

KPIT

io_uring In The Automotive World



About Myself

- Name: T.V.R.Prasad
- Company: KPIT Technologies
- Career: 12 years of experience in Android and Linux based IVI systems
- On the AGL mailing list
- Play around with my Odroid C2 and Dragonboard 410c



Agenda

- 1 `io_uring` Overview
- 2 `io_uring` In Automotive
- 3 Challenges
- 4 Questions

Abbreviations and Acronyms

Abbreviation	Definition/ Meaning
SQE	Submission Queue Entry
IVI	In Vehicle Infotainment
CQE	Completion Queue Entry
IO	Input-Output
ECU	Electronic Control Unit
OTA	Over The Air
CAN	Controller Area Network
VM	Virtual Machine

Thanks Stefano Garzarella

For letting me use slides from your talk

Speeding Up VM's I/O Sharing Host's io_uring Queues With Guests

Primary Goal

Handle IO Effectively To Extract Maximum Performance
From The Hardware

An aerial photograph of a two-lane road with a white dashed center line. A white car is driving in the right lane. The road is flanked by green trees and grass. A large black rectangle is overlaid on the left side of the image, partially obscuring the road and the trees. The text "What Is io_uring?" is written in white on the black rectangle.

What Is io_uring?

io_uring Making Headlines

The rapid growth of io_uring

By **Jonathan Corbet**
January 24, 2020

One year ago, the [io_uring subsystem](#) did not exist in the mainline kernel; it showed up in the 5.1 release in May 2019. At its core, io_uring is a mechanism for performing asynchronous I/O, but it has been steadily growing beyond that use case and adding new capabilities. Herein we catch up with the current state of io_uring, where it is headed, and an interesting question or two that will come up along the way.



Jens Axboe
@axboe

...

122M IOPS in 2U, with > 80% of the system idle. Easy.

How io_uring and eBPF Will Revolutionize Programming in Linux



By Glauber Costa
May 5, 2020

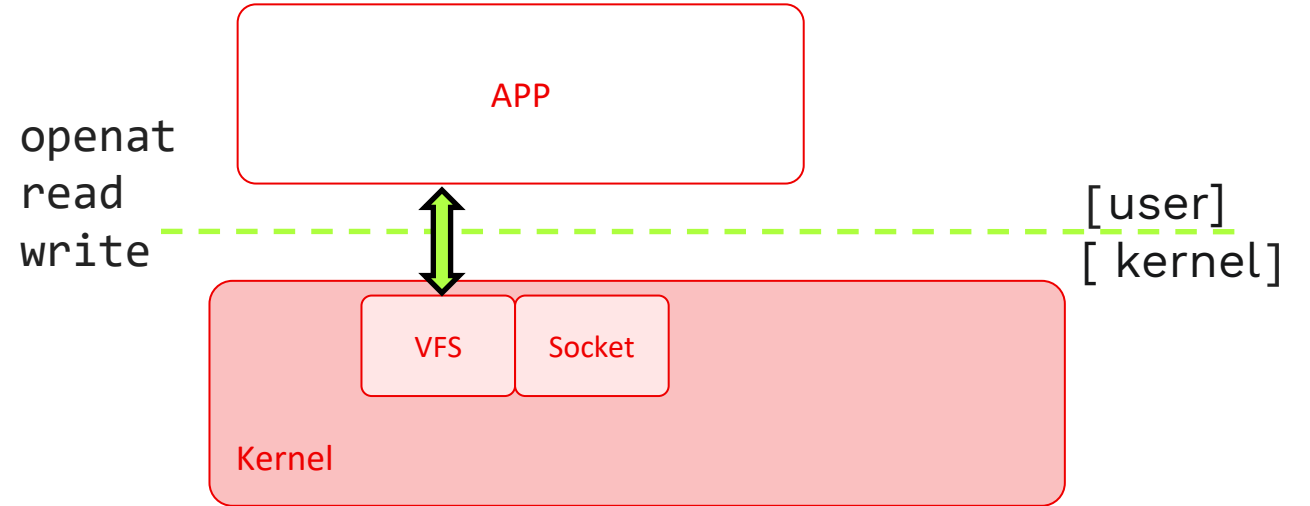
f t e in

IO_uring Continues Advancing In 2022 With New Features For Pushing Linux I/O

Written by [Michael Larabel](#) in [Linux Storage](#) on 1 June 2022 at 03:00 PM EDT. [9 Comments](#)

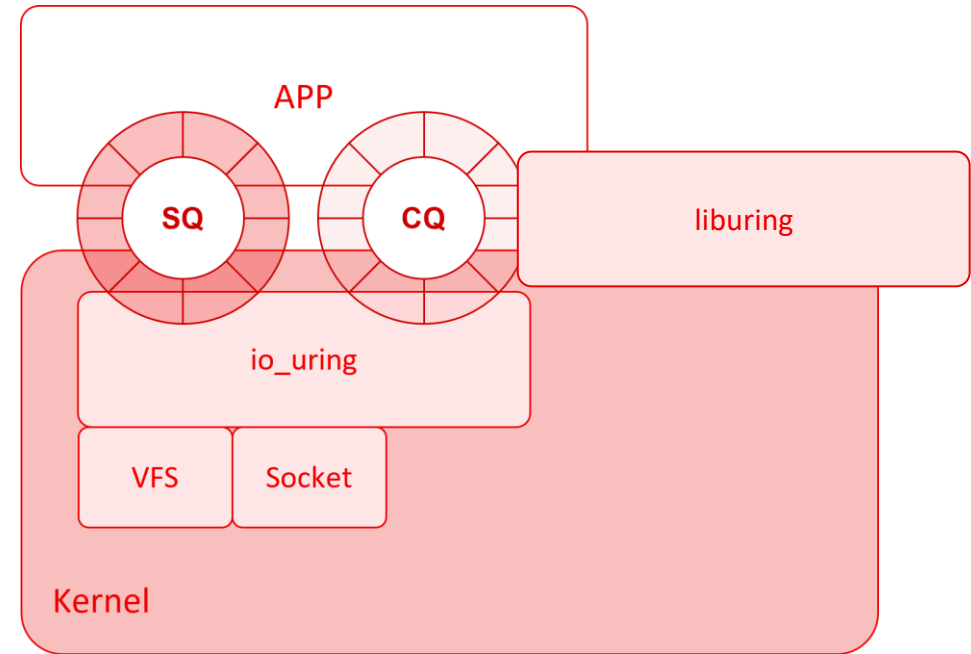
Traditional Linux I/O Model

- Synchronous I/O based on read/write
- aio exists but not used extensively
- Threads used for async behaviour
- epoll/poll/select based I/O used extensively



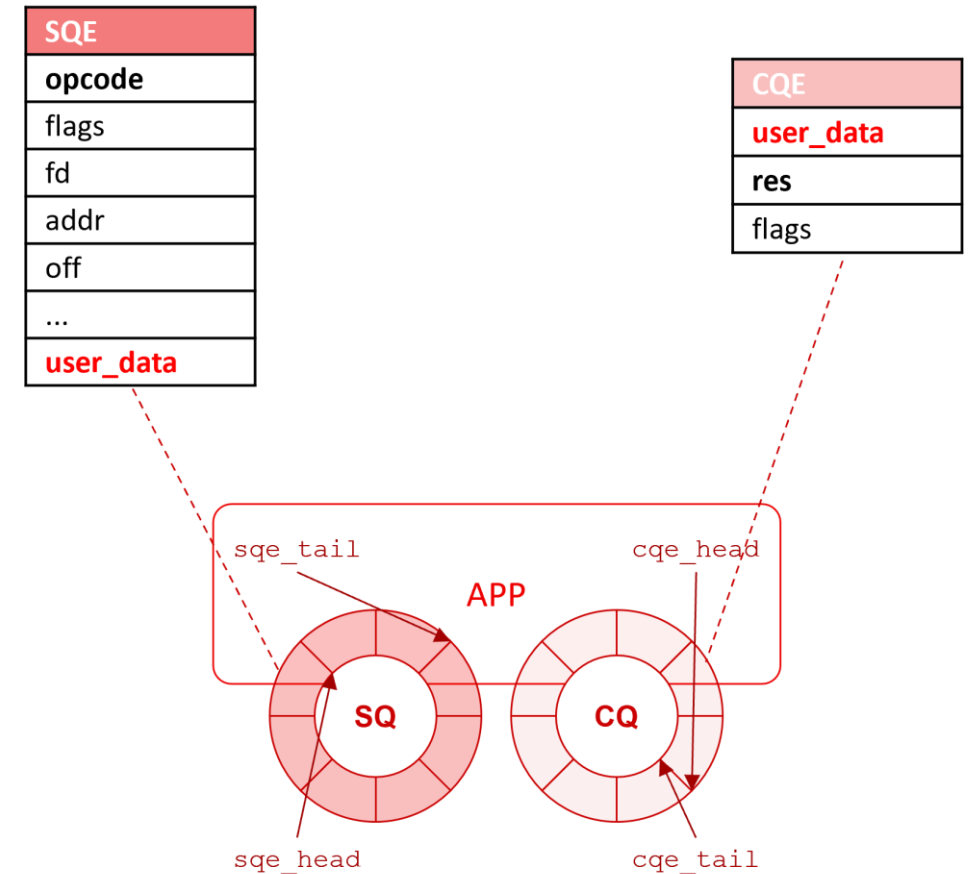
io_uring Overview

- A new **Linux** interface for **asynchronous I/O**
 - Not only for block oriented I/O
- A **pair of rings** shared between kernel and application
 - Submission Queue (**SQ**)
 - Completion Queue (**CQ**)
- Three system calls
 - `io_uring_setup`
 - `io_uring_register`
 - `io_uring_enter`
- Use `liburing` instead of the system calls directly
- Initial version merged in Linux kernel version 5.1



io_uring In Operation

- **Submission Queue (SQ)**
 - Application
 - produces SQEs (SQ Entry)
 - operation to do (opcode)
 - updates sqe_tail
 - invokes io_uring_enter(2)
 - Kernel
 - consumes SQEs
 - updates sqe_head
- Kernel processes the operation
- **Completion Queue (CQ)**
 - Kernel
 - produces CQEs (CQ entry)
 - operation result (res) and user_data
 - updates cqe_tail
 - Application
 - consumes CQEs
 - updates cqe_head



Ref: Speeding Up VM's I/O Sharing Host's io_uring Queues With Guests – Stefano Garzarella

io_uring Resource Registration, Linked Commands ...

- **Resource Registration**

- Eliminates the need for pin/unpin pages in the submission and completion paths
 - User buffers
 - File Descriptors
- Use the _FIXED variant of functions for preparing the SQE
 - IORING_OP_READ_FIXED
 - IORING_OP_WRITE_FIXED

- **Linked Commands**

- Arbitrary Chain of syscalls/commands
- Pick up the SQE only if the previous SQE succeeds
- System call reductions
- accept -> recv -> send -> close (example chain)

- **Polled I/O**

- IORING_SETUP_SQPOLL
- IORING_SETUP_IOPOLL

- **LSM Auditing support**

- SELinux
- SMACK

io_uring Adoption ...

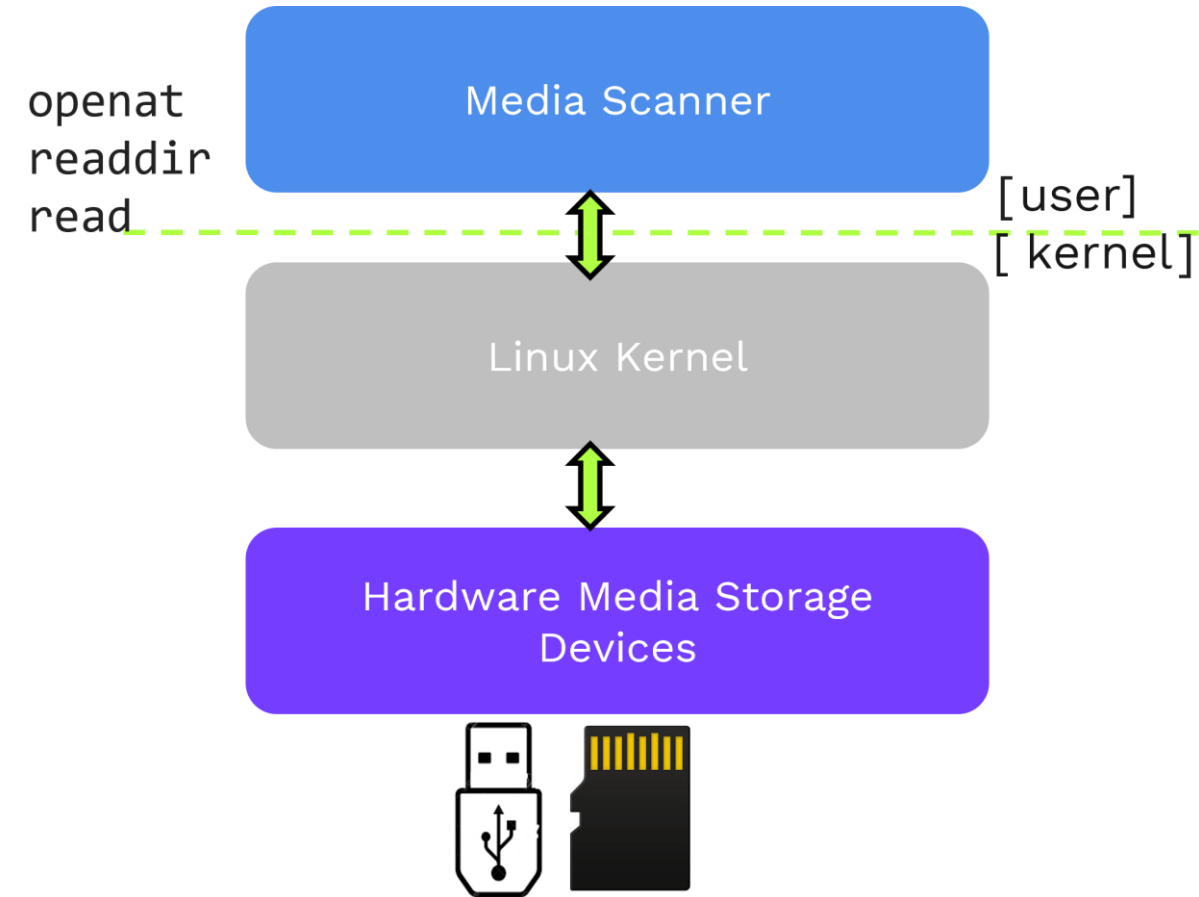
- **Android for software update**
 - Snapshot merging
 - Uses a UBD (User Space Block Driver based on io_uring)
 - fastboot utility also shows code pertaining to io_uring
- **Rust, C++ I/O executors**
- **Various other databases**

An aerial photograph of a two-lane road with a white dashed center line. A white car is driving in the right lane. The road is flanked by green trees and grass. Overlaid on the entire image is a grid of green rectangular sensor beams, resembling a LiDAR point cloud or sensor field. A semi-transparent black rectangle is positioned on the left side of the image, containing the text 'io_uring In Automotive'.

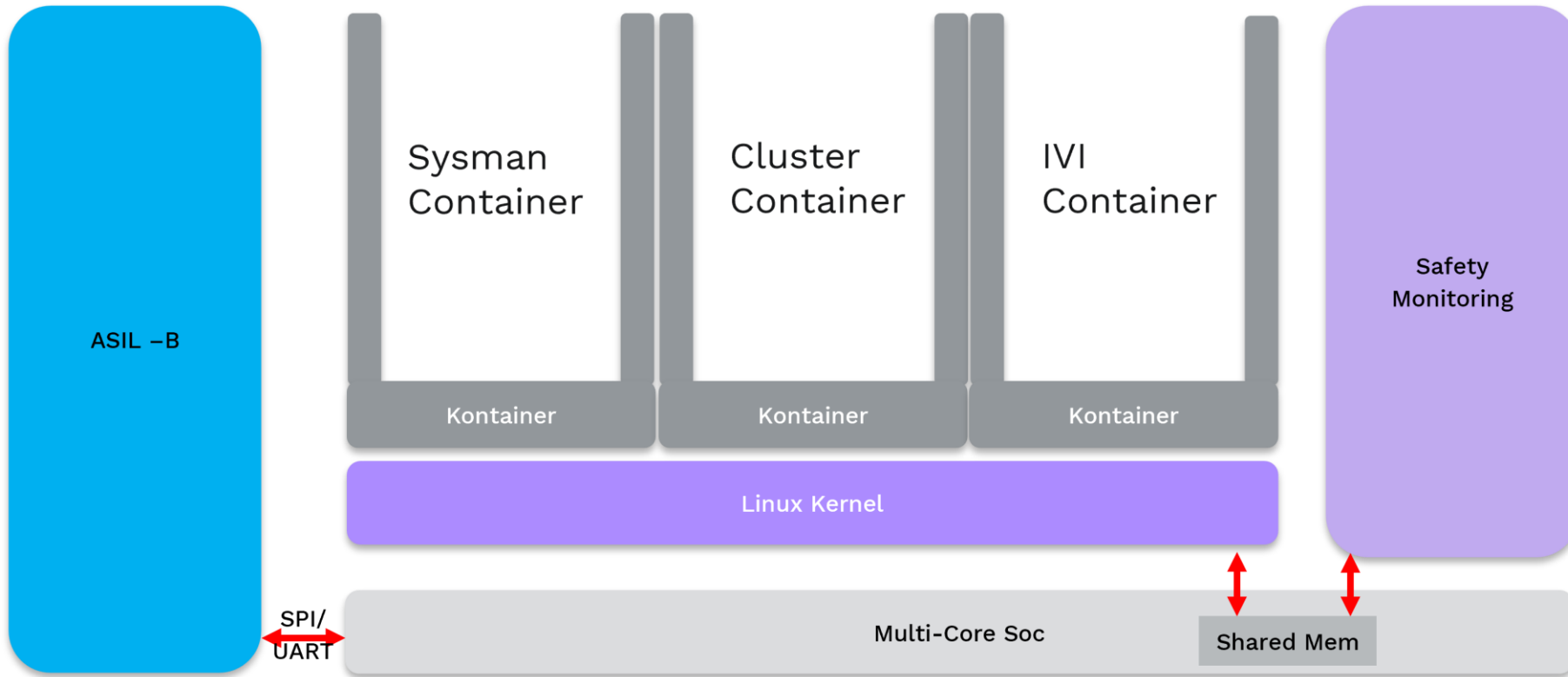
io_uring In Automotive

Media Indexing

- Media storage devices typically have GB's of media content
- Need to extract the metadata from the media files
- Huge amount of IO involved to scan through the directories and read the metadata
- **readdir and lseek support missing in io_uring**



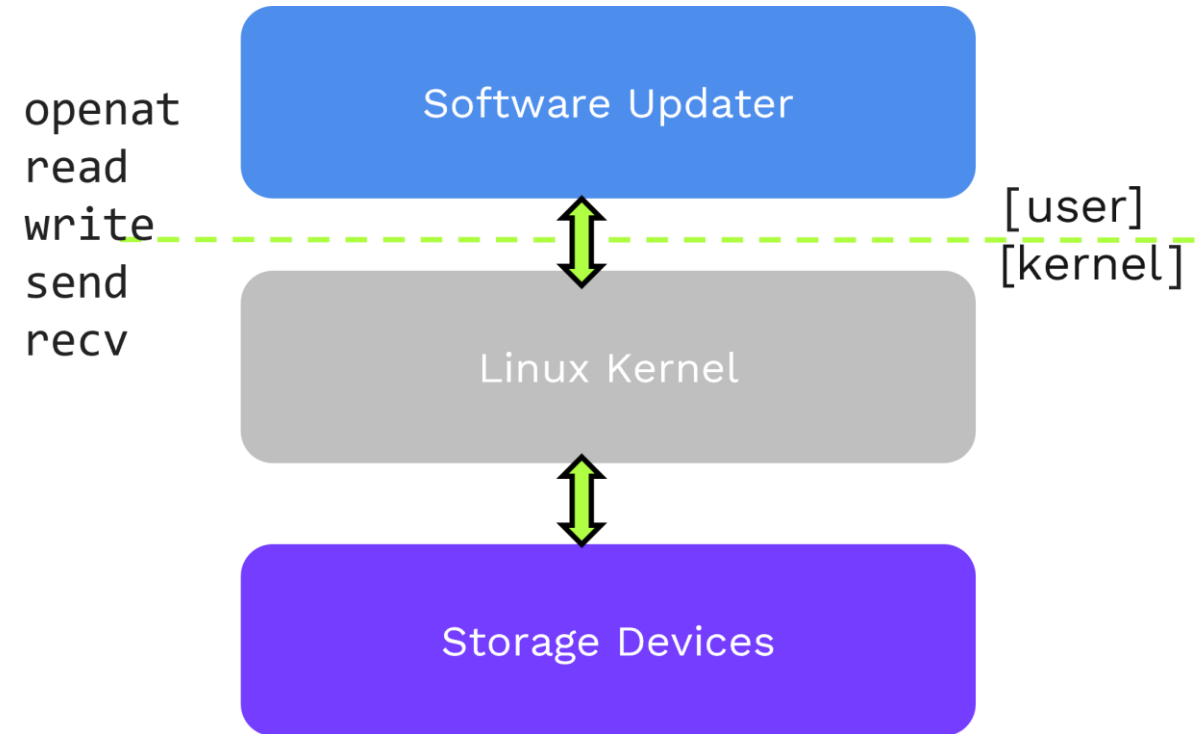
Reference Architecture For Container Based eCockpit



kontainer: KPIT Proprietary Container Runtime

Software Update

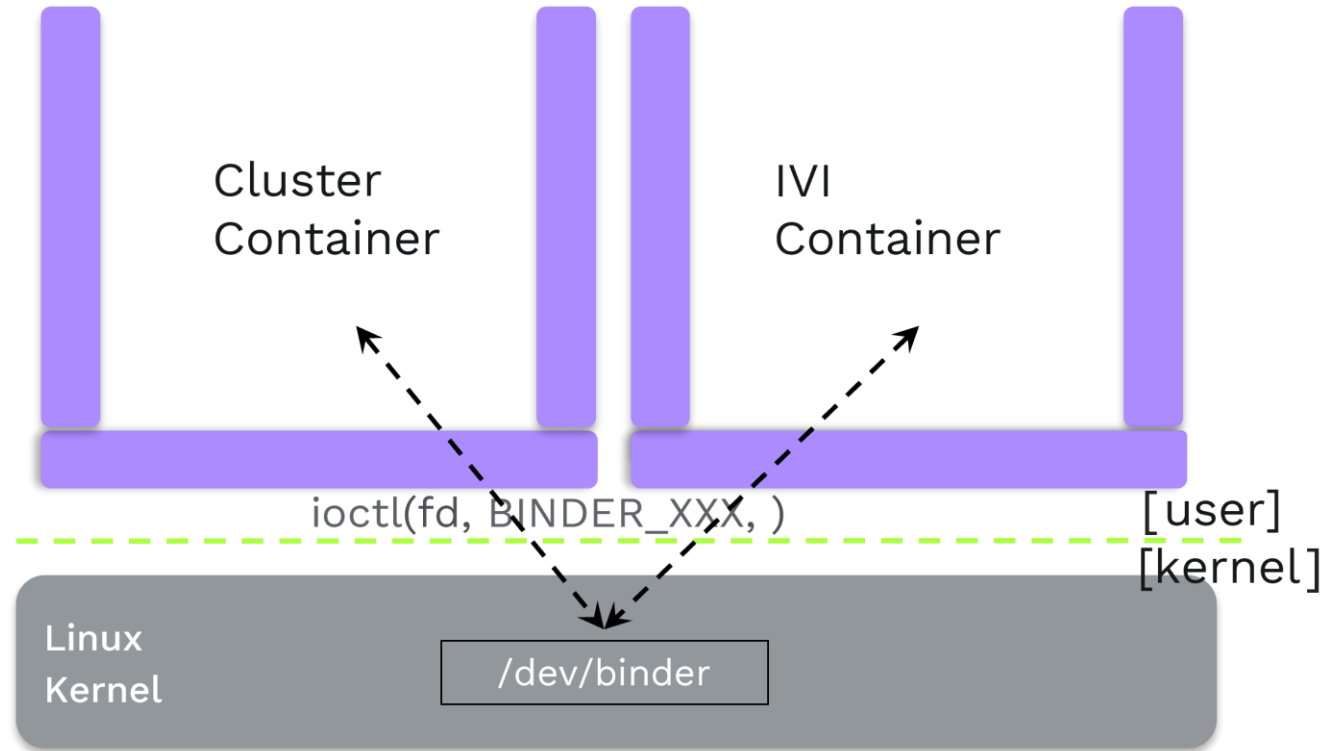
- Update various ECU's
- ECU's connected on various busses
- Involves huge amount of IO to download, flash and verify the image
- A/B updates requires the backup partition also to be updated
- Cockpits require container or VM update



Binder And Inter-Container RPC

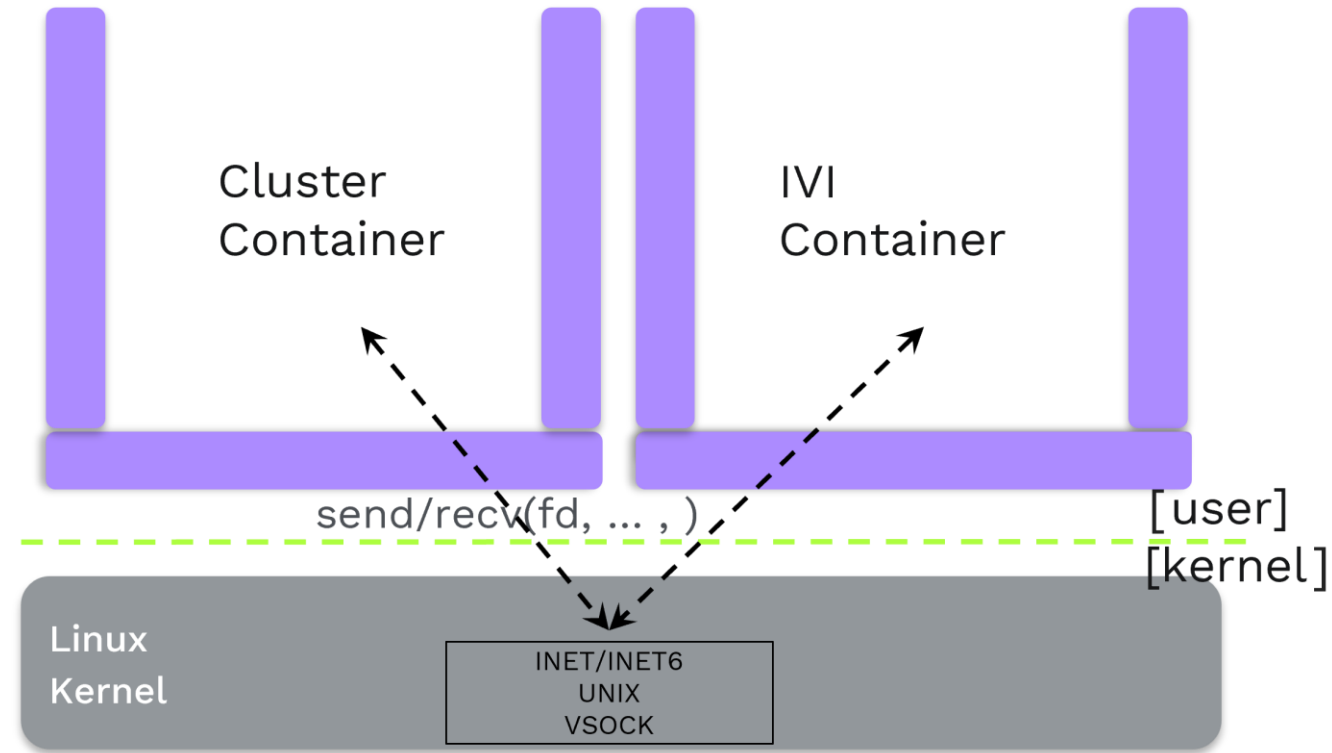
- Binder used for Inter-container communication
- Binder transaction one or more IOCTL's
- Evaluation in progress to use the IORING_OP_URING_CMD for Binder IOCTL's
- Using the NVMe implementation as reference

```
static const struct file_operations nvme_dev_fops = {  
    .owner      = THIS_MODULE,  
    .open       = nvme_dev_open,  
    .release    = nvme_dev_release,  
    .unlocked_ioctl = nvme_dev_ioctl,  
    .compat_ioctl = compat_ptr_ioctl,  
    .uring_cmd  = nvme_dev_uring_cmd,  
};
```



io_uring And Other RPC

- RPCs like D-Bus, capnproto etc using sockets can benefit from io_uring
- capnproto supports Unix domain and INET sockets for the transport layer
- Register the buffers and the file descriptors the RPC needs with io_uring
- **Evaluating the performance of io_uring ZEROCOPY send on capnproto**
- **VSOCK sockets considered for inter-VM RPC**



CAN-Over-WLAN Bridge

```
(..) /* some source code – don't worry */
int can; /* socket handle */
int wlan;
struct can_frame mymsg; /* data structure for CAN frames */

can = socket(PF_CAN, SOCK_DGRAM, CAN_RAW); /* CAN RAW Socket */
wlan = socket(PF_INET, SOCK_DGRAM, 0); /* UDP/IP Socket */

(..) /* set addresses and CAN filters */
bind(can, (struct sockaddr *)&can_addr, sizeof(can_addr));
connect(wlan, (struct sockaddr *)&in_addr, sizeof(in_addr));

while (1) {
    read(can, &mymsg, sizeof(struct can_frame));
    write(wlan, &mymsg, sizeof(struct can_frame));
}
```

Ref: Presentation for Automotive Grade Linux F2F, 2017-04-04, Microchip (Karlsruhe)

io_uring:CAN-Over-WLAN Bridge

```
while (1) {  
    sqe = io_uring_get_sqe(ring);  
    io_uring_prep_readv(sqe, can, &data->iov, 1, 0);  
    sqe->flags |= IOSQE_IO_LINK;  
    io_uring_sqe_set_data(sqe, data);  
  
    sqe = io_uring_get_sqe(ring);  
    io_uring_prep_writev(sqe, wlan, &data->iov, 1, 0);  
    io_uring_sqe_set_data(sqe, data);  
    io_uring_submit(ring);  
}
```

Misc. Use Cases

- EAVB based use cases
- Cockpit related use cases (Hypervisor and container)
- Camera based use cases

An aerial photograph of a two-lane road with a white dashed center line. A white car is driving in the right lane. To the right of the road is a dense line of green trees. To the left of the road is a large black rectangle that covers the left side of the image. The word "Challenges" is written in white text on the black rectangle.

Challenges

Challenges

- Requires recent kernel versions (Very few SoC vendors support latest versions of the kernel)
- Dependent open source libraries also need to adapt to io_uring
- Developers not comfortable with the async IO programming model

Conclusion

- Automotive systems involve lot of IO and io_uring helps make IO performant
- The automotive industry and the community need to come together to drive the development of new features in io_uring and trigger widespread adoption
- io_uring is not the solution for everything , use it with care

Let Us Know Your io_uring Use Cases

eMail:Prasad.T.V.R@kpit.com

Phone:+91 9849506168



Thank You