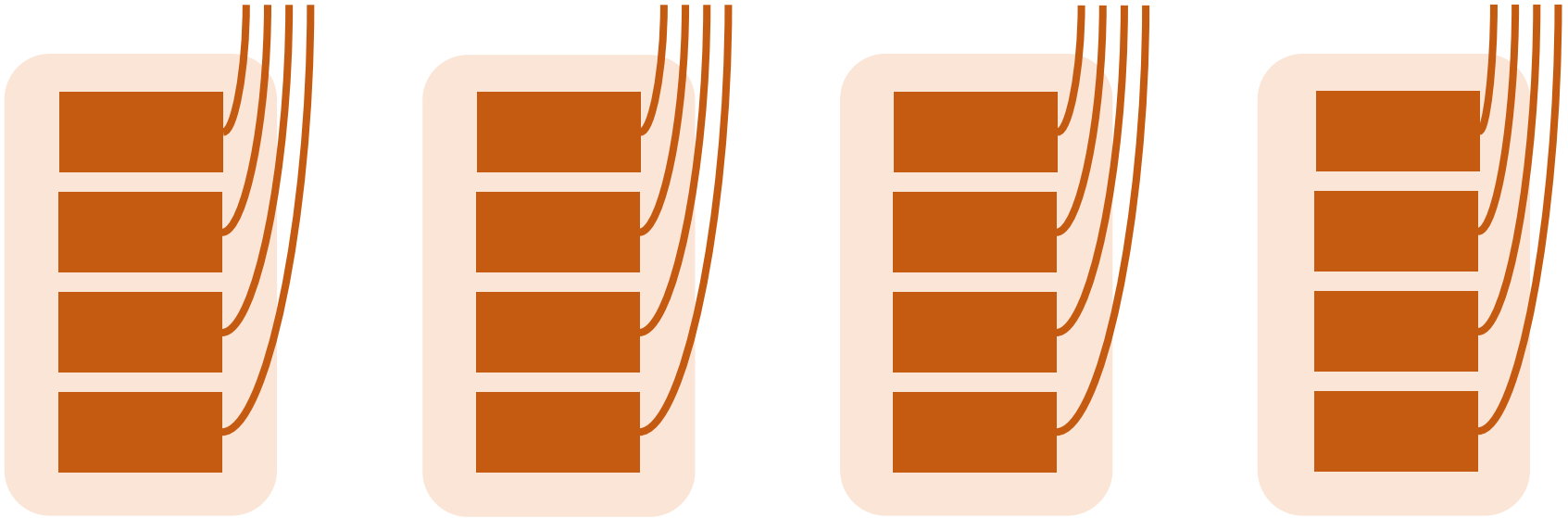
RDC: Energy-Efficient Data Center Network Congestion Relief with Topological Reconfigurability at the Edge

Weitao Wang, *Dingming Wu, Sushovan Das, Afsaneh Rahbar, Ang Chen, T. S. Eugene Ng

Top-of Rack Design: Almost Every Data Center Uses

Servers are organized in racks.



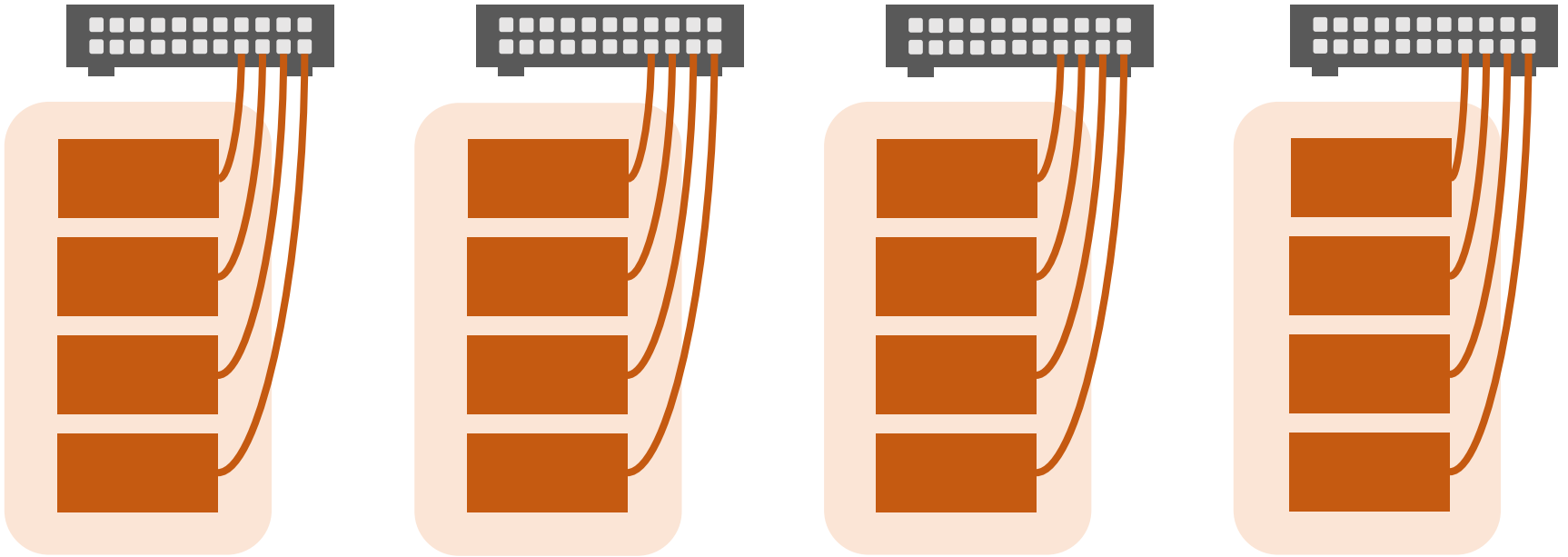
Top-of Rack Design: Almost Every Data Center Uses

Add ToR Switches to connect all the servers under the same rack.

Independent power supply and others

Reduced cabling complexity and cable length

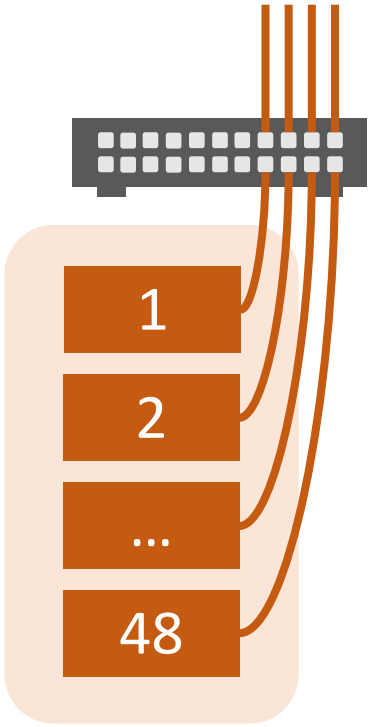
ToR Switches



Over-subscription: A Common but NOT Favorable Design

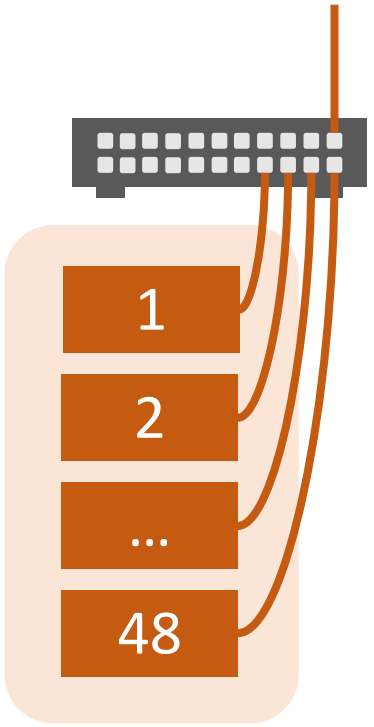
With a non-blocking network, the cost for building the network core is very high.
To reduce the cost, **over-subscription** is employed to attach more devices.

Non-blocking:
48 uplinks
48 downlinks



Rack

Over-subscribed:
12 uplinks
48 downlinks
(1:4 – o.s.)

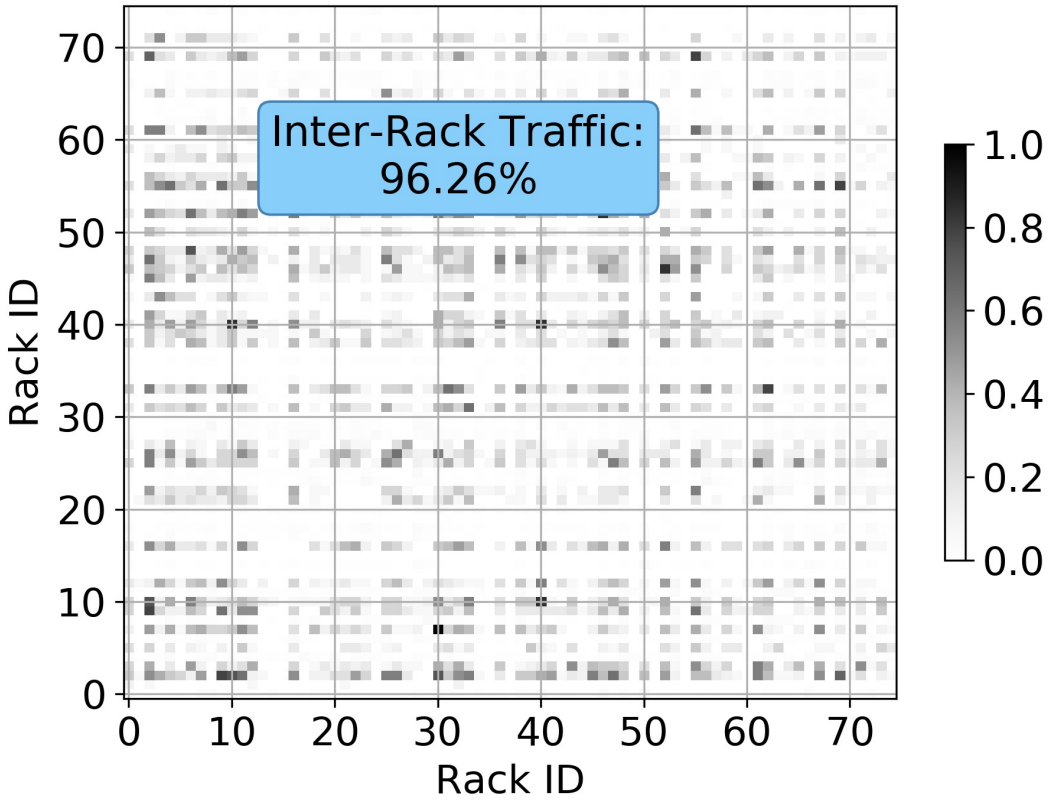


Rack

Over-subscripted ToRs: Bottleneck For **Inter-rack Traffic**

Increasing **inter-rack traffic** demand:

- ToR become bottleneck for inter-rack traffic;
- Bandwidth under the same ToR are wasted;



Facebook cluster packet trace:

- Cluster type: **Web-frontend**
- Duration: 24 hours
- Sampling rate: 1:30k

Key Observation: Traffic are Skewed at Server-level

For those inter-rack traffic:
During a short time period, we can find many server groups across different racks.
Communications are **intensive among servers within the same group**.

Key Observation: Traffic are Skewed at Server-level

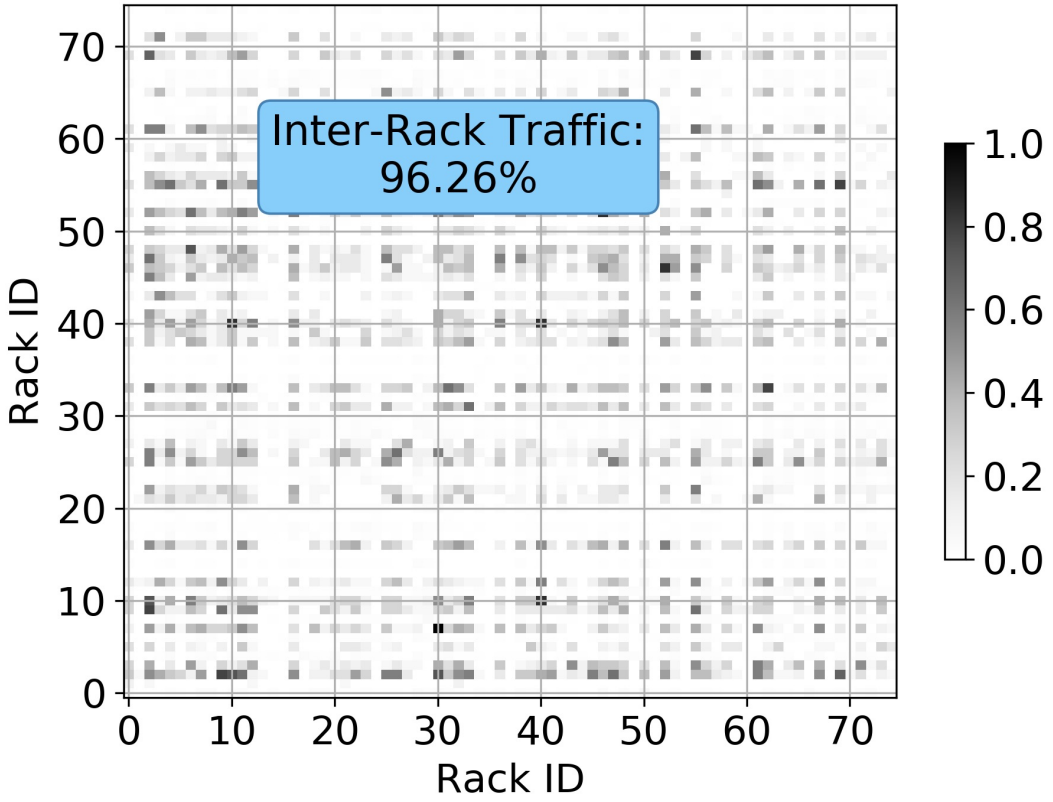
For those inter-rack traffic:
During a short time period, we can find many server groups across different racks.
Communications are **intensive among servers within the same group**.

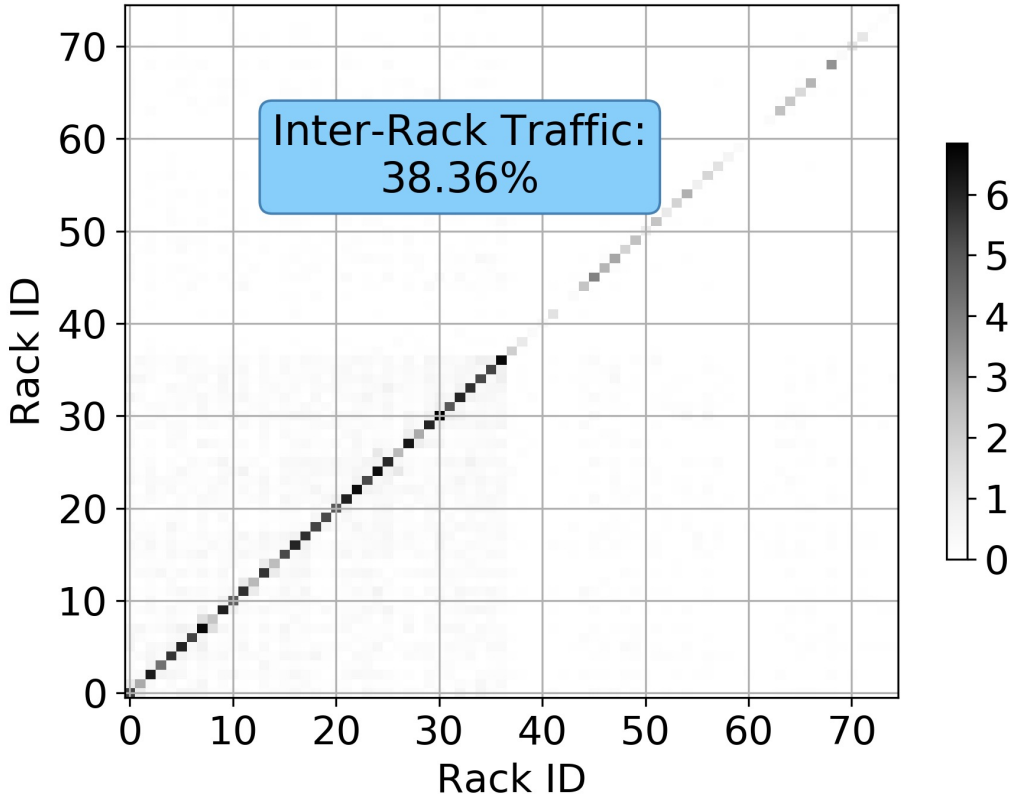
What if we can regroup those servers to be under the same ToR?

Power of Regrouping Servers to Break the Rack Boundary

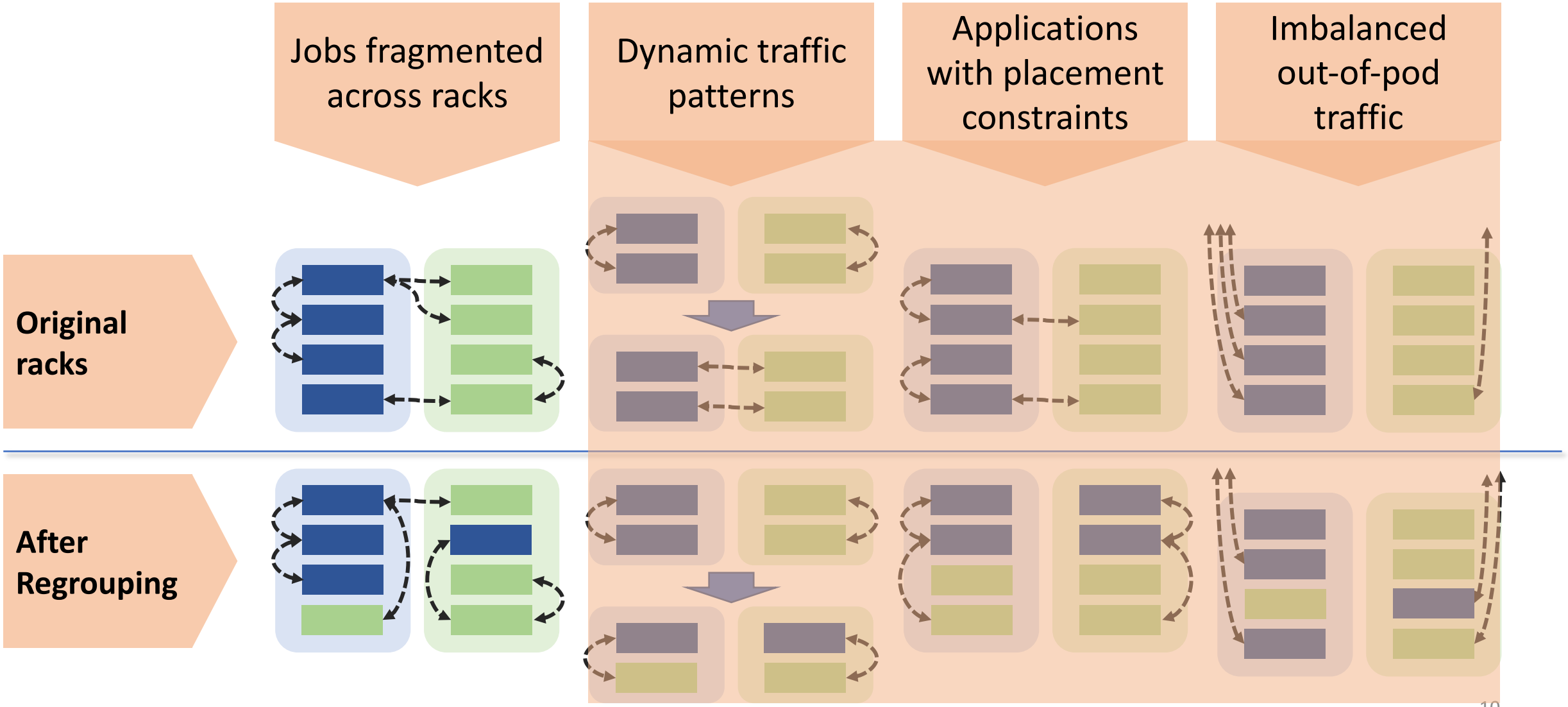
After regrouping the servers in the Facebook trace:

- Localize most of the inter-rack traffic, only 38.36% of inter-rack traffic remains;

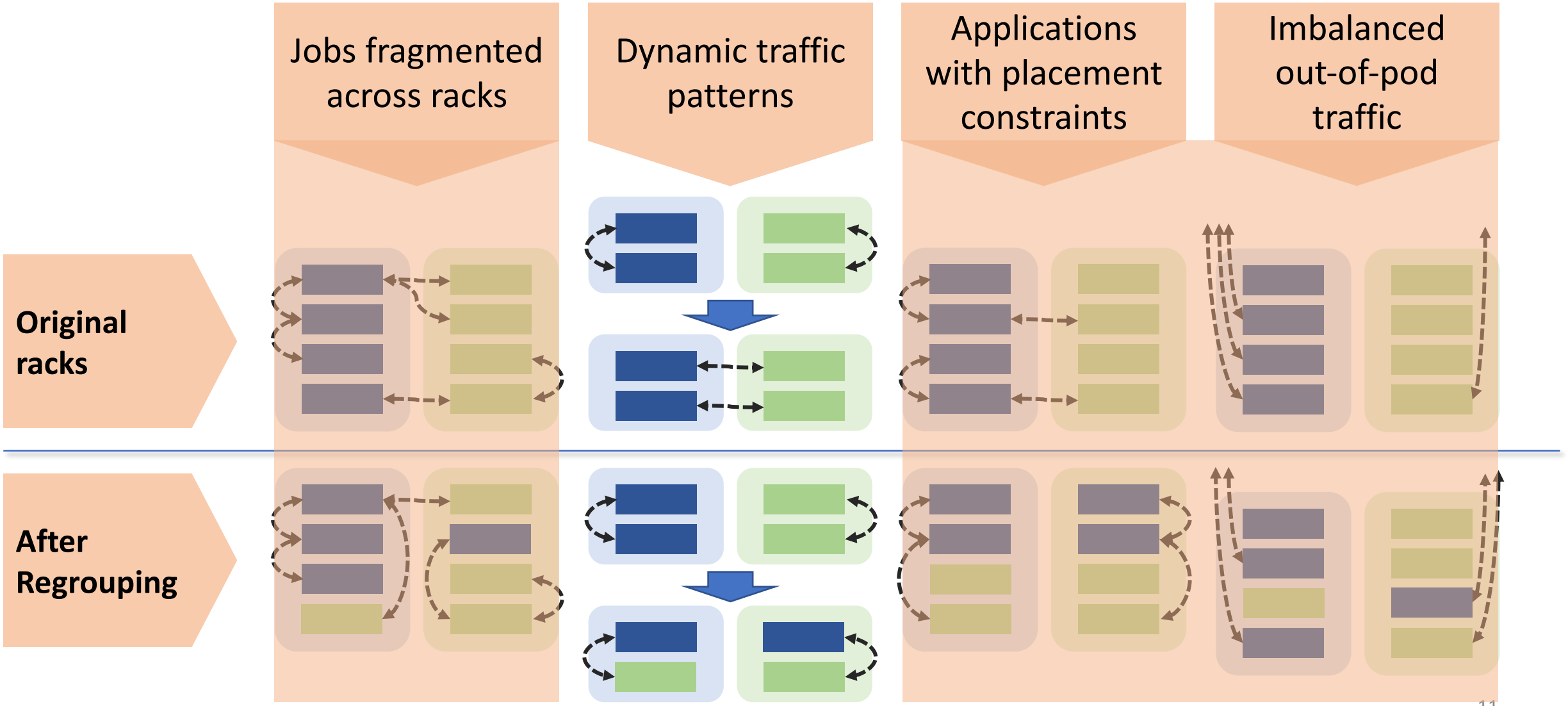




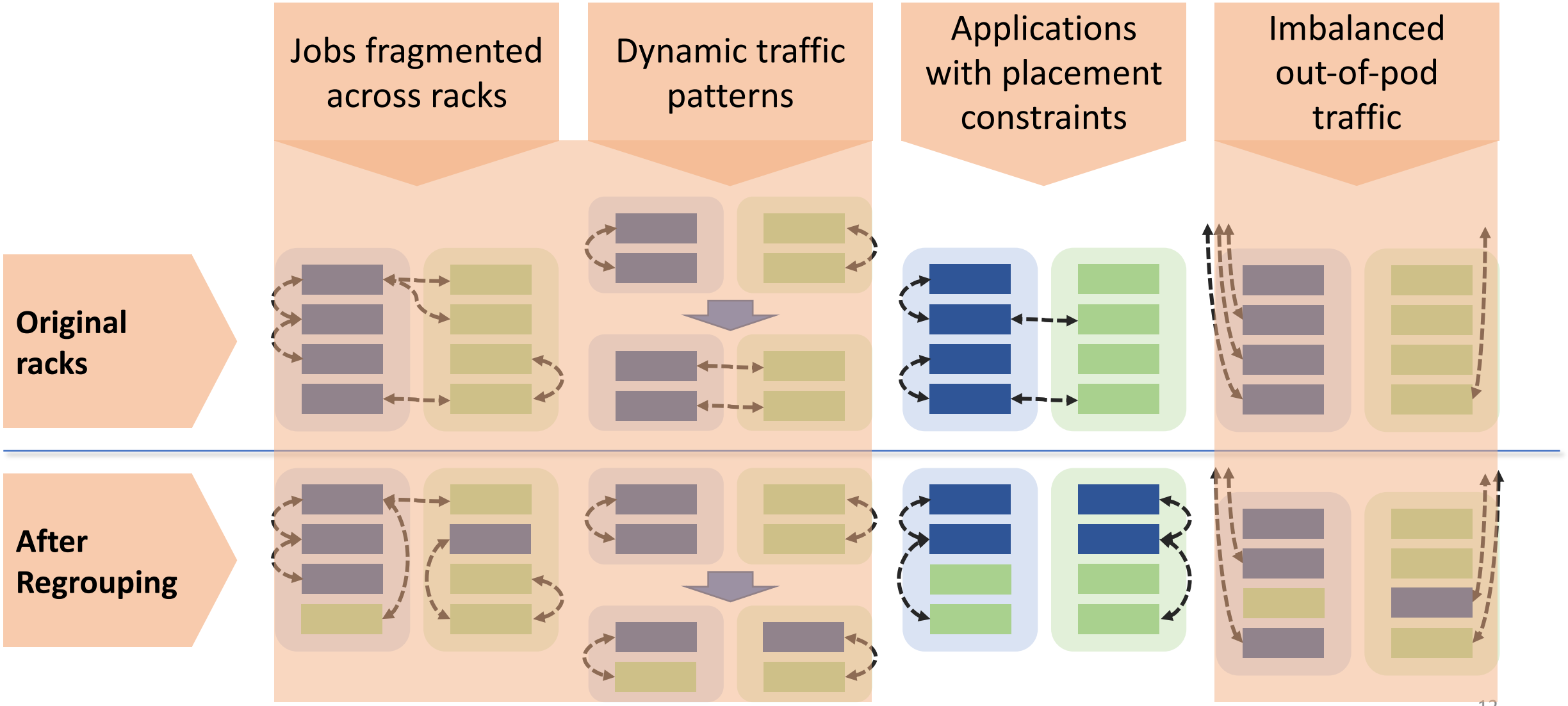
Power of Regrouping Servers



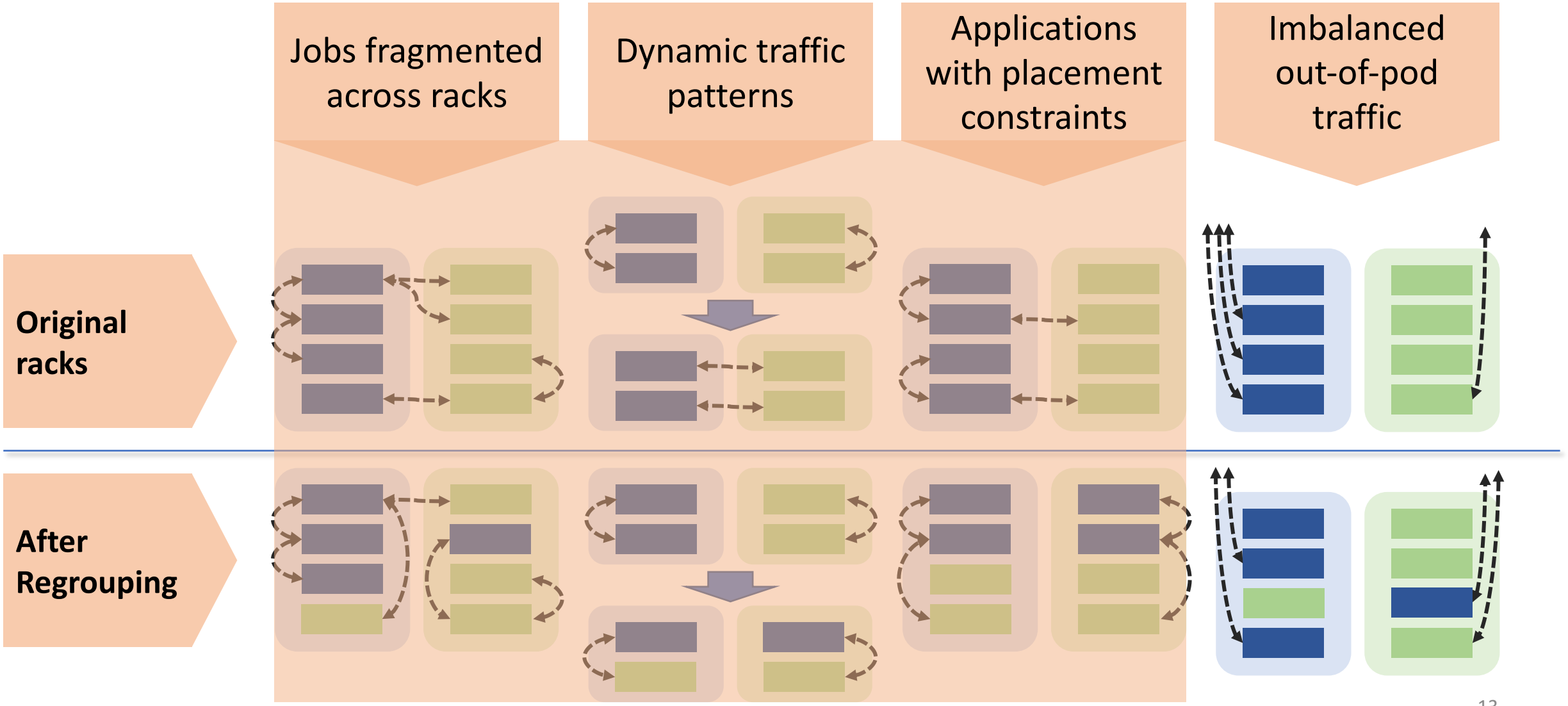
Power of Regrouping Servers



Power of Regrouping Servers



Power of Regrouping Servers



Outlines

- RDC (Rackless Data Center): regroup the servers into logical racks
  - RDC Architecture
  - RDC Control Algorithm
- Evaluation with real cluster applications on testbed
- Evaluation with packet-level simulator

Circuit Switching: Key Building Block

Circuit Switches has some unique benefits:

- High bandwidth and high port count
- Power efficient than packet switches
- **Reconfigurable**: exchange the connections between ports

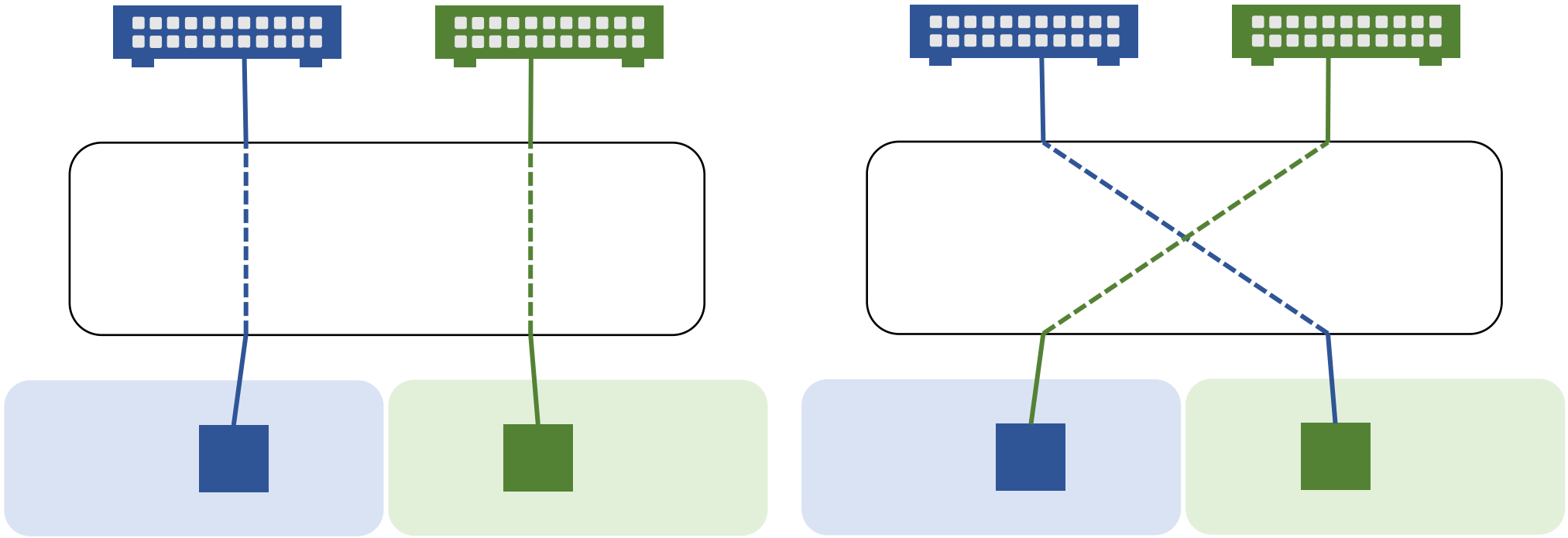
Circuit Switches are widely used to build a reconfigurable network **CORE**

- RotorNet (SIGCOMM'17)
- Sirius (SIGCOMM'20)
- …

But RDC tries to build a reconfigurable network **EDGE**

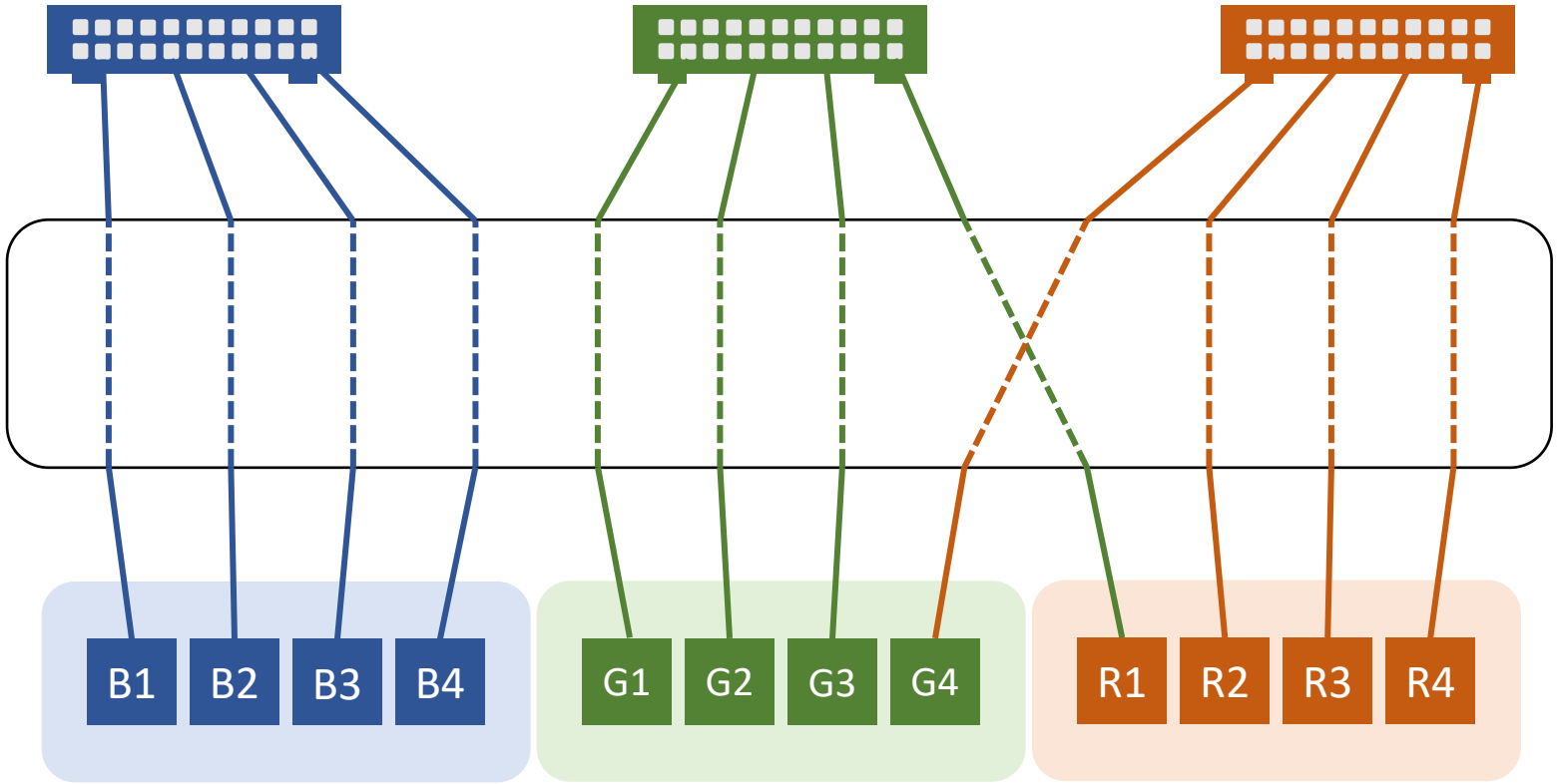
Circuit Switching: Key Building Block of RDC

If multiple ToR switches are connected to the same circuit switch:
Exchanging the connection can exchange servers' ToR switches.



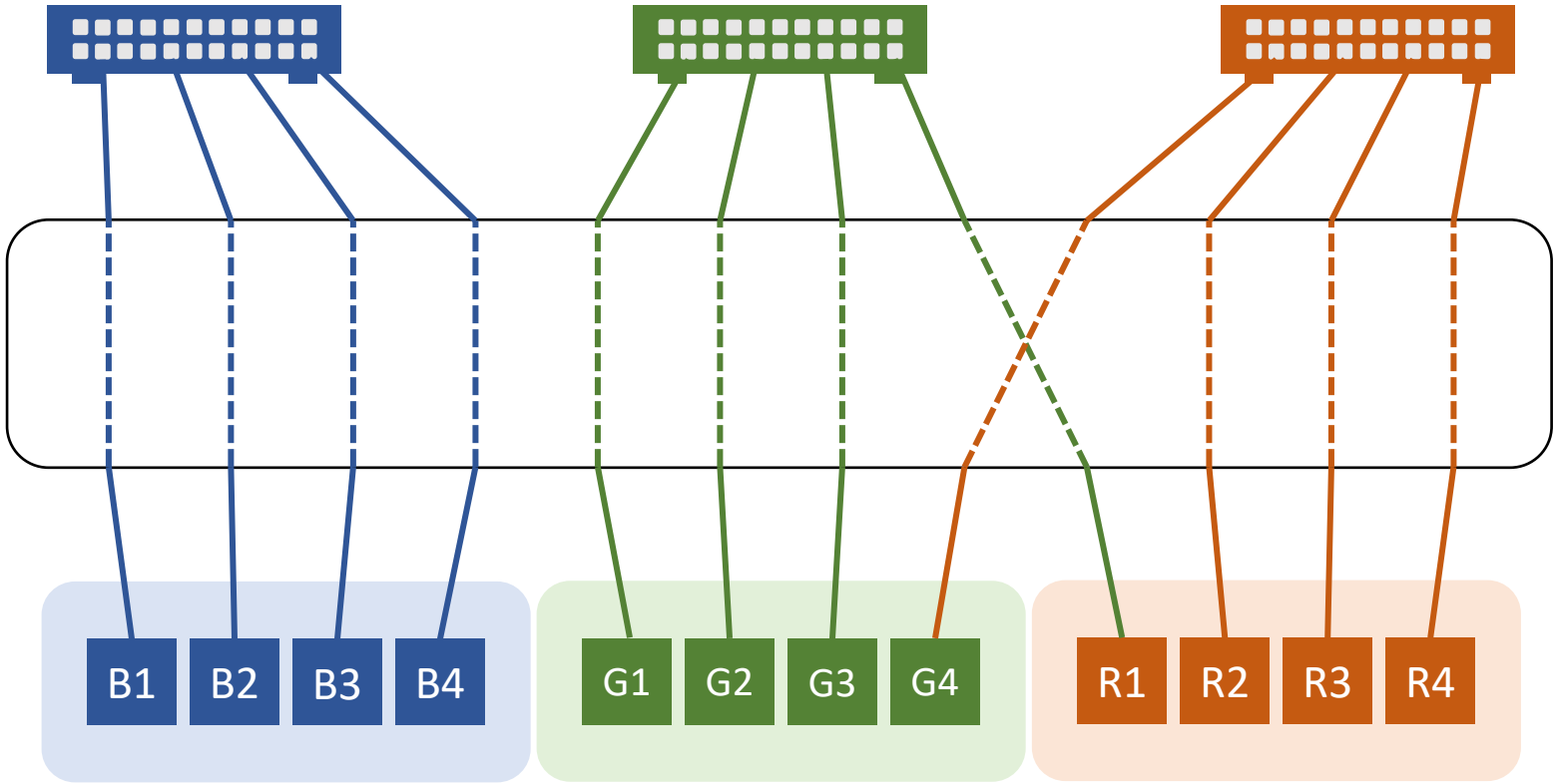
RDC Architecture

Use a circuit switch to inter-connect multiple ToR Switches and the related servers.



RDC Architecture

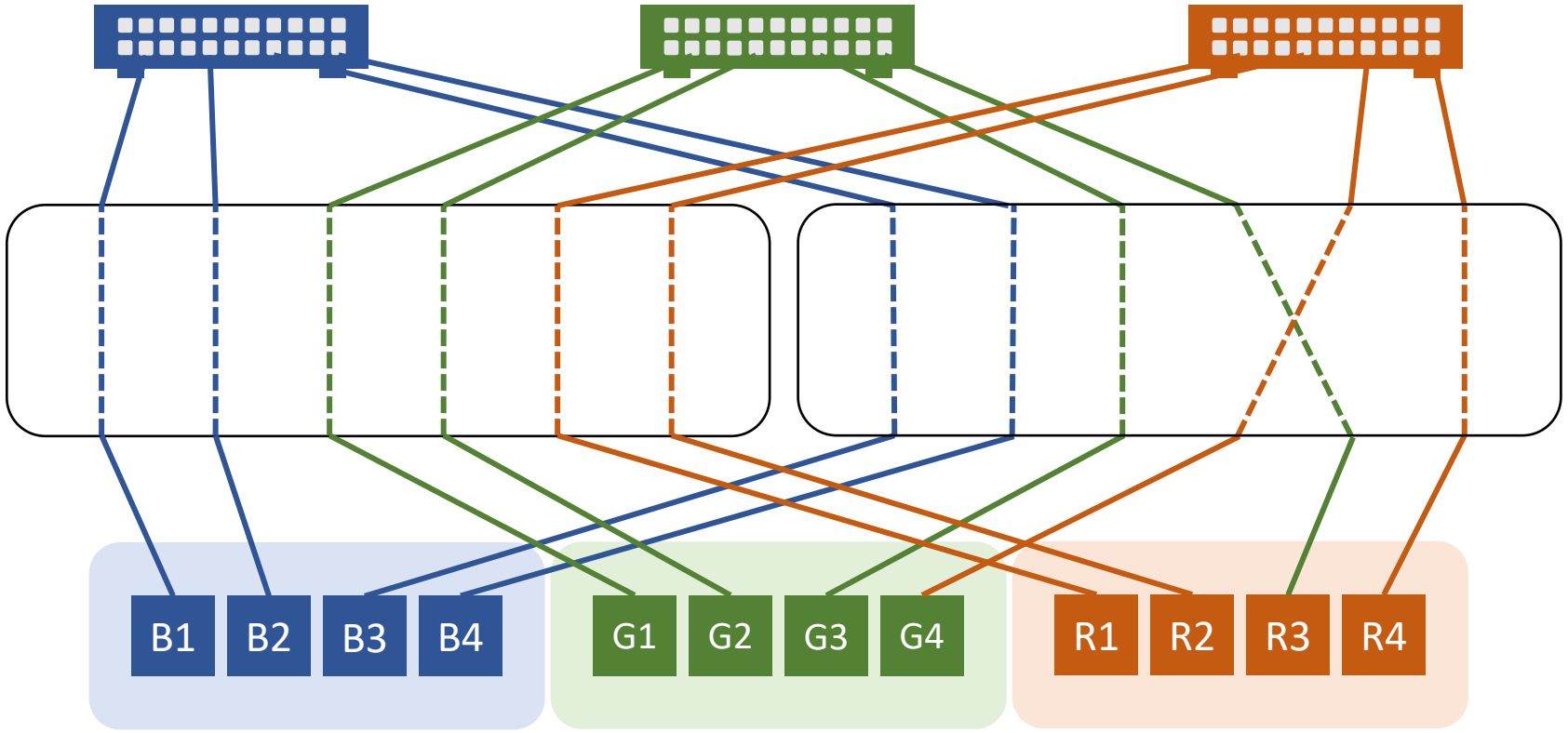
Use a circuit switch to inter-connect multiple ToR Switches and the related servers.



Cannot support many ToRs and hosts due to the port limit.

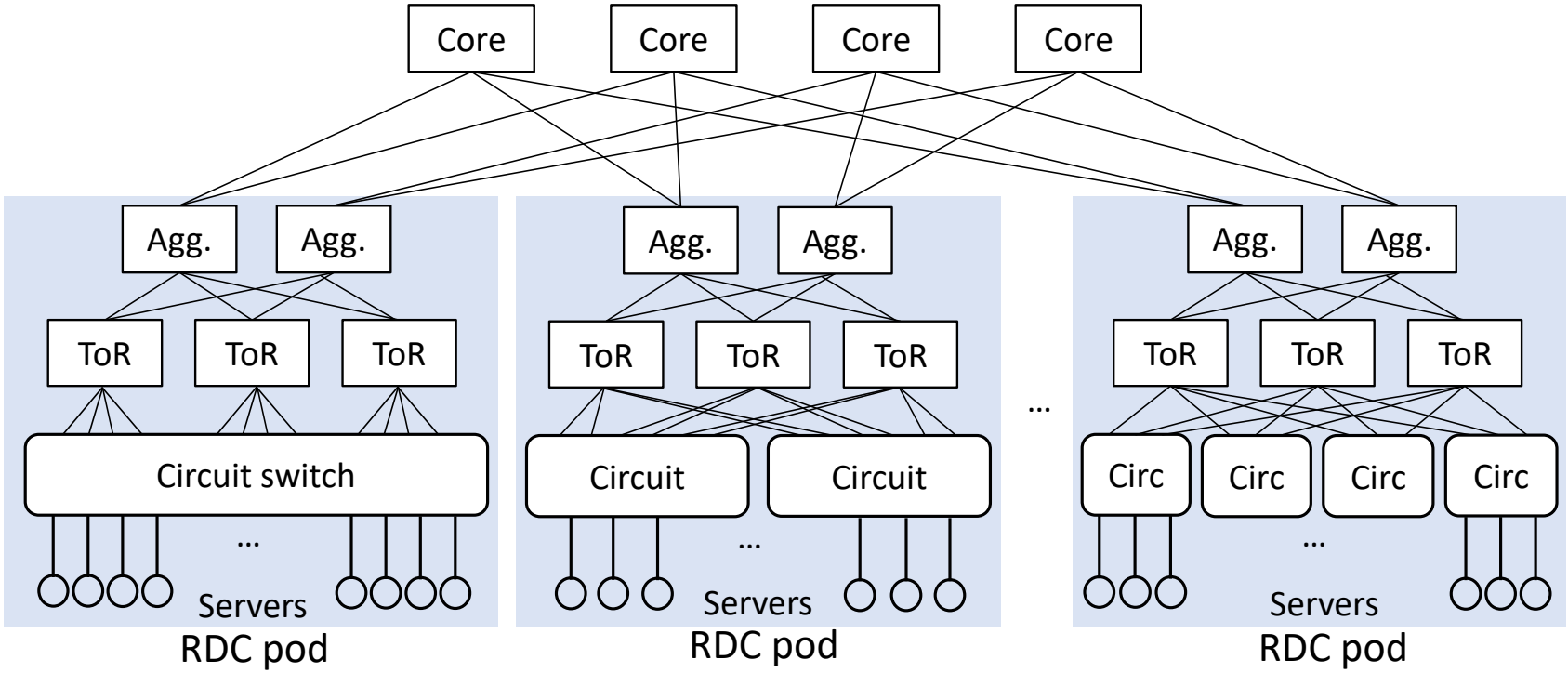
RDC Architecture

Use multiple circuit switches to **inter-connect a part of every ToR**.



RDC Architecture

For each pod, one or more circuit switches is inserted to regrouping all the servers.



RDC Controller

RDC configuration procedure:

1. Collect flow pattern (proactive-mode, reactive-mode)
2. Determine the optimized topology
3. Reconfigure the network to the optimized topology

1. Collect Flow Pattern

Proactive Mode

Reactive Mode

In a pod shared by multiple applications:

- Send the traffic demand matrix

In a pod used by one application only

- Send the new topology directly

1. Collect Flow Pattern

Proactive Mode

In a pod shared by multiple applications:

- Send the traffic demand matrix

In a pod used by one application only

- Send the new topology directly

Reactive Mode

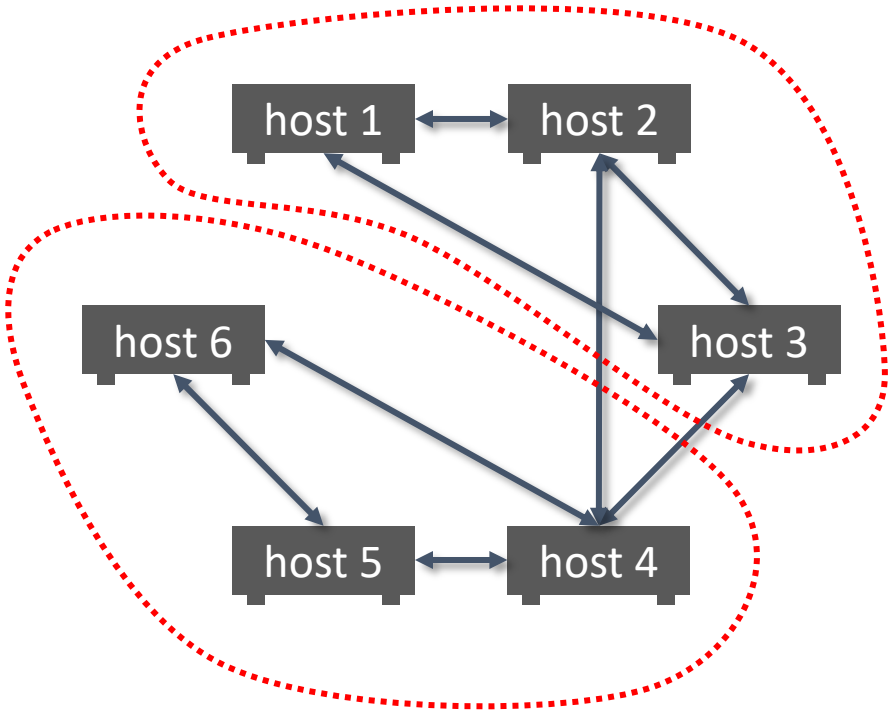
1. Monitor the flow counters on ToR
2. Estimate the demand of each flow

2. Topology Optimization (Traffic Localization)

RDC pod with 1 circuit

Multiple circuit switches

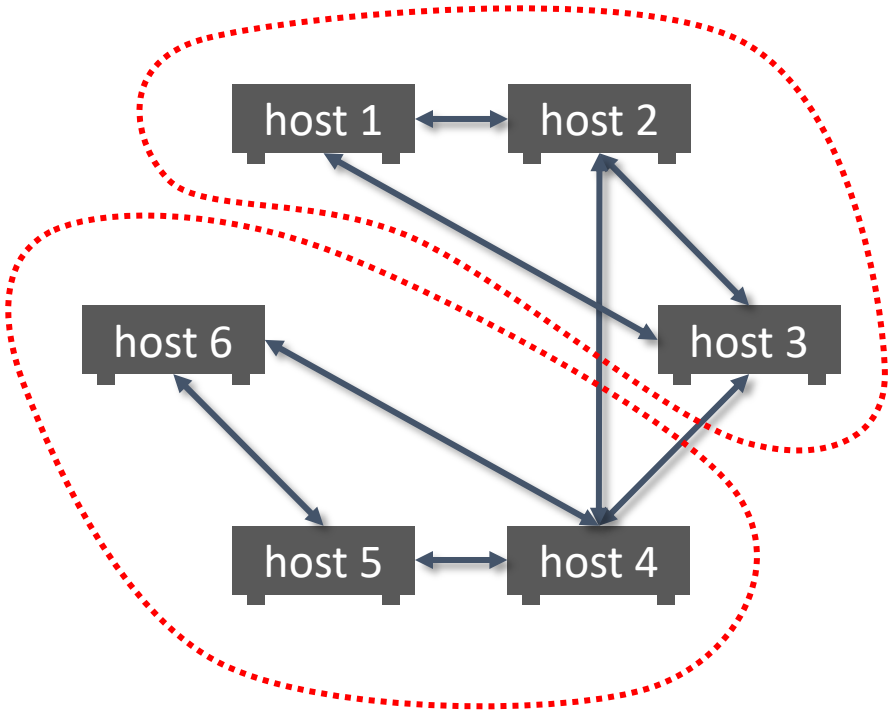
Balanced graph partition (BGP) algorithm:
Maximize edge **weights** insides subgraphs



2. Topology Optimization (Traffic Localization)

RDC pod with 1 circuit

Balanced graph partition (BGP) algorithm:
Maximize edge **weights** insides subgraphs



Multiple circuit switches

Add one extra constraints:
The number of links between a ToR and any circuit switch is fixed.

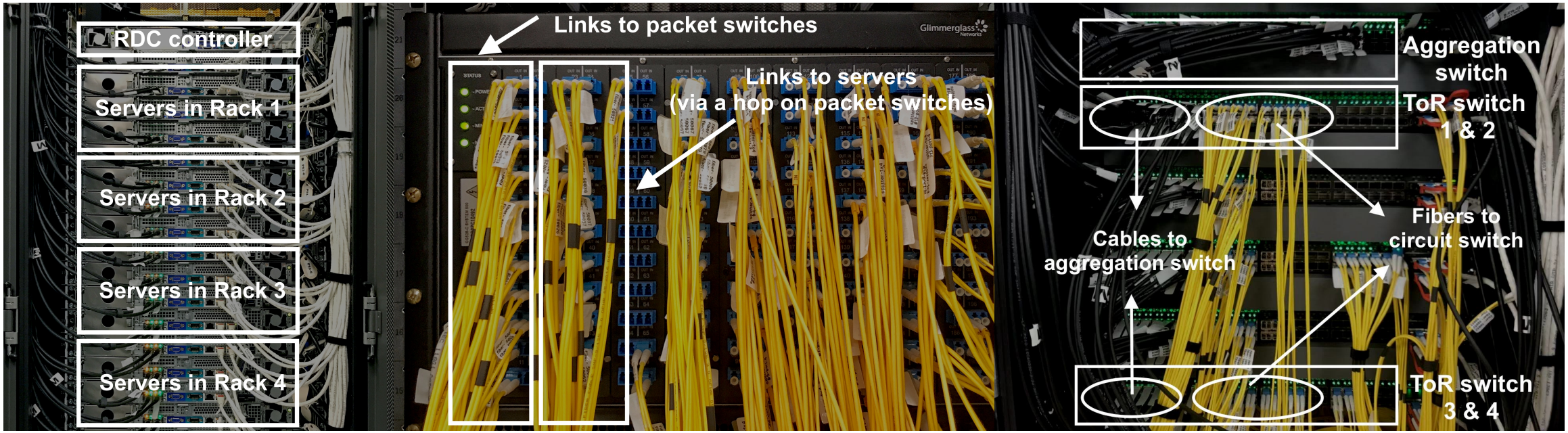
3. Topology Reconfiguration

1. **Reconfigure the circuit connection**
2. **Reconfigure the routing rules on ToR switches**

More details in the paper.

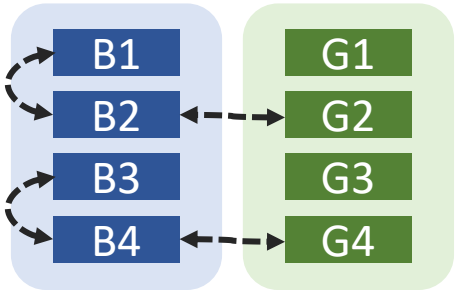
Testbed Setup

- 16 servers, 4 logical racks, 1 aggregation switch;
  - Each ToR has 4 downlinks and 1 uplink (4:1 - oversubscribed);
- 1 Circuit Switches: 192-port Glimmerglass 3D-MEMS switch

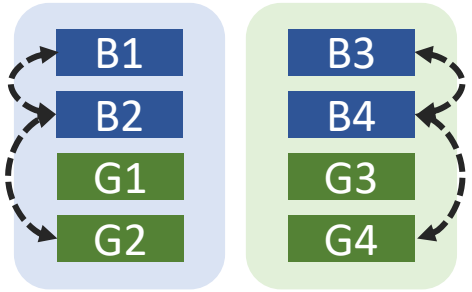


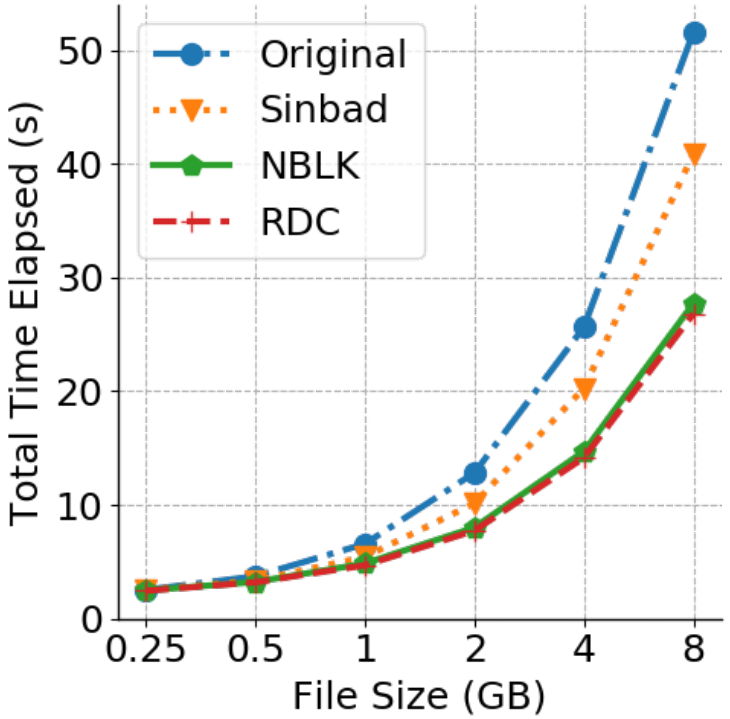
HDFS Write Need to Place the 3rd Copy onto a Different Rack

Without RDC



After reconfig





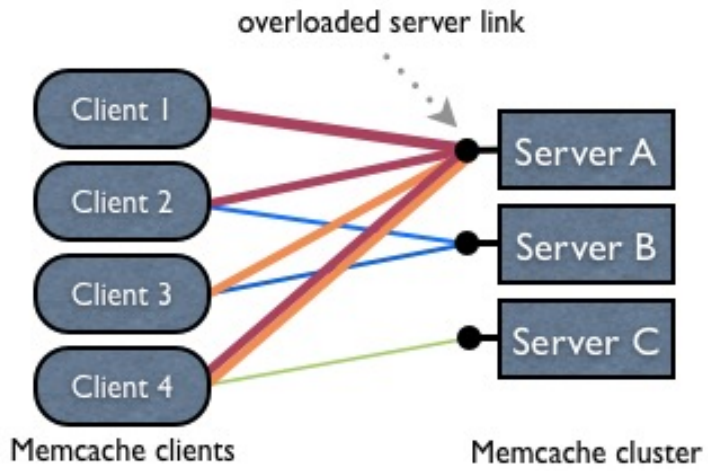
Job placement solution, like Sinbad, is hard to optimize this application.

RDC achieves similar performance with non-blocking network.

HDFS is aware of the topology, so that it can localize all the replica write without violating the policy.

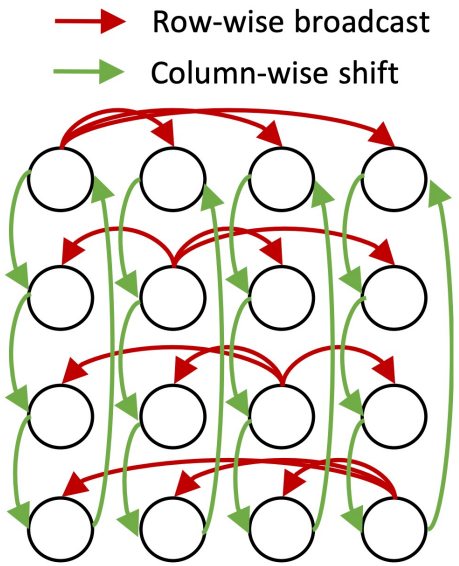
Other Applications

Memcached



Improve query throughput by **1.78x**
Reduce client latency to **0.48x**

Distributed matrix multiplication



Reduce communication time by up to **3.9x**

Simulation Setup

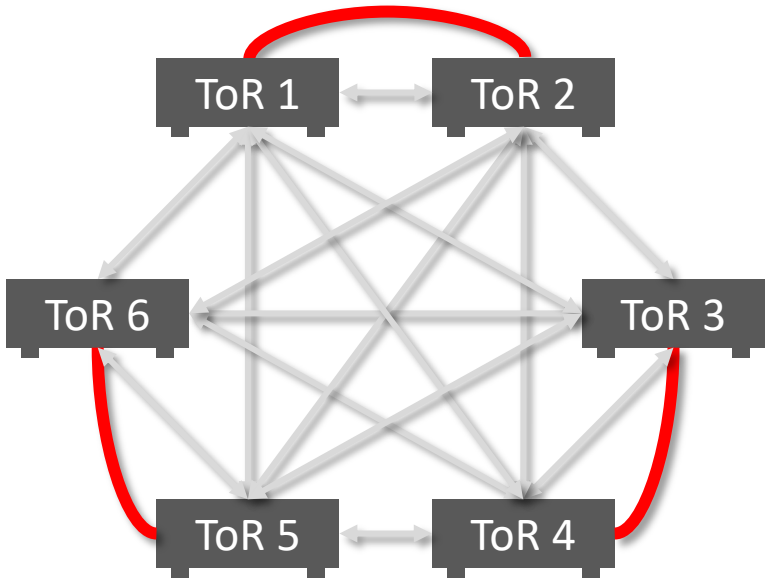
- 1 pod: 512 servers in 16 logical racks
  - Each ToR has 32 servers
  - Tunable oversubscription ratio
- Htsim: packet-level simulator with TCP and ECMP

- First set of baselines (previous work):
  - Hybrid network with reconfigurable ToR-pair links, like C-Through and Firefly
  - A novel circuit-core network, RotorNet

C-Through and RotorNet

C-Through:
Additional links between ToRs

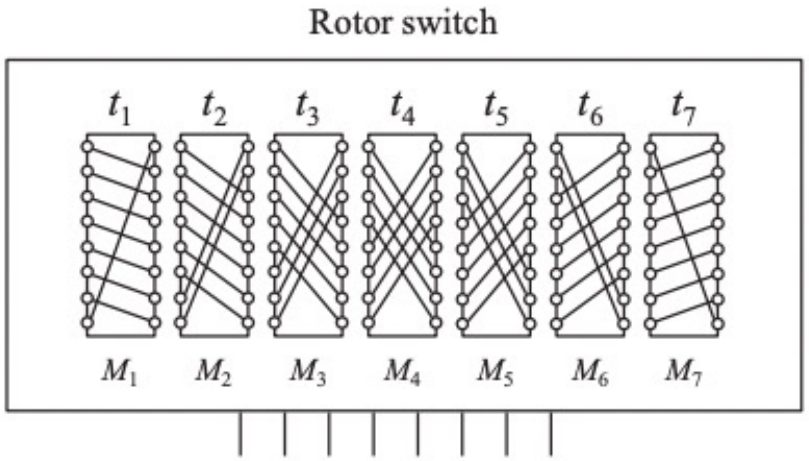
Best for serving **a small number of hot-ToR-pairs**.



1:1 C-Thr: non-blocking link
4:1 C-Thr: oversubscribed link

RotorNet:
Network core with rotating topo

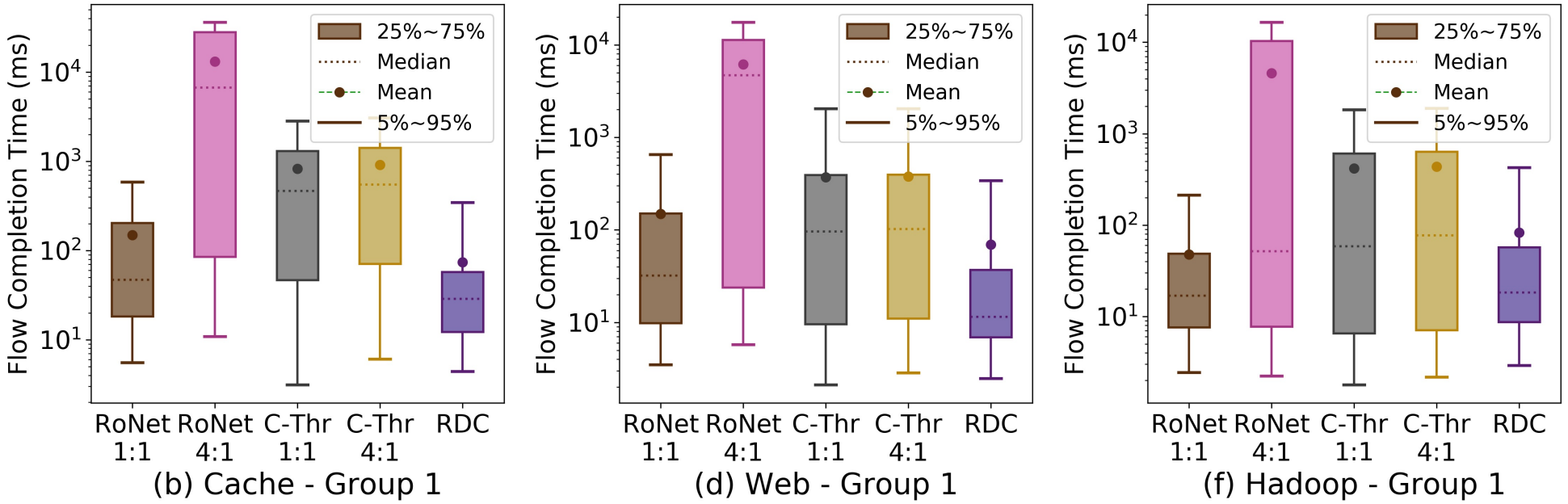
Best for serving **uniform traffic among all ToRs**.



1:1 RoNet: non-blocking core
4:1 RoNet: oversubscribed core

RDC Outperforms All Other Solutions

We test three different Facebook traces and compared the flow completion time.



RDC has shorter flow completion time in most of the cases.

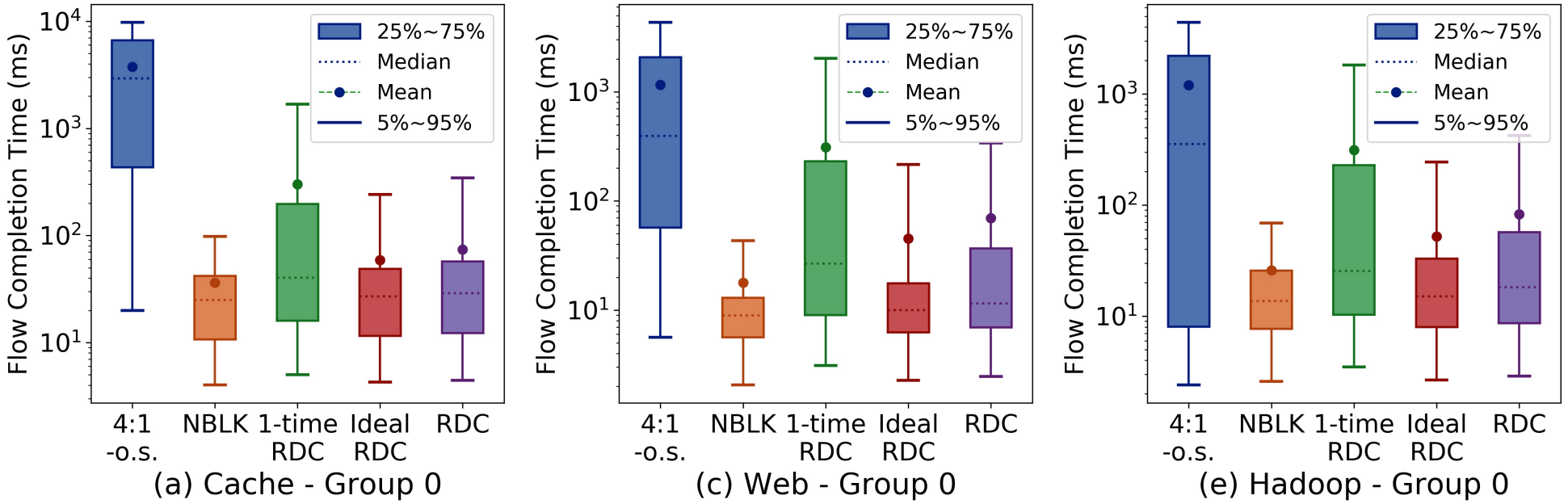
RDC benefits most from the intensive communication between a group of servers.

Simulation Setup

- Second set of baselines (NBLK and different versions of RDC):
  - Non-blocking Network (NBLK)
  - Static network with 4:1 – oversubscription (4:1-o.s.)
  - RDC with future traffic demand information (ideal RDC)
  - **Apply RDC's algorithm only for one time (1-time RDC)**

RDC Achieves Similar Median Completion Time With NBLK

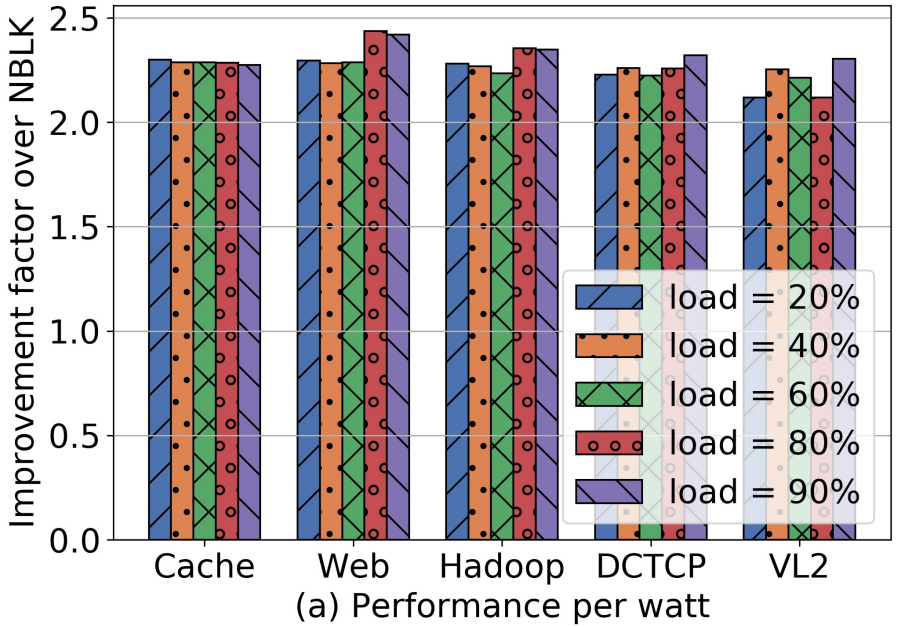
We test three different Facebook traces and compared the flow completion time.



RDC > 1-time RDC, because reconfiguration frequency benefits **changing traffic patterns**.
RDC achieve similar median FCT as NBLK, but tail is longer due to configuration.

Energy Consumption Analysis

We varied the traffic load and show the relative performance per watt results for RDC.



RDC (4:1-o.s.) has more than 2x improvements in **performance per watt** than NBLK.

Summary

The rackless data center (RDC) is a novel network architecture

- Logically removes the rack boundaries;
- Substantial performance benefits for real-world applications;
- More energy efficient than NBLK network.

Summary

The rackless data center (RDC) is a novel network architecture

- Logically removes the rack boundaries;
- Substantial performance benefits for real-world applications;
- More energy efficient than NBLK network.

Thank you!

(Weitao Wang: wtwang@rice.edu)