An Introduction to Compute Express Link (CXL)

Steve Scargall
Sr. Product Manager, MemVerge
Growth of Data led by AI/ML

2,600,000x
Growth of EDA data since 1984

Cerberus WSE-2
The Largest Chip Ever Built

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>mm² silicon</td>
<td>46,225</td>
</tr>
<tr>
<td>Trillion transistors</td>
<td>2.6</td>
</tr>
<tr>
<td>AI optimized cores</td>
<td>850,000</td>
</tr>
<tr>
<td>Gigabytes on chip memory</td>
<td>40</td>
</tr>
<tr>
<td>Petabyte/s memory bandwidth</td>
<td>20</td>
</tr>
<tr>
<td>Petabit/s fabric bandwidth</td>
<td>220</td>
</tr>
<tr>
<td>Terabit/s ingest bandwidth</td>
<td>1.2</td>
</tr>
<tr>
<td>Process technology at TSMC</td>
<td>7nm</td>
</tr>
</tbody>
</table>

Source: VentureBeat

1,000x
Greater cell data since 2009

1,000x
Larger model data in last 2 years

Source: VentureBeat

Source: Analytical Biosciences

Source: VentureBeat
Driving Need for more Memory & NAND

Hyperscale adoption of AI

- 2020: AI servers (2x) vs. Other servers
- 2025: AI servers (2x) vs. Other servers

Drives memory & storage growth

- Compute optimized server
- AI optimized server
- 7x storage
- 6x memory

NAND storage

HBM memory

DDR memory

Global memory market

- 2020: $20B
- 2025: $100B
- 2030: $180B

16% CAGR

Source: Micron
Memory is >=50% of Server BOM

PowerEdge R750 Rack Server (1 TiB)

- **Intel® Xeon® Gold 6354 3G, 18C/36T, 11.2GT/s, 39M Cache, Turbo, HT (205W) DDR4-3200**
  - Selected
- **32GB RDIMM, 3200MT/s, Dual Rank, 16Gb BASE x8**
  - Qty: 32, $1,061.58 /ea.
- **800GB SSD SAS ISE Mix Use 12Gbps 512e 2.5in Hot-plug AG Drive, 3 DWPD**
  - Qty: 8, $1,467.36 /ea.
- **BOSS-S2 controller card + with 2 M.2 240GB (RAID 1)**
  - Selected
- **Intel X710-T2L Dual Port 10GBase-T Adapter, PCIe Full Height**
  - Qty: 2, $591.58 /ea.

**Estimated Value:** $93,292.76

**Total Savings:** $34,765.20

**Shipping:** Free

**MemVerge Dell Price:** $58,526.56


Prices from Sept 2022
The Stranded Memory Problem

“the average utilization of the DRAM across its (Google) clusters is somewhere around 40 percent.”

“In Azure, up to 25% of memory is stranded.

…

∼50% of all VMs never touch 50% of their rented memory

…

our pooling approach incurs a configurable performance loss between 1-5%.

…

disaggregation can achieve a 9-10% reduction in overall DRAM, which represents hundreds of millions of dollars in cost savings for a large cloud provider”

Borg: the Next Generation [Google Whitepaper]

First-generation Memory Disaggregation for Cloud Platforms [Azure Whitepaper]
Compute Express Link™
The Breakthrough CPU-to-Device Interconnect

• Compute Express Link™ (CXL™) is an industry-supported Cache-Coherent Interconnect for Processors, Memory Expansion and Accelerators.

• The CXL Consortium is an open industry standard group formed to develop technical specifications that facilitate breakthrough performance for emerging usage models while supporting an open ecosystem for data center accelerators and other high-speed enhancements.

• Over 200 Consortium Members

• 3 major specification versions publicly available
Top Use Cases

FSI
- Pub/Sub using Shared Memory amongst nodes
- Prevent Out-of-Memory conditions during trading hours
- Analytics/Quants: Keep weeks/months of data in-memory
  - Reduce the cloud egress costs
- Bigger *SQL Databases & Caches
  - Reduce the Storage and Network overhead
  - Reduce tail latencies
  - Cache held in a local memory or a shared memory pool

AI/ML
- Accommodate bigger models in-memory
- Faster training times

Cloud
- Composable Memory
  - Eliminate stranded/frozen memory
- Increase Container/VM Density

HPC
- Checkpoint to pooled memory (vs disk)
- Shared Memory vs MPI/RDMA
Compute Express Link Protocols

CXL uses PCIe Physical & Transport Layers and new protocols:

- **CXL.io** is used for initialization, link-up, device discovery and enumeration, and register access.
- **CXL.cache** defines interactions between a Host (usually a CPU) and Device (such as a CXL memory module or accelerator).
- **CXL.mem** provides a Host processor with direct access to Device-attached memory using load/store commands.
Usage Models (Device Types)
CXL v1.1 – Direct Attach
CXL 2.0 Single Level Switching

Benefit of CXL 2.0 Switching
Expansion

Host

CXL 2.0 Switch
D1 D2 D3 D4

CXL 2.0 Switch
D1 D2 D3

CXL 2.0 Switch
D4 D5 D6

Copyright | CXL™ Consortium 2020
CXL 2.0 Device Pooling

Benefit of CXL 2.0 Switching
Pooling

Memory/Accelerator Pooling with Single Logical Devices

CXL 2.0 Switch

Memory Pooling with Multiple Logical Devices

CXL 2.0 Switch

Standardized CXL Fabric Manager
CXL 3.0: A Flexible Fabric

- Pool memory across multiple xPUs
- Solve stranded memory
- Scale memory independent of xPUs

- Attach various resources into the fabric
- Scalable
- Serviceable
- Fully Composable Infrastructure
CXL 3.0 Multi-Layer Switching
Peer-to-Peer (P2P)

CXL 3.0: DEVICE TO DEVICE COMMS

CXL 3.0 enables peer-to-peer communication (P2P) within a virtual hierarchy of devices:
- Virtual hierarchies are associations of devices that maintain a coherency domain.
# CXL Feature Summary

<table>
<thead>
<tr>
<th>Features</th>
<th>CXL 1.0 / 1.1</th>
<th>CXL 2.0</th>
<th>CXL 3.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release date</td>
<td>2019</td>
<td>2020</td>
<td>1H 2022</td>
</tr>
<tr>
<td>Max link rate</td>
<td>32GTs</td>
<td>32GTs</td>
<td>64GTs</td>
</tr>
<tr>
<td>Flit 68 byte (up to 32 GTs)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Flit 256 byte (up to 64 GTs)</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Type 1, Type 2 and Type 3 Devices</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Memory Pooling w/ MLDs</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Global Persistent Flush</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>CXL IDE</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Switching (Single-level)</td>
<td></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Switching (Multi-level)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Direct memory access for peer-to-peer</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Enhanced coherency (256 byte flit)</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Memory sharing (256 byte flit)</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Multiple Type 1/Type 2 devices per root port</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Fabric capabilities (256 byte flit)</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
A Rapidly Growing CXL Ecosphere

Switches: Marvell, Xilinx, Elastics, LIqid, GigaIO

Memory Systems: Samsung, SK hynix, Micron, KIOXIA, SMART, Western Digital, Broadcom, Seagate

Big Memory Software

Memory Machine
- Memory Snapshot
- Memory Tiering
- Memory Sharing
- Hardware API Integration

Memory Viewer
- Memory Monitoring
- Memory Visualization

GFAM Orchestrator & Fabric Manager
- Data Protection
- Security
- Global Insights
- Memory Provisioning & Sharing
- Capacity Optimization

Processors: Intel, AMD, NVIDIA, ARM

Servers: Dell, Lenovo, Hewlett Packard Enterprise, Supermicro, Penguin Computing

Standards Bodies: CXL, SNIA, Open Compute Project

Big Memory Apps: Synopsys, Cadence, Hazelcast

Clouds: AWS, Azure, Alibaba Cloud, Tencent Cloud, Google, ByteDance, Meta, Global IT Services
Lots of Software is Needed
MemVerge Vision & Goals

Deliver CXL features early
• Shared Memory
• Integration with existing workflows/schedulers/resource managers
• Host-side Memory Tiering
• Memory Pools (Expansion)
• Memory Pool Orchestration between consumers and pools
• Global CXL Fabric Management
• Memory Pool Data Management - Protection, H/A, RAS, Security
• Advance Memory Services – Snapshots & Replication
• Datacenter Management: Day 0 deployment/Day 1 operations

Not in priority or specific order
Futuristic CXL Memory Management Software Model

Host-based Software
- Auto-Tiering
- Migration
- App Profiling
- Snapshot
- Monitoring
- Sharing

Pool-based Software
- Data Protection
- Security
- Global Insights
- Provisioning & Sharing
- Capacity Optimization

Operating Systems

Computing Hosts

Memory Pool (All Memory Array, DCI, Memory Server)

CXL Switch

Pool Server
### CXL Timeline

*Where we are now & Where are we going?*

<table>
<thead>
<tr>
<th>Year</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2021</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2022</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2023</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **CXL Spec**
  - 1.0
  - 1.1
  - 2.0
  - 3.0

- **Linux***
  - v5.12
  - v6.0

- **Platform/OEM***
  - 1.1

- **CXL Devices***
  - 2.0
  - 3.0

- **MemVerge***
  - 2.0
  - 3.0

* Estimates of future roadmap projections are not accurate or guaranteed. Refer to individual vendors for more detailed information.
### Memory Monitor Overview

#### Top Memory Consumers

<table>
<thead>
<tr>
<th>Name</th>
<th>PID</th>
<th>DRAM Usage</th>
<th>CPU</th>
<th>Start Time</th>
<th>User</th>
<th>Monitoring Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>libnl</td>
<td>504466</td>
<td>68.73 MB</td>
<td>0%</td>
<td>23:37:21</td>
<td>Jul 06, 2022</td>
<td>root</td>
</tr>
<tr>
<td>mount</td>
<td>2958</td>
<td>49.06 MB</td>
<td>0.03%</td>
<td>16:48:19</td>
<td>Jul 06, 2022</td>
<td>mmmm</td>
</tr>
<tr>
<td>mmux</td>
<td>2243190</td>
<td>47.26 MB</td>
<td>52.38%</td>
<td>09:20:19</td>
<td>Jul 31, 2022</td>
<td>root</td>
</tr>
<tr>
<td>sad_zes</td>
<td>2580</td>
<td>42.01 MB</td>
<td>0%</td>
<td>23:43:10</td>
<td>Jul 06, 2022</td>
<td>root</td>
</tr>
<tr>
<td>pcm-memory.t</td>
<td>306817</td>
<td>37.80 MB</td>
<td>0.23%</td>
<td>17:34:07</td>
<td>Jul 06, 2022</td>
<td>mmmm</td>
</tr>
<tr>
<td>polkd</td>
<td>2460</td>
<td>36.40 MB</td>
<td>0%</td>
<td>14:30:09</td>
<td>Jul 06, 2022</td>
<td>polkd</td>
</tr>
<tr>
<td>pcm-memory.t</td>
<td>2493429</td>
<td>35.82 MB</td>
<td>0.22%</td>
<td>17:00:50</td>
<td>Jul 15, 2022</td>
<td>songhe</td>
</tr>
<tr>
<td>gdm-media-tes</td>
<td>8642</td>
<td>22.14 MB</td>
<td>0%</td>
<td>14:45:21</td>
<td>Jul 06, 2022</td>
<td>gdm</td>
</tr>
<tr>
<td>gdm-power</td>
<td>8643</td>
<td>20.03 MB</td>
<td>0%</td>
<td>14:45:21</td>
<td>Jul 06, 2022</td>
<td>gdm</td>
</tr>
<tr>
<td>gdm-color</td>
<td>8653</td>
<td>20.60 MB</td>
<td>0%</td>
<td>14:45:21</td>
<td>Jul 06, 2022</td>
<td>gdm</td>
</tr>
</tbody>
</table>

#### Memory Usage Summary

- **Peak Memory Used:** 61.46 MB
- **Average Memory Used:** 64.60 MB
- **Memory Used Standard Deviation:** 2.44 MB (4.47%)
- **Memory Used Variance:** 4.95 MB (25.93%)

**Memory Used Graph**

- Y-axis: Memory Used (MB)
- X-axis: Time

---

**Memory Monitor - mmux (2243190)**

- **Process Monitor**
  - **Process:** mmux
  - **Start Time:** 09:20:19, Jul 31, 2022
  - **CPU Usage:** 52.38%
  - **Memory Usage:** 47.26 MB

**Memory Usage Graph**

- Y-axis: Memory Used (MB)
- X-axis: Time
Call To Action

• Don’t Wait. **Start** your CXL journey now!
• Download **Memory Viewer** for insights today
• Collaborate with **MemVerge** for your CXL adoption strategy and solutions
Questions?

Q&A