Rack-Scale Memory Disaggregation over Ethernet
Weigao Su, Vishal Shrivastav

Memory disaggregation via Ethernet

- Using Ethernet fabric
- High compute density
- Fine-grained provisioning
- Seamless scaling

<table>
<thead>
<tr>
<th>Compute Node</th>
<th>Application</th>
<th>Memory Traffic</th>
<th>Memory Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/IP or RDMA</td>
<td>Ethernet MAC</td>
<td>Ethernet PHY</td>
<td>Ethernet PHY</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Switch</th>
<th>Memory Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP/IP or RDMA</td>
<td>Ethernet MAC</td>
</tr>
</tbody>
</table>

Requirements

- Latency: Transmission delay needs to be as close to local memory access as possible (NUMA takes around 280ns)
- Utilization: Header encapsulation needs to be efficient since memory flows are extremely small, often less than 64B and potentially a single byte.

Existing Limitations

1. Minimum frame size overhead
2. Inter-frame gap (IFG) overhead
3. No intra-frame preemption
4. Layer 2 switching overhead
5. Transport layer overhead
6. Queueing delay at switch

FPGA Prototype

Added latency:
- 268.8ns for read;
- 262.4ns for write

Comparable to one hop NUMA; 4x faster than the raw Ethernet.

Network Simulation

EDM keeps the end-to-end latency within 1.2x and 1.3x the ideal unloaded latency for READs and WRITeS respectively.

EDM (Ethernet Disaggregated Memory)

- Centralized flow scheduling: zero network queuing + high bandwidth utilization.
  - Maximal-matching: At most one sender sending to a receiver at any time.
  - Zero-delay forwarding: No processing needed because of matched circuit.
  - Reduced transport overhead: A no-loss environment is guaranteed by matching.
  - Near-optimal flow completion time: Achieved by a configurable priority queue.
- Bypassing higher layers: minimum latency + header encapsulation overhead
  - Host: Tx interacts with local application via notification queue, while Rx asynchronously updates a grant queue for responding remote requests. States are stored in a flow state table.
  - Switch: maintains a notification (priority) queue to proactively shape traffic.

Latency Source

<table>
<thead>
<tr>
<th>Raw Ethernet</th>
<th>EDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>READ</td>
</tr>
<tr>
<td>Compute Node</td>
<td>268.8ns</td>
</tr>
<tr>
<td>MAC</td>
<td>2 * 19.2ns</td>
</tr>
<tr>
<td>PHY (PCS)</td>
<td>2 * 19.2ns</td>
</tr>
</tbody>
</table>

Switch

Layer 2 fwd | 2 * 400ns | 400ns | 0 | 0 |
MAC | 4 * 19.2ns | 2 * 19.2ns | 0 | 0 |
PHY (PCS) | 4 * 19.2ns | 2 * 19.2ns | 4 * 12.8 ns + 70.4ns | 4 * 12.8 ns + 70.4ns |

Memory Node

MAC | 2 * 19.2ns | 19.2ns | 0 | 0 |
PHY (PCS) | 2 * 19.2ns | 19.2ns | 2 * 12.8 ns +64ns | 12.8 ns +19.2ns |

Network Stack Latency

1.11μs | 553.6ns | 268.8ns | 262.4ns |
Transmission Delay | 4 * 51.2ns | 2 * 51.2ns | 6.4 + 51.2ns | 12.8 +51.2ns |
Propagation Delay | 4 * 10ns | 2 * 10ns | 4 * 10ns | 4 * 10ns |
Total Latency | 1.35μs | 676ns | 364.4ns | 366.4ns |