Integrating SPDK in the NAS gateway

Erlang Li

Senior Storage Software Engineer

Integrated Storage Division
XSKY

XSKY

Outline

- Background
- Old Design of the NAS Gateway
- New Design of the NAS Gateway with SPDK
  - iSCSI based on SPDK
  - NVMe-of based on SPDK
- Performance comparison
  - Local Kernel VS NVMe-of VS iSCSI
  - Intel Optane VS NAND NVMe
  - Old System VS New System
- Add JSON-RPC Methods for NVMe SSD status

Outline

- Background

The “Appliance” Vs. “Software-only” crazy illogical circle

“Can I have an integrated Hardware/ Software Appliance?”

你能提供软件和硬件集成的一体机吗？

“I don 't want an appliance, I want software I can deploy on any industry-standard hardware”

我不想要一体机，我想要能够部署在标准服务器上的软件

“What hardware do you as the software vendor recommend?”

作为软件厂商，有推荐的硬件吗？

“What hardware have you tested on?”

你们的软件在什么硬件上测试过？

“If I have a problem with the hardware, who will I call?”

如果硬件有问题，我应该咨询谁？

“Can you provide industry-standard hardware that you would support?”

你能提供你们支持的硬件吗？

来源：https://virtualgeek.typepad.com/virtual_geek/2015/03/is-the-dress-white-and-gold-or-blue-and-black-sds-server-or-appliance.html

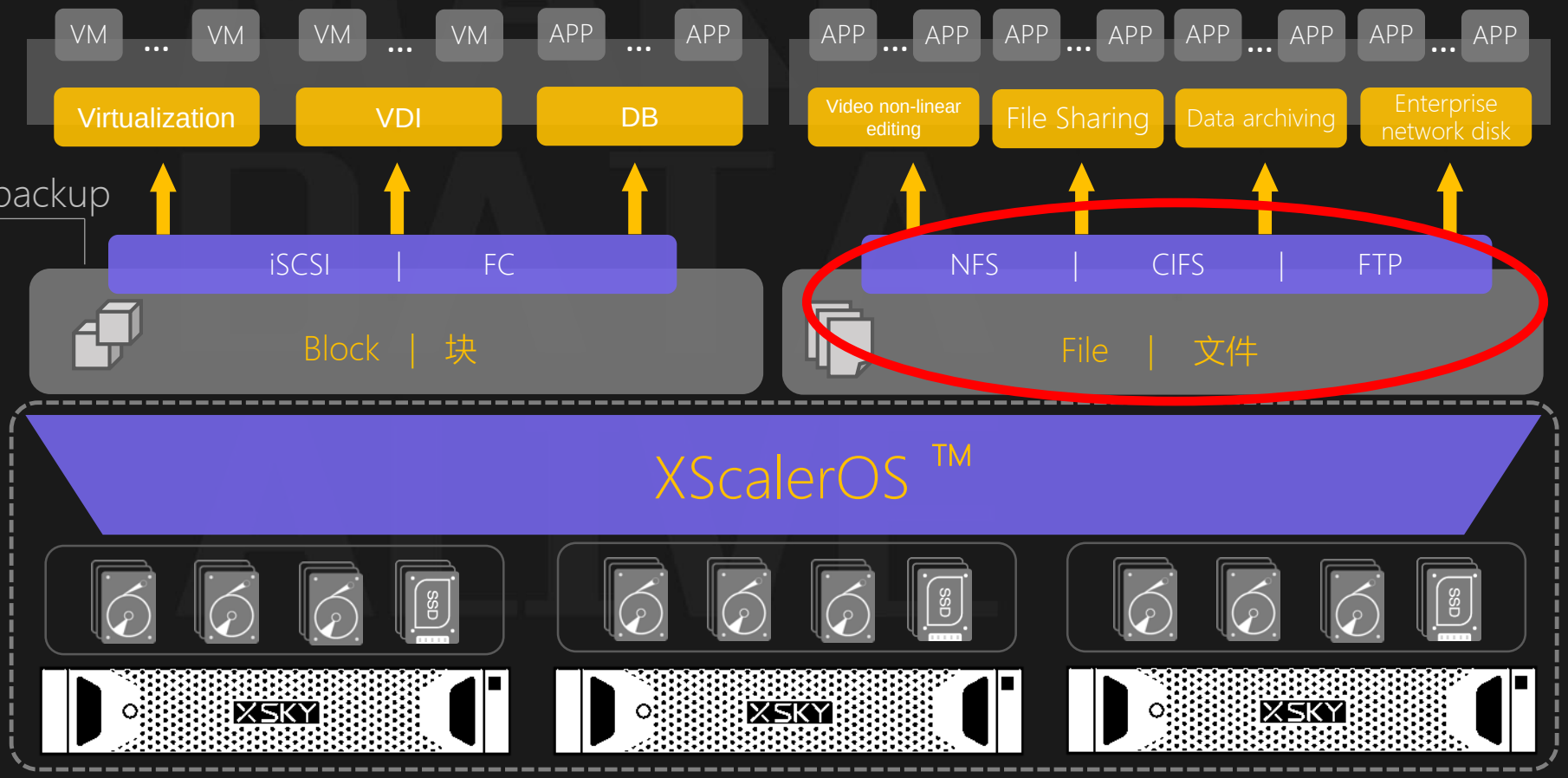
通向敏捷 IT 的理想选择

XSCALER EXPRESS 2000系列
软件定义存储一体机

- 软件定义存储引领IT变革
- 无缝对接主流企业级应用
- 经济易捷助力IT快速转型

XE2000 series focuses on the storage needs of multi-scenarios in growing companies

Public cloud backup



Typical scenes

- Virtualization and VDI
- Unified storage
- Data backup
- File management and sharing

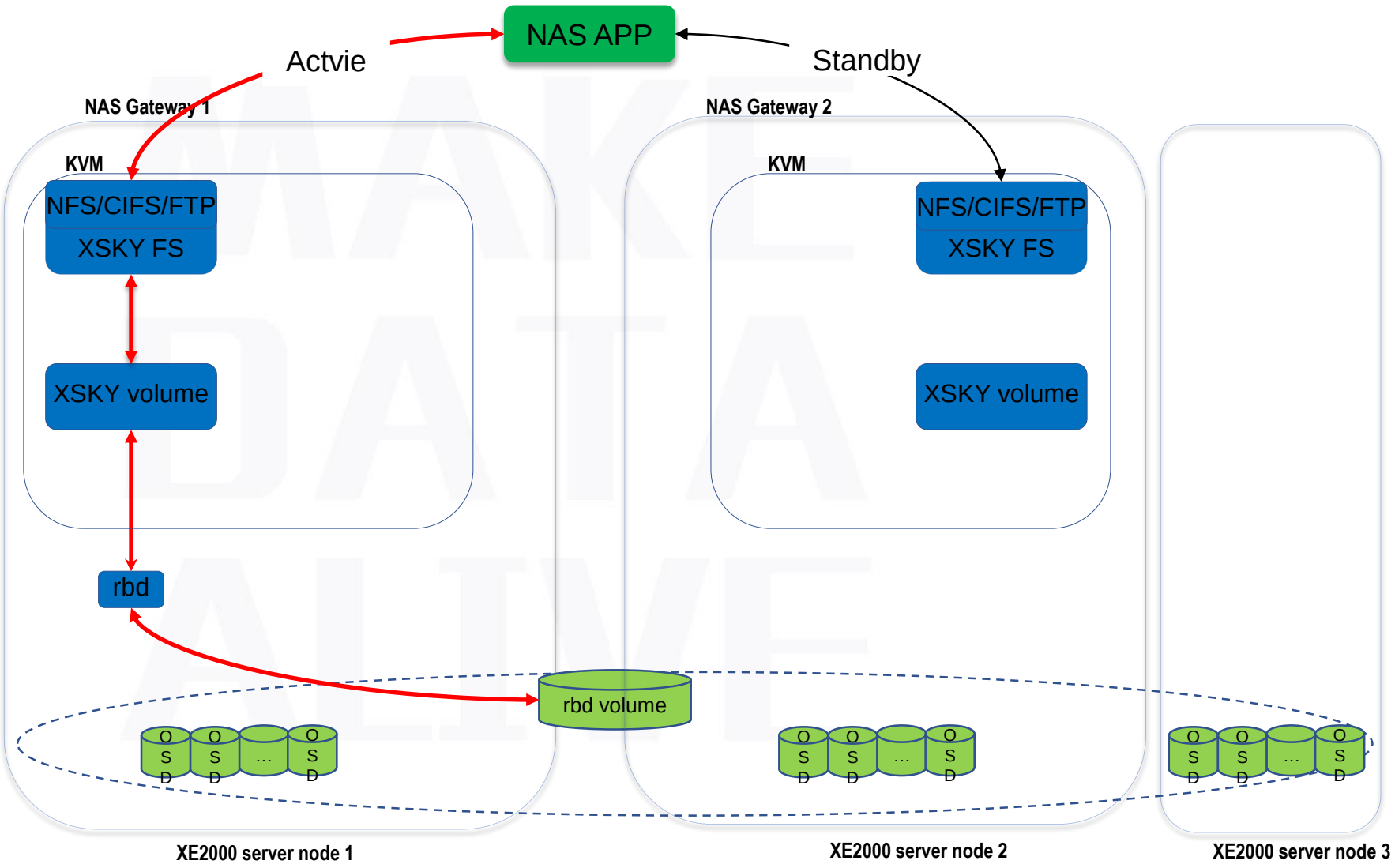
XSKY

Outline

- Background
- Old Design of the NAS Gateway

Old Design of the NAS Gateway

- AS-NAS (Active-Standby)
- KVM based
- Performance based on cluster

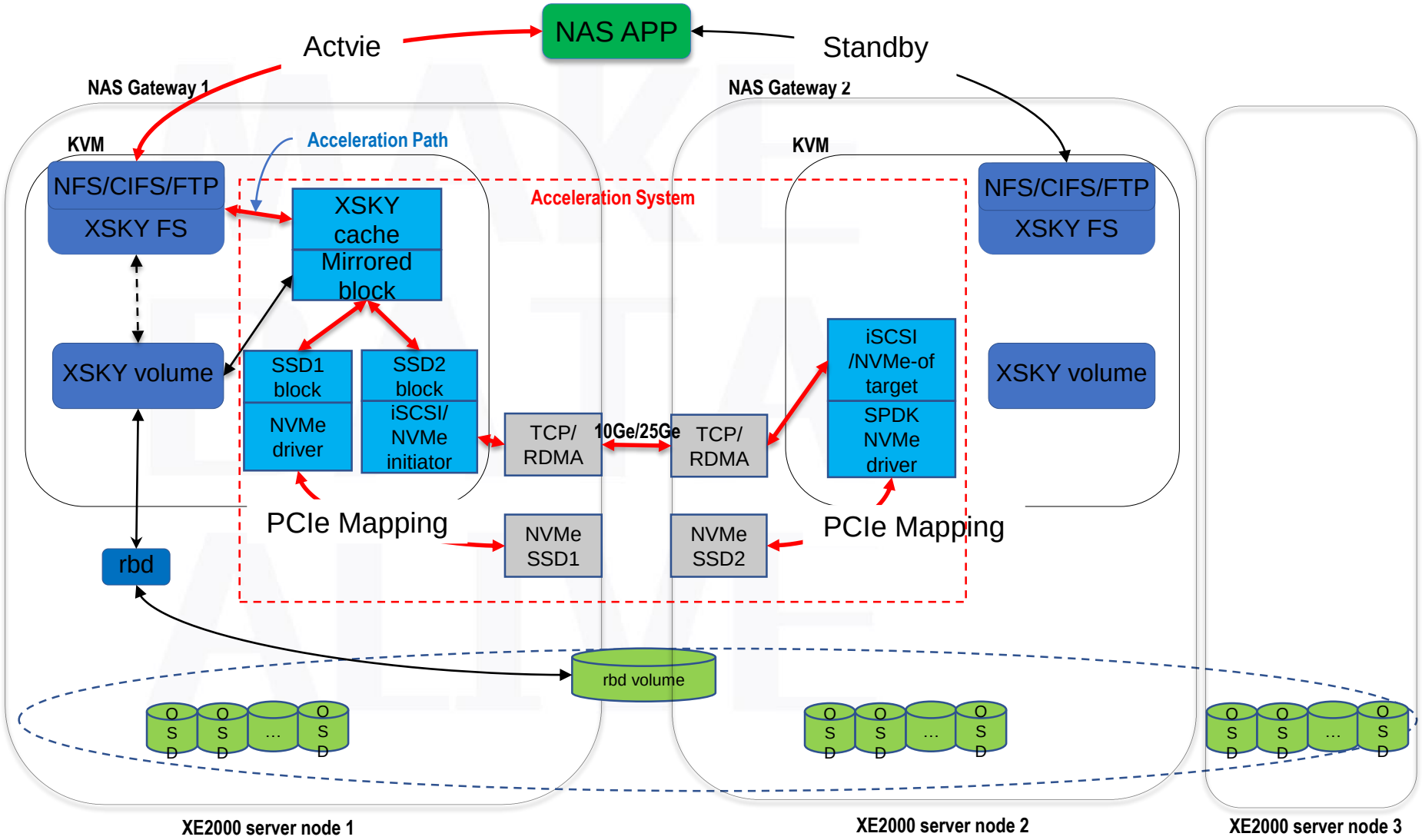


Outline

- Background
- Old Design of the NAS Gateway
- New Design of the NAS Gateway with SPDK
    - iSCSI based on SPDK
    - NVMe-of based on SPDK

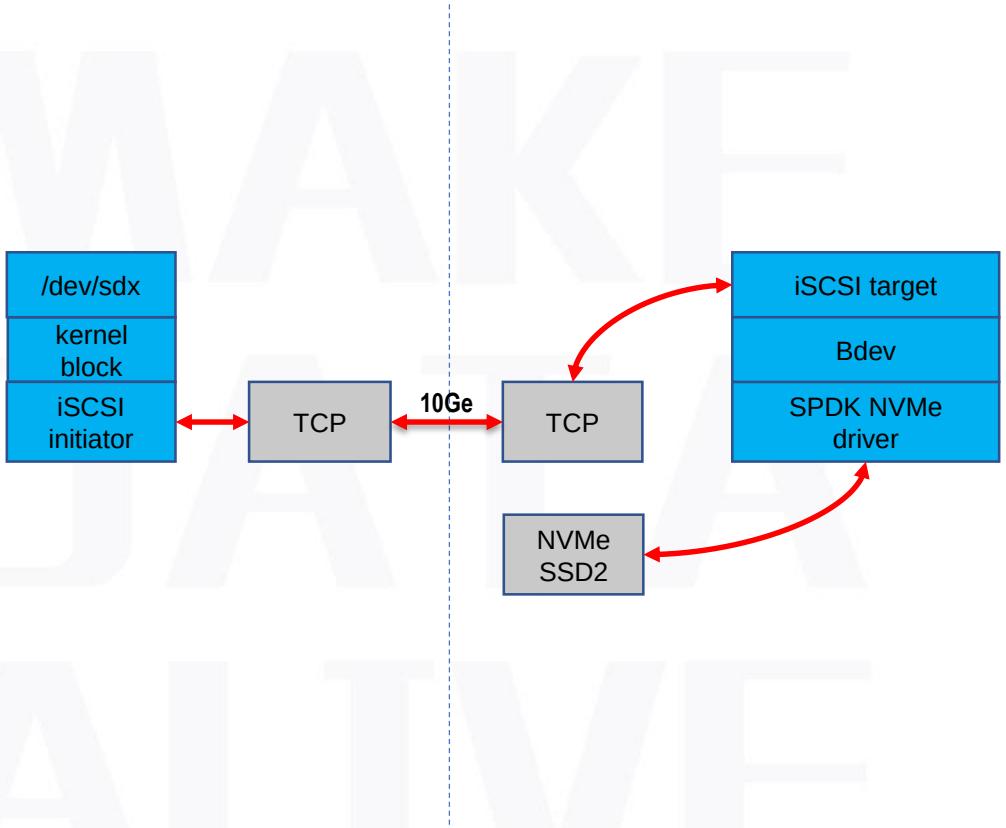
New Design of the NAS Gateway with SPDK

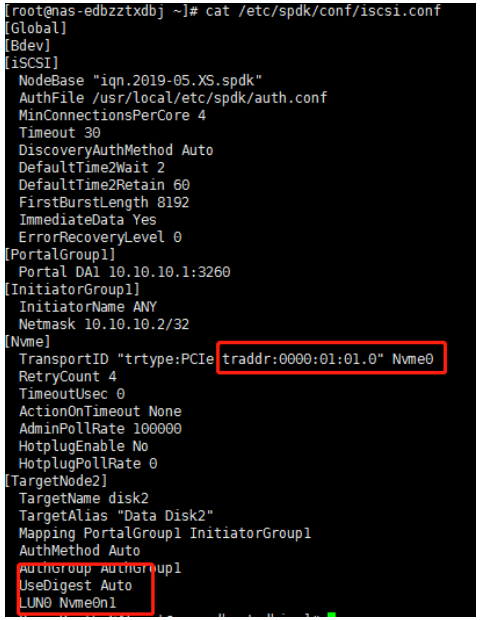
- Performance based on NVMe&&Network
- HA
- Optional protocol stacks(iSCSI or NVMe-of) with different hardware



iSCSI based on SPDK

- iSCSI protocol based on TCP
- Hardware:
  - bonding 10Ge Network
  - NAND NVMe SSD
- Soft:
  - SPDK iscsi tgt
  - Kernel iSCSI initiator





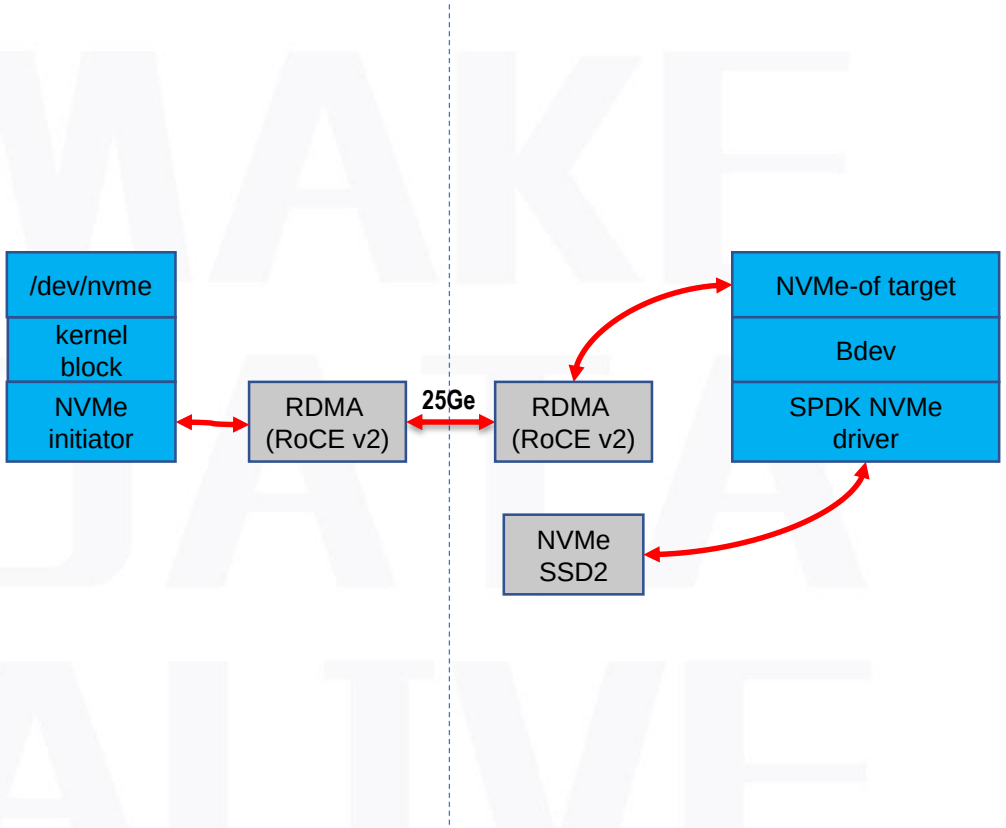
```
[root@nas-edbzztxdbj ~]# cat /etc/spdk/conf/iscsi.conf
[Global]
[Bdev]
[iSCSI]
  NodeBase "iqn.2019-05.XS.spdk"
  AuthFile /usr/local/etc/spdk/auth.conf
  MinConnectionsPerCore 4
  Timeout 30
  DiscoveryAuthMethod Auto
  DefaultTime2Wait 2
  DefaultTime2Retain 60
  FirstBurstLength 8192
  ImmediateData Yes
  ErrorRecoveryLevel 0
[PortalGroup1]
  Portal DA1 10.10.10.1:3260
[InitiatorGroup1]
  InitiatorName ANY
  Netmask 10.10.10.2/32
[Nvme]
  TransportID "trtype:PCIe traddr:0000:01:01.0" Nvme0
  RetryCount 4
  TimeoutUsec 0
  ActionOnTimeout None
  AdminPollRate 100000
  HotplugEnable No
  HotplugPollRate 0
[TargetNode2]
  TargetName disk2
  TargetAlias "Data Disk2"
  Mapping PortalGroup1 InitiatorGroup1
  AuthMethod Auto
  AuthGroup AuthGroup1
  UseDigest Auto
  LUN0 Nvme0n1
```

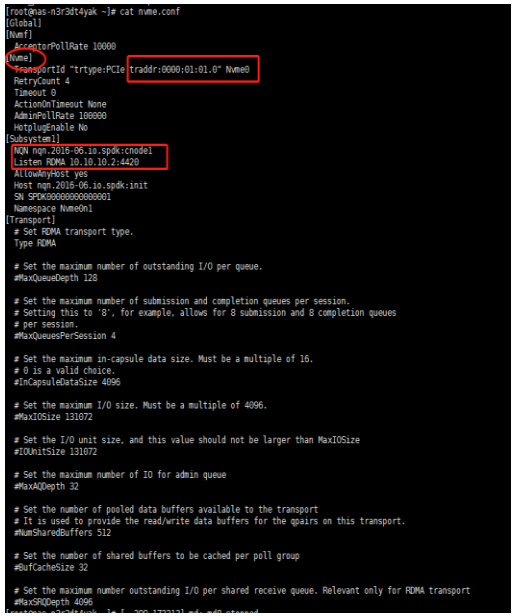
Limited cost reach considerable performance

NVMe-of based on SPDK

- NVMe-of protocol based on RDMA(RoCE v2)
- Hardware:
  - bonding 25Ge/40Ge Network
  - NAND NVMe SSD or Optane SSD
- Soft:
  - SPDK nvme-of tgt
  - Kernel nvme-of initiator

Pursue better performance





Outline

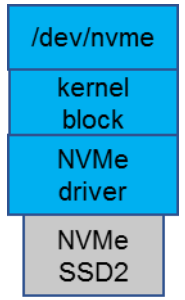
- Background
- Old Design of the NAS Gateway
- New Design of the NAS Gateway with SPDK
  - iSCSI based on SPDK
  - NVMe-of based on SPDK
- Performance comparison
  - Local Kernel VS NVMe-of VS iSCSI
  - Intel Optane VS NAND NVMe
  - Old System VS New System

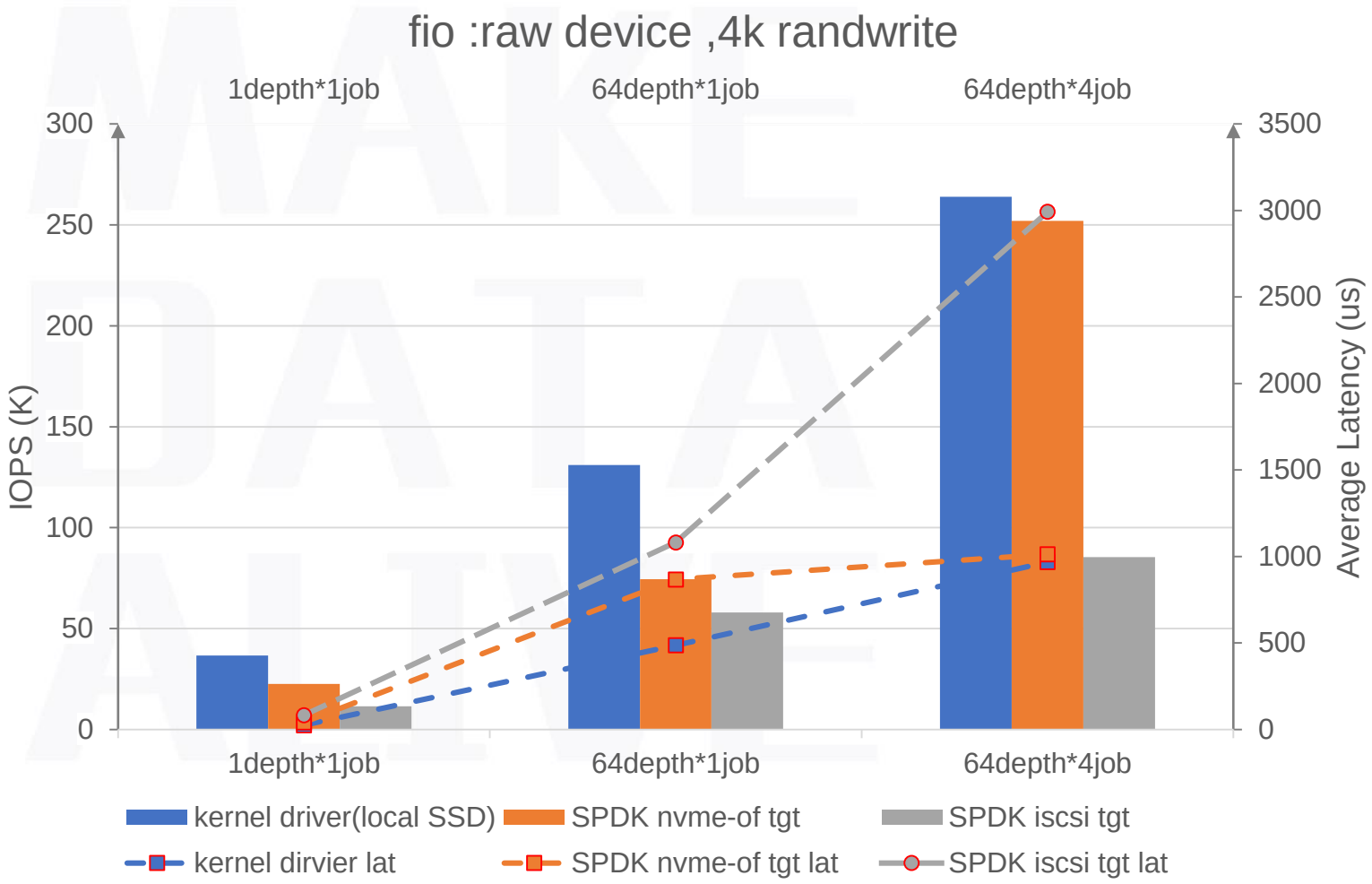
Local Kernel VS NVMe-of VS iSCSI

➢ Hardware:

- 25Ge Ethernet controller
- NAND NVMe SSD

Local kernel is just baseline:

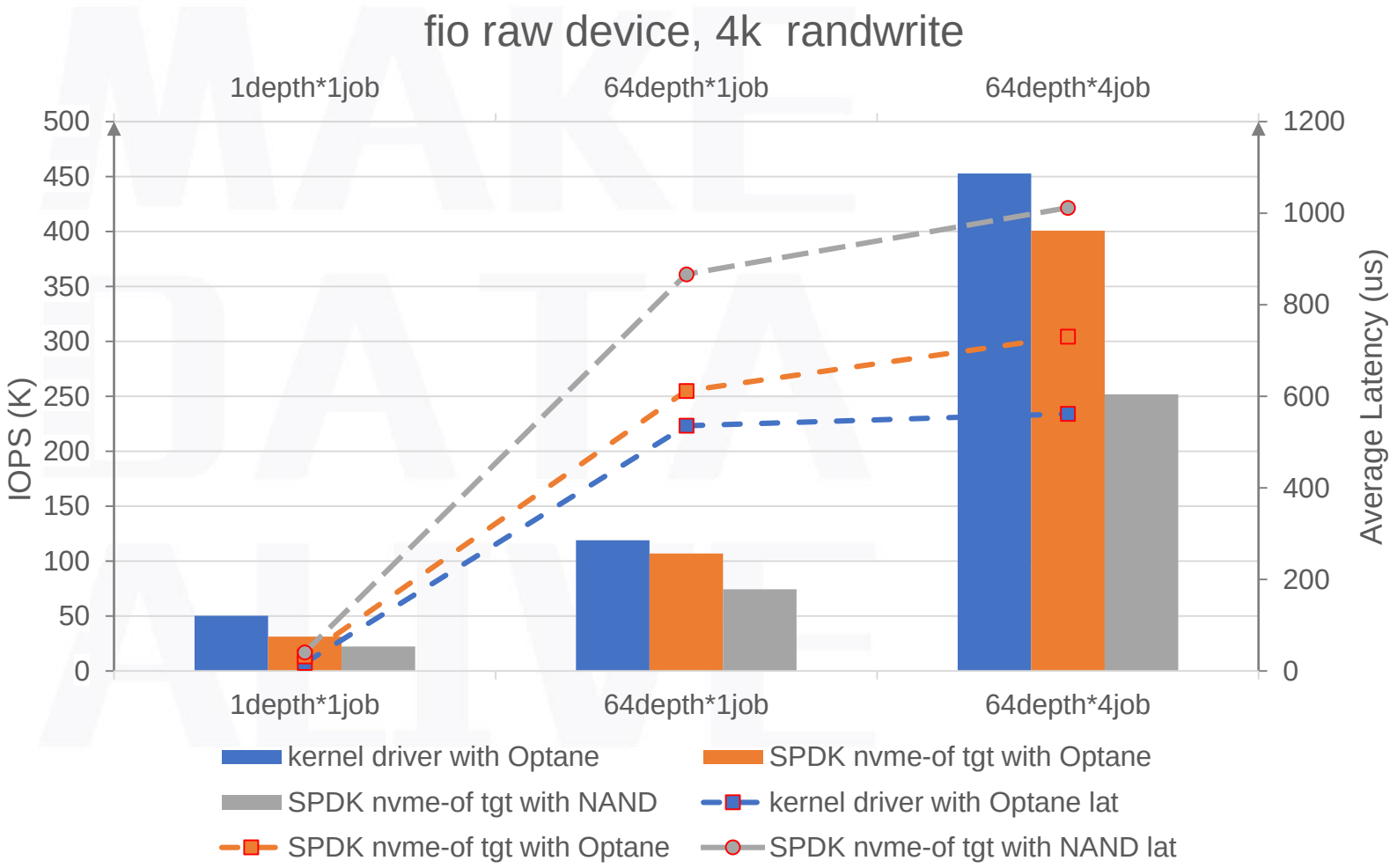




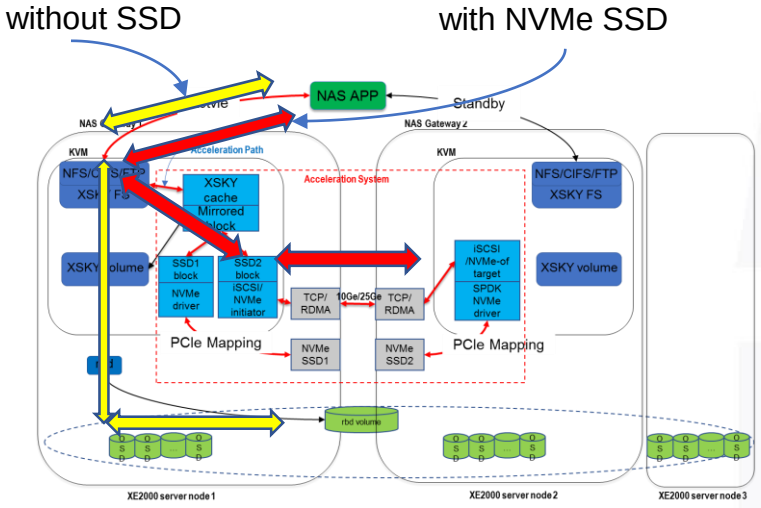
Intel Optane VS NAND NVMe

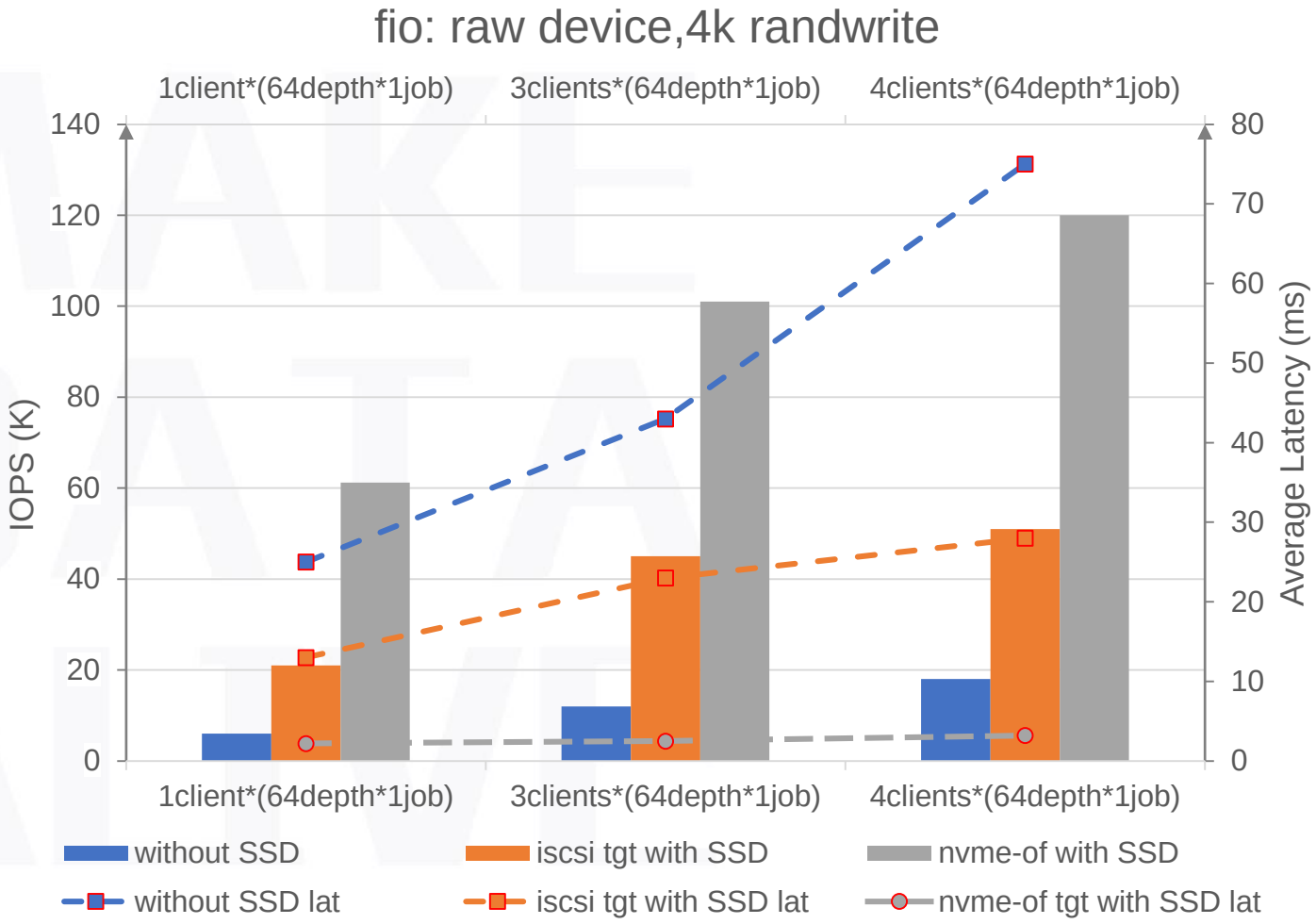
➢ Hardware:

- 25Ge Ethernet controller
- NAND NVMe SSD
- Intel Optane SSD



Old System VS New System





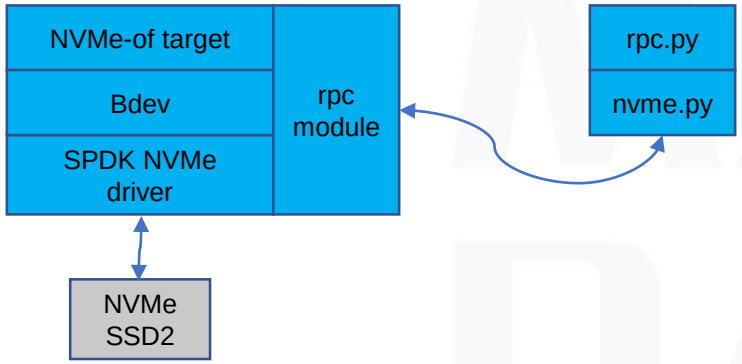
End-to-end performance (NFS):

① without NVMe SSD(Old system)
② iscsi tgt with NVMe SSD
③ nvme-of tgt with NVMe SSD

Outline

- Background
- Old Design of the NAS Gateway
- New Design of the NAS Gateway with SPDK
  - iSCSI based on SPDK
  - NVMe-of based on SPDK
- Performance comparison
  - Local Kernel VS NVMe-of VS iSCSI
  - intel Optane VS NAND NVMe
  - Old System VS New System
- Add JSON-RPC Methods for NVMe SSD status

Add JSON-RPC Methods for NVMe SSD status




```
[root@nas-n3r3dt4yak ~]# spdk-rpc get_nvme_info -r 'trtype:PCIe traddr:0000:01:01.0' -i health
{
  "Available Spare Space": "OK",
  "Temperature": "OK",
  "Current Temperature": "303 Kelvin (30 Celsius)",
  "Volatile Memory Backup": "OK",
  "Available Spare": "100%",
  "Available Spare Threshold": "5%",
  "Temperature Threshold": "343 Kelvin (70 Celsius)",
  "Read Only": "No",
  "Device Reliability": "OK",
  "Life Percentage Used": "1%"
}
[root@nas-n3r3dt4yak ~]# ps -ef |grep tgt
root      10577 10981 99 06:48 ttyS0    00:03:39 ./app/nvmf_tgt/nvmf_tgt -c /root/nvme.conf
root      10683 10641  0 06:52 pts/0    00:00:00 grep --color=auto tgt
[root@nas-n3r3dt4yak ~]#
```

- Add rpc API to get information of the nvme SSD attached in iscsi/nvme-of tgt.
- Just like “identify” example in SPDK, but for running process.

Beijing | Shanghai | Shenzhen | Chengdu | Nanjing | Wuhan

Thank You

MAKE DATA ALIVE

XSKY