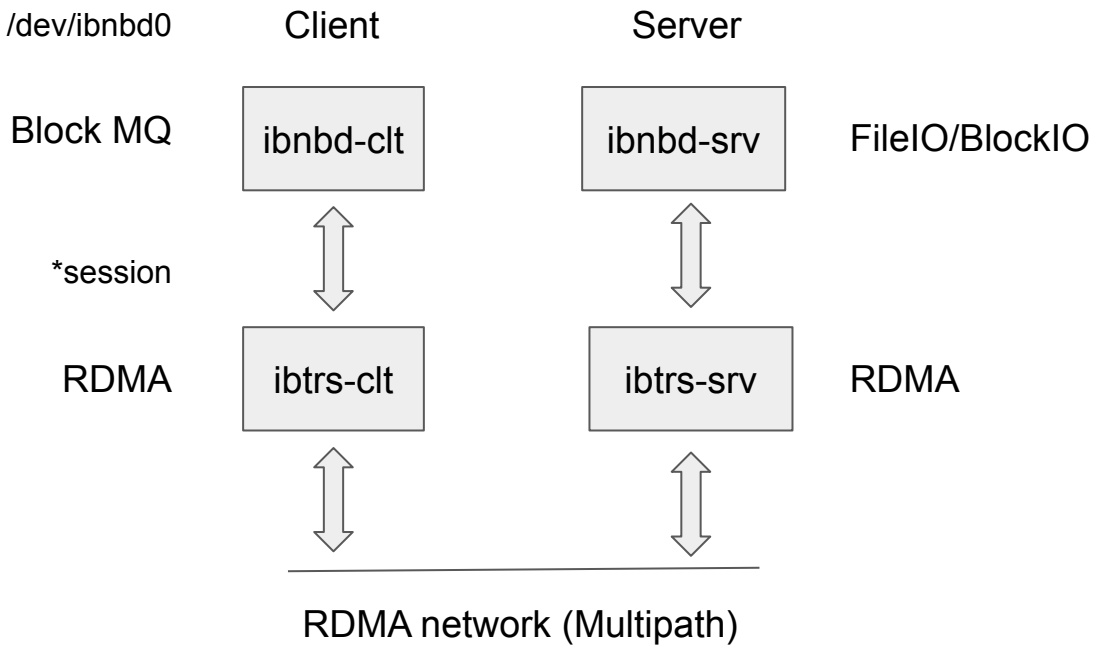
# IBNBD/IBTRS Upstreaming: Action Items

Jack Wang, jinpu.wang@cloud.ionos.com
Danil Kipnis, danil.kipnis@cloud.ionos.com

Contributors: Roman Penyaev, Jack Wang, Fabian Holler, Kleber Souza, Danil Kipnis, Swapnil Ingle

# IBNBD/IBTRS: Overview




## Github

https://github.com/ionos-enterprise/ibnbd

## Performance Evaluation

https://dcd.ionos.com/ibnbd-performance-report/

## Last patchset

https://lwn.net/Articles/791690/

# IBTRS: main features

- Client side server memory management
- Only RDMA writes with immediate
- No registration/unregistration on server side in io path
- Multipath and Failover policies: “Min Inflight”, Round Robin
- One rdma connection per cpu. (Separate cq_vector per connection - allows to “pin” IO on client side to a cpu if setting IRQ affinity accordingly)
- Good performance numbers on variety of test systems
- Memory preallocation on server for better performance

# IBNBD: main features

- MQ devices on client side
- Block or File IO interface on server side
- Minimal user interface

# Production usage and test coverage

Platforms

- AMD Opteron 6xxx/EPYC Naples
- Intel Haswell/Broadwell/Skylake/Cascadelake

Infiniband HCAs

- Mellanox ConnectX2 MT26428
- Mellanox ConnectX3 MT4099
- Mellanox ConnectX4 MT4115
- Mellanox ConnectX5 MT4119

# Patchsets

| | |
|---|---|
| V0 RFC | 2017/03/24 |
| V1 Multipath, less code | 2018/02/02 |
| V2 RQ removal, FR only, MR invalidation, docs, etc | 2018/03/18 |
| V3 Sparse fixes, sysfs changes | 2018/06/06 |
| V4 IO prio, CX4/CX5 support, benchmark, bugfixes | 2019/06/20 |

# Planned next steps

- Rename IBNBD to RNBD (RDMA Network Block Device)
- Process send completions for read path or set retry_cnt to 0. (Lost IB Acknowledgements)
- Finish user-space ibnbd-tool
- Test ROCE support

# Thank you!

Open questions:

- What’s the right place to put documentation?
- Rename driver to R(dma)NBD?
- New AMD Rome has 256 cpus, but HCA only supports only 128 MSI-X, some are internal use?

Backup slides:

- Links
- Changelogs for different patchsets
- Not implemented community requests
- Some performance numbers

# Links

- Github https://github.com/ionos-enterprise/ibnbd
- Last patchset https://lwn.net/Articles/791690
- Performance evaluation v4 https://dcd.ionos.com/ibnbd-performance-report/. Links to performance results for each version can also be found in the cover letters of corresponding patchsets.
- Vault 2017 presentation http://events.linuxfoundation.org/sites/events/files/slides/Copy%20of%20IBNBD-Vault-2017-5.pdf

# Backup: Changelogs (v4, v3)

V4: https://lwn.net/Articles/791690

- Extend protocol to transport IO priorities
- Support Mellanox ConnectX-4/X-5
- Extend sysfs: display access mode on server side
- Bug fixes: clean up sysfs folders, fix race on deallocation of resources
- Style fixes

V3: https://lwn.net/Articles/756994/

- Sparse fixes:
  - le32 -> le16 conversion
  - pcpu and RCU declaration
  - sysfs: dynamically alloc array of sockaddr structures to reduce size of a stack frame
- Rename sysfs folder on client and server sides to show source and destination addresses of the connection, i.e.: .../<session-name>/paths/<src@dst>/
- Remove external inclusions from Makefiles.

# Backup: Changelog v2 (https://lwn.net/Articles/755075/)

- No legacy request IO mode, only MQ is left. (IBNBD)
- No FMR registration, only FR is left.
- Don’t created pd with IB_PD_UNSAFE_GLOBAL_RKEY by default.
- Always register memory on server. Send MRs dma addresses to client.
- Client side (initiator) has `noreg_cnt` module option, which specifies sg number, from which read IO should be registered. By default 0 is set, i.e. always register memory for read IOs. (IBTRS protocol does not require registration for writes, which always go directly to server memory).
- Proper DMA sync with ib_dma_sync_single_for_(cpu|device) calls.
- Do signalled IB_WR_LOCAL_INV.
- Avoid open-coding of string conversion to IPv4/6 sockaddr, inet_pton_with_scope() is used instead.
- Introduced block device namespaces configuration on server side (target) to avoid security gap in not trusted environment, when client can map a block device which does not belong to it.
- README is extended with description of IBTRS and IBNBD protocol, e.g. how IB IMM field is used to acknowledge IO requests or heartbeats.
- IBTRS/IBNBD client and server modules are registered as devices in the kernel in order to have all sysfs configuration entries under /sys/devices/virtual/ in order not to spoil /sys/kernel directory.

# Backup: Changelogs (v1, v0)

V1: https://lwn.net/Articles/746342/

- IBTRS: load-balancing and IO fail-over using multipath features were added.
- Major parts of the code were rewritten, simplified and overall code size was reduced by a quarter.

V0: https://lwn.net/Articles/718181/

- Initial submission

# Backup: Not implemented community requests

- Bart Van Assche and Sagi Grimberg suggested to use sbitmap instead of calling find_first_zero_bit() and friends.  We found calling pure bit API is more explicit in comparison to sbitmap - there is no need in using sbitmap_queue and all the power of wait queues, no benefits in terms of LoC as well.

- Roman Penyaev did several attempts to unify approach of wrapping ib_device with ULP device structure (e.g. device pool or using ib_client API), as Sagi Grimberg suggested, but it turns out to be that none of these approaches bring simplicity, so IBTRS still creates ULP specific device on demand and keeps it in the list.

- Sagi Grimberg suggested to extend inet_pton_with_scope() with gid to sockaddr conversion, but after IPv6 conversion (gid is compliant  with IPv6) special RDMA magic should be done in order to setup IB port space range, which is very specific and does not fit to be some generic library helper.

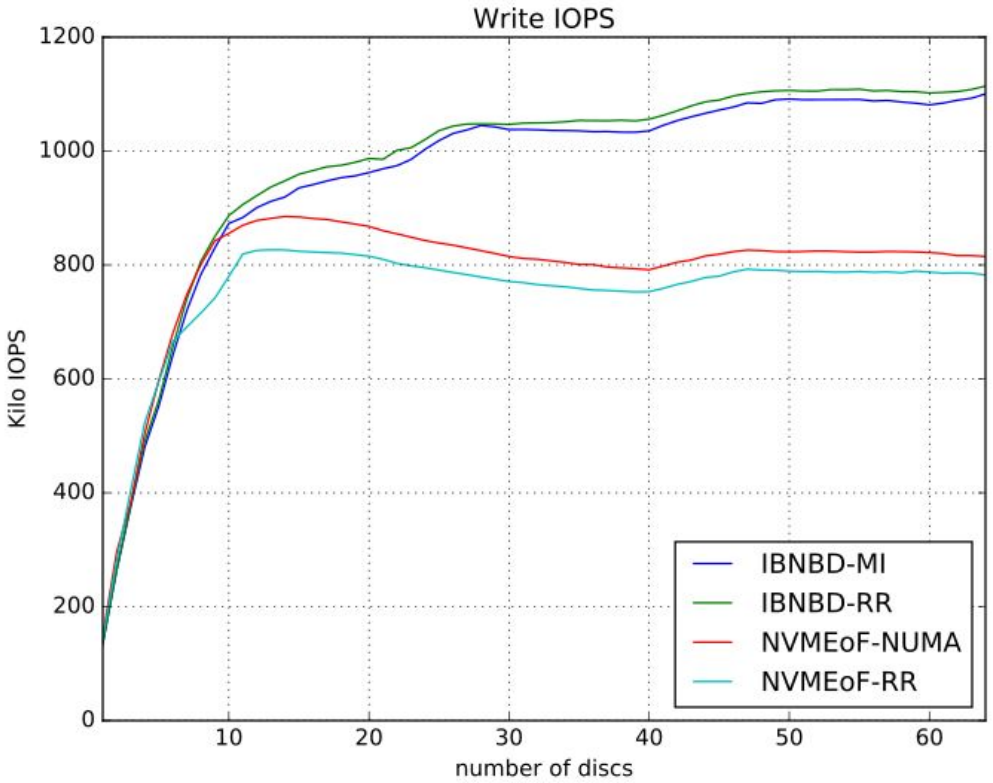
# Backup: IBTRS - Functionality and applications

- Transceive sg_lists with read/write semantics over RDMA
- Connection establishment, Multipath, Auto-Reconnects

(Potential) Applications:

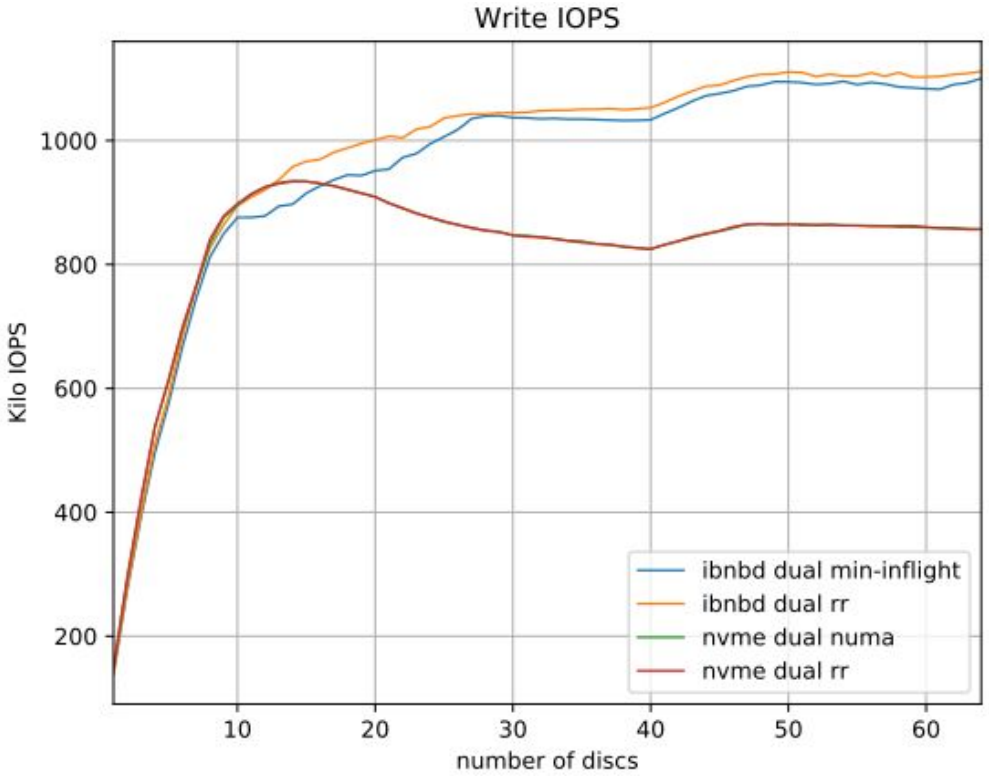
- Block IO over RDMA (BIO/SCSI/NVME)
- In-Kernel RDMA Transport for CephFS and RBD: RADOS messages
- Distributed Computations (Write a chunk of data to a compute node, receive the result of the computation back)
- Distributed Databases (Write for update, read for select)

# Backup: null_blk, Write IOPS




- Linux kernel v5.2-rc3
- 40 CPUs Intel Xeon Silver 4114 CPU 2.20GHz
- Mellanox MT27700 Family ConnectX-4 100Gb/s adaptors
- bssplit 512/20:1k/16:2k/9:4k/12:8k/19:16k/10:32k/8:64k/4
- NVMEoF param_inline_data_size **4096 (default)**

# Backup: null_blk, Write IOPS




- Linux kernel v5.2-rc7
- 40 CPUs Intel Xeon Silver 4114 CPU 2.20GHz
- Mellanox MT27700 Family ConnectX-4 100Gb/s adaptors
- bssplit 512/20:1k/16:2k/9:4k/12:8k/19:16k/10:32k/8:64k/4
- NVMEoF param_inline_data_size **16384**

# Backup: NVMEoF-related questions

- Where do performance differences between IBNBD and NVMEoF come from?