

Simple and Cheap

Theia: Networking for Ultra-Dense Data Centers

meg walraed-sullivan, Jitendra Padhye, David A. Maltz
Microsoft

HotNets 2014



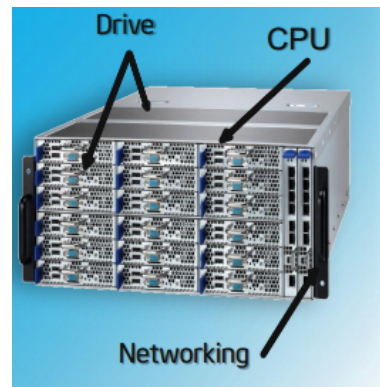
Ultra-Dense Data Centers (UDDCs)

- Data centers are expensive to build
- So we try to pack more hardware into existing data centers
- One way: pack more CPUs into a rack

SeaMicro



Intel RSA



HP Moonshot



FireBox

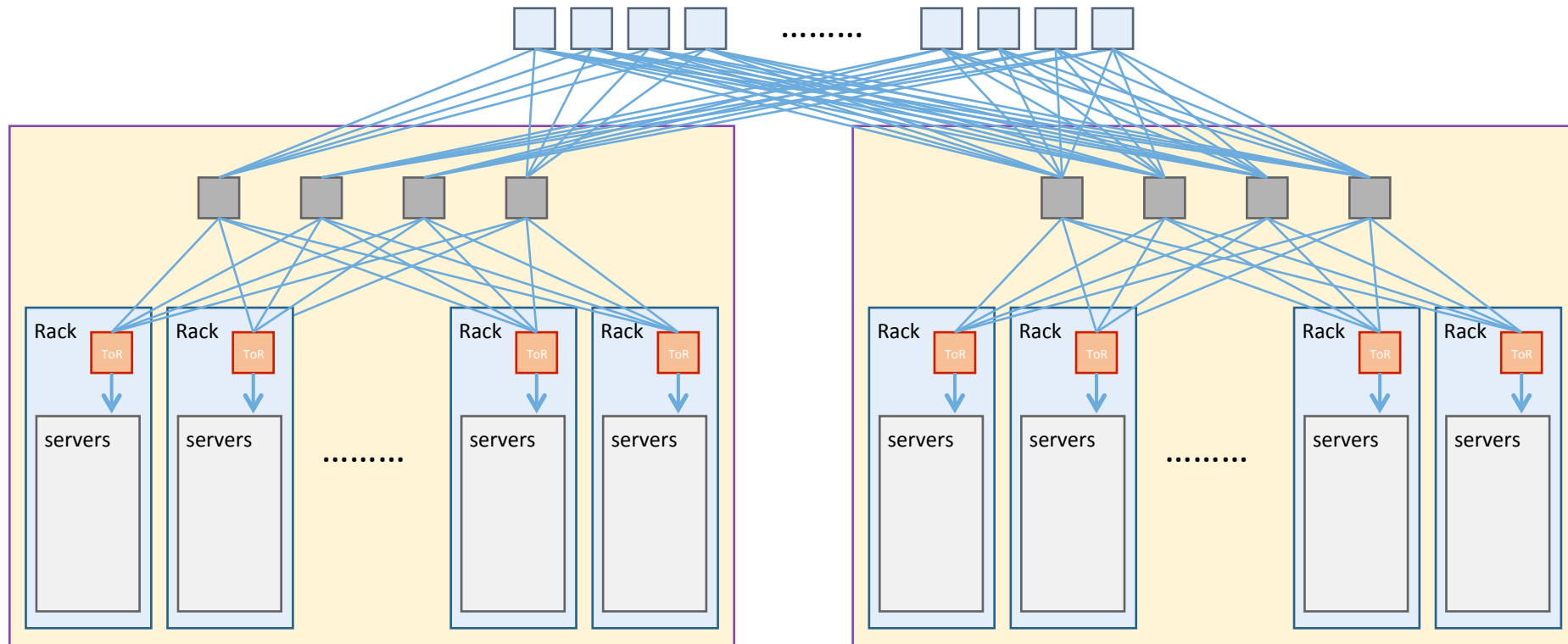


UDDC Challenges

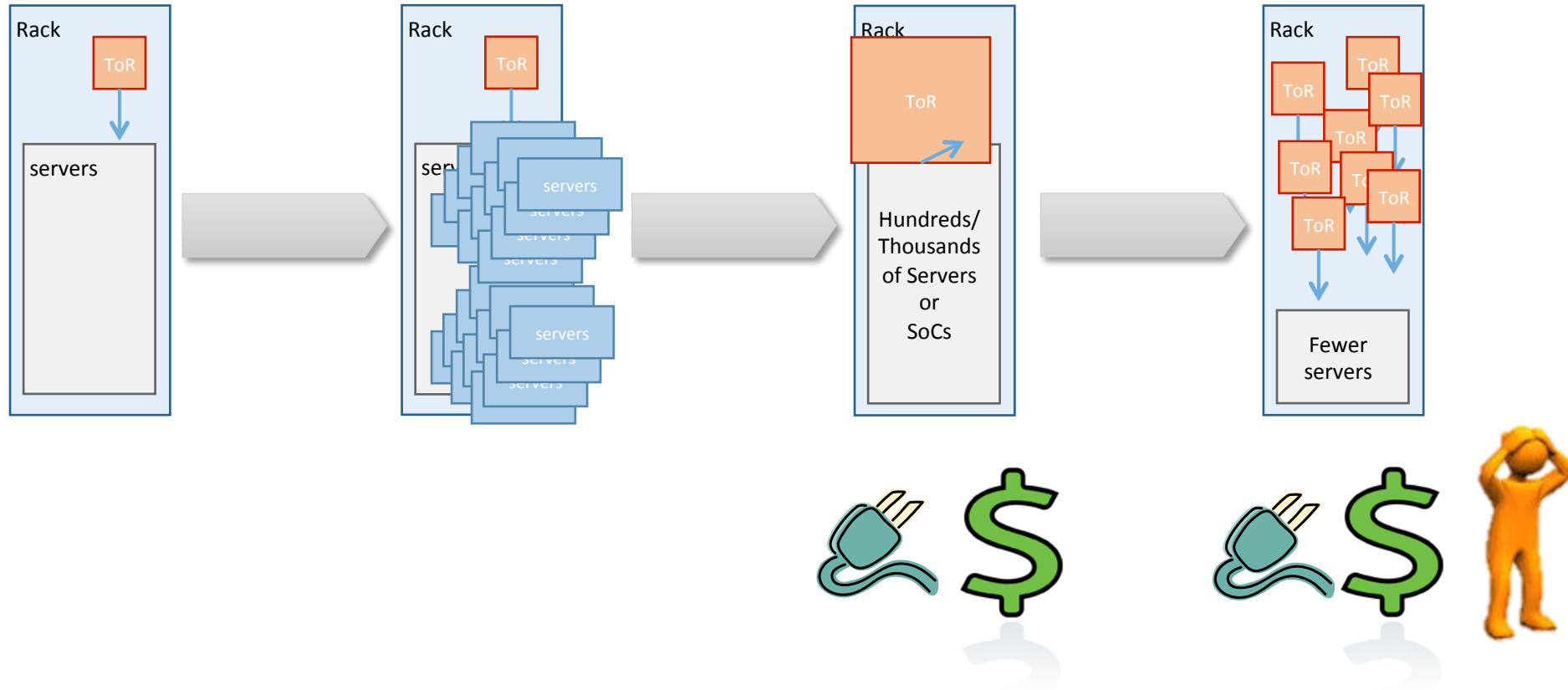
- System management
- Power and cooling
- Failure recovery
- How to tailor applications
- **Networking**

Traditional ToR-based architectures no longer appropriate due to
monetary cost
physical space requirements
oversubscription

Data Center Networks Today

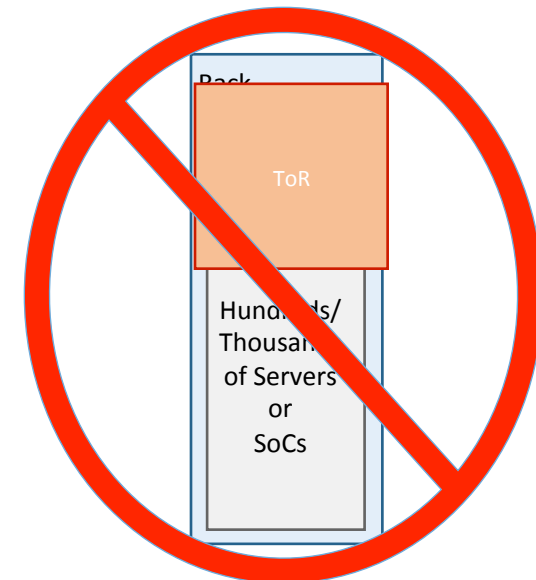
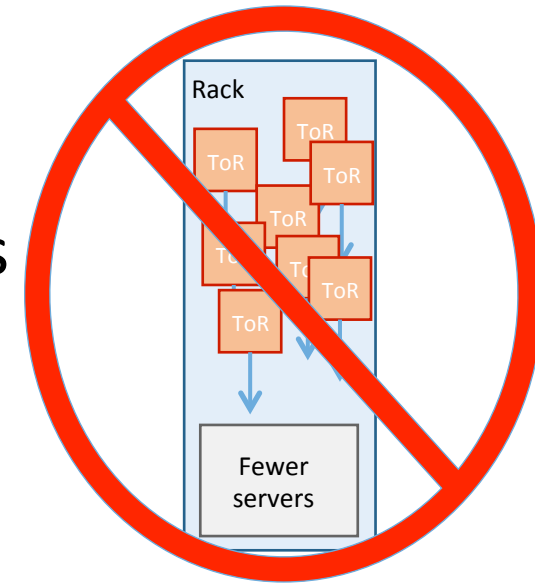


Why Rethink the Architecture?



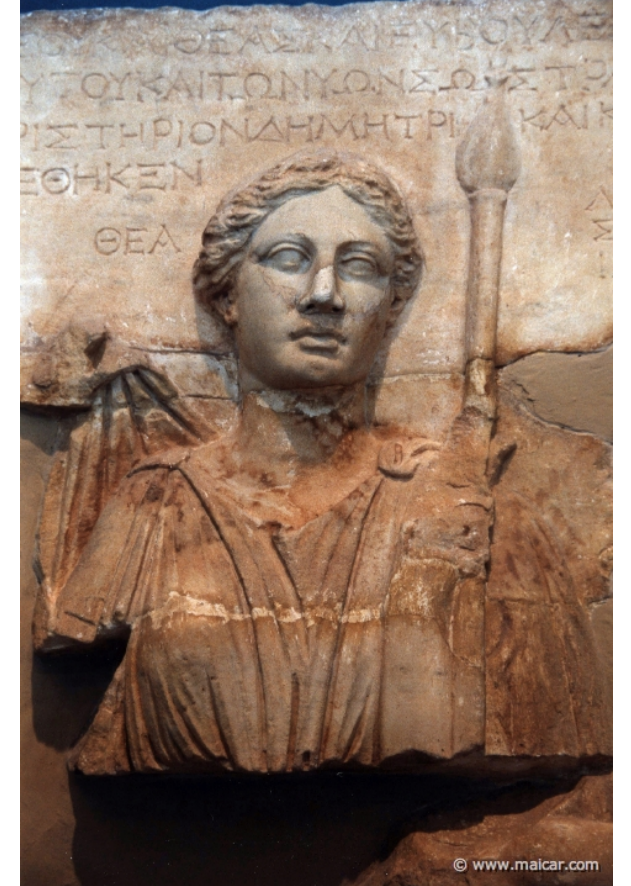
Why Rethink the Architecture?

- Problem: need to connect hundreds/thousands of servers
 - To each other
 - To rest of data center
- Naïve solutions won't work (cost, power, space)
 - Can't build a thousand-port ToR
 - Can't add many ToRs per rack
- Trade star topology for fixed, direct-connect topology
 - Upside: cheap, no power, small physical space
 - Downside: lose full bisection bandwidth, flexible topology



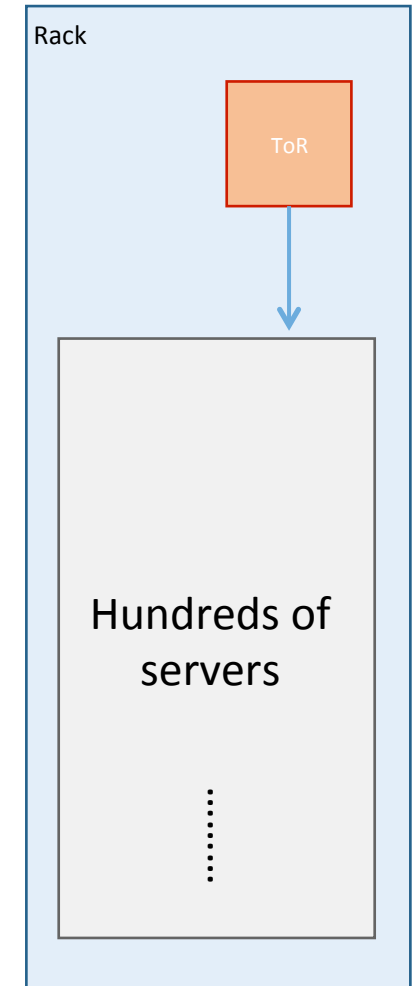
Theia

- *Preliminary* design for UDDC network architecture
 - Building out and evaluating new vendor hardware
 - Design will undoubtedly change as we progress
- Goal is a simple, practical, cheap design
 - Beg, borrow, and steal from existing technologies
 - Throw hardware at the problem when it is cheap, software when not
- Theia is meant to start a conversation about UDDCs



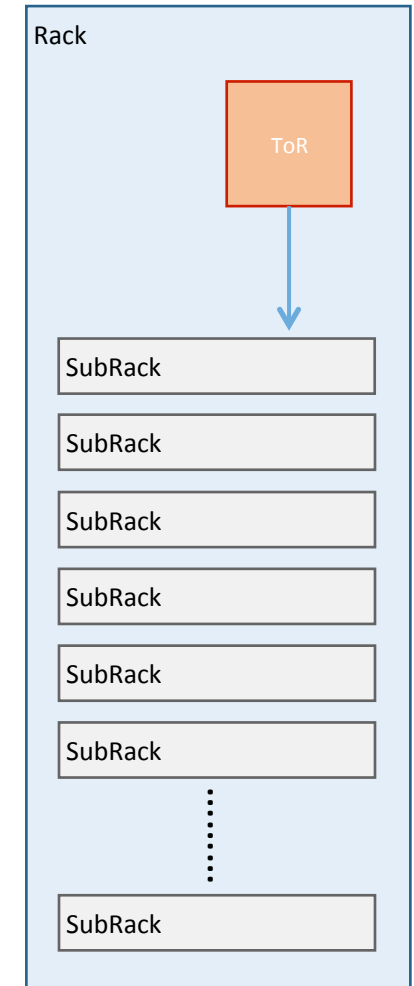
The Theia Architecture

- Start with traditional rack



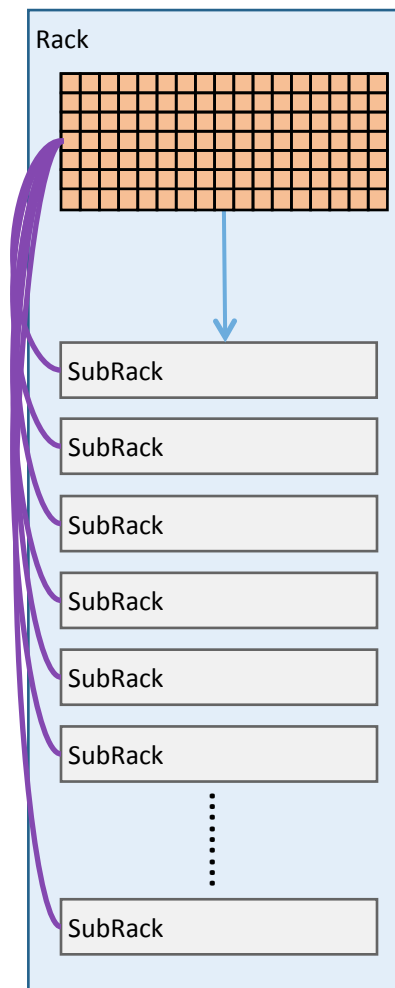
The Theia Architecture

- Start with traditional rack
- Divide servers into SubRacks



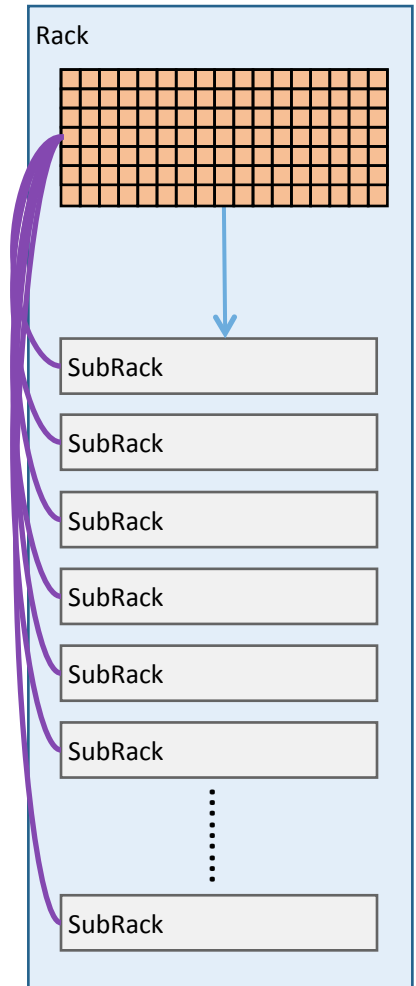
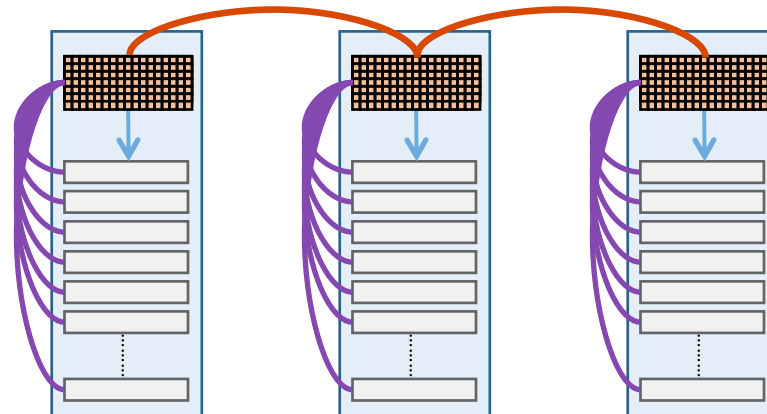
The Theia Architecture

- Start with traditional rack
- Divide servers into SubRacks
- Replace ToR with fixed circuit interconnect (patch panel)



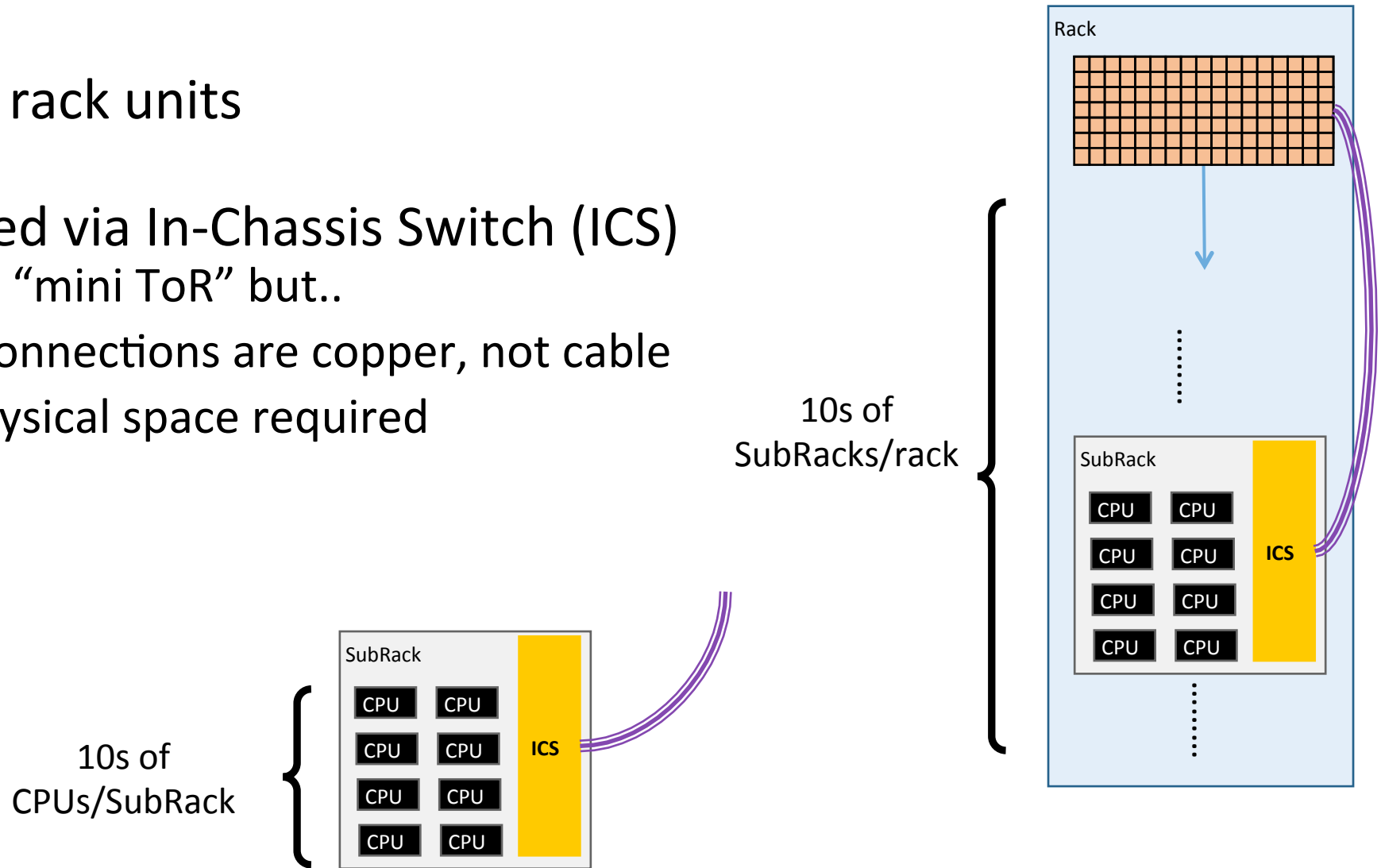
The Theia Architecture

- Start with traditional rack
- Divide servers into SubRacks
- Replace ToR with fixed circuit interconnect (patch panel)
- Connect racks to one another using spare patch panel ports



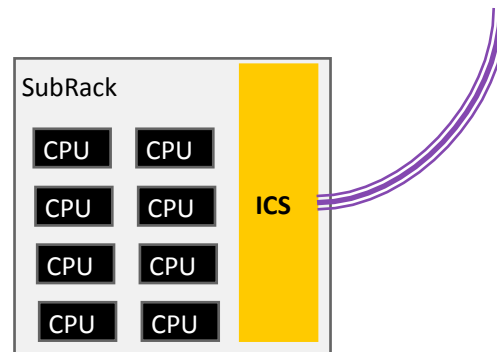
Theia Architecture: SubRacks

- SubRack \approx 1-2 rack units
- CPUs connected via In-Chassis Switch (ICS)
 - Like our own “mini ToR” but..
 - ICS-to-CPU connections are copper, not cable
 - Very little physical space required

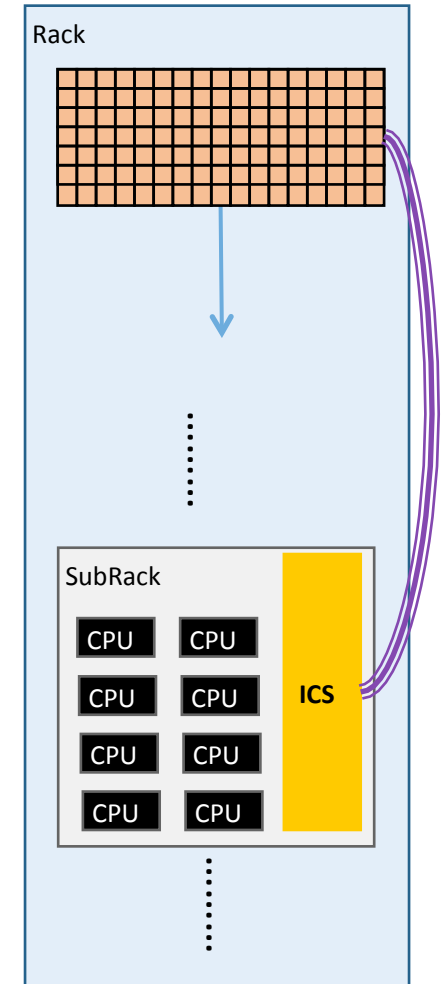


Theia Architecture: SubRacks

- Can tune (at deployment time) number of downlinks (ICS-CPU) vs. uplinks (ICS-patch panel and rest of rack)
- Tradeoff at ICS: aggregation vs. oversubscription
 - Oversubscription ratio: # uplinks : # CPUs

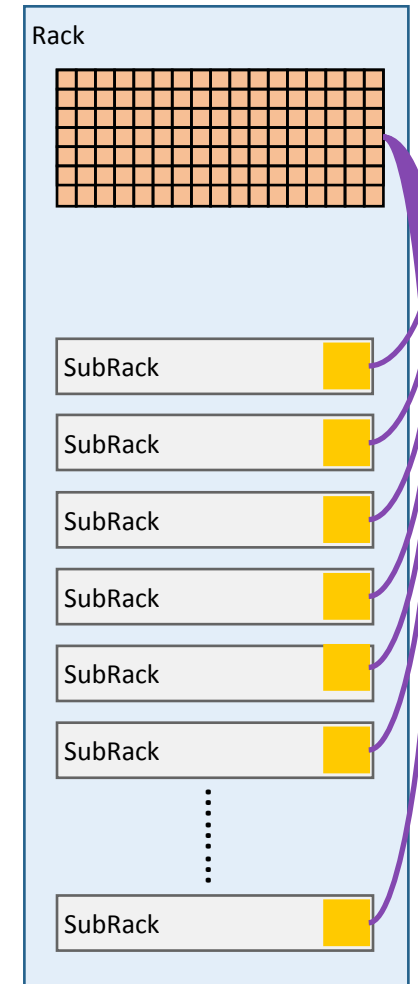
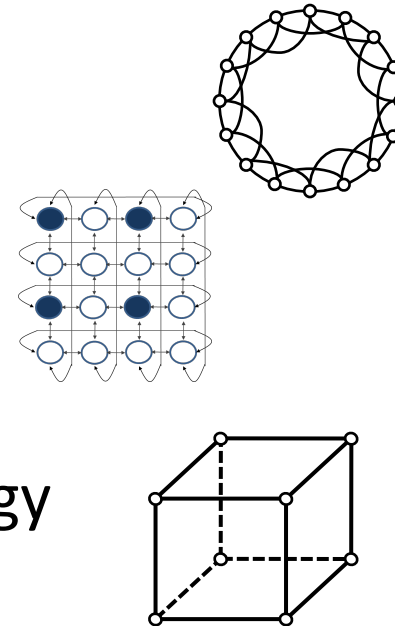


Initially, \leq ten uplinks



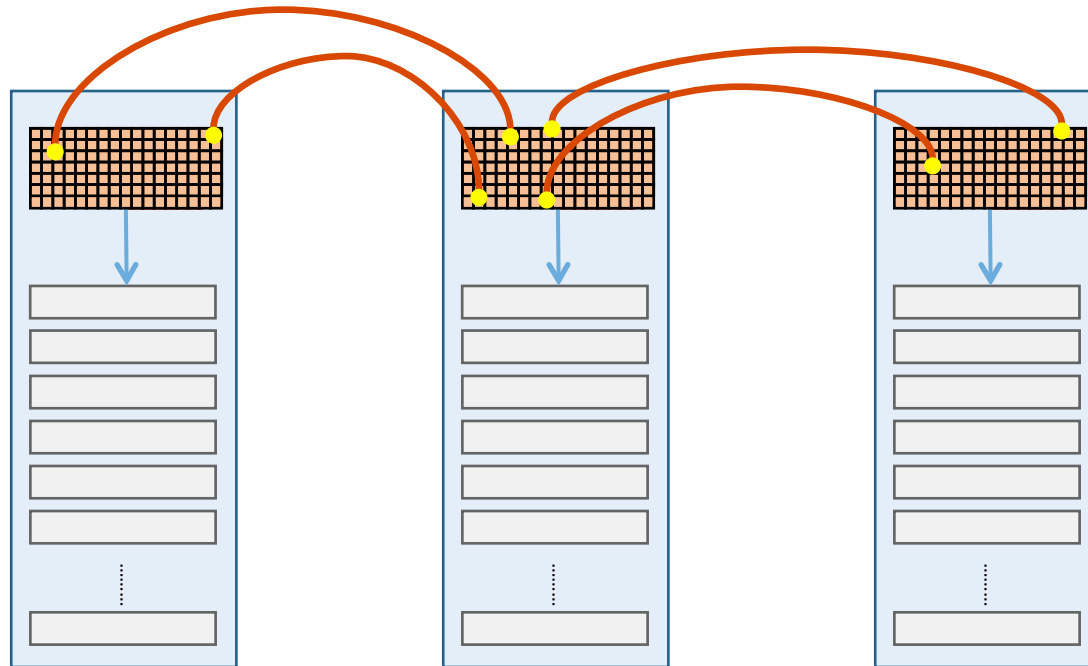
Theia Architecture: Patch Panel

- Patch panel connects *SubRacks* to one another
 - 10s of SubRacks with ~10 uplinks = hundreds of ports
- Optical patch panel implements a fixed circuit topology
 - No active components
 - Draws no power
 - Compact
 - Adds no queuing delay
 - Cabling is simple (underlying topology is hidden)
- Tradeoff:
cost (power, space, \$) vs. fixed, direct topology



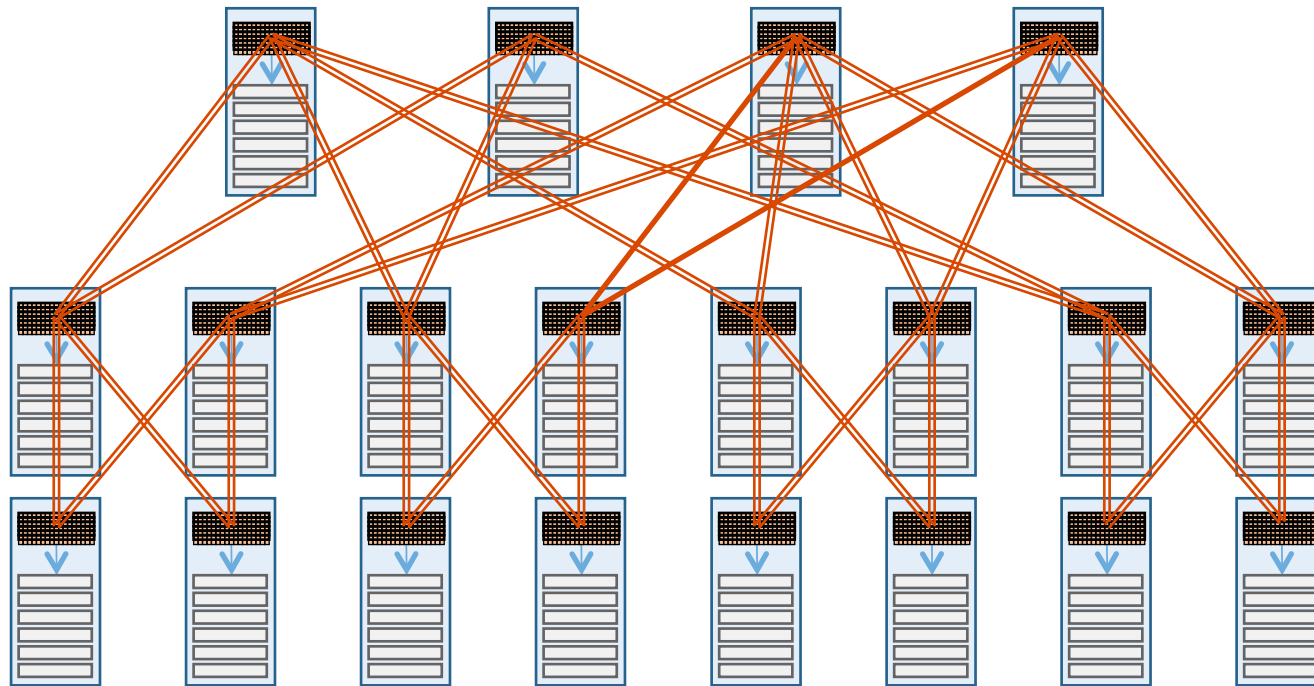
Theia Architecture: Inter-Rack Connectivity

- Repurpose “leftover” patch panel ports to interconnect racks
- Link between 2 racks may be groups of multiple links



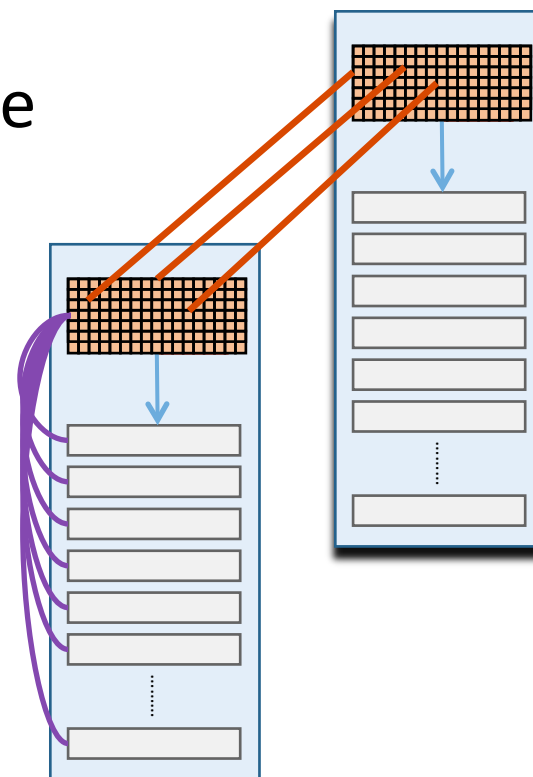
Theia Architecture: Inter-Rack Connectivity

- Repurpose “leftover” patch panel ports to interconnect racks
- Link between 2 racks may be groups of multiple links
- Build larger topology w/ each rack as a “super node”



What about oversubscription?

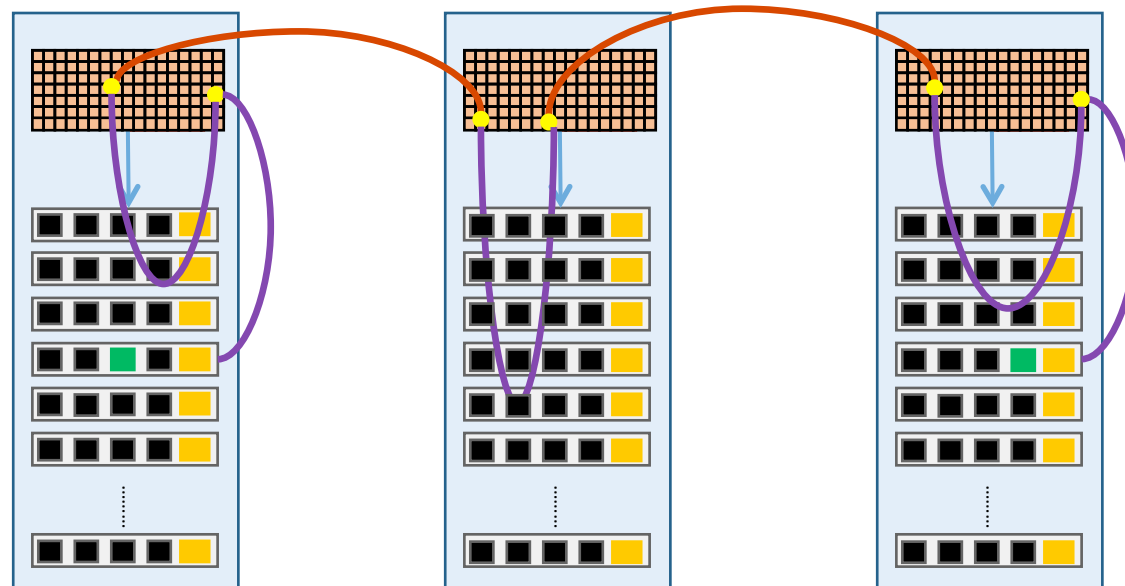
- At this scale, oversubscription is unavoidable
- More rack-locality can be expected



Tune this oversubscription by allocating patch panel ports to in-rack interconnect (purple) or inter-rack interconnect (red)

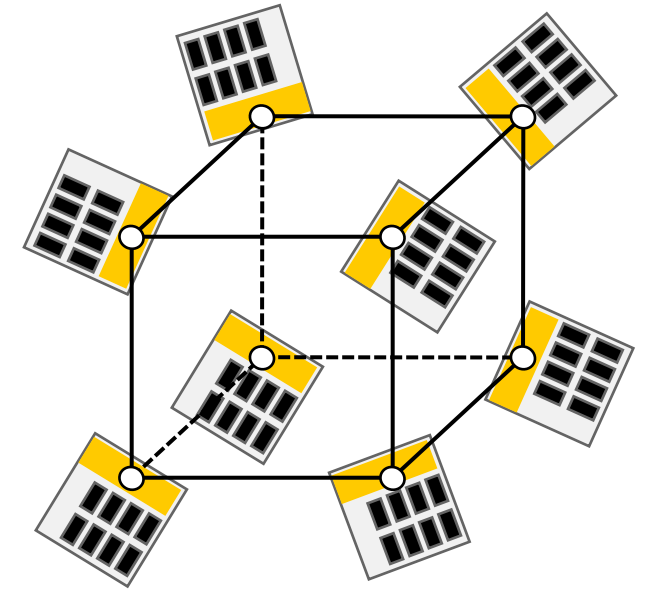
What about through-traffic?

- Traffic passes through intermediate racks
- Traffic traverses the patch panel (and therefore ICSs)

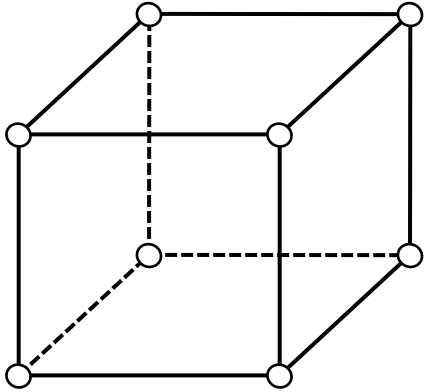


Patch Panel Topology

- Graph in which
 - Each node is an ICS (and its corresponding SubRack)
 - Links are implemented by patch panel internals
- What we care about:
 - Minimize through-traffic: latency and failure resilience
 - Support wide-range of graph sizes: UDDCs are still new
 - No dependency between number of nodes and ports per node
 - Reduce disruptions caused by failures and miscablings

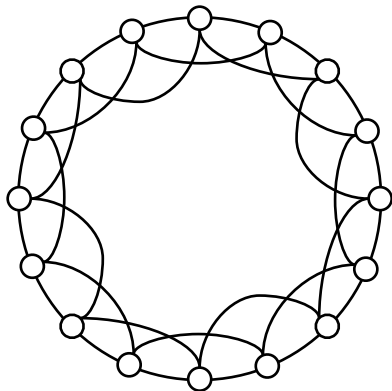
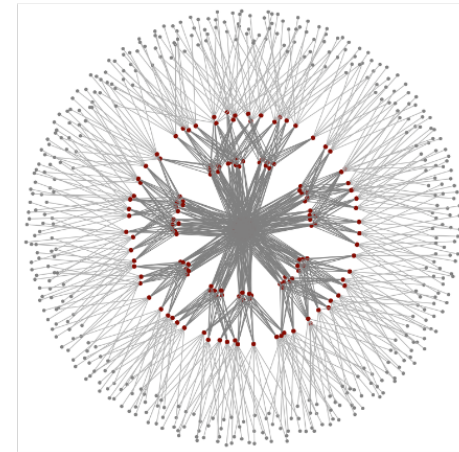


Patch Panel Topology Options



Hypercube: constraints on number of nodes, port counts, dependency between the two (similar for torus, Dcell, Bcube, etc)

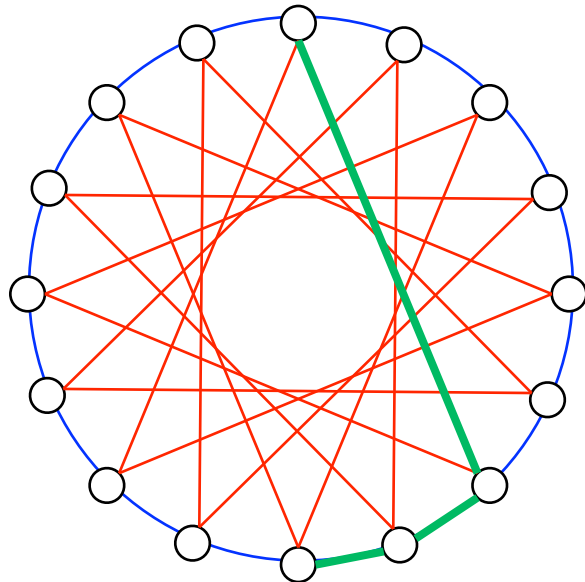
Jelly fish: allows for organic growth, but this is not needed with fixed topology patch panel



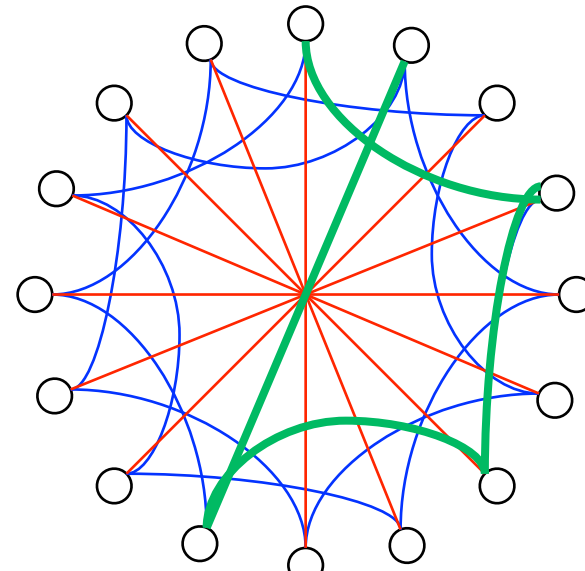
Circulant Graph: Can build a performant graph w/ any number of nodes, port counts.

Initial Topology: Circulant Graph

- Nodes $N=\{0,\dots,N\}$
 - With p ports/node
- Strides $S=\{\dots s\dots\}$ s.t. node i connects to nodes $i\pm s$
- ...A ring with “short cuts”
 - Key is to pick good shortcuts given N and p



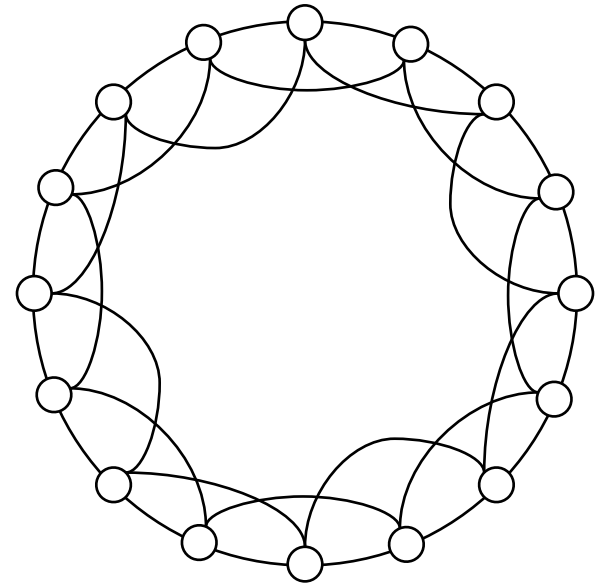
$S=\{1,6\}$
Avg Path Len = 1.933
 $\frac{1}{2}$ are 2-hops
Worst = 3 hops



$S=\{3,8\}$
Avg Path Len = 2.6
~Even split btwn 1,2,3,4

Circulant Graph

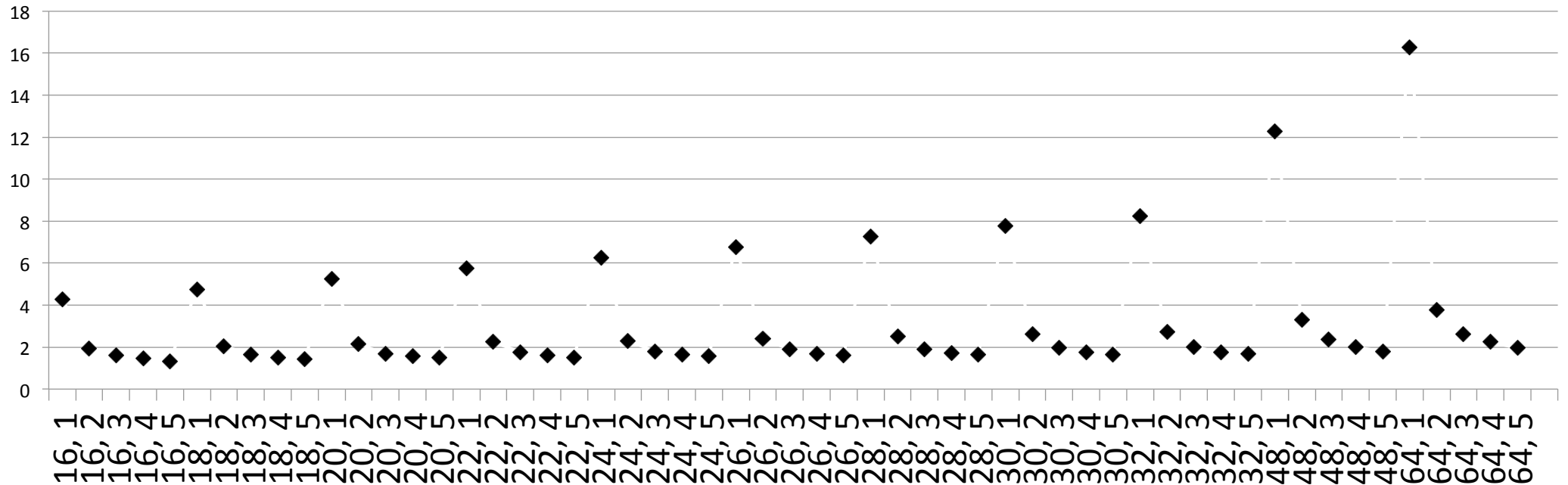
- Initial reasons for choosing: Flexibility
 - Wide range of graph sizes
 - No dependency between port count and number of nodes
- Turns out to be quite performant
 - Low amount of “through” traffic
 - Resilient to failure in connectivity, performance, and consistency
 - Simple, elegant routing and forwarding
 - Miswirings likely to cause isomorphic graphs



Circulant Graph Average Path Lengths

Latency and Through-Traffic

Best Avg. Path Length Across
Stride Sets



Circulant Graph Size <# Nodes, # Strides>

Summary

- ToR-based architecture won't work for UDDCs
- Theia: Preliminary architecture to support 1000s of CPUs/rack
 - Flexibility of packet-switched network over fixed circuit topology
- Just the beginning of this conversation:
 - Other in-rack topologies...
 - Inter-rack connectivity: will our proposal scale to data center size?
 - Routing and addressing: different protocols for inter- and intra- rack?
 - Tailoring topology to workload and workload to (dense) topology