NVMe/TCP Standards-Based, Fault-Tolerant Clustered Storage with LightOS

Alex Shpiner
System Architect
Lightbits Labs
alex@lightbitslabs.com
About Lightbits Labs

- **Founded** in Q1 2016
- **Key milestones:** NVMe/TCP, LightOS 1.0, LightOS 2.0
- **80 Employees** (90% Engineering):
  - EX: MLNX, PMCS, IBM, EMC, APPLE
  - First NVMe SSD and NVMe-oF products
- **Locations:**
  - Israel: Kfar Saba, Haifa
  - US: San Jose, NYC
  - Europe
  - China
- **Funding:** $55M, two rounds
  - Strategic investors: Cisco Investments, Dell Technologies Capital, Micron
  - Angel investors: Avigdor Willenz, Lip-Bu Tan, Marius Nacht and others
  - VCs: SquarePeg Capital, Walden International

**We are hiring!**
From direct-attached storage to disaggregated storage servers

- Efficient scalability
- Maximal utilization—support more users
- Easy maintenance and operation
LightOS v1.x

NVMe/TCP target with data services

- First commercially-available, standards based NVMe/TCP
- Software-defined storage
- Standard servers, SSDs and networking
- High throughput, consistent low latency
- Data protection: drives fault tolerance
- Data services including compression and thin provisioning

![Diagram showing LightOS Server, NVMe/TCP target, Global FTL with Rich Data Services, and SSDs with optional hardware acceleration for SSD management and data services.](image-url)
Optional hardware acceleration for SSD management and data services

High performance, low latency Global Flash Translation Layer with data services

High performance, low latency NVMe/TCP target

Standard TCP/IP Network (no RDMA required)

Standard NVMe/TCP client driver

Application Server 1

Application Server 2

Application Server N

Application
Cassandra, MongoDB ...

Application
Cassandra, MongoDB ...

Application
Cassandra, MongoDB ...

OS (Linux) with NVMe/TCP

OS (Linux) with NVMe/TCP

OS (Linux) with NVMe/TCP

NVMe/TCP target

Global FTL with Rich Data Services

SSDs

LightOS Server 1

LightOS Server 2

LightOS Server N

SSDs

NVMe/TCP target

Global FTL with Rich Data Services

SSDs

Lightbits Labs | 5
Increased Blast Radius with Disaggregated Storage

- LightOS is a great alternative to DAS.
- But, it increases the blast radius from a single compute node to multiple compute nodes.
- For some applications this is a non-issue since the application itself replicates.
  - Although some still prefer a fault tolerant infrastructure.
- For those applications that want fault-tolerant storage:
  - How can we provide the economics, flexibility, and performance of disaggregation with durable disaggregated storage?

<table>
<thead>
<tr>
<th>With Application Replication</th>
<th>No Application Replication</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="LightOS™ v1.x" /></td>
<td><img src="image" alt="Applications that do replicate still prefer a fault tolerant infrastructure solution." /></td>
</tr>
</tbody>
</table>
Surviving Any Blast

- With LightOS v2.x:
  - In case of server failure, computation nodes (clients) continue working (“business as usual”).

<table>
<thead>
<tr>
<th></th>
<th>With Application Replication</th>
<th>No Application Replication</th>
</tr>
</thead>
<tbody>
<tr>
<td>LightOS™ v1.x</td>
<td><img src="green-check" alt="Green Check" /></td>
<td><img src="red-x" alt="Red X" /></td>
</tr>
<tr>
<td>LightOS™ v2.x</td>
<td><img src="green-check" alt="Green Check" /></td>
<td><img src="green-check" alt="Green Check" /></td>
</tr>
</tbody>
</table>
## Extending the Scope of Failure Protection

### v1.x
- **SSD level protection**
  - SSD failure via Global FTL Erasure Coding

### v2.x
- **Storage server level protection**
  - Storage server failure via LightOS Clustering
Surviving Any Blast with LightOS Clustering

- Clients are connected to multiple storage servers.
- In case of server failure, the service continues from another server.
  - During failover some clients might suffer from performance hit.
  - All clients continue working!
Inherit storage services from LightOS 1.x
High performance and low latency
  - Single hop reads
  - Two hop writes (user + replications)

Standard unmodified clients and network
  - Leveraging standard NVMe-1.4 and NVMeoF 1.1
  - Transparent failover via multipath with Asymmetric Namespace Access (ANA)

Distributed and fault tolerant storage servers
  - Automatic volume assignment
  - Failure domains
  - Management
  - Discovery service
Volume Assignment to Storage Servers

- Multi-replica volumes
- Each replica is stored on a separate storage server
Different groups of storage servers can be impacted by common elements that share a point of failure:

- Network
- Power
- Geographical
User-defined server assignments to specific failure domain groups.
- Configured via labels assigned to servers, reflecting common dependencies.
  - rack_01, rack_02, ...
  - power_0, power_1, ...
- Replicas are placed in different failure domains.
Data Flow

- One of the replicas is defined as a primary.
- The remaining replicas are secondary.
- Client communicates with primary replica only.
- Read requests are served from primary replica.
- Write requests are sent to primary replica. Then, primary replica replicates to secondaries.
- Write requests are acknowledged after the request was replicated.
Server failure handling

- If a server fails, its data is rebuilt from other replicas.
  - Temporary failures use **partial rebuild**—only the data that was changed during a failure is re-sent.
  - Temporary Failure = network disconnection, SW upgrade, FW upgrade, etc.
- Rebuild operations are transparent to clients.
NVMe Asymmetric Namespace Access (ANA)

- NVMe Multipath IO defines access to NVMe namespace across two or more NVMe controllers (represent network paths).
- Multipath Namespace Access schemes:
  - **Symmetric**: All paths are equal
  - **Asymmetric**: Path state informs on access semantics
    - Optimized: Preferred accessible path
    - Non-Optimized: Non-preferred accessible path
    - Inaccessible
- **LightOS leverages NVMe ANA for Clustering**
  - Primary server reports "Optimized" ANA State
  - Secondary servers report "Inaccessible" ANA State
  - **Failure Handling**: Controller changes state in ANA report
NVMe Asymmetric Namespace Access (ANA)

- NVMe Multipath IO defines access to NVMe namespace across two or more NVMe controllers (represent network paths).
- Multipath Namespace Access schemes:
  - **Symmetric**: All paths are equal
  - **Asymmetric**: Path state informs on access semantics
    - Optimized: Preferred accessible path
    - Non-Optimized: Non-preferred accessible path
    - Inaccessible
- **LightOS Leverages NVMe ANA for Clustering**
  - Primary server reports “Optimized” ANA State
  - Secondary servers report “Inaccessible” ANA State
  - **Failure Handling**: Controller changes state in ANA report
NVMe Asymmetric Namespace Access (ANA)

- NVMe Multipath IO defines access to NVMe namespace across two or more NVMe controllers (represent network paths).
- Multipath Namespace Access schemes:
  - **Symmetric**: All paths are equal
  - **Asymmetric**: Path state informs on access semantics
    - Optimized: Preferred accessible path
    - Non-Optimized: Non-preferred accessible path
    - Inaccessible
- **LightOS Leverages NVMe ANA for Clustering**
  - Primary server reports “Optimized” ANA State
  - Secondary servers report “Inaccessible” ANA State
  - **Failure Handling**: Controller changes state in ANA report
Clustering Services

- **API Service**
  - REST API service for cluster control and volume definitions
  - Ibcli - command line utility
- **Cluster Management**
  - Managing cluster operation, configuration.
  - Monitoring, failure recovery orchestration
  - Replicas management, volume placing, capacity balancing, primary and secondary servers allocation.
- **Discovery Service**
  - Informs clients on accessible cluster volumes
  - NVMeoF standard
- **All services are fault-tolerant**
Demo: Cluster Status

- Status of storage servers:
  - state, IP address, failure domains, rebuild status

```
bash-4.2# lbccli list nodes

Name          UUID                     State      NVMe endpoint          Failure domains                  Local rebuild progress
server00-0    174d6fcdf-42f2-4c7a-834d-899045d2c7dc   Active  10.17.124.4:4420       [rack06-server52-vm05]            None
server01-0    c162151f-3869-4468-bb79-d5f80385c3d9     Active  10.17.124.5:4420       [rack06-server53-vm07]            None
server02-0    fdd672fc-394e-4080-bf5a-af2bc66aee7     Active  10.17.124.7:4420       [rack01-server64-vm07]            None
```

- Status of NVMe devices:
  - name, size, serial number, model, server

```
[root@rack12-server03 ~]# lbccli list nvme-devices

NAME        SIZE           NUMA-ID SERIAL              MODEL                 SERVER-UUID
nvme0n1     10002448b0816 0        PHLF728560UJ1IPCGN     INTEL SSDPE2X0010T7  9b6f26f6af-3983-55e4-b6b0-7757b7358897
nvme1n1     10002448b0816 0        PHLF728560UD1IPCGN     INTEL SSDPE2X0010T7  9b6f26f6af-3983-55e4-b6b0-7757b7358897
```
## Demo: Creating Volume

- Volume creation command:
  - Name, compression, replicas, size, ACL

```bash
$ bash-4.2# lbccli create volume --name=demo_vol --compression=true --acl=demo --replica-count=3 --size=1MiB
```

- Listing volumes:
  - Name, protection state, size, replicas, rebuild progress

```bash
$ bash-4.2# lbccli list volumes
```
Demo: Secondary Replica Disconnection

- Disconnecting server storing secondary replica
  - Automatically detected in several seconds
- Server state marked as “inactive”
- Path state updated
Demo: Primary Replica Disconnection

- Disconnecting server storing primary replica
  - Automatically detected in several seconds
- Server state marked as “inactive”
- Path state updated

Client:

```
[root@rack12-server01 ~]# nvme list-subsys /dev/nvme0n1
nvme_subsys - NQN=nqn.2014-05.org.nvmeexpress:nvme:uuid:00000000-0000-0000-0000-000000000000
  + nvme4 tcp traddr=10.23.20.1 trsvcid=4420 live inaccessible
  + nvme5 tcp traddr=10.23.20.2 trsvcid=4420 live inaccessible
  + nvme6 tcp traddr=10.23.20.3 trsvcid=4420 live optimized

[root@rack12-server01 ~]# nvme list-subsys /dev/nvme0n1
nvme_subsys - NQN=nqn.2014-05.org.nvmeexpress:nvme:uuid:00000000-0000-0000-0000-000000000000
  + nvme4 tcp traddr=10.23.20.1 trsvcid=4420 live inaccessible
  + nvme5 tcp traddr=10.23.20.2 trsvcid=4420 live optimized
  + nvme6 tcp traddr=10.23.20.3 trsvcid=4420 connecting optimized
```

Cluster:

```
[root@rack12-server03 demo]# lbcli list peers
NAME     UUID             State       NVME-Endpoint     Failure-Domains     In-Local-Rebuild Local-Rebuild-Progress
node00-0 14cfa509-96a5-5e01-98bd-1a5751156fcf Active 10.23.20.1:4420 [u'rack12-server03' u'nod... False 0
node01-0 4b21ff8a-d39a-522b-9401-60a25b1e65f7e Active 10.23.20.2:4420 [u'rack12-server09' u'nod... False 0
node02-0 8a86b8bc-24d5-5966-b33a-00a3c3bca4e Inactive 10.23.20.3:4420 [u'rack12-server03' u'nod... False 0
```
Demo: Replica Re-connection and Catch-up

- Replica is “behind”
- Partial rebuild - only missing data (that was written during the failure) is resent
- Replica catches up
Summary

- Direct attached storage to disaggregated storage architecture.
- LightOS v1.x: first NVMe/TCP based storage server with data services.
- LightOS v2.x: first NVMe/TCP based clustering solution for fault-tolerance in a storage server level.
- Volume replicas assignment considering failure domains.
- Data flow and server failover using NVMe multipath with ANA.
- Clustering services and management capabilities.

Contact information:
www.lightbitslabs.com
alex@lightbitslabs.com
Thank you!
Backup slides
Client Configuration

- Standard upstream kernel
  - Available in RHEL/CentOS 8.1 and Ubuntu 19.10
  - Available via ELREPO and HWE
- Standard ‘nvme-cli’
- Client ID - hostnqn, similar to iSCSI IQN
- Automation of connection during boot is possible
Client Connections

- Connecting to all cluster nodes
  - Discovery service allows connection using a single command

```
#!/bin/bash
IPS="10.23.20.1 10.23.20.2 10.23.20.3"
for IP in $[IPS]
do
nvme connect -t tcp -s 4420 -a ${IP} \
  --ctrl-loss-tmo 1 \
  -n nqn.2014-08.org.nvmeexpress:NVMe:uuid:00000000-0000-0000-0000-000000000000 \
  -q nqn.2014-08.org.nvmeexpress:uuid:00b84736-0594-4924-963a-41fe1f0ad4a
done
```
Client Connection Status

- Device is visible
  - via `lsblk` and `nvme list`
- Connections, active (optimized) and backup (inaccessible) are visible
  - via `nvme list-subsys <dev>'

```
[root@rack12-server51 demo]# nvme list
Node  SN  Model    Namespace  Usage
/dev/nvme4n1 6659d524cbb88a8a  LightBox  3  1.10 TB / 1.10 TB

[root@rack12-server51 demo]# nvme list-subsys /dev/nvme4n1
nvme-subsys4 - NQN=nqn.2014-08.org.nvmeexpress:N VMf:uuid:00000000-0000-0000-0000-000000000000
  + nvme4 tcp traddr=10.23.20.1 trsvcid=4420 live inaccessible
  + nvme5 tcp traddr=10.23.20.2 trsvcid=4420 live inaccessible
  + nvme6 tcp traddr=10.23.20.3 trsvcid=4420 live optimized
[root@rack12-server51 demo]# 
```
Write Traffic

- Running random-write from a single client
- One storage server shows in/out traffic
  - 23 Gbs in, 46 Gbs out
- Two storage servers show only incoming traffic
  - 22 Gbs in, 42 Mbs out
Read Traffic

- Running random-read from a single client
- One storage server shows outgoing traffic
  - 23 Gbs outgoing
- Two storage servers with secondary replicas are idle