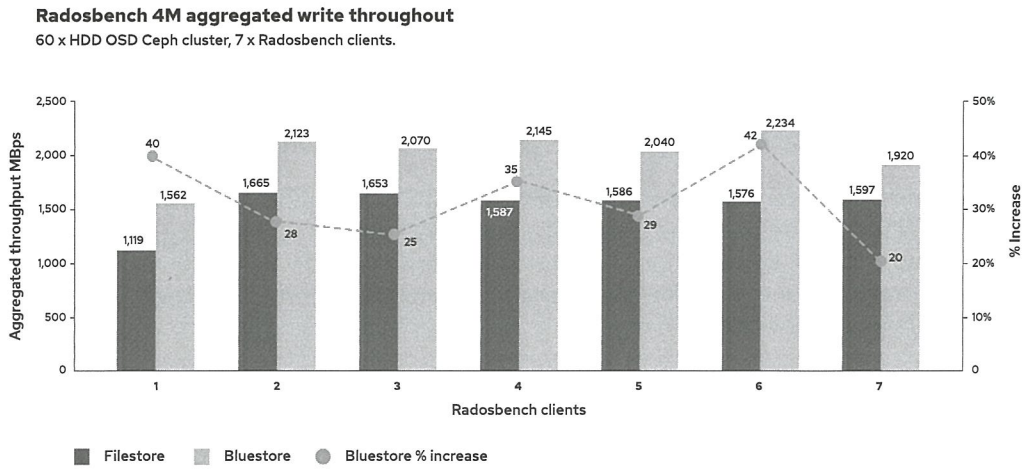

Figure 4.2: FileStore versus BlueStore write throughput

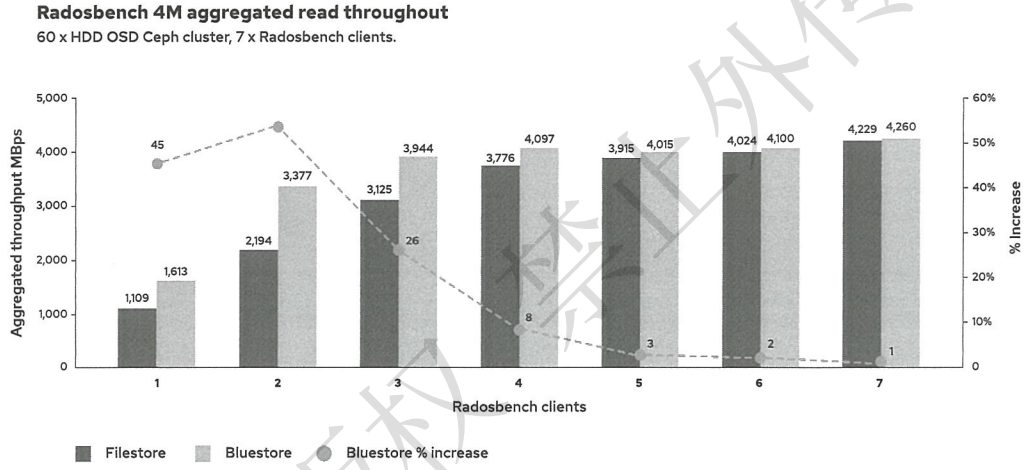

Figure 4.3: FileStore versus BlueStore read throughput

BlueStore runs in user space, manages its own cache and database, and can have a lower memory footprint than FileStore. BlueStore uses RocksDB to store key-value metadata. BlueStore is self-tuning by default, but you can manually tune BlueStore parameters if required.

The BlueStore partition writes data in chunks of the size of the `bluestore_min_alloc_size` parameter. The default value is 4 `KiB`. If the data to write is less than the size of the chunk, BlueStore fills the remaining space of the chunk with zeroes. It is a recommended practice to set the parameter to the size of the smallest typical write on the raw partition.

It is recommended to re-create FileStore OSDs as BlueStore to take advantage of the performance improvements and to maintain Red Hat support.

Introducing BlueStore Database Sharding

BlueStore can limit the size of large *omap* objects stored in RocksDB and distribute them into multiple column families. This process is known as *sharding*. When using sharding, the cluster groups keys that have similar access and modification frequency to improve performance and to save disk space. Sharding can alleviate the impacts of RocksDB compaction. RocksDB must reach a certain level of used space before compacting the database, which can affect OSD