Introduction of Multicore Architecture of Crimson

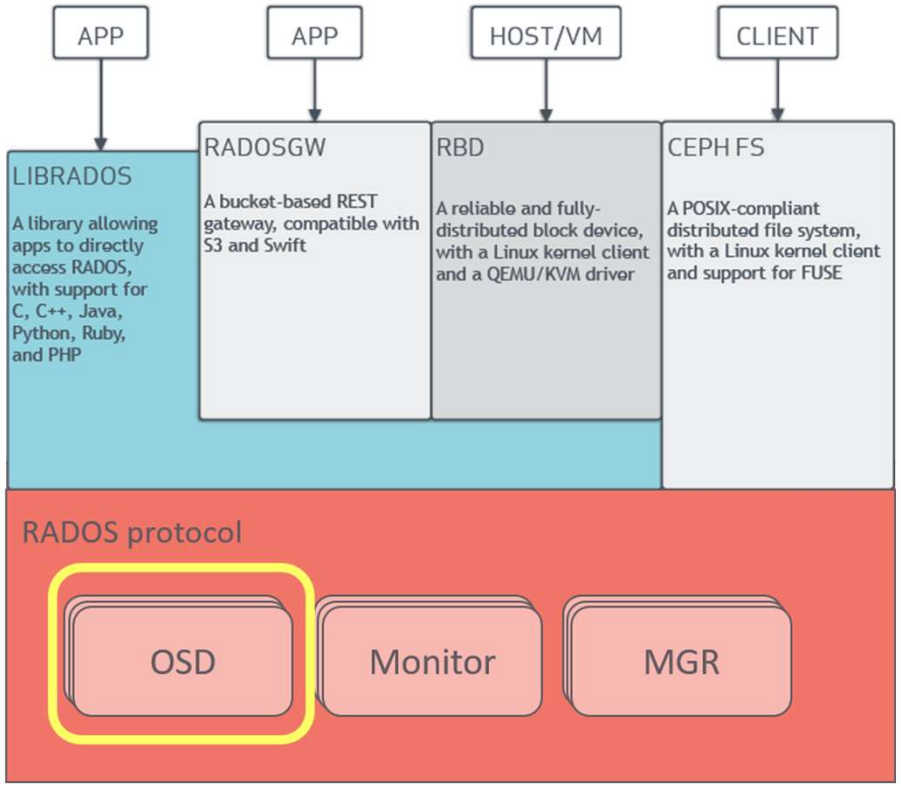
Liu, Chunmei chunmei.liu@intel.com

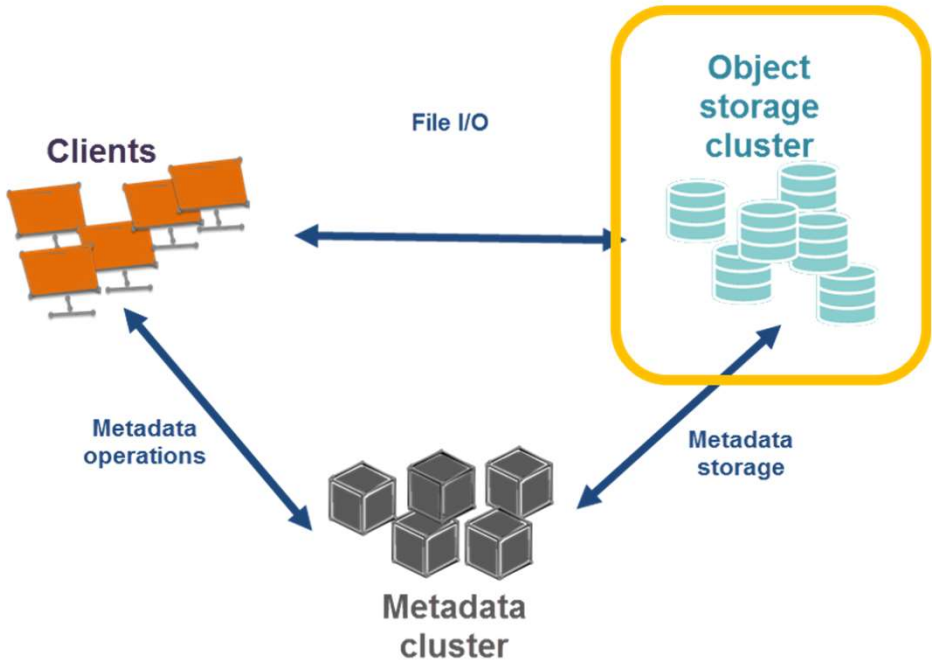
Cheng, Yingxin yingxin.cheng@intel.com

Contents

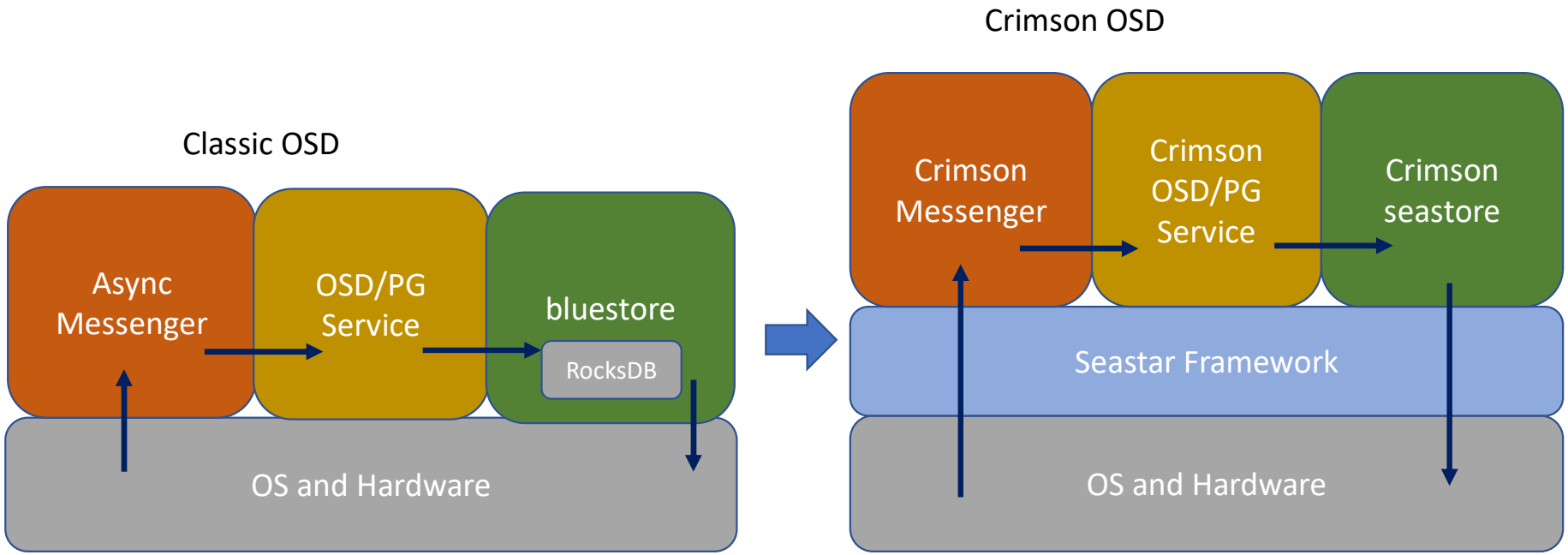
- Crimson Software Stack
- Crimson threads Model
- Crimson multicore messenger workflow
- Crimson multicore osd workflow
- Crimson multicore seastore workflow

Crimson position in Ceph

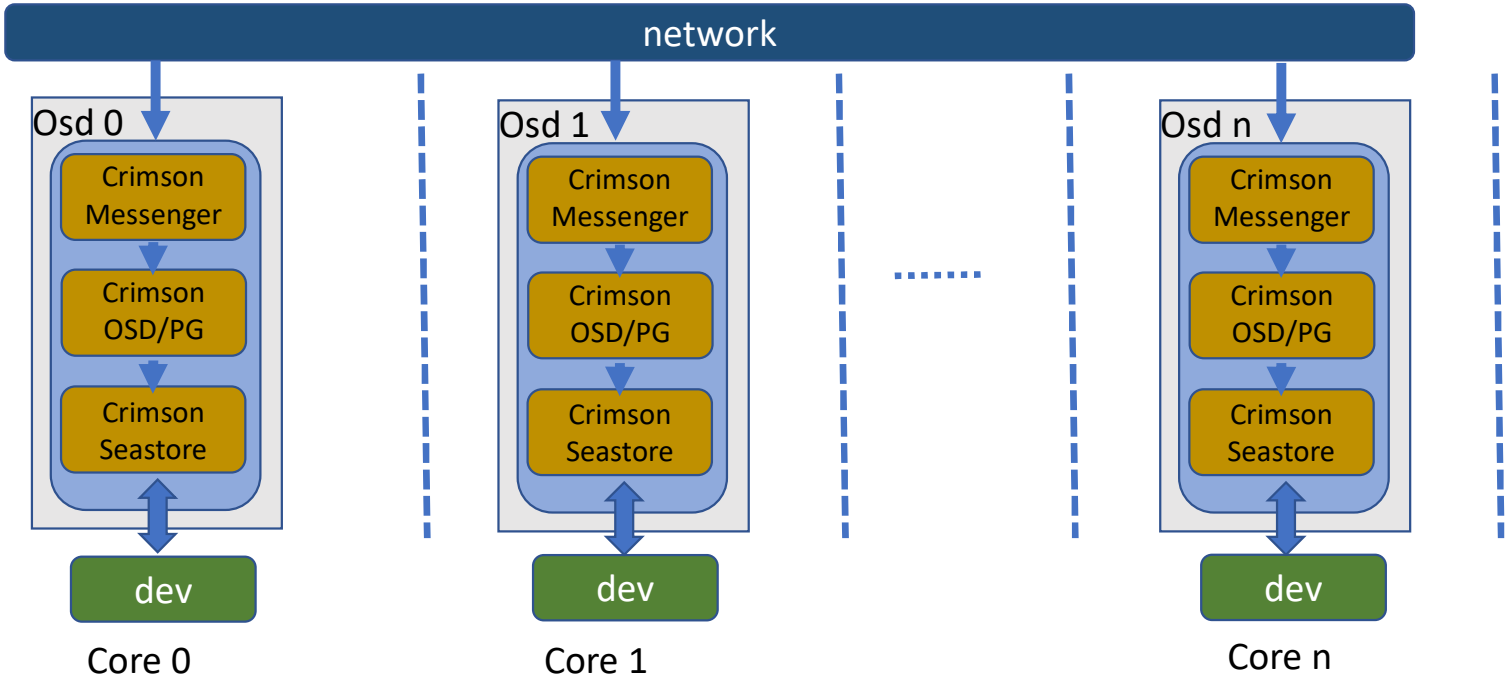




Crimson OSD Software Stack



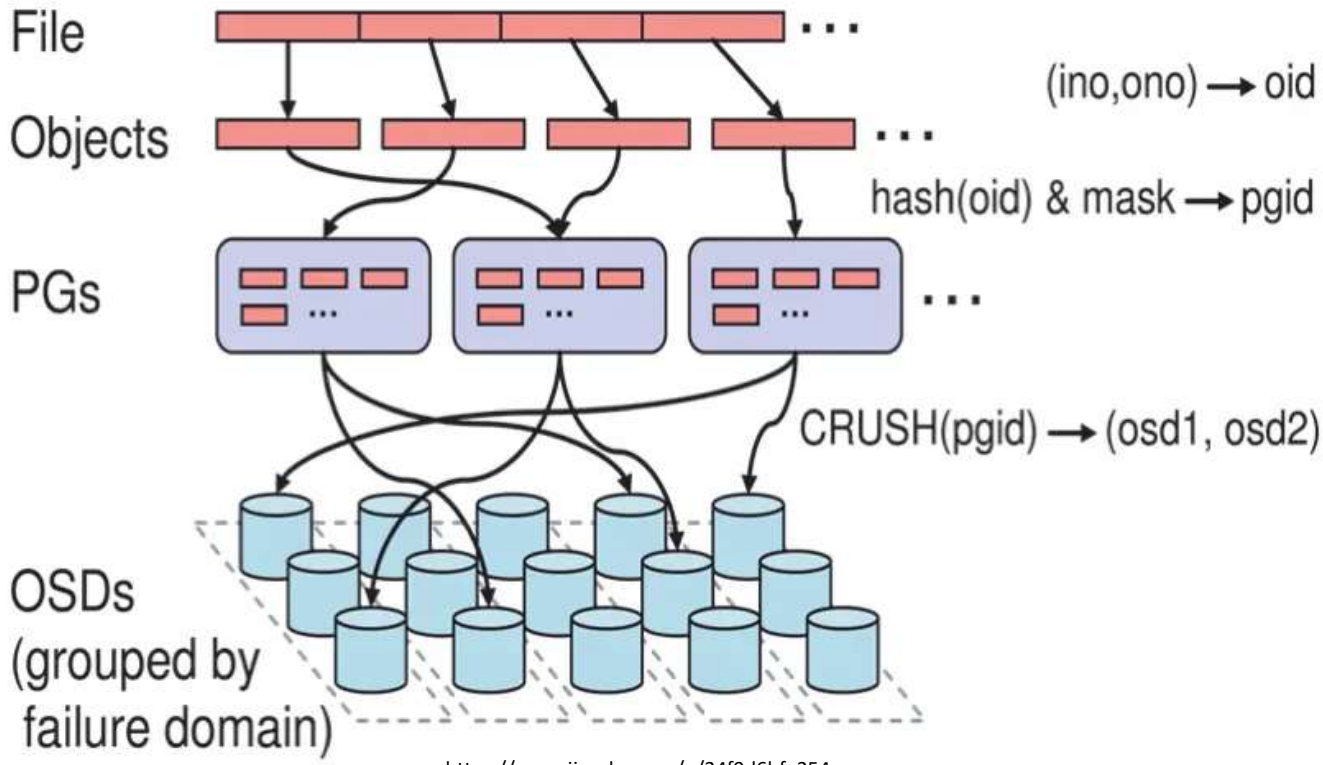
Single-core thread model



- Initial implement, per thread per OSD
- Simple, needn't cross-core communication for logic process
- Performance is low since it is single core
- No enough storage device to map each OSD
- Osdmap size is enlarged since osd instance is bigger

- Run-to-completion model

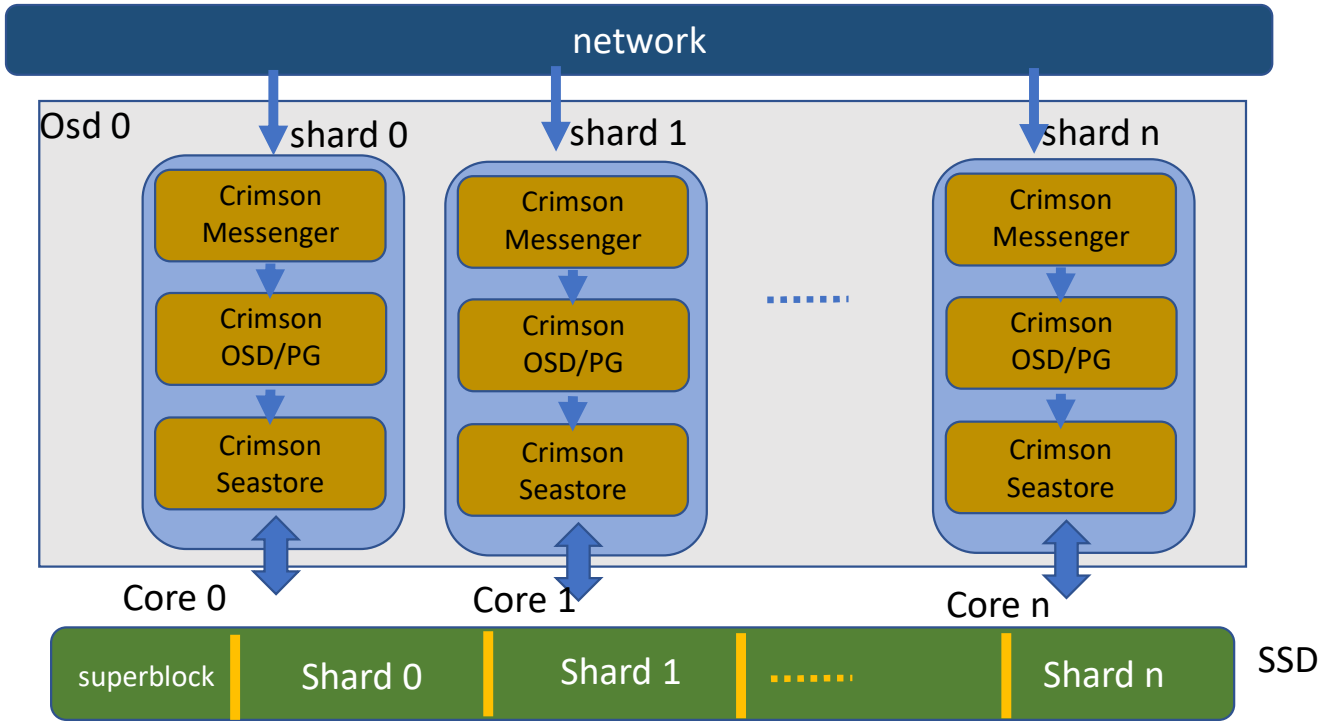
Ceph IO placement



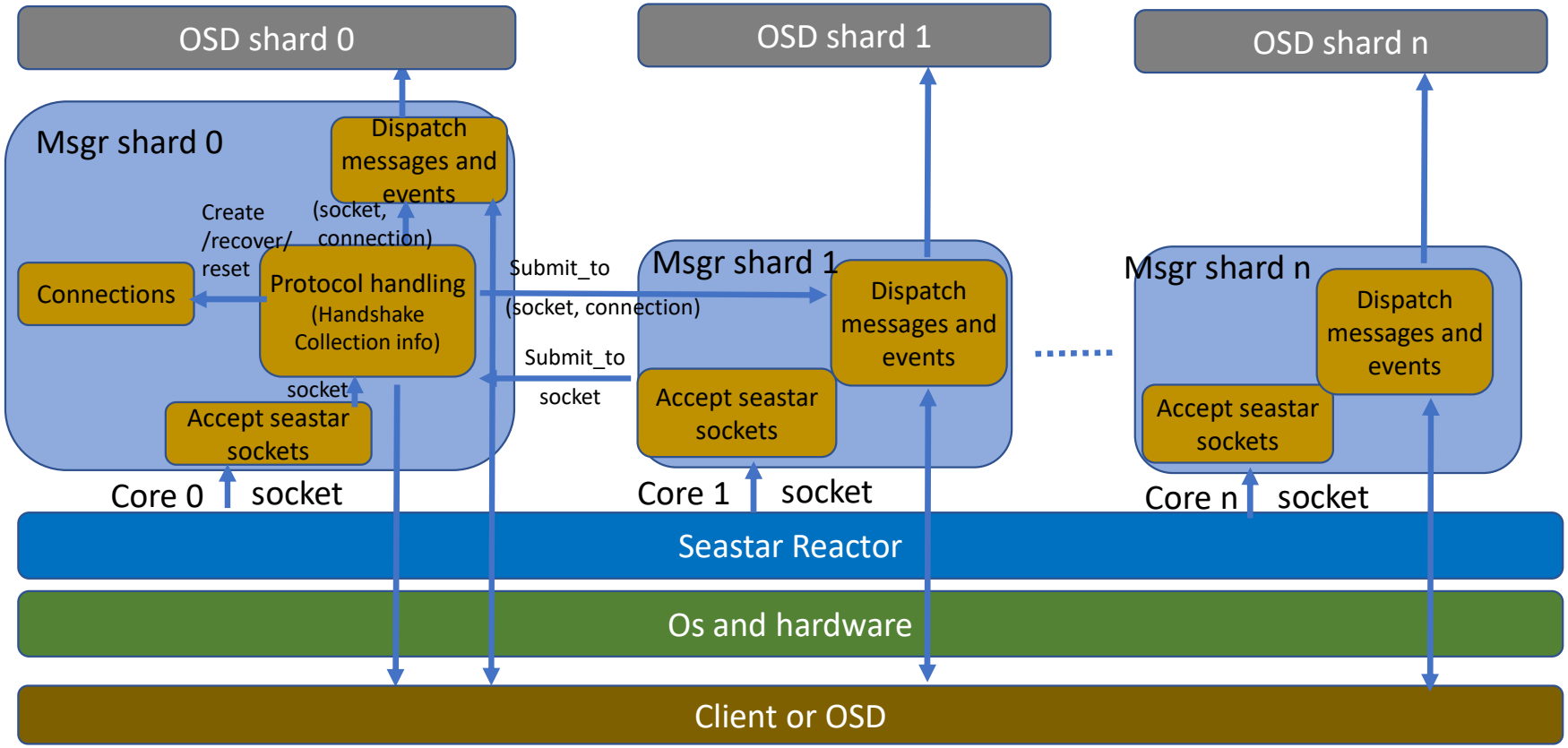
https://www.jianshu.com/p/24f9d6bfe254

Multicore threads model

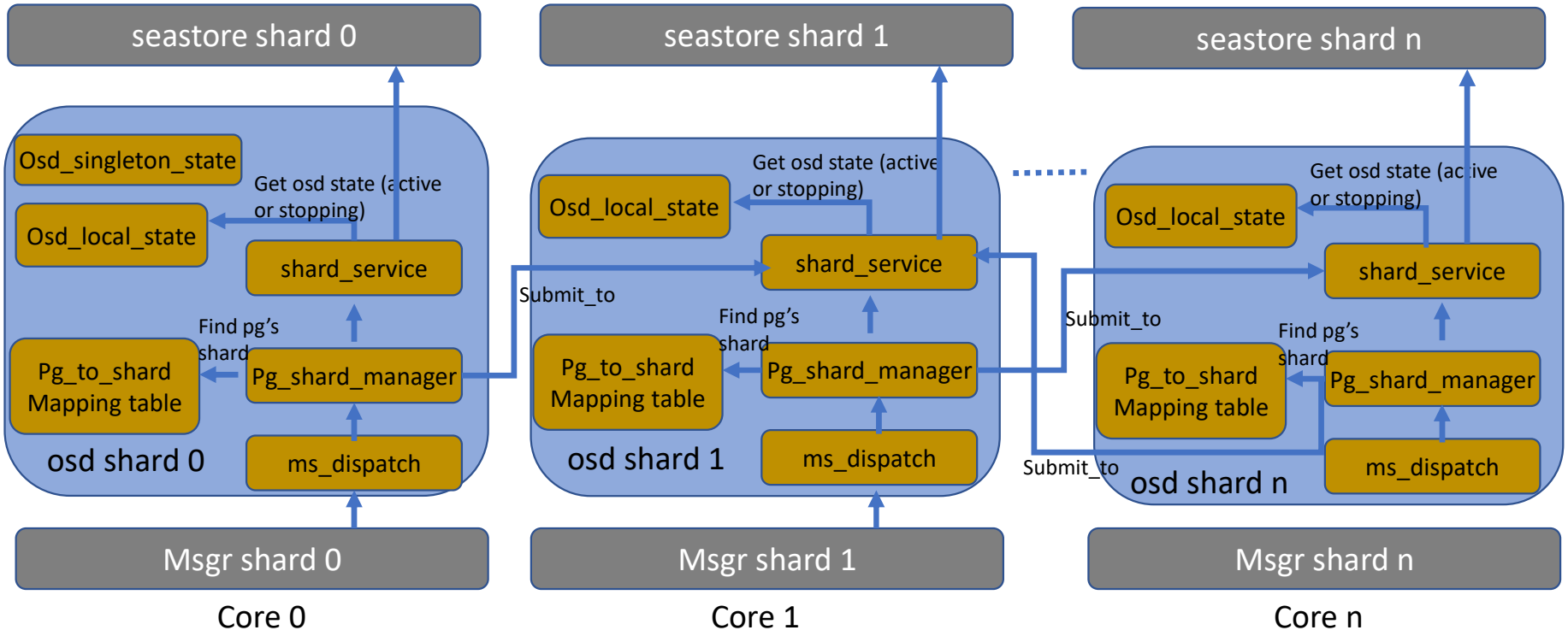
- final implement, multiple threads per OSD based on seastar sharded framework
- need cross-core communication
- Performance is better than single core since use more cores for per OSD
- Each shard share one device



Multicore messenger workflow



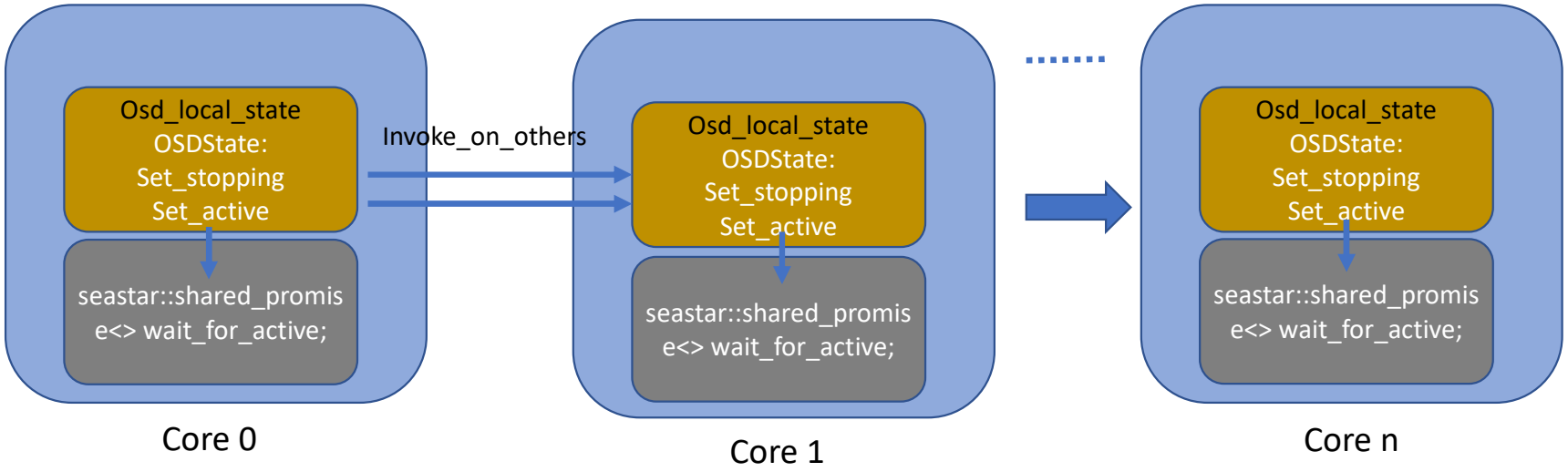
Multicore osd workflow



- According to pg to shard mapping, transfer request to another shard

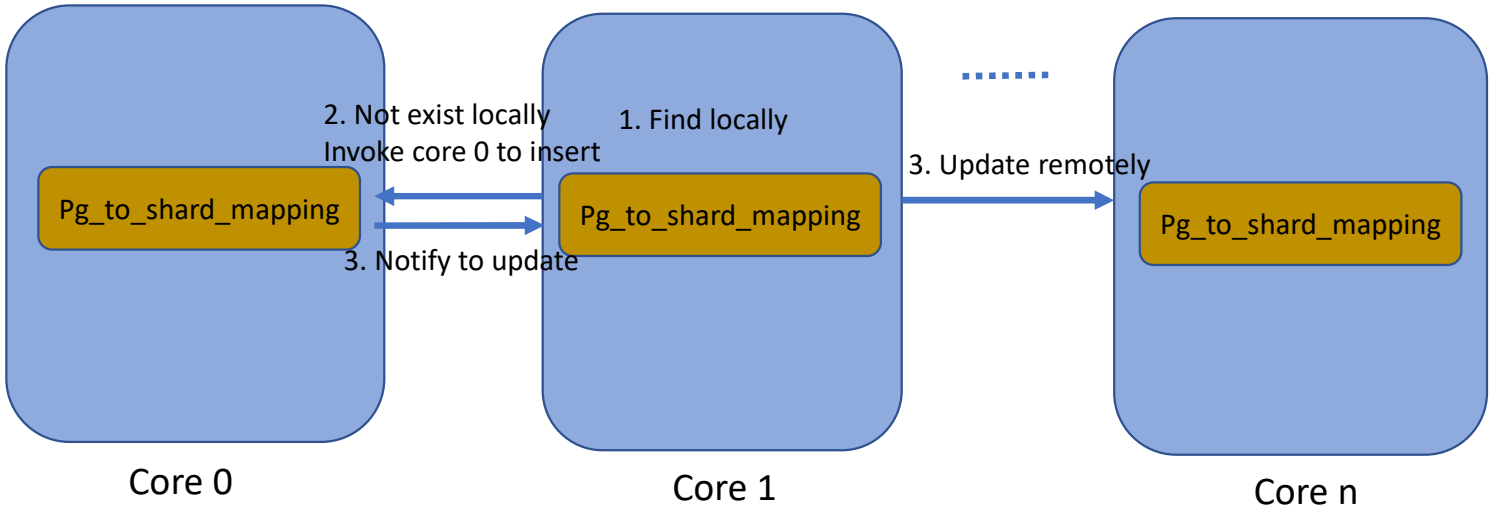
How global data sharing between shards

- Primary core update and notify others

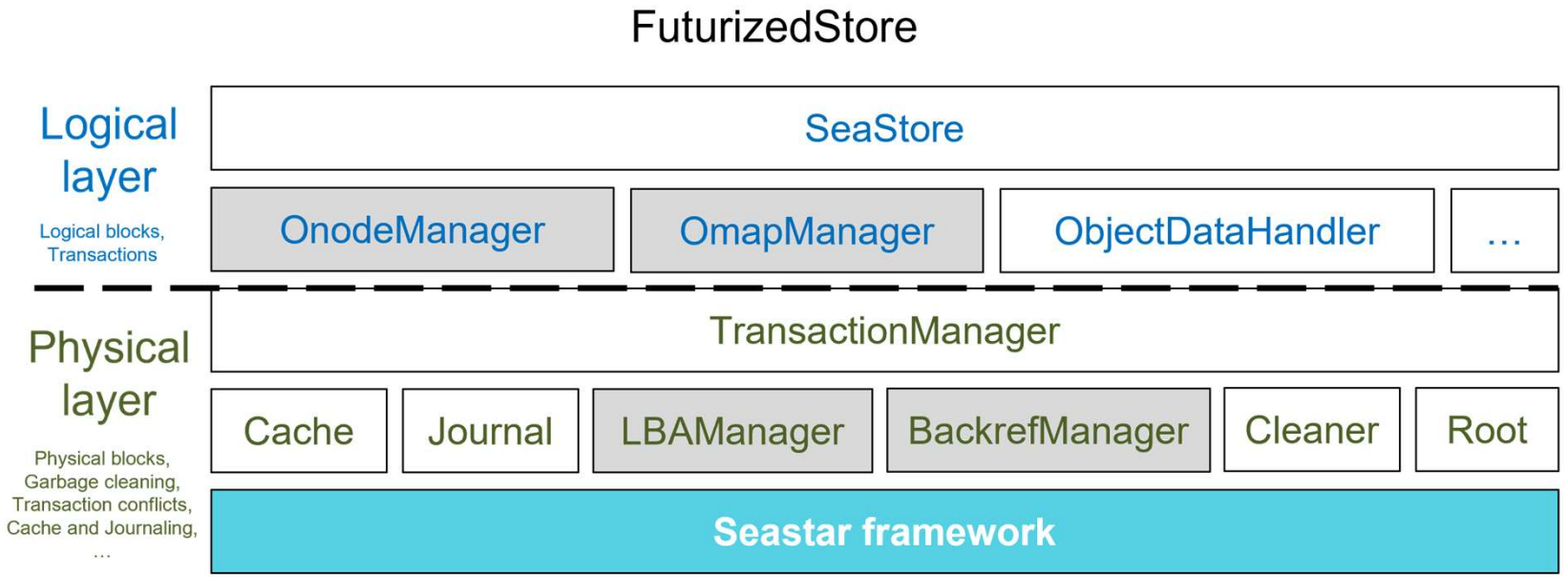


How global data sharing between shards

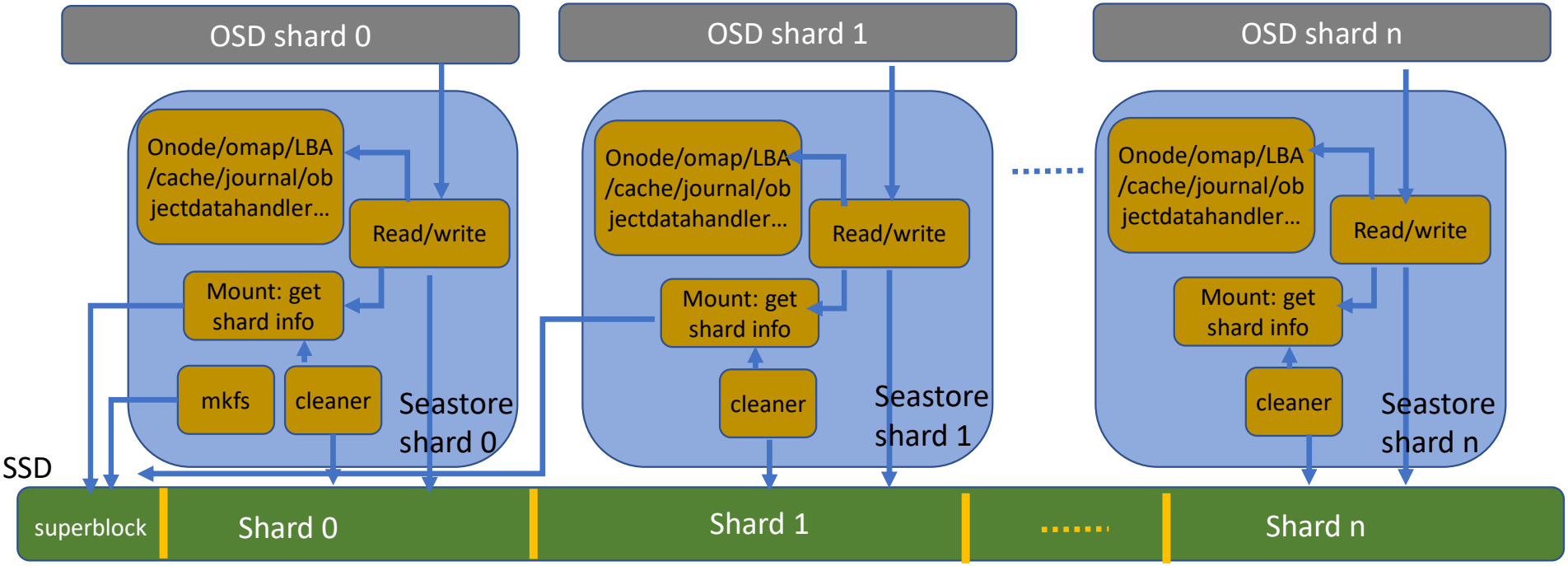
- Ask primary core to update and notify the others



Seastore workflow



Multicore Seastore workflow



Thank you