Implementation of block device drivers in userspace of modern general-purpose operating systems, although possible, is fairly uncommon, poorly supported and usually achieves only low performance. Being able to implement high-performance drivers in userspace with ease would allow for faster iterations in storage research and would make it possible to design block devices which operate in radically different ways.

In this thesis, we present Block Device in Userspace (BUSE), a Linux kernel module and communication protocol which makes it easy to develop userspace block-device drivers. Compared to the existing approaches, BUSE can scale on modern multicore architectures and provides at least 7x higher throughput with significantly simpler setup. Furthermore, the kernel module communicates with the userspace driver through shared memory, eliminating an extraneous memory copy. BUSE also solves the write-after-write and read-after-write consistency issues which stem from the use of multiple hardware queues in the Linux storage stack, allowing the implementation to focus on the domain of the problem.

As a proof-of-concept, we implemented Block Device in S3 (BS3), a userspace block device implementation backed by Amazon S3 (or any other S3-compatible storage) on top of BUSE. BS3 can be used as a generic disk providing a throughput of more than 10 GB/s, making it faster than the fastest possible locally-attached PCIe 3.0 4x NVMe SSDs. It is up to 130x faster than CloudBD, a commercial product advertising high performance. S3's durability of 99.999999999 % (eleven nines) translates into the durability of BS3 block devices. Furthermore, BS3 is prefix-consistent under all failure conditions, preserving file system crash consistency guarantees.