I/O middleware for artificial intelligence: real intelligence required

Glenn K. Lockwood, Ph.D.
Principal Product Manager, Azure Storage
Microsoft Corporation
Traditional storage for high-performance storage

**Scratch storage**
- higher performance-per-terabyte
- higher cost-per-terabyte too
- data resident for hours/days/weeks

**Project storage**
- lower performance-per-terabyte
- lower cost-per-terabyte
- data resident for weeks/months/years
Azure model for high-performance storage

Storage accelerators
- Enhance performance of core storage
- Optimized for specific workload patterns
- Provisioned on-demand
- Active, ephemeral storage for hot data

Core storage
- Durable, available, and secure
- Rich data management capabilities
- Scalable and elastic
- Permanent home for data
How can middleware accelerate AI training?
Training models directly from core storage
Training models with an intermediate accelerator

InfiniBand Cluster

500 GB/s

Accelerator

50 GB/s

Core storage

all training data

GPU node

shard0

shard1

shard2

shard3

shard4

shard5

shard6

shard7
Using middleware for client-side acceleration

InfiniBand Cluster

GPU node

5,000 GB/s

Core storage

50 GB/s

shard0 shard1 shard2 shard3 shard4 shard5 shard6 shard7

all training data
Checkpointing with application-specific middleware

Core Storage (Azure Blob)
How can middleware accelerate AI inferencing?
Loading models directly from core storage

- InfiniBand Cluster
- GPU node
- GPU node
- GPU node
- GPU node
- 50 GB/s
- Core storage
- model
- model data (e.g., 500 GiB)
Using middleware for inferencing acceleration

InfiniBand Cluster

5,000 GB/s

GPU node

copy1

GPU node

copy3

GPU node

GPU node

model data (500 GiB)

Core storage

50 GB/s

model

using middleware for inferencing acceleration
<table>
<thead>
<tr>
<th>Activity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Loading training data</td>
<td>Multilevel caching to track where different tensor components are cached (core storage, local SSD, neighbor node)</td>
</tr>
<tr>
<td>Checkpointing models while training</td>
<td>Multilevel checkpointing to asynchronously, progressively flush data to slower but more durable storage</td>
</tr>
<tr>
<td>Loading models for inferencing</td>
<td>Fan-out broadcasting of critical datasets to avoid severe incast-type traffic</td>
</tr>
</tbody>
</table>