Decibel: Isolation and Sharing in Disaggregated Rack-Scale Storage

Mihir Nanavati, Jake Wires, and Andrew Warfield
(Coho Data and University of British Columbia)
Redundancy and high availability
Transparent device aggregation
Cluster-accessibility
Decoupling and migration
Splendid cheeses they were with a two hundred horse-power scent that might have been warranted to knock a man over at two hundred yards.

Extract from “Three Men in a Boat”, Jerome K Jerome (1889)
“... because the data is remote (and often replicated under the covers), there’s typically a noticeable latency cost involved in using it rather than local storage [...] the ideal would be to use local disk for each replica for the sake of lower latency.”

Low throughput

“EBS has got a lot better... [At Netflix] we [still] don’t quite trust it for Kafka workloads [...] going for instance type”

Latency overheads

Shared ➔ Crosstalk

“... unless you want to add more complexity for your operations team, choose [...] direct-attached storage ....”
<table>
<thead>
<tr>
<th></th>
<th>Standard persistent disks</th>
<th>SSD persistent disks</th>
<th>Local SSDs</th>
<th>Cloud Storage buckets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Storage type</strong></td>
<td>Efficient and reliable block storage</td>
<td>Fast and reliable block storage</td>
<td>High-performance local block storage</td>
<td>Affordable object storage</td>
</tr>
<tr>
<td><strong>Price per GB/month</strong></td>
<td>$0.04</td>
<td>$0.17</td>
<td>$0.218</td>
<td>$0.007 - $0.026</td>
</tr>
<tr>
<td><strong>Maximum space per instance</strong></td>
<td>64 TB</td>
<td>64 TB</td>
<td>3 TB</td>
<td>Almost infinite</td>
</tr>
<tr>
<td><strong>Scope of access</strong></td>
<td>Zone</td>
<td>Zone</td>
<td>Instance</td>
<td>Global</td>
</tr>
<tr>
<td><strong>Data redundancy</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Standard persistent disks</th>
<th>SSD persistent disks</th>
<th>Local SSD (SCSI)</th>
<th>Local SSD (NVMe)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Maximum sustained IOPS</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read IOPS per GB</td>
<td>0.75</td>
<td>30</td>
<td>266.7</td>
<td>453.3</td>
</tr>
<tr>
<td>Write IOPS per GB</td>
<td>1.5</td>
<td>30</td>
<td>186.7</td>
<td>240</td>
</tr>
<tr>
<td>Read IOPS per instance</td>
<td>3,000</td>
<td>40,000</td>
<td>400,000</td>
<td>580,000</td>
</tr>
<tr>
<td>Write IOPS per instance</td>
<td>15,000</td>
<td>30,000</td>
<td>280,000</td>
<td>360,000</td>
</tr>
</tbody>
</table>
“[with Local SSDs] You cannot stop and restart an instance ... only suitable for temporary storage ...”

(Roz Chast, The New Yorker, April 2001)
**DECIBEL STORAGE GOALS**

*Managed Storage*

- Redundancy and high availability
- Transparent device aggregation
- Cluster-accessibility
- Decoupling and migration

*Local Storage*

- Throughput and latency
- No crosstalk
- Tight host coupling
Make all local storage cluster-accessible!

(Adapted from Intel Rack Scale Architecture)
Decibel is a storage service that remotely serves “local” storage

Challenges

- Workload dynamism
- Virtualization
- Maintain device performance

Virtualized Storage Abstraction + Runtime
“They are neither man nor woman
They are neither brute nor human
They are Ghouls”

Extract from “The Bells”, Edgar Allan Poe (1849)
From the application perspective, storage resembles a network-attached disk. A client library based TCP-RPC call, read(block_addr, len), would be sent. Virtualizing only the capacity of devices is insufficient!

Preserve device performance. Virtualizing only the capacity of devices is insufficient! Performance isolation and sharing Multiplex on fast hardware!

Network Storage Service

Runs locally on

Virtual device
Run-time

Virtual Machines

Client Library

Access control
dVols are bound to a single core and a single device

Limits mimic local SSDs

Decouple placement from scheduling

(Mirador, FAST 2017)

Capacity Migrations

Performance Migrations
Device Partitioning

Contestion and fragmentation

Per-core caches (Hoard/tcmalloc)

Scheduling

Device throughput and latency
dVol isolation
dVols provide hooks for conveniently implementing policies
THE DECIBEL RUNTIME

Resource Management
Scheduling
Address Virtualization
Access Control
Atomicity and Data Integrity
Decibel vs. Local Devices

Performance

Single-core Performance
1U Client
1U Server
2 x Fortville 40GbE
4 x P3700
Clients/Core : 2
2 x Fortville 40GbE
QD/client : 16

4K random requests

4.8 M IOPS (60 Gbps)

Independent remote dVol
Performance (70/30 Mixed Workload)

Throughput (K IOPS)

Latency (μs)

422 vs 450 vs 490μs

Number of Cores

Number of Cores
Performance (All Reads Workload)

Throughput (K IOPS)
- Local
- Decibel (DPDK)
- Decibel (Legacy)

Latency (μs)
- Local
- Decibel (DPDK)
- Decibel (Legacy)

203 vs 221 vs 290μs
SINGLE CORE PERFORMANCE

<table>
<thead>
<tr>
<th>Condition</th>
<th>Latency (μs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read (Idle)</td>
<td>79, 91</td>
</tr>
<tr>
<td>Read (Saturation)</td>
<td>209, 223</td>
</tr>
<tr>
<td>Mixed 70/30 (Idle)</td>
<td>67, 78</td>
</tr>
<tr>
<td>Mixed 70/30 (Saturation)</td>
<td>333, 346</td>
</tr>
</tbody>
</table>

- Local
- Decibel
Local storage as a service

Encapsulate full-system resources

20-30μs overhead to local devices

Isolation on shared storage
Backup Slides
Performance (DRAM Backed Storage)

Throughput (K IOPS)

Latency (μs)

Saturation: 23 vs 68μs

Decibel (DPDK) vs Decibel (Legacy)

Number of Cores

Throughput (K IOPS) and Latency (μs) for Decibel (DPDK) and Decibel (Legacy) with varying numbers of cores. The plots show an increasing trend in both throughput and latency as the number of cores increases. The saturation is noted at 23 vs 68μs.
LATENCY CONSISTENCY (READS)

Throughput (K IOPS)

Latency (μs)

Queue Depth

Throughput Avg. Latency 95th Percentile

Latency

0 50 100 150 200 250 300 350 400 450

Throughput

0 50 100 150 200 250 300 350 400 450

1 2 4 8 16 32 64 128
LATENCY CONSISTENCY (WRITES)

Throughput (K IOPS)

Queue Depth

<table>
<thead>
<tr>
<th>Throughput</th>
<th>Avg. Latency</th>
<th>95th Percentile</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>20</td>
</tr>
<tr>
<td>8</td>
<td>20</td>
<td>40</td>
</tr>
</tbody>
</table>

Latency (μs)

- 0
- 50
- 100
- 150
- 200
- 250