Network Requirements for Resource Disaggregation

Peter Gao (Berkeley), Akshay Narayan (MIT), Sagar Karandikar (Berkeley), Joao Carreira (Berkeley), Sangjin Han (Berkeley), Rachit Agarwal (Cornell), Sylvia Ratnasamy (Berkeley), Scott Shenker (Berkeley/ICSI)
Disaggregated Datacenters

Current Datacenter: Server-Centric

Future datacenter: Disaggregated?

HP – The Machine

Intel – RSD

Facebook

Huawei – NUWA

SeaMicro

Berkeley – FireBox
Disaggregation Benefits (Architecture Community)

- Overcome memory capacity wall
- Higher resource density

- Simplify Hardware Design
- Relax Power & Capacity Scaling
Do we need specialized hardware? e.g.: Silicon photonics, PCI-e
• What end-to-end latency and bandwidth must the network provide for legacy apps?
  • Do existing transport protocols meet these requirements?
  • Do existing OS network stacks meet these requirements?
  • Can commodity network hardware meet these requirements?

Current OS and network stack are not (Solutions are feasible)

Commodity hardware solutions may be sufficient
## Assumptions

<table>
<thead>
<tr>
<th>Cache Coherence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Datacenter Network</td>
</tr>
</tbody>
</table>

### CPU
- Limited cache coherence domain
- Small amount of local cache (how much?)

### Memory
- Page-level remote memory access

### Storage
- Block-level distributed data placement

### Scale
- Rack-scale?
- Datacenter-scale?
Methodology: Workload Driven

- 10 workloads on 8 applications
- ~ 125 GB input data
- 5 m3.2xlarge EC2 nodes
- Virtual Private Cloud enabled
Disaggregated Datacenter Emulator

OS

Special Swap Device (Handles Page Fault)

Local RAM  Emulated Remote RAM

Free to access
Via swap device

Inject latency and bandwidth constraints
- Using special swap device
- Delay = latency + request size / bandwidth
- Akin to a dedicated link between CPU and memory

- Backed by the machine’s own memory
- Partition the memory into local and remote
**Latency and Bandwidth Requirement**

\[ \text{Delay} = \text{latency} + \frac{\text{request size}}{\text{bandwidth}} \]

- ~3us latency / 40Gbps bandwidth is enough, ignoring queueing delay

*Note: Delay = latency + request size / bandwidth*
Performance degradation is correlated with application memory bandwidth
• 3us end-to-end latency
• 40Gbps dedicated link (no queueing delay)
Transport Simulation Setting

Special Swap
Instrumentation

Flow Trace

Network Simulator

Flow completion time distribution

Need new transport protocols

Graph

Slowdown

Hadoop Wordcount
Hadoop Sort
Graphlab CF
Memcached YCSB
Spark Wordcount
Spark Sort
TimelyDataflow PageRank
SparkSQL BDB

TCP
Application Performance Degradation

100Gbps network

DC scale for some apps, rack scale for others
• 3us end-to-end latency
• 40Gbps dedicated link

• Efficient Transport
• 100Gbps network
Is 100Gbps/3us achievable?
Feasibility of end-to-end latency within a rack

*Numbers estimated optimistically based on existing hardware*
Feasibility of end-to-end latency within a rack

*Numbers estimated optimistically based on existing hardware*
Feasibility of end-to-end latency within a rack

Application | Propagation | Transmission | Switching | Data Copying | OS | Remote Resource
---|---|---|---|---|---|---
0.32us | 0.8us | 3us Target | 2us | | 1.9us |

Application | Cut-through Switch | NIC Integration | Copying | OS | Remote Resource
---|---|---|---|---|---
0.32us | 0.48us | 1us | | 1.9us |

*Numbers estimated optimistically based on existing hardware*
Feasibility of end-to-end latency within a rack

Feasible to meet target across the datacenter?

*Numbers estimated optimistically based on existing hardware
- 3us end-to-end latency
- 40Gbps dedicated link

- Efficient Transport (pFabric, SIGCOMM’13, pHost, CoNEXT’15)
- 100Gbps network (Available)

- Kernel bypassing (RDMA common)

- CPU–NIC Integration (Coming soon)
- Cut-through switch (Common?)
- 100Gbps links (Available)
What's next?

Application Design

Rethinking OS Stack
- Storage
- Network Stack
- Failure Models

Network Fabric Design

Please refer our paper for evaluations on improving application performance in disaggregated datacenters.
Thank You!

Peter X. Gao
Akshay Narayan
Sagar Karandikar
Joao Carreira
Sangjin Han
Rachit Agarwal
Sylvia Ratnasamy
Scott Shenker