OPENFABRICS ALLIANCE

13th ANNUAL WORKSHOP 2017

CEPH RDMA UPDATE

Haomai Wang, CTO

XSKY

[March 28, 2017]

AGENDA

- **About**
- **Ceph Introduction**
- **Ceph Network Evolement**
- **Ceph RDMA Support**

ABOUT

- **I am Haomai Wang**
- **XSKY(A China Storage Startup)**
- **Active Ceph Developer**
- **Maintain AsyncMessenger and NVMEDevice module in Ceph**
- **haomaiwang@gmail.com**

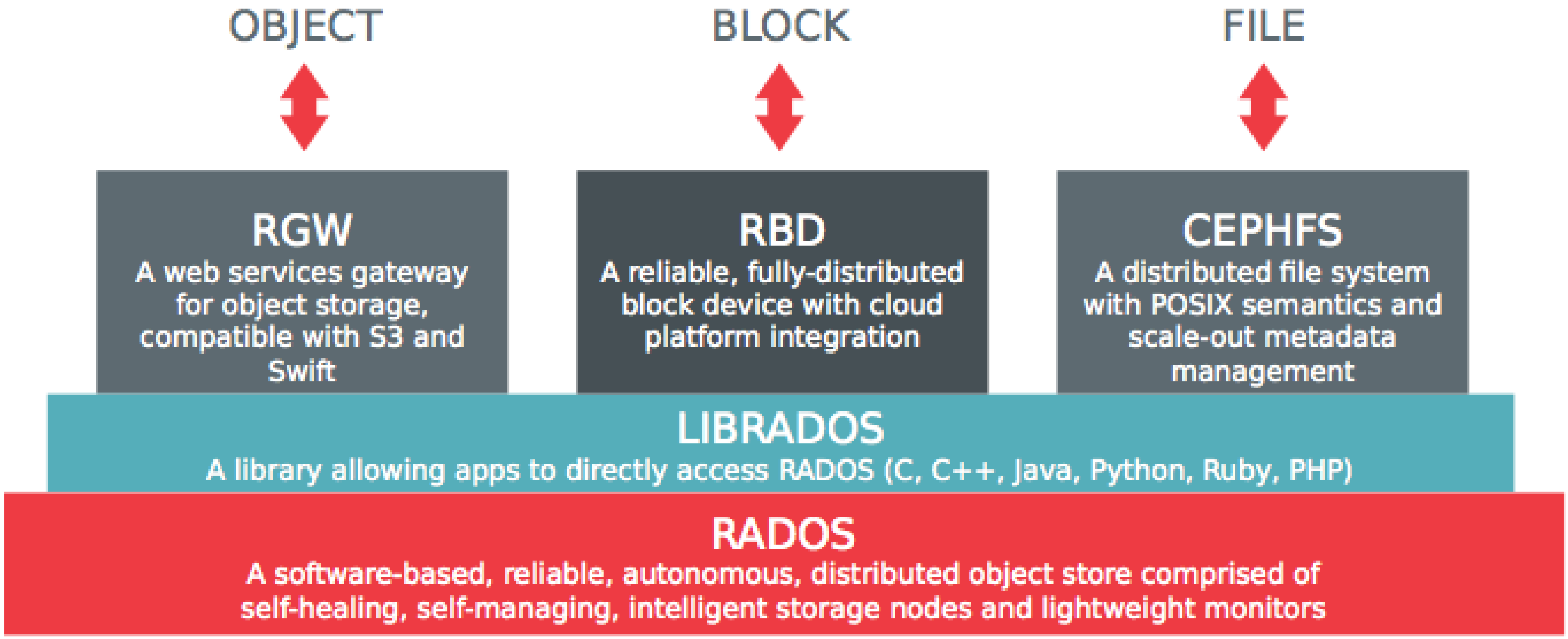
CEPH INTRODUCTION

CEPH INTRO

- Object, block, and file storage in a single cluster
- All components scale horizontally
- No single point of failure
- Hardware agnostic, commodity hardware Self-manage whenever possible
- Open source

- “A Scalable, High-Performance Distributed File System”
- “performance, reliability, and scalability”
- “Create The Ecosystem To Become The Linux Of Distributed Storage”

CEPH INTRO



OBJECT

BLOCK

FILE

RGW
A web services gateway for object storage, compatible with S3 and Swift

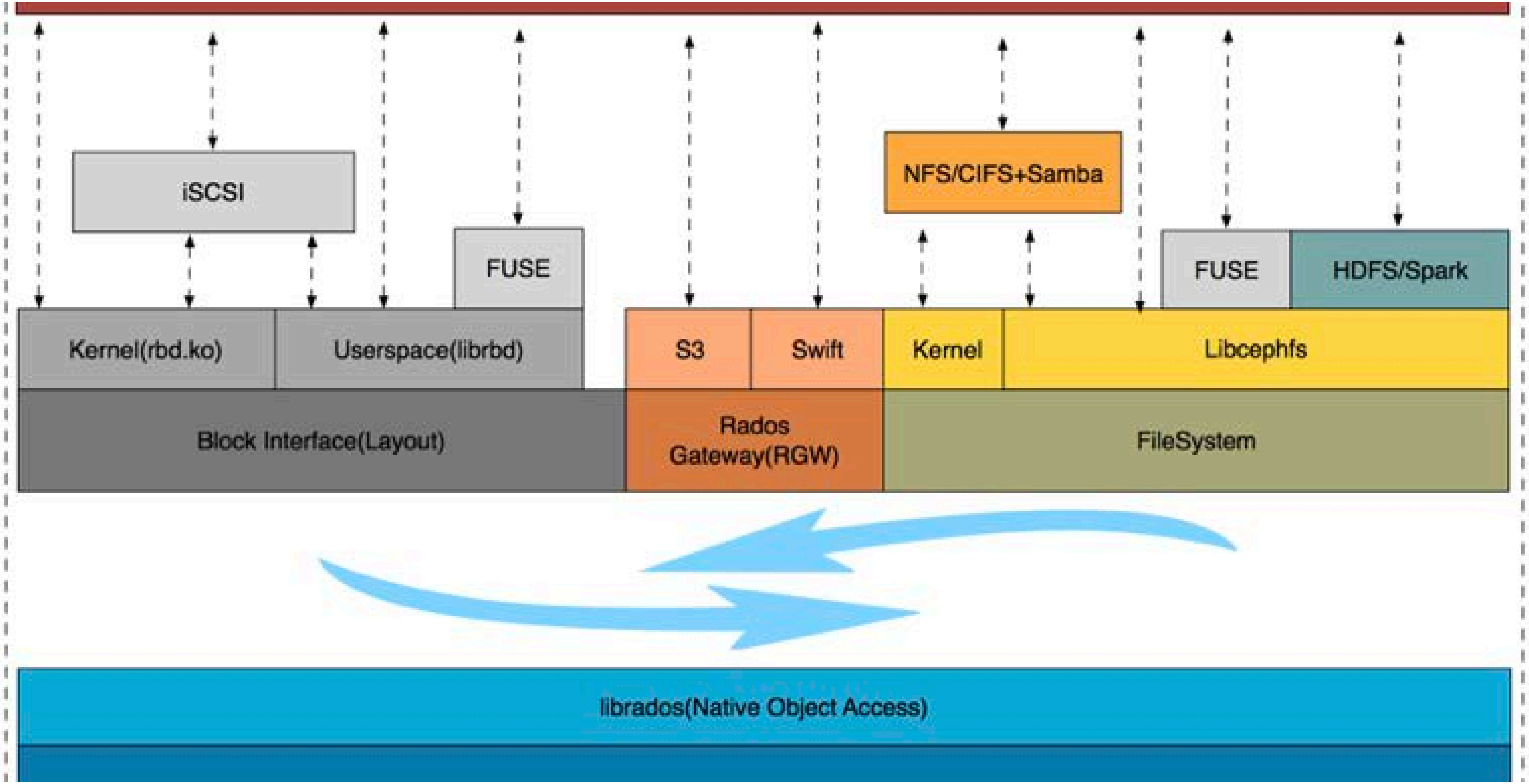
RBD
A reliable, fully-distributed block device with cloud platform integration

CEPHFS
A distributed file system with POSIX semantics and scale-out metadata management

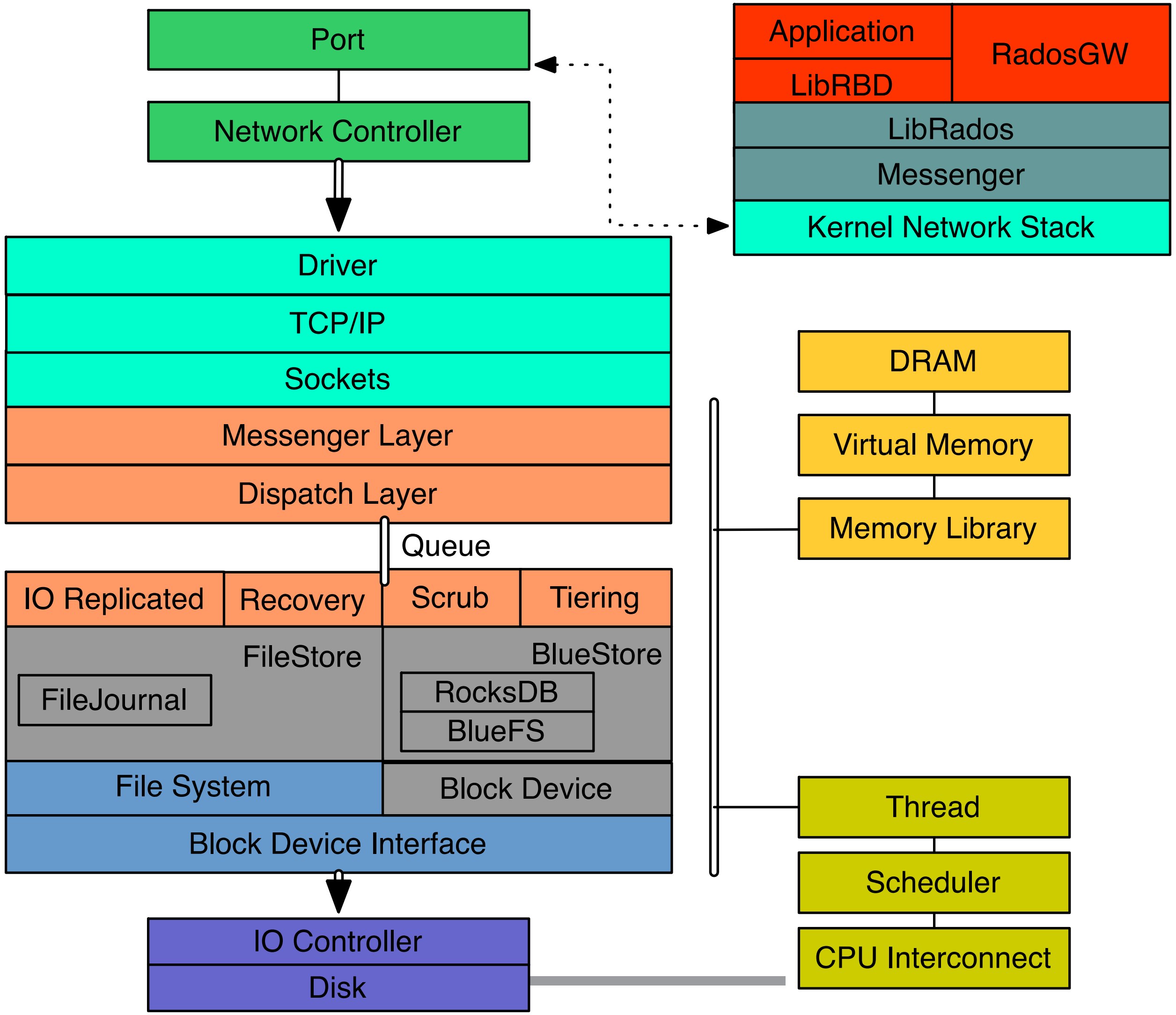
LIBRADOS
A library allowing apps to directly access RADOS (C, C++, Java, Python, Ruby, PHP)

RADOS
A software-based, reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes and lightweight monitors

CEPH INTRO

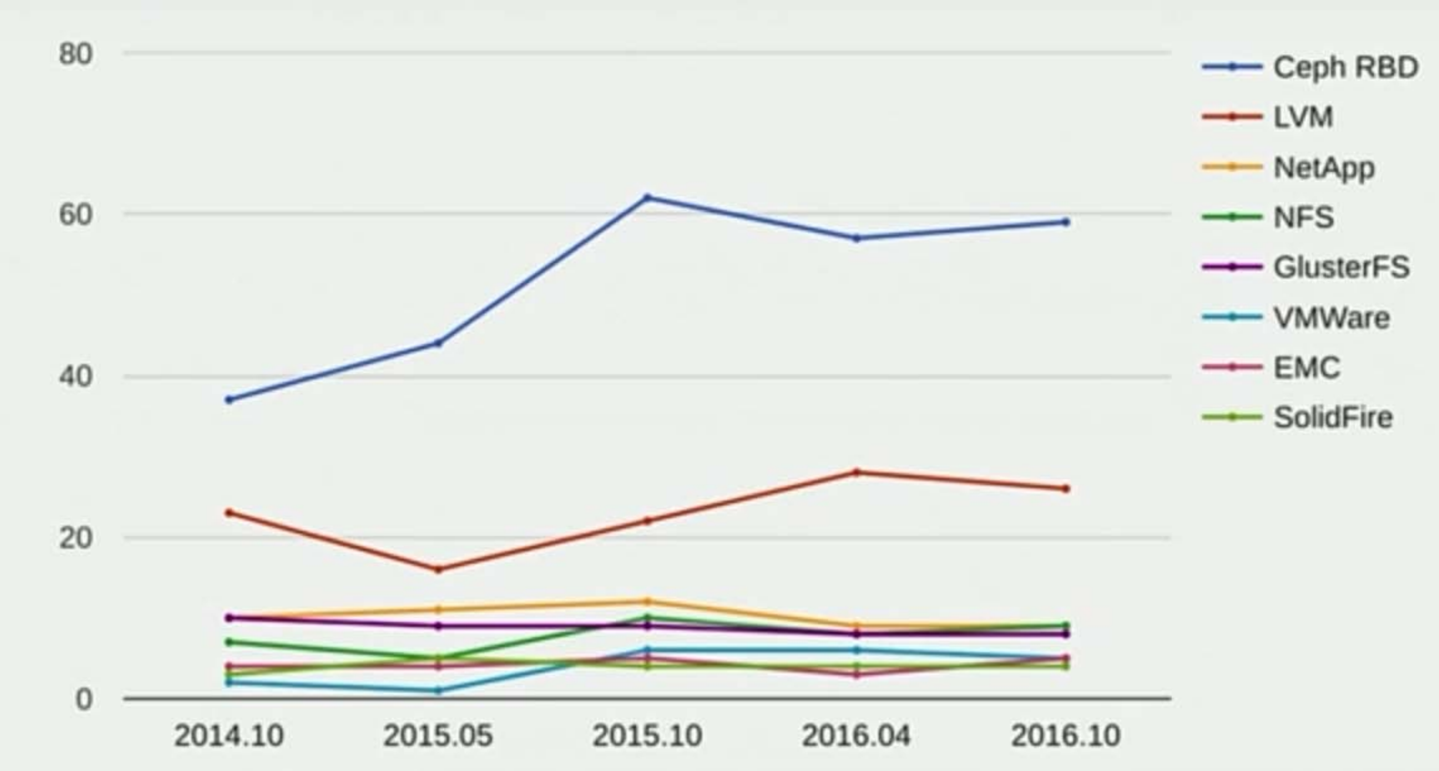

iSCSI
FUSE
NFS/CIFS+Samba
FUSE
HDFS/Spark
Kernel(rbd.ko)
Userspace(librbd)
S3
Swift
Kernel
Libcephfs
Block Interface(Layout)
Rados Gateway(RGW)
FileSystem
librados(Native Object Access)

CEPH INTRO


Port
Network Controller
Application
RadosGW
LibRBD
LibRados
Messenger
Kernel Network Stack
Driver
TCP/IP
Sockets
Messenger Layer
Dispatch Layer
Queue
IO Replicated
Recovery
Scrub
Tiering
FileStore
FileJournal
BlueStore
RocksDB
BlueFS
File System
Block Device
Block Device Interface
IO Controller
Disk
DRAM
Virtual Memory
Memory Library
Thread
Scheduler
CPU Interconnect

CEPH INTRO

- **User Cases**
  - OpenStack
  - KVM
  - Backup
  - Object Storage



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut iaculis interdum posuere. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut vel dignissim nisl. Donec egestas, urna a gravida varius, magna velit interdum lacus, eget vehicula enim leo et turpisLorem ipsum dolor sit amet, consectetur adipiscing elit. Ut iaculis interdum posuere.

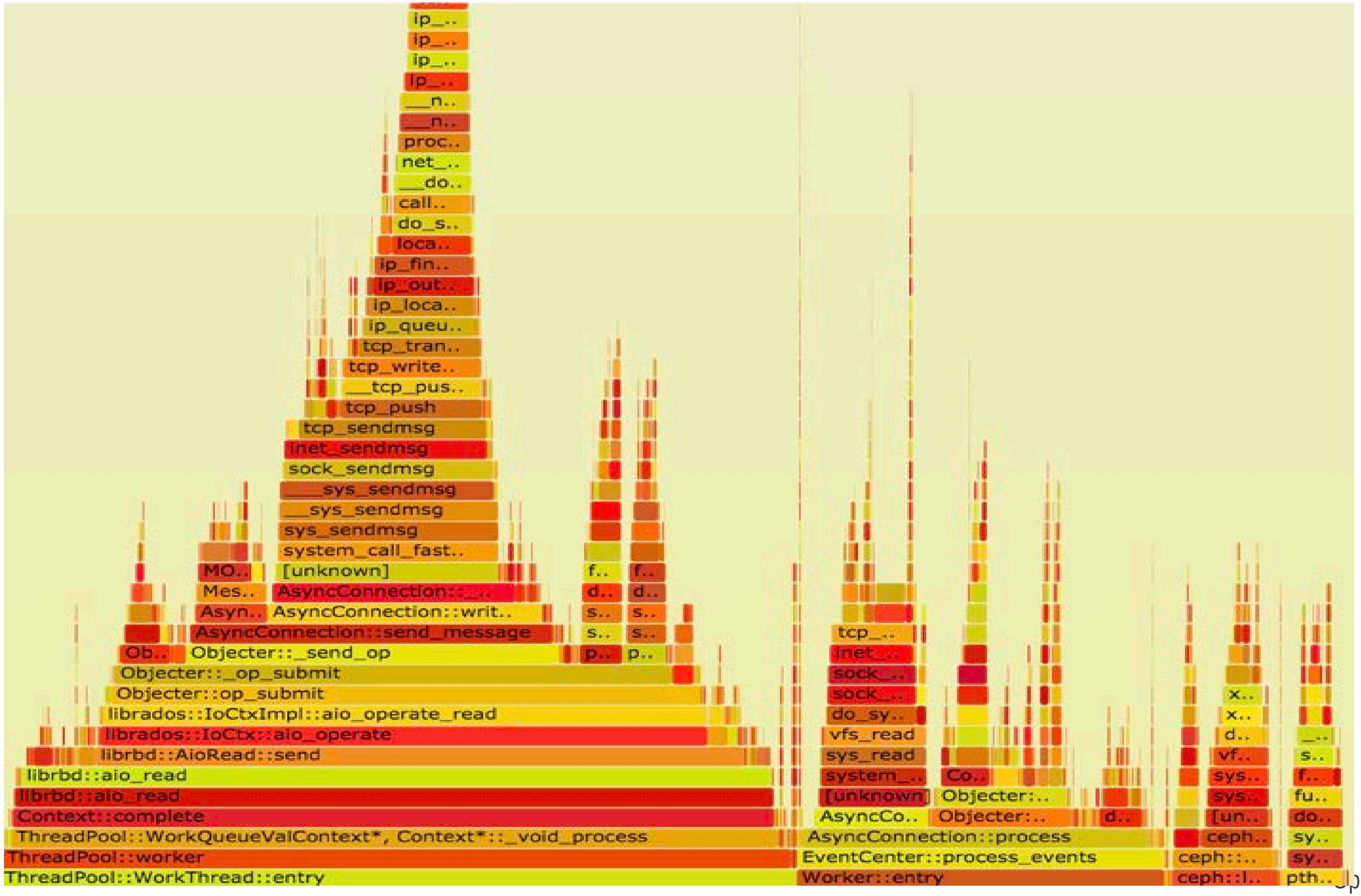
CEPH NETWORK EVOLVEMENT

CEPH NETWORK EVOLVEMENT

- **AsyncMessenger**
  - Core Library included by all components
  - Kernel TCP/IP driver
  - Epoll/Kqueue Drive
  - Maintain connection lifecycle and session

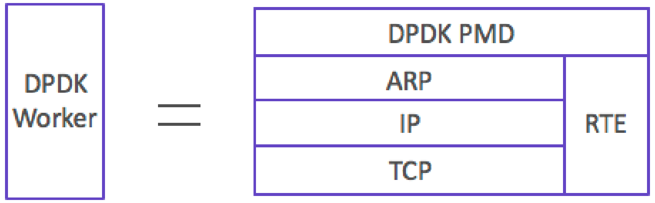
- **Performance Bottleneck:**
  - Non Local Process of Connections
    - RX in interrupt context
    - Application and system call in another
  - Global TCP Control Block Management
  - VFS Overhead
  - TCP protocol optimized for:
    - Throughput, not latency
    - Long-haul networks (high latency)
    - Congestion throughout
    - Modest connections/server

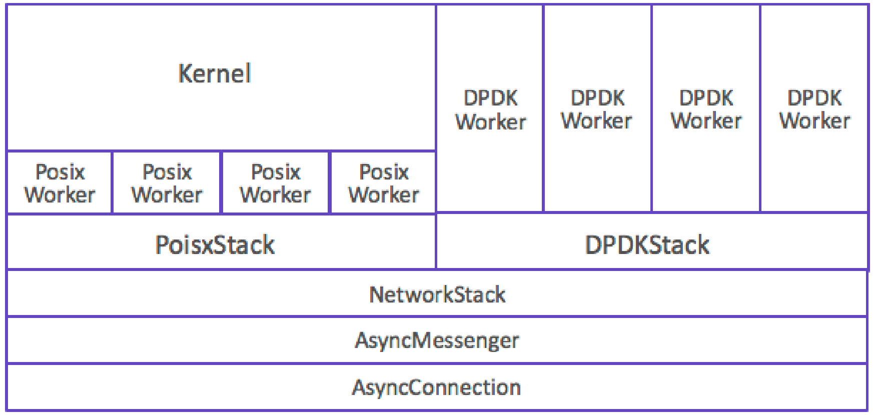
CEPH NETWORK EVOLVEMENT

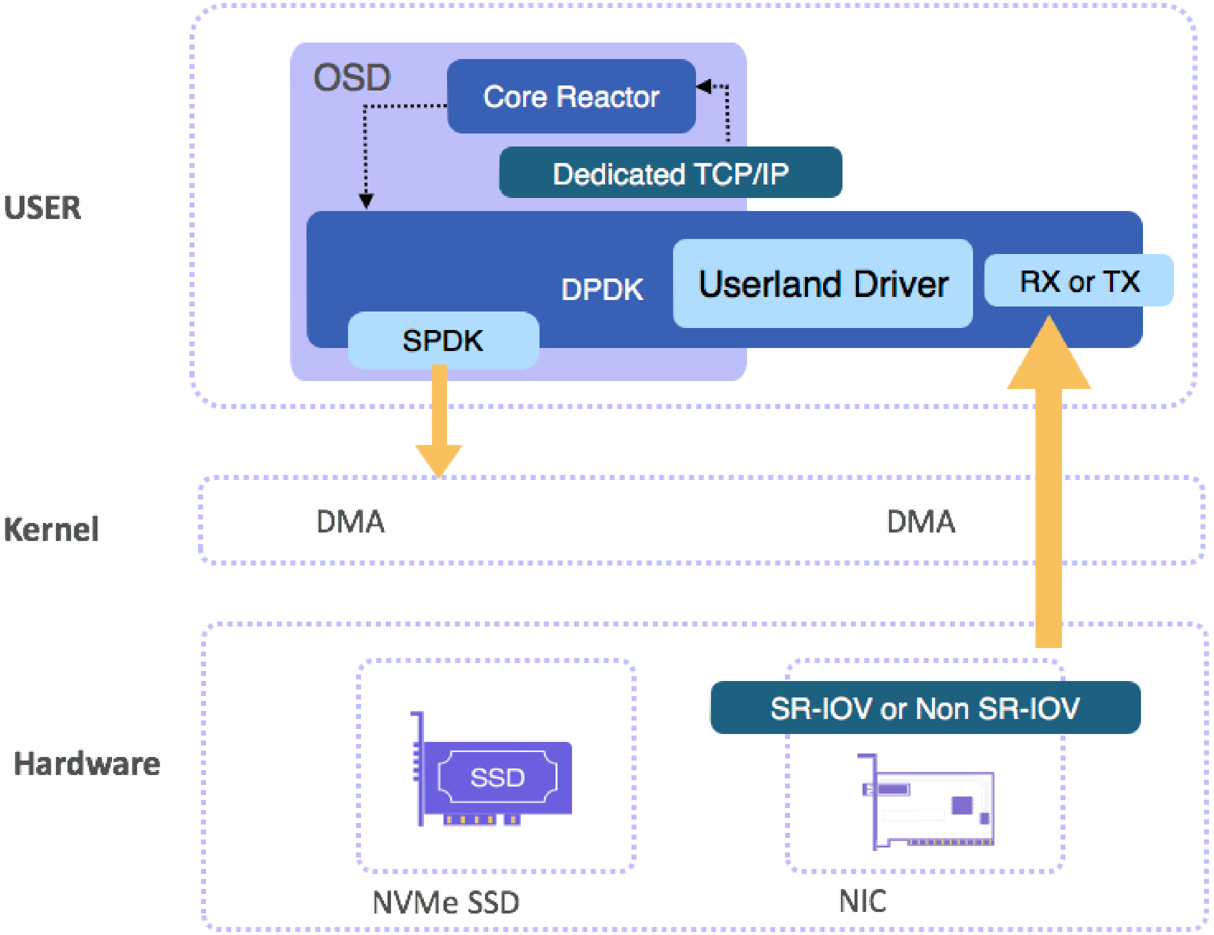


CEPH NETWORK EVOLVEMENT

- **Built for High Performance**
  - DPDK
  - SPDK
  - Full userspace IO path
  - Shared-nothing TCP/IP Stack(Seastar refer)







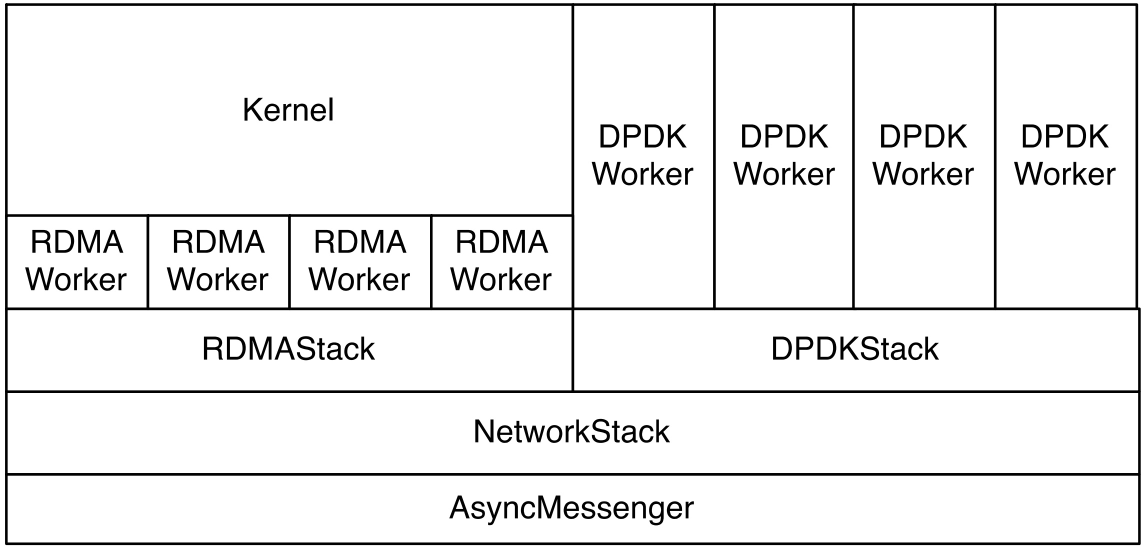
CEPH NETWORK EVOLVEMENT

- **Problems**
  - OSD Design
    - Each OSD own one disk
    - Pipeline model
    - Too much lock/wait in legacy
  - DPDK + SPDK
    - Must run on nvme ssd
    - CPU spining
    - Limited use cases

CEPH RDMA SUPPORT

CEPH RDMA

- **RDMA backend**
  - Inherit NetworkStack and implement RDMAStack
  - Using user-space verbs directly
  - TCP as control path
  - Exchange message using RDMA SEND
  - Using shared receive queue
  - Multiple connection qp's in many-to-many topology
  - Built-in into ceph master
  - All Features are fully avail on ceph master



Kernel

RDMA Worker | RDMA Worker | RDMA Worker | RDMA Worker

DPDK Worker | DPDK Worker | DPDK Worker | DPDK Worker

RDMAStack | DPDKStack

NetworkStack

AsyncMessenger

- **Support:**
  - RH/centos
  - INFINIBAND and ETH
  - Roce V2 for cross subnet
  - Front-end TCP and back-end RDMA

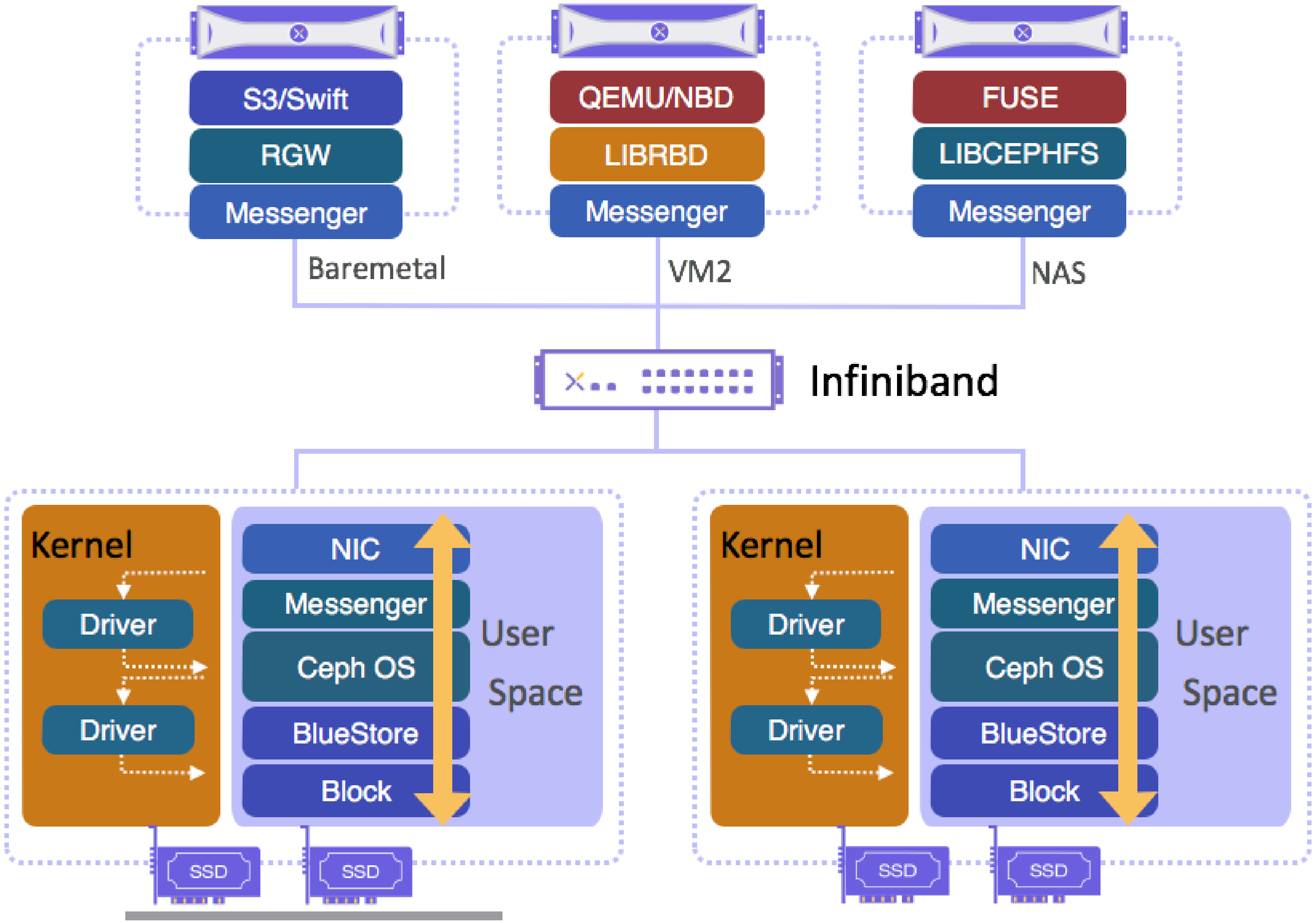
CEPH RDMA

- **Work in progress:**
  - RDMA-CM for control path
    - Support multiple devices
    - Enable unified ceph.conf for all ceph nodes
  - Ceph replication Zero-copy
    - Reduce number of memcpy by half by re-using data buffers on primary OSD
  - Tx zero-copy
    - Avoid copy out by using reged memory

- **ToDo:**
  - Use RDMA READ/WRITE for better memory utilization
  - ODP – On demand paging
  - Erasure-coding using HW offload

CEPH RDMA SUPPORT

- **Usages**
  - QEMU/KVM
  - NBD
  - FUSE
  - S3/Swift ObjectStorage
  - All ceph ecosystem



OPENFABRICS ALLIANCE

13th ANNUAL WORKSHOP 2017

THANK YOU

Haomai Wang, CTO

XSKY