Rockies: A Network System for Future Data Center Racks

PURDUE UNIVERSITY

Shin-Yeh Tsai, Linzhe Li, Yiying Zhang

WukLab

Future Data Center Racks

Disaggregated Resources
- Break monolithic machines into disjoint resource components
- Flexible combination, addition, removal, and upgrade of hardware components
- E.g., HP "The Machine", Berkeley Firebox

Fast, Large-Scale Components
- Non-Volatile Memory, NVMe-based SSDs
- CPU, GPGPU
- 100-1000 components

Why a New Rack Network?

Requirements of Future Rack Network
- OS operations go through network
- Low-latency, high-bandwidth
- Scale to ~1000 connection points

Limitations of Traditional Rack Networks
- OS-oblivious, do not handle OS operations
- Do not offer low latency
- Do not scale

Our Solution: Rockies

New RDMA-Based Rack Network Layer

Co-Design Network and OS

Exploit Scale of Rack

OS-Aware Topology and Routing

Benefits
- Low latency, even under heavy traffic
- High throughput
- Low monetary cost
- Scale to ~1000 components

80% better performance than 3D-switch

50% lower $ than 3D-DC

3D-switch: switches connected with 3D-Torus
3D-DC: components directly connected with 3D-Torus

Rockies Architecture

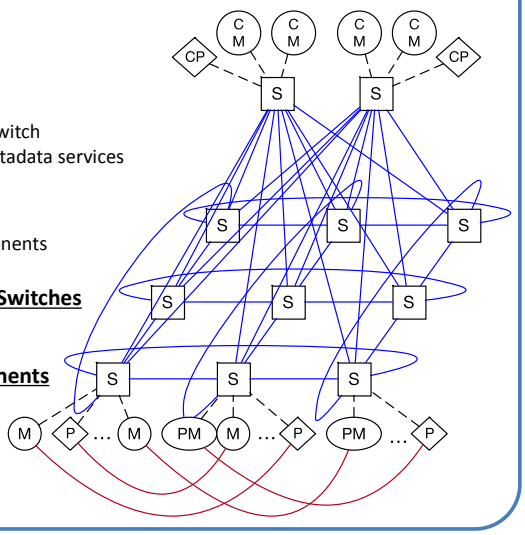
Two Layers of Switches

Top-layer switches
- Connect to all bottom-switches
- Few processors and memories per switch
- Used for central management or metadata services

Bottom-layer switches
- Form a 2D-torus
- Most ports used for resource components

All Components Directly Connect to Switches
- Fast intra-switch communication

Direct Connections Between Components
- Processors connect to memories on other switches
- Avoid congestions in switch links
- Tolerate switch and switch link failures

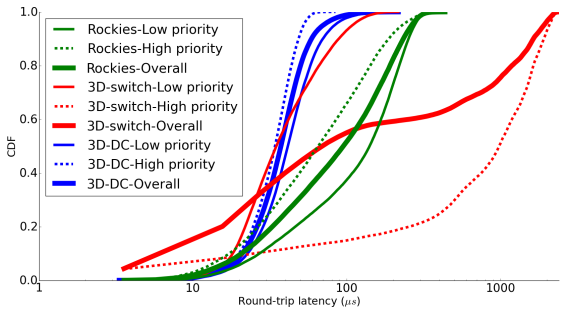


Initial Results

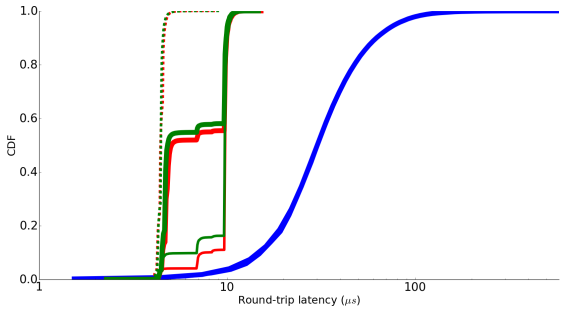
Research Questions
- How does Rockies compare with other networks?
- How does Rockies perform under data center workloads?
- Does Rockies ensure QoS of different OS operations?

Environment
- Implemented on top of OMNeT++ simulator
- 810 components, 27 switches
- Compare with 3D-switch and 3D-DC

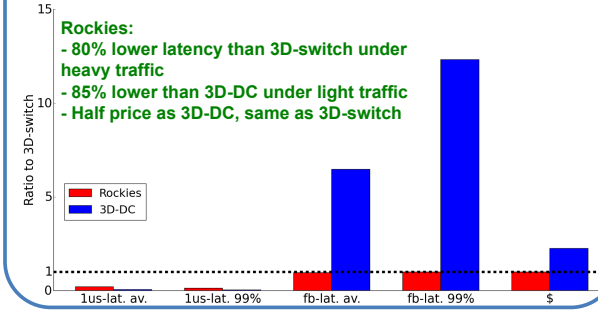
Key-Value Get Latency under Heavy Traffic
- 1us inter-arrival time, 4B keys, 4KB values



Facebook Key-Value Get Latency (Light Traffic)
- Modeled according to Facebook distributions [1]



Comparison of Rockies, 3D-switch, and 3D-DC



CO-Design Network with Operating System

Valuable OS Information
- Priorities
- Dependencies
- Type-based

Pass Information from OS to Network
- Explicit (tagging requests)
- Implicit (using hints)

OS-Aware Routing and Congestion Control

Priority-Based Routing and CC
- Use different types of connection
  - High priority → switch links
  - Low priority → direct links
- Use different Infiniband QoS channels
- Dynamic deadlock-free min-hop algorithm

Exploit Scale of Rack and Infiniband
- Small-scale broadcast
- Lossless communication with Infiniband
- Infiniband congestion control signals

[1] B. Atikoglu, Y. Xu, E. Frachtenberg, S. Jiang, and M. Paleczny. Workload Analysis of a Large-Scale Key-Value Store. In Proceedings of SIGMETRICS '12.

Rockies: A Network System for Future Data Center Racks

Shin-Yeh Tsai*, Linzhe Li*, Yiying Zhang
Purdue University (are students)*

Two recent trends are reshaping data center racks and software systems that manage them. First, there is an increasing interest in organizing racks with disaggregated resources instead of traditional monolithic servers [1, 3]. The disaggregated racks enable flexible combination, addition, removal, and upgrade of various hardware components. Meanwhile, the disaggregated architecture puts traffic that used to be inside individual servers on rack network, requiring higher bandwidth and low latency from the network layer.

Second, individual resource components are also evolving. Fast, byte-addressable persistent memories [4] and NVMe-based SSDs are coming to market. Rack networks need to improve their performance to keep up with these new storage devices.

Both these trends point to one need: a fast, efficient, and flexible rack-scale network layer to enable communication between components in future racks. Traditional rack-scale network systems such as those that connect all servers to a Top-of-Rack switch will not fit future racks because they do not provide the bandwidth, latency, and scale requirements future rack systems demand. Moreover, traditional rack network layers are OS-oblivious and only handle application traffic. With disaggregated racks, operating system operations between resource components have to go through the same rack network, dramatically changing the landscape of rack network.

We propose *Rockies*, an RDMA-based network system for future data center racks. Our key idea is to exploit the close relationship of the operating systems and the network in future racks by co-designing the network layer with the rack operating systems and management systems. We also exploit the specific scale of our targeted racks, 100 to 1000 components [2], throughout our design. With these design principles, Rockies aims to achieve low latency, high bandwidth, quality-of-service, flexibility, failure tolerance, and low monetary costs.

Rockies Topology. We propose a new Infiniband-based topology designed for future racks by leveraging both direct connection and connection through switches.

First, based on our targeted rack scale, we organize switches into two layers. The top layer consists of a small number of switches (usually one or two). These switches directly connect to all switches in the bottom layer. We then connect the bottom-layer switches with a 2D torus. The maximum min-hop distance between any switches is two. This topology also restricts the number of cycles in the switch network, reducing the overhead of deadlock-free routing algorithms required by Infiniband.

Next, we use most ports of the bottom-layer switches to connect resource components, ensuring low-latency intra-switch communication. With most ports connecting to the bottom-layer switches, the top-layer switches use the few remaining ports to connect few processors and memory components that can serve for central management or metadata service, a common requirement by many data center software systems.

Finally, we directly connect processors of one switch to memories of other switches for path diversity and failure isolation. With our targeted rack scale, all components can be connected through direct connections. Thus, when the network between switches is congested or fails, the direct connections can handle all traffic.

OS-Aware Routing and Congestion Control. Rockies fully exploit operating system information to guide its routing and congestion control algorithms. Such information includes operation priorities and dependencies, and can be passed from the OS either explicitly (by tagging OS operations) or implicitly (by using hints from the OS). Based on this information and network congestion status, Rockies dynamically routes different types of traffic via different connections (direct or through switches), paths, and Infiniband QoS channels.

Rockies also takes advantage of the limited scale of a rack to reduce various routing and flow control overheads. Each component broadcasts its load, link usage, and congestion signals only to all other components on the same switch and the component it directly connects with. Because of the limited rack scale, the locally broadcasted information is sufficient for many Rockies decisions.

Initial Results. We have implemented Rockies on top of the OMNeT++ [5] simulator and evaluated it with micro- and macro-benchmarks. Our initial results with simulation show that Rockies outperforms a typical 3D-torus network by 70% under heavy traffic, a likely situation in disaggregated racks. Its performance is on par with a directly-connected network under heavy traffic and 85% better under lighter traffic, and only requires around half the monetary cost. We are currently implementing Rockies in Linux to evaluate it with real systems.

References

[1] K. Asanovi. FireBox: A Hardware Building Block for 2020 Warehouse-Scale Computers, February 2014. Keynote talk at FAST '14.
[2] Hewlett Packard. HP Moonshot SYstem. http://www8.hp.com/us/en/products/servers/moonshot/.
[3] Hewlett Packard. The Machine: A New Kind of Computer. http://www.hpl.hp.com/research/systems-research/themachine/.
[4] Intel Corporation. Intel Non-Volatile Memory 3D XPoint. http://tinyurl.com/phom9x8.
[5] O. Ltd. OMNeT++ Discrete Event Simulator. https://omnetpp.org/.