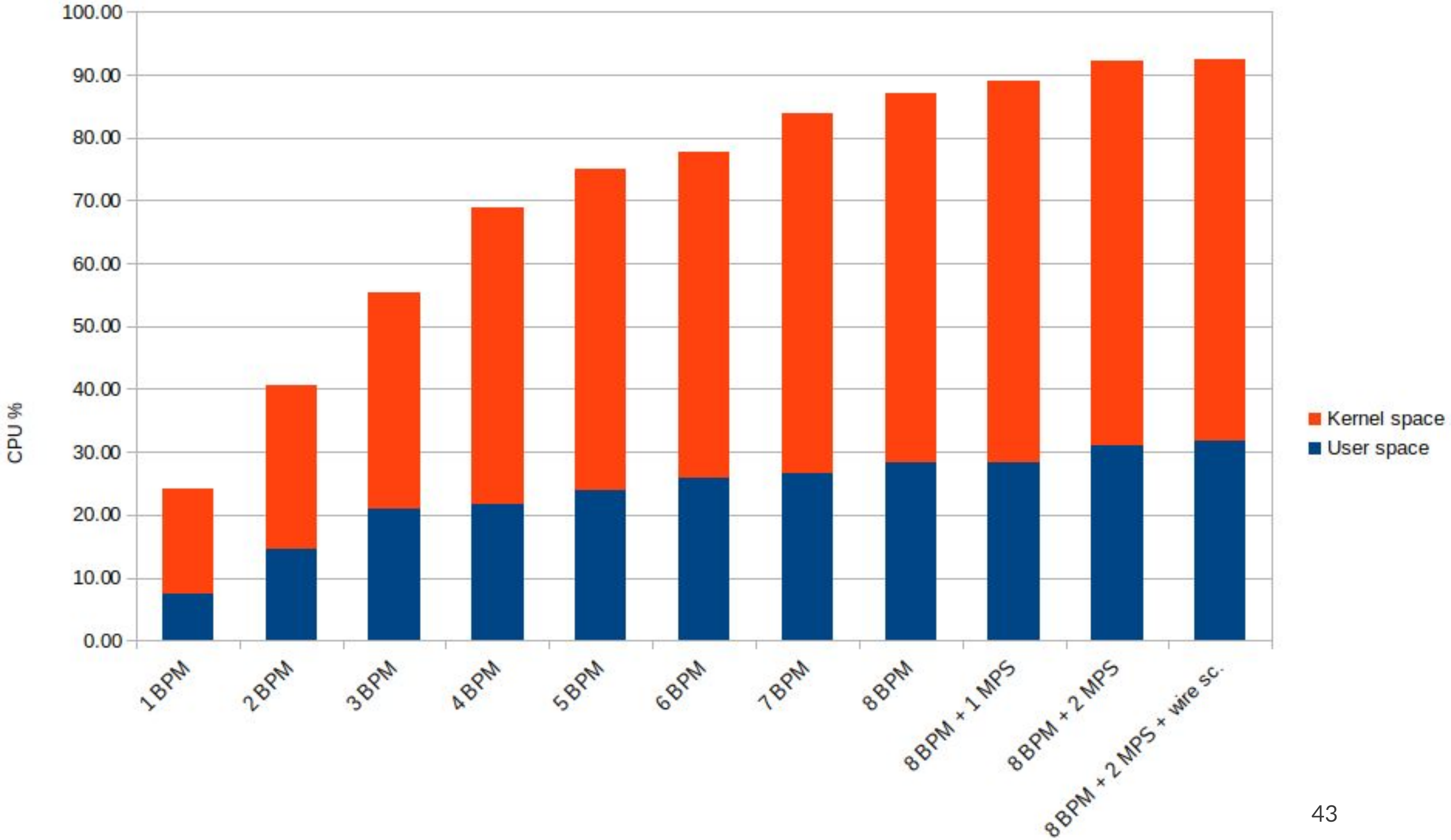
Results

Adding one IOC at a time

11 IOCs use approximately 93% of the CPU (including all cores)

Kernel space tasks takes twice more CPU than user space ones.

This is due to the high activity in the network stack.

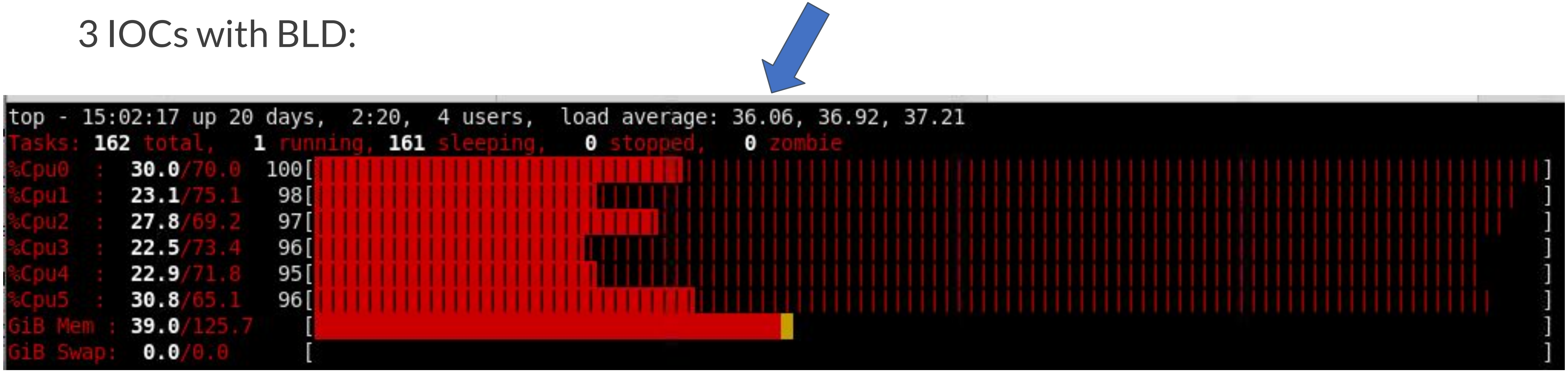


Results

BLD has high load

Activating one BLD configured at 1 MHz at a time.
We observed that BLD uses a lot of processing time.

3 IOCs with BLD:

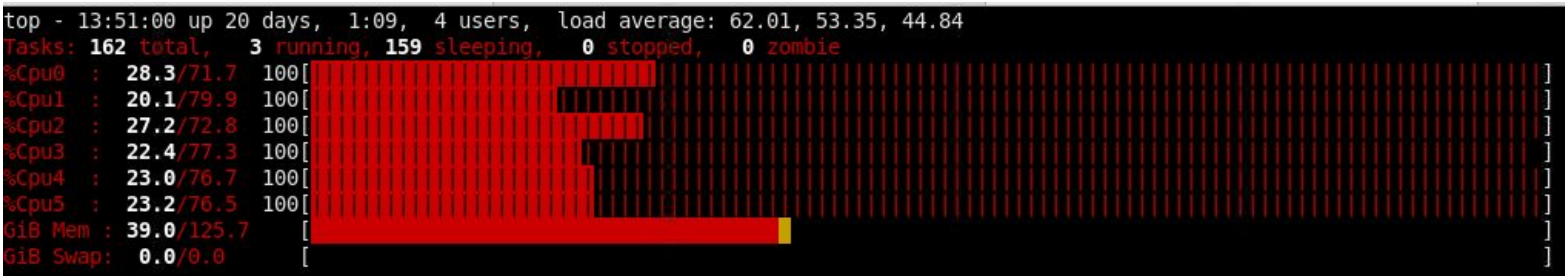


```
top - 15:02:17 up 20 days,  2:20,  4 users,  load average: 36.06, 36.92, 37.21
Tasks: 162 total,   1 running, 161 sleeping,   0 stopped,   0 zombie
%Cpu0  :  30.0/70.0  100[                                                              ]
%Cpu1  :  23.1/75.1   98[                                                              ]
%Cpu2  :  27.8/69.2   97[                                                              ]
%Cpu3  :  22.5/73.4   96[                                                              ]
%Cpu4  :  22.9/71.8   95[                                                              ]
%Cpu5  :  30.8/65.1   96[                                                              ]
GiB Mem :  39.0/125.7     [                                                              ]
GiB Swap:   0.0/0.0       [                                                              ]
```

Results

BLD has high load

4 IOCs with BLD:



```
top - 13:51:00 up 20 days,  1:09,  4 users,  load average: 62.01, 53.35, 44.84
Tasks: 162 total,   3 running, 159 sleeping,   0 stopped,   0 zombie
%Cpu0  :  28.3/71.7   100[                                                          ]
%Cpu1  :  20.1/79.9   100[                                                          ]
%Cpu2  :  27.2/72.8   100[                                                          ]
%Cpu3  :  22.4/77.3   100[                                                          ]
%Cpu4  :  23.0/76.7   100[                                                          ]
%Cpu5  :  23.2/76.5   100[                                                          ]
GiB Mem :  39.0/125.7    [                                                          ]
GiB Swap:   0.0/0.0      [                                                          ]
```

At this scenario:

- "top" takes a lot of time to start.
- ifconfig is slow.
- Autocomplete with Tab is slow.
- Accessing the IOC console is extremely slow (almost 1 minute after issuing iocConsole).
- iocManager showing heartbeat status changing: PRESENT - INTERMITTENT - ABSENT, meaning that the IOC was unable to post monitors for seconds.

Results

Curious side effect used as a tool

We discovered that the trigger channel calculated rate is affected when the system is too busy:

```
$ caget TPR:{COL0,DIAG0}:BP0{1,2,3,4,5}:0:CH0{1,2}_RATE | grep -v onnected
TPR:COL0:BP01:0:CH01_RATE   203.5
TPR:COL0:BP02:0:CH01_RATE   111.5
TPR:COL0:BP02:0:CH01_RATE   111.5
TPR:COL0:BP03:0:CH01_RATE   173.5
TPR:COL0:BP03:0:CH01_RATE   173.5
TPR:COL0:BP01:0:CH01_RATE   205.0
TPR:DIAG0:BP02:0:CH01_RATE   226.0
TPR:DIAG0:BP02:0:CH01_RATE   226.0
TPR:DIAG0:BP04:0:CH01_RATE   279.0
TPR:DIAG0:BP04:0:CH01_RATE   279.0
TPR:DIAG0:BP05:0:CH01_RATE   153.0
TPR:DIAG0:BP05:0:CH01_RATE   153.0
```

Triggers were configured to 100 Hz, so the calculated triggers should be close to it.

Results

Curious side effect used as a tool

We discovered that the trigger channel calculated rate is affected when the system is too busy.

This happens because the software is calculating it, not the firmware. Pseudo-code:

```
1. initial_count = get_trigger_count();
2. wait(1 second);
3. final_count = get_trigger_count();
4. calculated_rate = final_count - initial_count;
```

If the program gets to line 3 after more than 1 second due to general system slowness, the calculated rate will be greater than what was expected.

Results

Curious side effect used as a tool

We started using the calculated trigger rate to evaluate if the IOC was viable.

Although all IOCs survive if 3 of them are running BLD, the only scenario when the calculated trigger rates are right is when only 1 of them runs BLD.

In summary:

- 11 IOCs in one CPU.
- Fault buffers being read.
- BSAS working with 1000 rows.
- The system can handle **only one** IOC with BLD.

Stress test with fault buffers

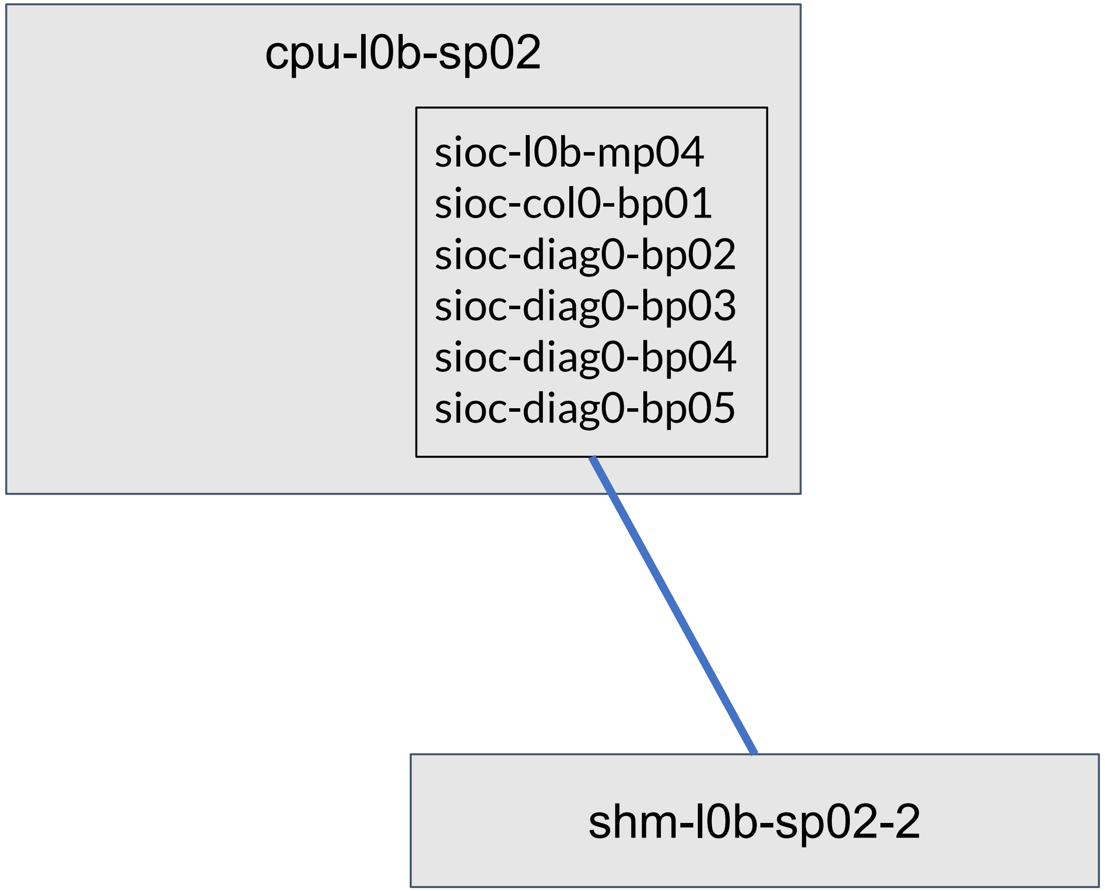
Testing IOCs with a single ATCA crate

This time, 6 IOCs running in cpu-l0b-sp02 (production area):

- 1 MPS
- 5 BPMs

Plan is to check cases with only one ATCA crate per CPU.

All previous test were repeated, with the same conditions.



cpu-l0b-sp02

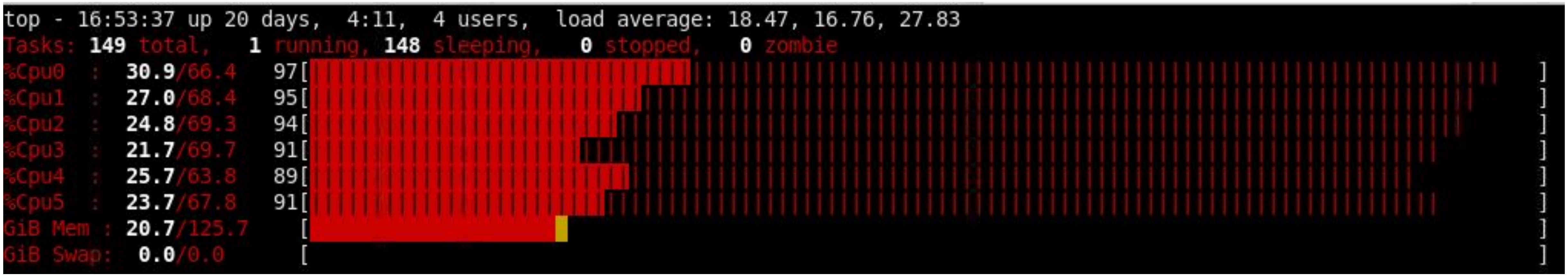
sioc-l0b-mp04
sioc-col0-bp01
sioc-diag0-bp02
sioc-diag0-bp03
sioc-diag0-bp04
sioc-diag0-bp05

shm-l0b-sp02-2

Results

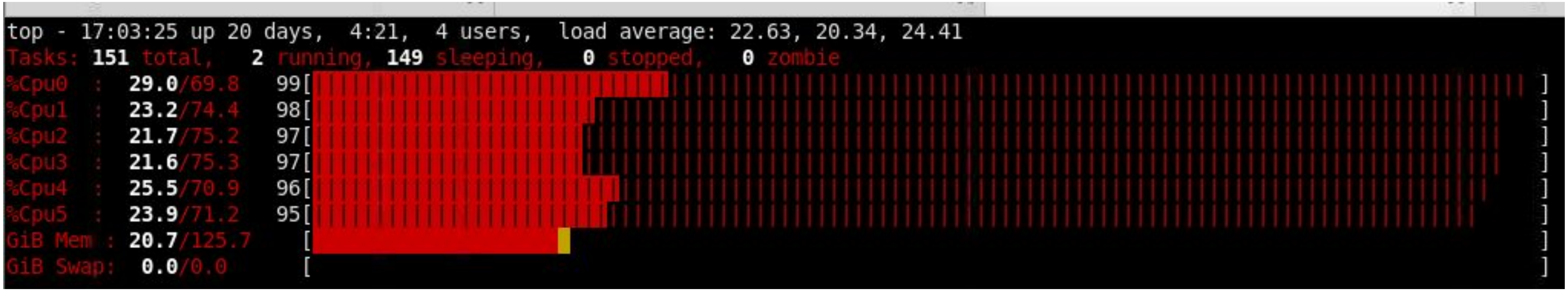
Testing IOCs with a single ATCA crate

3 BLDs running



```
top - 16:53:37 up 20 days,  4:11,  4 users,  load average: 18.47, 16.76, 27.83
Tasks: 149 total,   1 running, 148 sleeping,   0 stopped,   0 zombie
%Cpu0  :  30.9/66.4   97[                                                        ]
%Cpu1  :  27.0/68.4   95[                                                        ]
%Cpu2  :  24.8/69.3   94[                                                        ]
%Cpu3  :  21.7/69.7   91[                                                        ]
%Cpu4  :  25.7/63.8   89[                                                        ]
%Cpu5  :  23.7/67.8   91[                                                        ]
GiB Mem :  20.7/125.7     [                                                        ]
GiB Swap:   0.0/0.0       [                                                        ]
```

4 BLDs running



```
top - 17:03:25 up 20 days,  4:21,  4 users,  load average: 22.63, 20.34, 24.41
Tasks: 151 total,   2 running, 149 sleeping,   0 stopped,   0 zombie
%Cpu0  :  29.0/69.8   99[                                                        ]
%Cpu1  :  23.2/74.4   98[                                                        ]
%Cpu2  :  21.7/75.2   97[                                                        ]
%Cpu3  :  21.6/75.3   97[                                                        ]
%Cpu4  :  25.5/70.9   96[                                                        ]
%Cpu5  :  23.9/71.2   95[                                                        ]
GiB Mem :  20.7/125.7     [                                                        ]
GiB Swap:   0.0/0.0       [                                                        ]
```

Results

Testing IOCs with a single ATCA crate

With 5 IOCs running BLD we got to the global system slowness state and IOCs started to crash.

With 4 IOCs with BLD, all IOCs running kept alive during days, but…

Calculated trigger rate was showing that the IOCs were not behaving properly.

The maximum number of BLDs running at this scenario was 2. With more than this, the calculated trigger rate was showing wrong values.

Results

Comparison between 2 ATCA versus 1 ATCA crate

1 ATCA	2 ATCAs
Maximum 2 BLD for having viable IOCs	Maximum 1 BLD for having viable IOCs
Maximum 4 BLDs before system instability	Maximum 3 BLDs before system instability
5 or more BLDs cause IOCs to crash	4 or more BLDs cause IOCs to crash

BLD takes so many resources that there's not much difference between 2 full crates and 1 full crate.

Results

BSAS can use a lot of resources, too

BSAS running with 1000 rows didn't add a significant load compared with what was already running in the IOCs.

Based on available fixed rate choices, the next configuration that can be made for Row Advance is 10 kHz. This produces tables with 10,000 rows.

When configuring this scenario, most of the IOCs crashed. Jeremy mentioned that he saw multiple MPS IOCs crashing throughout the accelerator, not only on cpu-l0b-sp02.

Although the software throw away the excess of 1000 rows by design, the kernel still has to deal with the network packets arriving from the firmware.

Results

camonitor script put a challenge to lcls-srv01

Script was run in lcls-srv01.

When the script opened 3,000 camonitors for waveform records, it used 7 GB of RAM memory and 6.5% of one CPU core of the server.

Eventually lcls-srv01 crashed. We are not sure if the script was responsible for the crash, but it is a prime suspect.

When opening camonitors using PyEpics, keep in mind that it uses around 10 bytes per element of the waveform. Rule of thumb:

- BSA buffer with 20,000 elements ~= 200 kB per buffer
- Fault buffer with 1 million elements ~= 10 MB per buffer

Results

camonitor script put a challenge to lcls-srv01

I tried again in lcls-srv03.

After 9 hours all my sessions were disconnected from lcls-srv01 and lcls-srv03. I believe that this was made on purpose because I was using too many resources from the servers.

9

Known issues

Known issues

BPMs still see a problem when accessing BSA buffers

This happened twice along 1 month of tests.

The observed effect is that the PV SC_BSA_STATUS changes frequently from NORMAL to FAULT and never recovers.

This means that one or more BSA buffers are not working properly.

Rebooting the IOC brings it back to normal.

Known issues

BSAS has no protection against excess of rows

More than 1000 rows make the IOCs to crash.

Although the software throw away what comes in excess of 1000 rows (which helps with memory usage) the kernel and CPSW still have to deal with the additional packets coming.

The solution is to make the firmware limit to 1000 rows and send to software at most 1000 rows.

10

Conclusions and recommendations

Conclusions and recommendations

IOCs are now stable

Although we still have a few issues to work on, the system is very stable now and we are confident that it is good to be used for operations.

Under out of ordinary heavy load, the IOCs stayed alive for days.

The amount of heavy load that we used during the tests are expected to happen only a few times per day in real life.

Conclusions and recommendations

Plan for BLD is not feasible with current CPUs

From the current plan, these are the CPUs that will work well with BLD:

- cpu-ltus-sp01 (2 IOCs, 1 ATCA):
  - sioc-ltus-bp03
  - sioc-ltus-bp05
- cpu-ltus-sp02:
  - sioc-ltus-bp12
- cpu-ltus-sp03:
  - sioc-ltus-bp14
- cpu-gunb-sp01:
  - sioc-gunb-bp01
- cpu-fees-sp01 (2 IOCs, 1 ATCA):
  - sioc-fees-gd01
  - sioc-fees-gd02

Conclusions and recommendations

Plan for BLD is not feasible with current CPUs

These are the CPUs that can't hold the planned BLDs:

- cpu-unds-sp03 (4 IOCs, 2 ATCA). With 2 ATCAs only one IOC with BLD is possible.
    - sioc-unds-bp02
    - sioc-unds-bp13
    - sioc-unds-bp14
    - sioc-unds-bp15
- cpu-bc1b-sp01 (2 IOCs, 2 ATCA). With 2 ATCAs only one IOC with BLD is possible.
    - sioc-bc1b-bl01
    - sioc-bc1b-bp02
- cpu-l2b-sp03 (2 IOCs, 2 ATCA). With 2 ATCAs only one IOC with BLD is possible.
    - sioc-bc2b-bp01
    - sioc-bc2b-bp02

Conclusions and recommendations

Plan for BLD is not feasible with current CPUs

To address the problem, 2 approaches can be used:

- Add more CPUs and distribute the IOCs between them.
- Use more powerful CPUs.

(more data on equipment cost will be shown later)

The problem happens when we have full rate.

We still have time to address the problem as 1 MHz beam is far away in the future.

No tests were made to measure the impact of BLD with intermediary rates, like 1 kHz or 10 kHz.

Conclusions and recommendations

One CPU connected to 2 ATCAs provides a fragile scenario

The kernel is very busy mainly because of:

- Network stack.
- Intensive memory access.

A CPU with more cores can help with the network stack.

But more cores can't help much with memory access. If you have 1 million cores and all your tasks are reading and writing to memory intensely, most of the tasks will sleep waiting for a chance to reach the memory.

The memory bus and memory speed are the same, no matter how many cores you have.

The ideal solution in this case would be to connect one ATCA to one CPU.

Conclusions and recommendations

One CPU connected to 2 ATCAs provides a fragile scenario

Currently there are 14 CPUs connected to 2 ATCA crates.

There are available space to install an additional 1U server in all of the 14 racks.

Cost for the splitting:

- Option 1:
  - SKY-8101 ($4,440) Gen2 Silver 4210R 2.4GHz Max 3.2GHz 10 Cores, 2x 64GB DDR4
  - $4,440 x (14+1 Backup) = $66,600
- Option 2:
  - SKY-8101 ($6,580) Gen2 Gold 6230R ,2.1GHz Max 4.0GHz 26 Cores, 2x 64GB DDR4
  - $6,580 x (14+1 Backup) = $98,700

Conclusions and recommendations

One CPU connected to 2 ATCAs provides a fragile scenario

Calculating the needed number of cores is not possible. We've seen that the amount of CPU usage increases in a non-linear way.

Measuring how much of the CPU usage is from network stack and how much is due to memory access is an expensive work because of the amount of time needed to analyse diagnostic reports from the kernel.

A cheaper and more efficient approach is to borrow one unit of a type from the vendor and repeat the stress test at SLAC to measure the load.

If this is not possible, the recommended approach is to buy the SKY-8101 ($6,580) Gen2 Gold 6230R, which is the most expensive CPU.

Conclusions and recommendations

Testing in production is a nightmare

Tests with the timing system in production affects several different teams from different departments:

- The tests can affect the work of other teams not involved in the test.
- Other teams' work can affect the tests that we wanted to perform.

Getting to production is very bureaucratic, and it has to be this way!

- Multiple meetings with different stakeholders.
- CATERs with multiple jobs.
- PJBs, JSAs.
- Authorization from control deputy.
- Limited time for tests due to multiple teams working with the accelerator.

Testing in production is expensive and less productive because we use a lot of time in administrative tasks instead of working on the tests themselves.

Conclusions and recommendations

Testing in production is a nightmare

Testing in dev is straightforward if you have a dedicated test-stand. It follows the Nike philosophy: Just do It.

We are implementing a test program that will check functionality and stability under stress conditions. The idea is that every time a module changes, we automatically test it to see if nothing was broken.

For this, a complete environment is needed:

- 2 full ATCA crates with boards for all systems: BPM, GMD, BLEN, BCM, MPS, wire-scanner.
- 1 or 2 CPUs.
- A dedicated TPG that we can use (and abuse) without coordinating with anyone else.

11

Backup slides

Introduction

CATERS 158662 and 160141

Different systems have been crashing in production, with BPMs more frequently

Error message on IOC console:

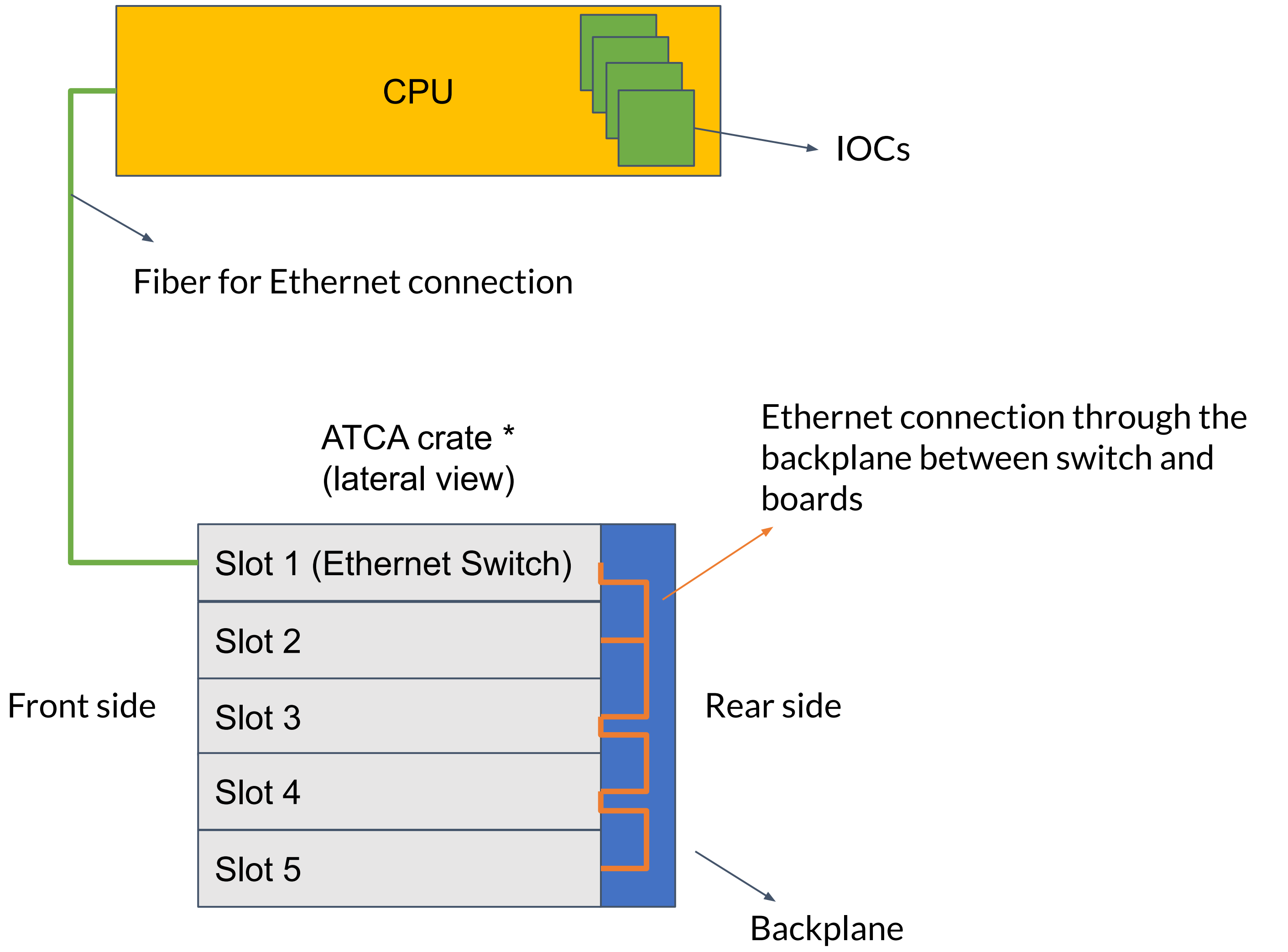
```
CPSW Error: CPSW Error: CPSW Error: No response -- timeout
(Retries=0, Last timeout=500000): Success:
/mmio/AmcCarrierCore/AmcCarrierBsa/Bsss/currPacketSize:
Success at ../BsssYaml.cc, line 69 terminate called after
throwing an instance of 'CPSWError'  what():  CPSW Error: CPSW
Error: No response -- timeout (Retries=0, Last
timeout=500000): Success:
/mmio/AmcCarrierCore/AmcCarrierBsa/Bsss/currPacketSize:
Success
```

Introduction

BSA pointer trap

Occasionally, the firmware would send a wrong memory position for the software to read.

Introduction



ATCA crate overview

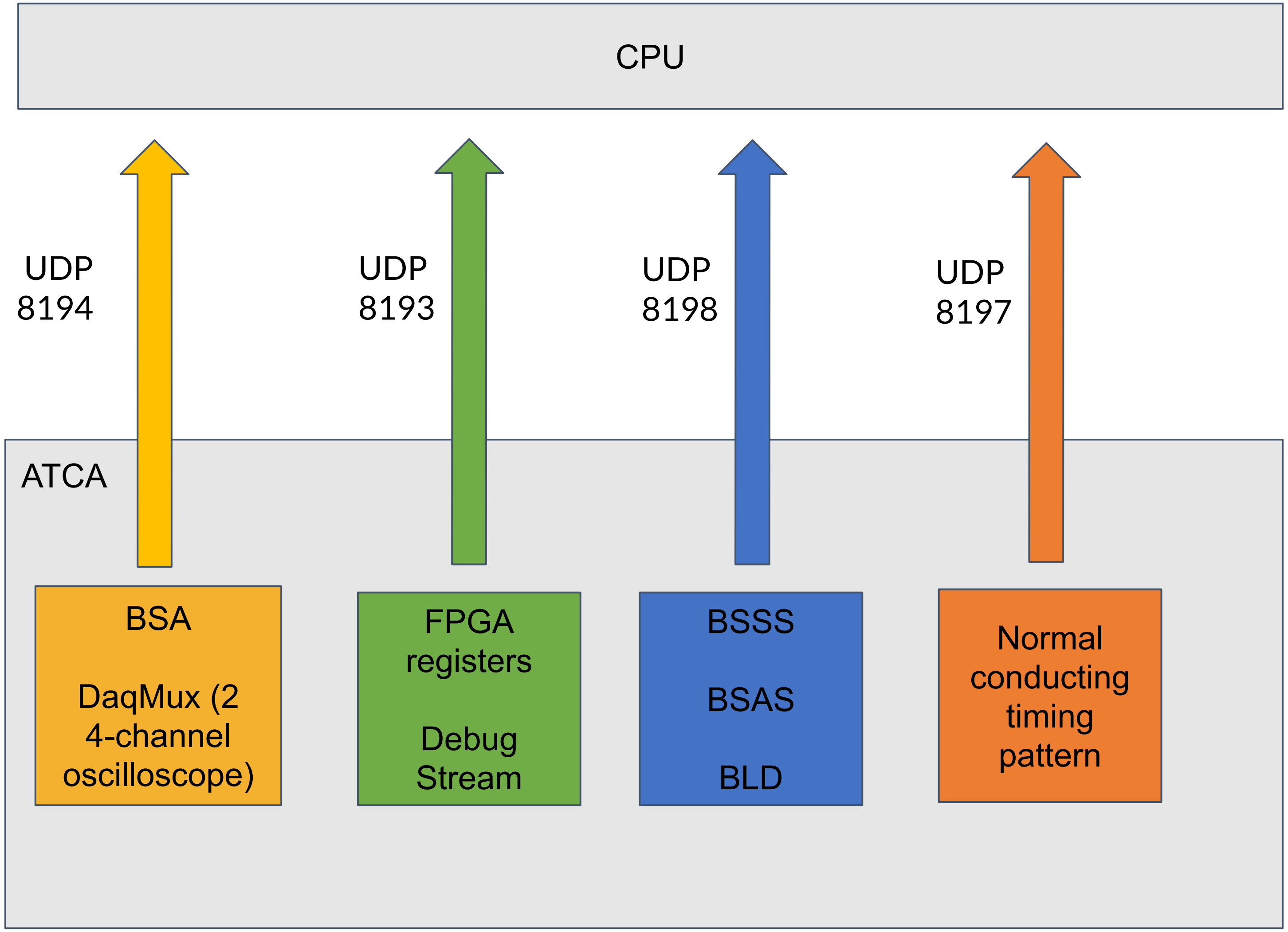
Communication between boards happen through the backplane which uses Ethernet.

Switch installed in slot 1 manages communication between boards and with the external world.

CPU where the IOCs run is connected to the ATCA crate through a 10 Gb/s fiber connection.

* RTM and all set of 7 slots not shown to make it easier to understanding

Introduction



ATCA crate overview

A few protocols were created at SLAC. All use UDP.

The firmware can handle only a few UDP servers, each using one port, due to limited resources of the FPGA chip.

The firmware developers defined the use of each UDP port, which can share different functions.

Reason for crashes and fixes

Issues solved since December 2022

BSA Buffer Freezing

- Reason: the memory address pointing to where to retrieve data (Write Pointer) was coming wrong from the firmware, pointing to a location outside of the BSA buffer boundary.
- Fix: bug fixing in firmware for the DMA boundary and BSA buffer boundary alignment.

CPSW exception: IOCs crashing when retrieving the fault buffer

- Reason: having too much CPU load and network bandwidth to access a large amount of data from the carrier board DRAM via RSSI communication.
- Fix: implemented throttling to slow down the fault buffer retrieving. Now it takes a bit longer to retrieve the fault buffer, but the IOCs are more stable.

Reason for crashes and fixes

Issues solved since December 2022

TPG got CPSW exception easily when reading FPGA registers

- Reason: TPG has tons of registers to be read and write. This made the system unstable and fragile when other heavy load processes are going on, for example when reading the fault buffer.
- Fix: merged multiple register access to a single array access to reduce the number of communication transactions.

Reason for crashes and fixes

Issues solved since December 2022

Found significant overload on the services: BSSS, BSAS, and BLD. Observed unstable BSSS PV update (missing updates).

- Reason: the listener thread was overloaded by heavy workload on the callback functions: BSSS callback and BSAS callback.
- Fix: implement the callback and callback queue. This way, the packet listener and worker threads for each services: BSSS, BSAS, BLD are separated. They can own a CPU core independently, being able to provide each service in parallel. The architecture is now prepared to support parallel callback (multiple worker threads for each services) if it is required.

Reason for crashes and fixes

Issues solved since December 2022

Overall CPU load was too high

- Reason: we have used asyn port driver for PVs and it has a significant overhead for updating large number of PVs with high rates. This was not good for BSA PVs and BSSS PVs
- Fix: implement own device support for SC BSA and BSSS. The own device support provided a dramatic performance improvement with optimizing code:
    - TPG: 225% CPU load down to 40%
    - AmcCarrierTpr: 185% CPU load down to 20%

Reason for crashes and fixes

Issues solved since December 2022

Overall memory usage was too high

- Reason: each layer in the software stack (ex, API, driver, and PV) used its own buffers. This requires memory copy (performance dragging) and also a large memory footprint.
- Fix: implemented zero copy buffer. This prevents memory copy (performance improvement), and also reduces the memory footprint. We reduced ~3.6 GB of memory usage for 31 bsa data signals.

Reason for crashes and fixes

Issues solved since December 2022

BSA processing error message appeared on the console. For some IOCs this happened mostly during the IOC start and, after a short period of time, the error messages would stop. BPM and GMD systems could take a longer time (multiple minutes to hours to get normal). When the message is being printed, the BSA PV update gets unstable. Some BSA buffers are ok, but other buffers stop updating.

- Reason: the firmware provides a wraparound flag to help handling the linear buffer as a ring buffer but, sometimes, the wraparound behaves wrongly.
- Fix: made software ignore the wraparound flag, and check up the relative location of read and write pointers to detect the wraparound condition without firmware's support.