OPENFABRICS ALLIANCE

RoCE and Dense Clusters

Paul Grun
April 5, 2011

Abstract

RoCE enables RDMA operations on a layer 2 Ethernet switched fabric.

This talk discusses some possible optimizations in server cluster design that take advantage of RoCE technology

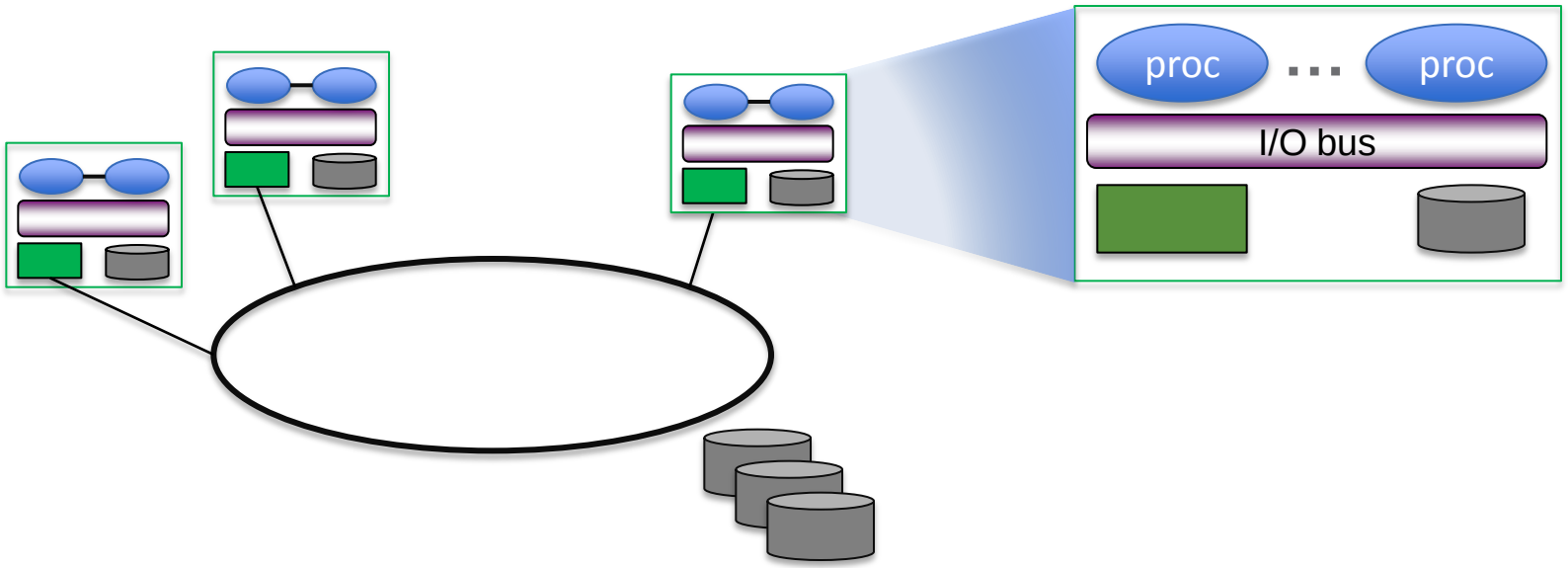
- RDMA provides low latency, and thus offers scalability
- RoCE allows us to run RDMA over Ethernet
- Now, what do we do with it?

An (non-profound) observation:

Every server has an Ethernet port

Scale-out clusters…

…are composed of *general purpose servers* interconnected by a *cluster fabric*

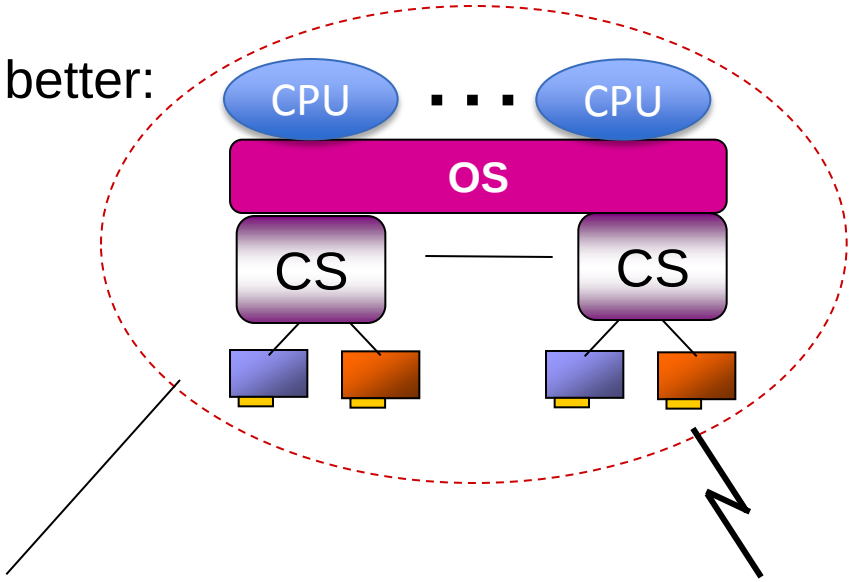


Look at both the endnodes and the interconnect

The canonical general purpose server

A *scale-up* architecture

To make it bigger, you plug in more and better:
- processor sockets,
- chipsets
- memory,
- I/O buses…



For clustering, add a PCIe adapter for a low latency cluster interconnect.
A variety of choices for storage interconnects
The one constant is the Ethernet port

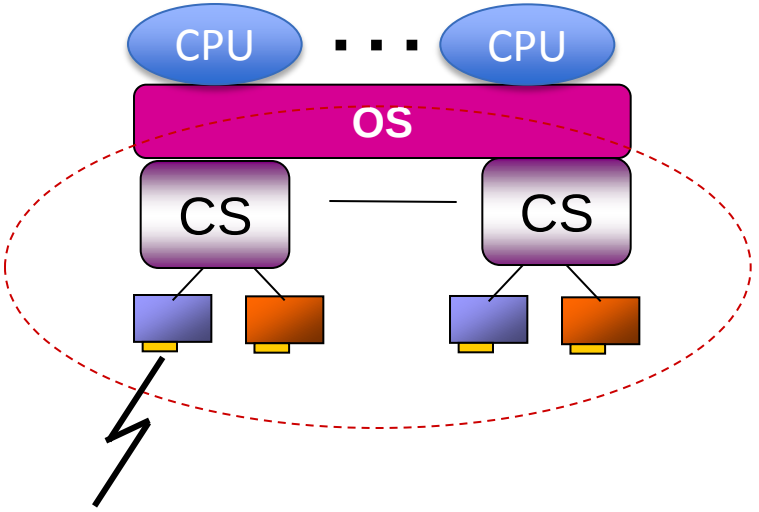
The canonical general purpose server

But the underlying I/O subsystem is sometimes hard to rationalize

- may require multiple I/O interconnects

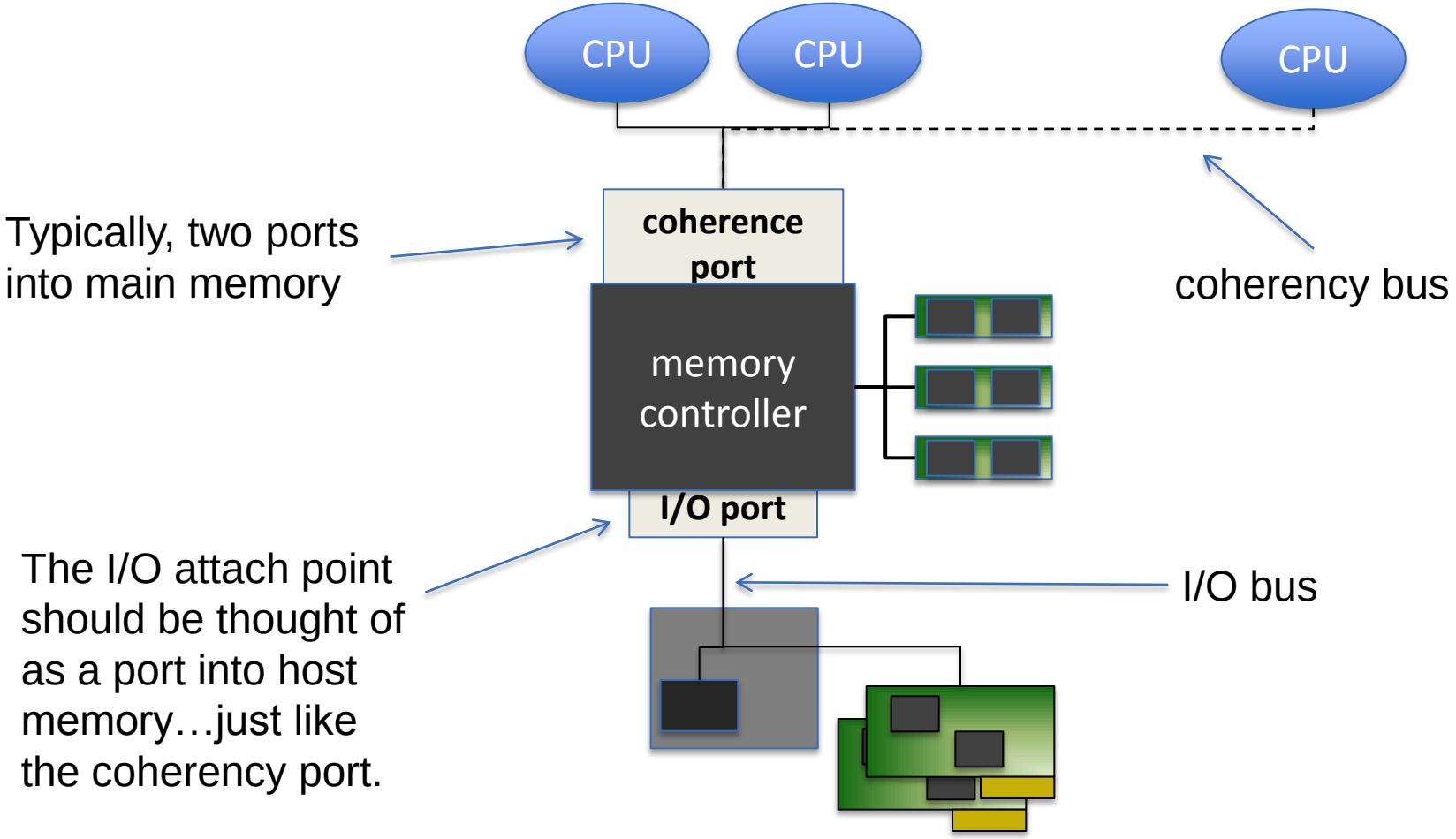
- difficult to keep all those cores fed

- hard to predict balance of I/O workloads across multiple I/O fabrics

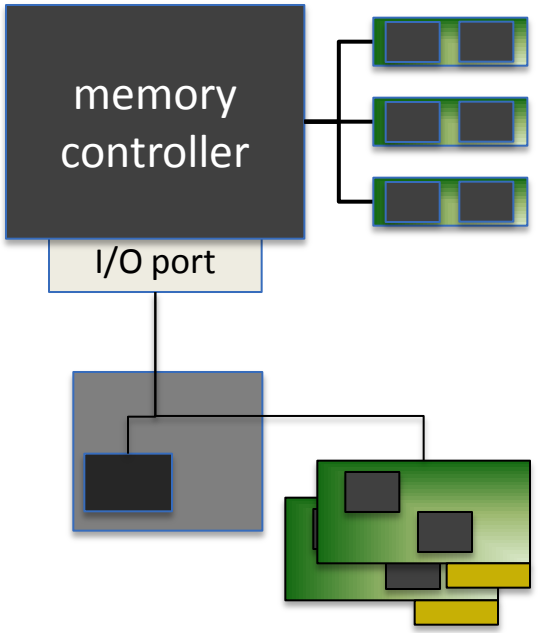


In short, it can be a challenge to keep the server architecturally balanced

Another view of the general purpose server



I/O memory port



The I/O bus is designed to connect various I/O protocol adapters to host memory.

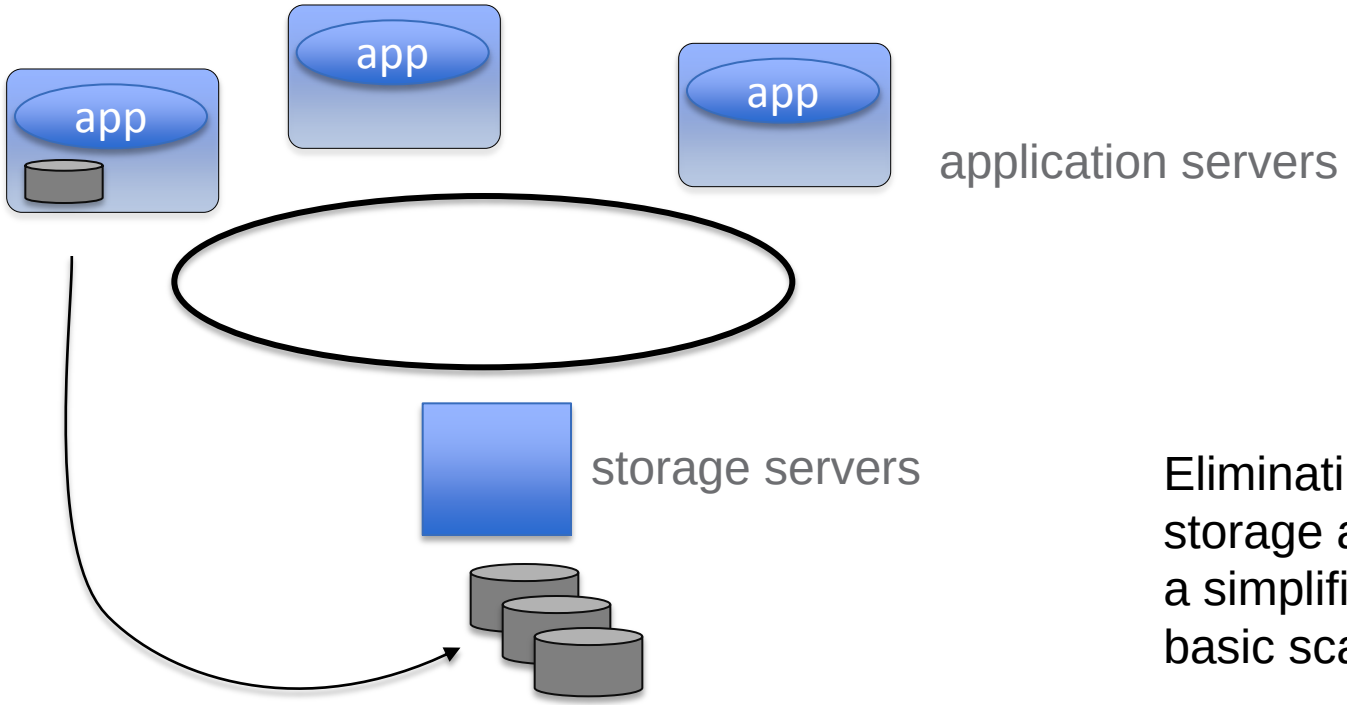
This is what makes the server platform general purpose.

Historically, each I/O protocol ran over a specific physical interconnect.

But that is becoming much less often the case.

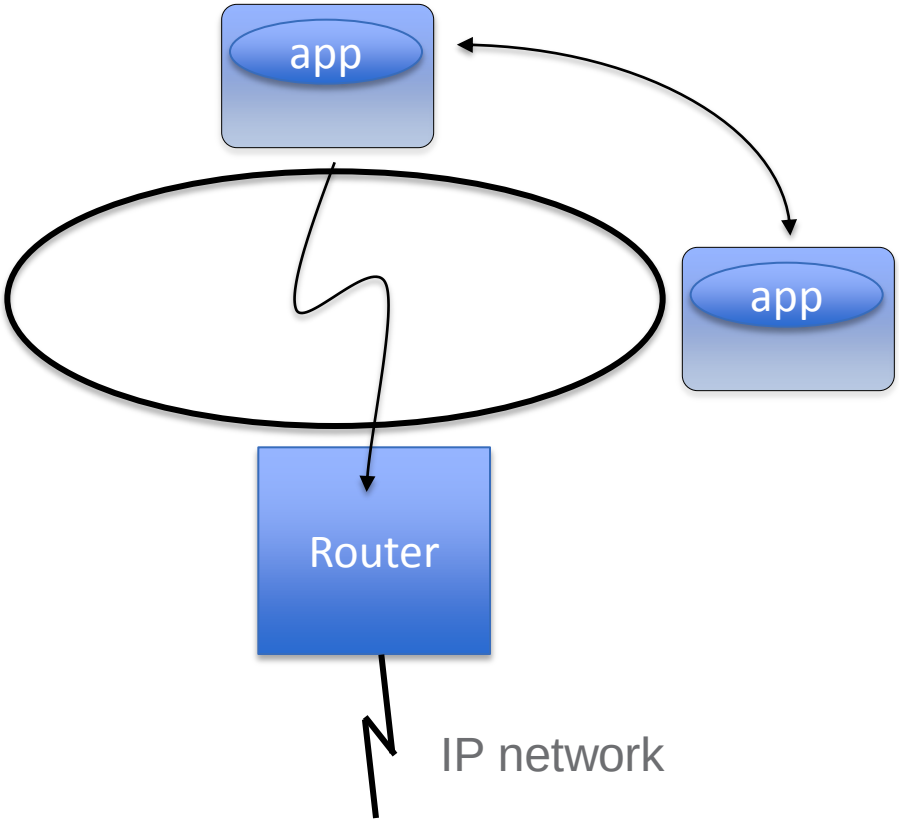
Storage in the modern era

Given the trend toward disaggregation of storage into the fabric, the requirement for on-board storage controllers is rapidly diminishing.



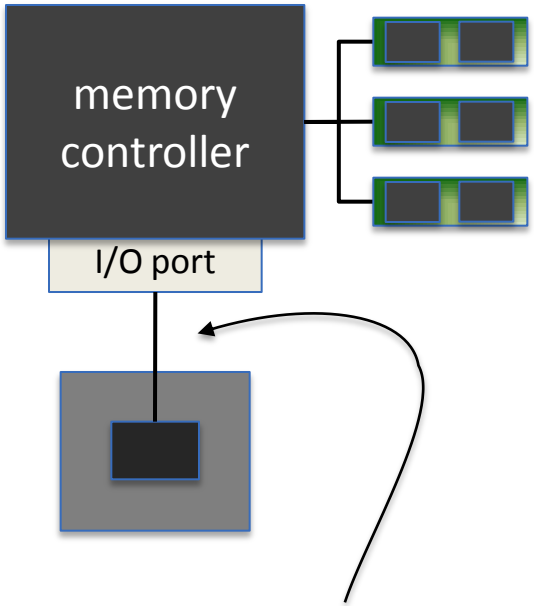
Eliminating the unique storage adapter allows a simplification of the basic scale-out server

IP networking, IPC



The obvious place to start here is with an Ethernet fabric

Optimizing the I/O subsystem



But how much bandwidth here??

- We know we can encapsulate storage protocols within messages

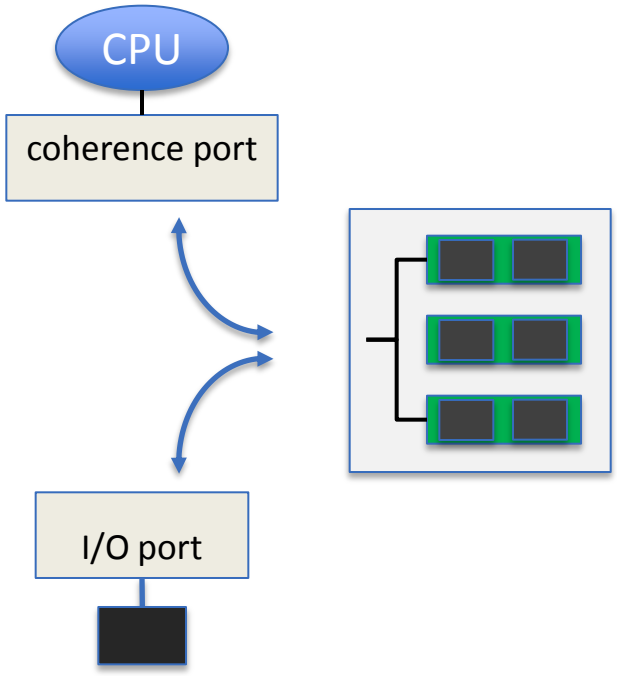
- We have a couple of options for IP and IPC traffic

- Maybe we can get away with only one I/O port into memory.

- Eliminate I/O protocol adapters?

- Ethernet is the obvious candidate for a unified I/O memory port

A balanced server

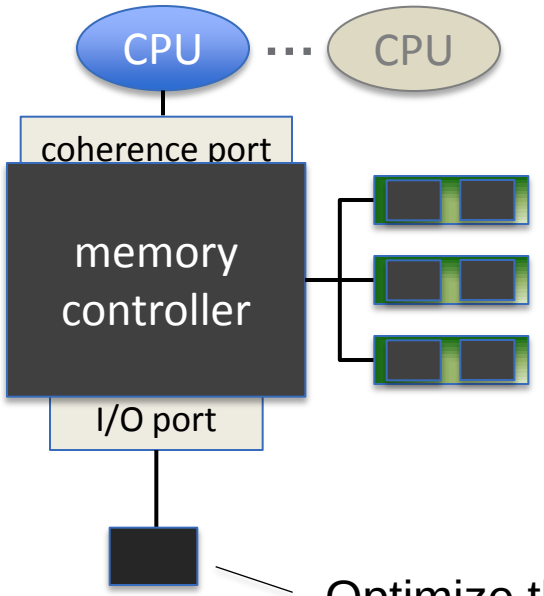


Choose the number of CPU cores to match the capacity of the I/O memory port (or vice versa)

As long as the sum of (storage traffic + IP traffic + IPC traffic) is less than or equal to the capacity of the I/O port, things are groovy.

(Provided we don't burn up too much memory bandwidth or CPU cycles in the process of delivering packets to applications)

A dense server cluster



About this point:

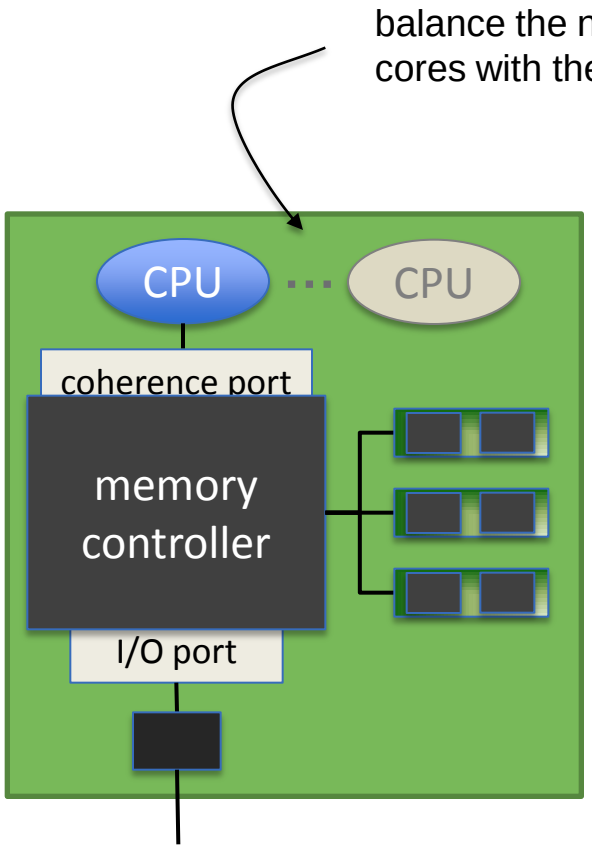
"Provided we don't burn up too much memory bandwidth or CPU cycles in the process of delivering packets to applications"

This, among other things suggests an RDMA-enabled I/O memory port.

Optimize the I/O stack, and associated adapter, for I/O message passing and IPC.

Since this talk is about RoCE, I'll ignore both IB and iWARP at present

A scale-down cluster processor

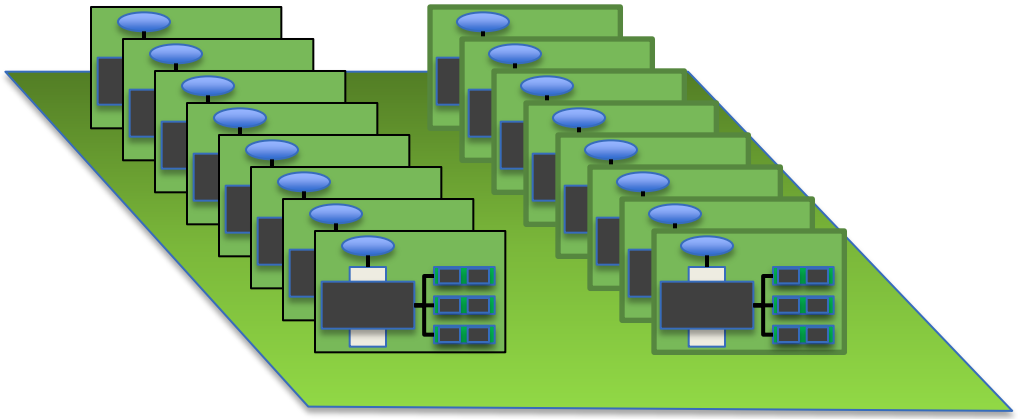


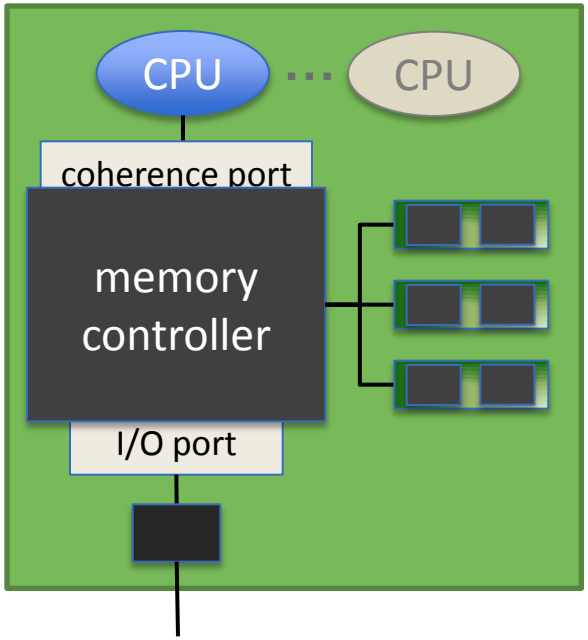
With all the benefits of RDMA:

1. CPU utilization.
2. Improve cluster performance
3. Reduce memory b/w demand
4. Balanced performance
5. Fine-grained compute particles

Fine-grained particles of compute consisting of a CPU/memory complex and a port into memory

Powerpoint scalability





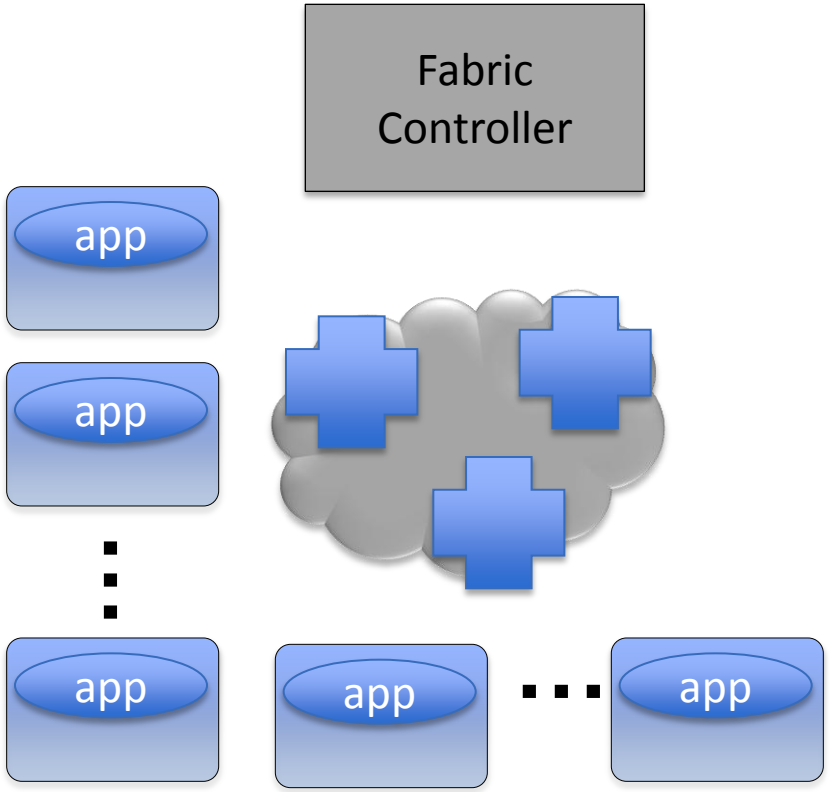
Summarizing the server platform

- If we disaggregate storage, what's left is IPC/RPC/client networking
- RDMA is very good at IPC/RPC
- implementing RDMA as the I/O memory port gives us a very efficient compute cluster with shared storage and bridging for client networking
- Conclusion: an RDMA port is sufficient!

Flat layer 2 fabrics

- We've already decided (by fiat, since this is a RoCE talk) that the fabric should be Ethernet
- But there are some well-known challenges:
  - spanning tree protocol
  - arbitrary topology support
  - latency optimization
  - efficient use of available bi-sectional bandwidth
  - 'hot-spot' avoidance…

A centrally-managed Ethernet fabric



- Very much like OpenSM or other centralized fabric managers

-Uses OpenFlow as the communication protocol

-arbitrary topology support

- efficient use of bi-sectional bandwidth

- congestion/hot-spot avoidance

- jitter reduction – very important in many environments

In other words, it is becoming practical to build large scale, flat address space layer 2 fabrics

These fabrics, as they emerge, should be capable of supporting large cluster sizes

A dense, scale-out cluster

1. traditional clusters are built of flexible monolithic servers
   - many core, many threads, highly integrated, complex I/O…
2. build ‘scale-down’ servers by rationalizing the I/O subsystem
   - use message passing service to conduct I/O protocols over a single fabric, eliminating much of the on-board complexity
   - no on-board storage
3. servers already have LOM, upgrade this to a RoCE packet processing engine,
4. Combine these with flat, centrally-managed layer 2 networks

To close…

efficient, fine-grained ‘scale-down’ endpoints

+

efficient layer 2 fabrics

Thank you