Minimal Rewiring: Efficient Live Expansion for Clos Data Center Networks

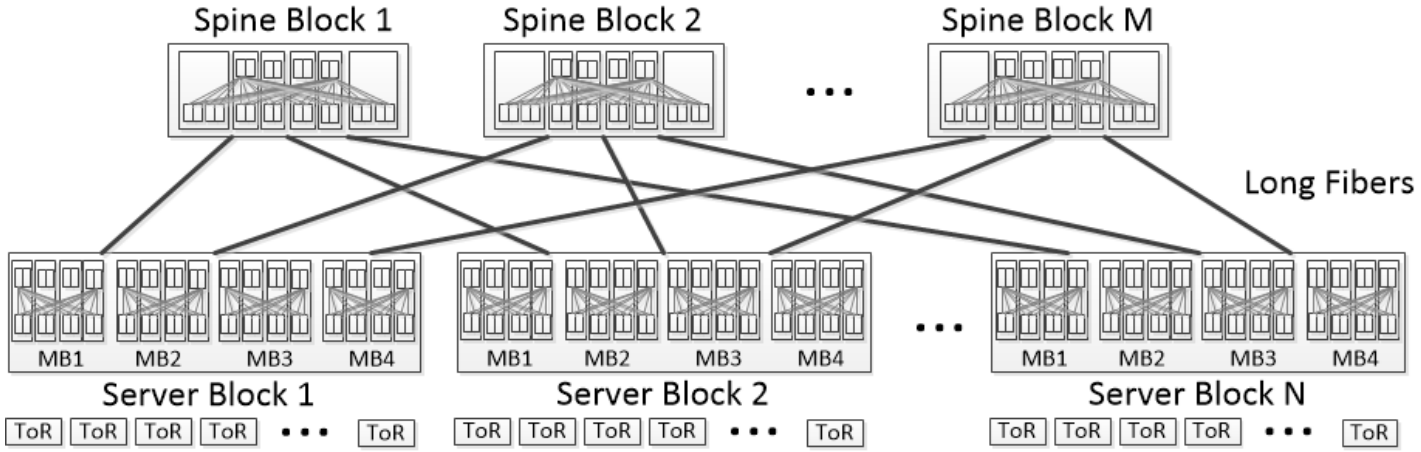
Shizhen Zhao[1, 2],

Rui Wang[1], Junlan Zhou[1], Joon Ong[1],Jeffrey C. Mogul[1], Amin Vahdat[1]

1. Google NetInfra; 2. Shanghai Jiao Tong University

Clos Topology for Commercial Data Centers

- Built with Commodity Switches
- High Path Diversity & Non-Blocking
- Simple Routing
- ***Widely Deployed*** in Commercial Data Centers!



Importance of Fine-grained Expansion

- What is Fine-grained Expansion?
  - Expand data center at server block granularity.
- Why Fine-grained Expansion is important?
  - Bandwidth requirement doubles every 15-18 months
  - Why not coarse-grained expansion?
    - Deploying large amount of capacity at once incurs significant opportunity cost, e.g., large idle capacity, technical refresh
- Unfortunately, ***Fine-grained Expansion is Very Difficult for Clos***

Challenges of Fine-Grained Expansion for Clos

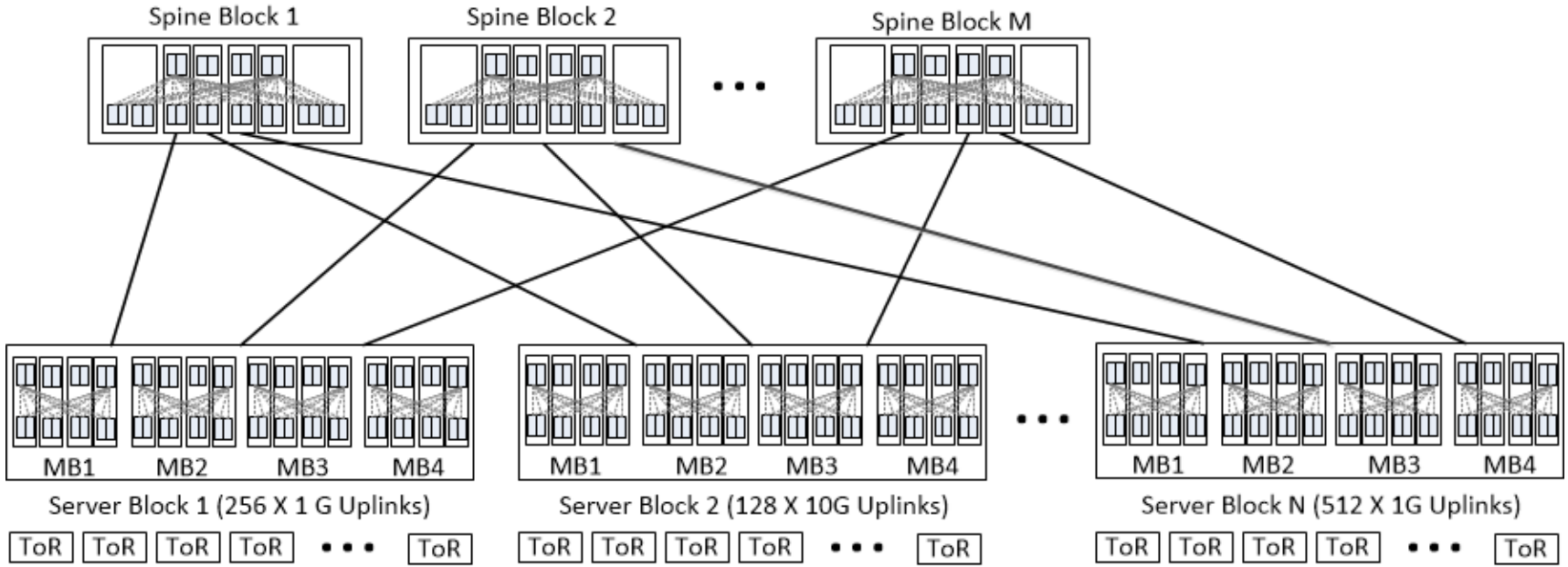
- Moving fibers around is labor intensive and error prone
  - Spine blocks and server blocks may not be co-located
  - Required fiber length may change in order to rewire
- Classical Clos was not designed for fine-grained expansion
  - E.g., FatTree [1] (limited sizes of 3456, 8192, 27648, 65536 corresponding to the commonly available port counts of 24, 32, 48, 64)
  - E.g., Rotation Striping for Clos [2] (need to rewire almostly all the links)
- Need live expansion
  - Cannot take the entire DCN offline to do an expansion
  - Data centers must be highly available. No packet loss is allowed

[1] M. Al-Fares *et. al.*, “A scalable, commodity data center network architecture,” ACM SIGCOMM 2008.
[2] J. Zhou *et. al.*, “WCMP: Weighted cost multipathing for improved fairness in data centers,” EuroSys 2014.

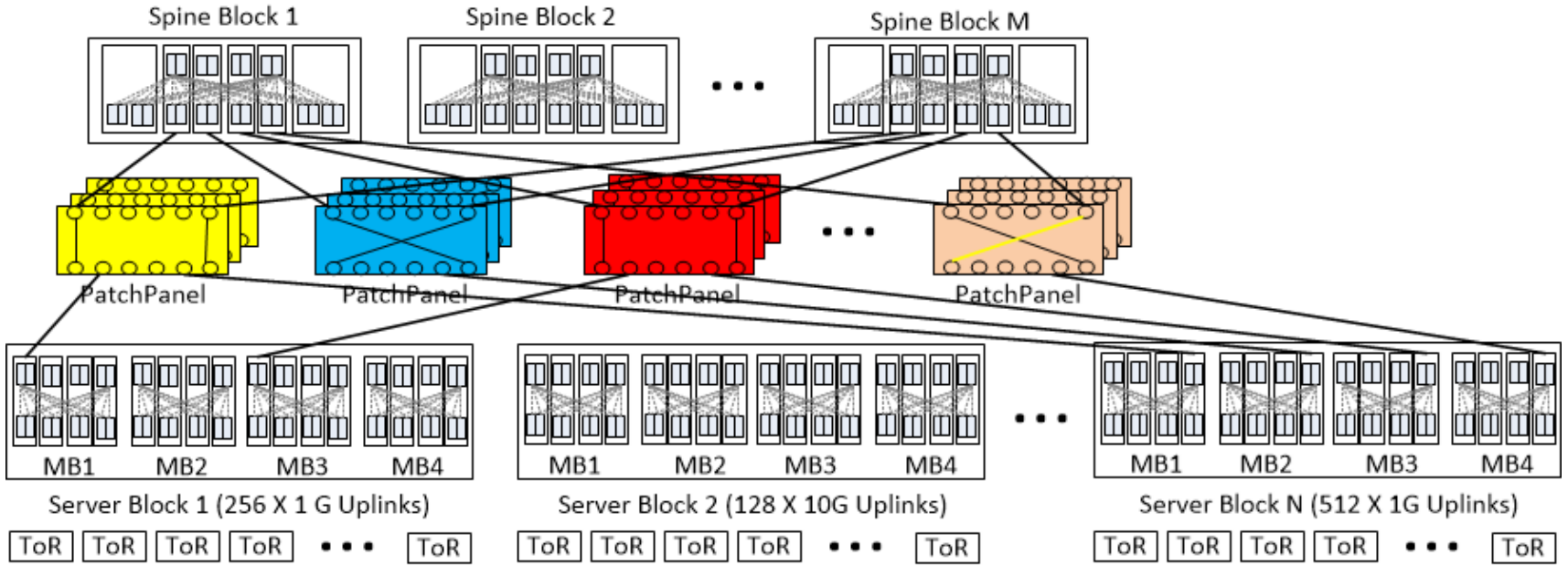
Our contribution

- Architecture Aspect:
  - A layer of patch panels to better handle fiber movements
  - A multi-stage pipeline for hitless live expansion
- Topology Design Algorithm Aspect:
  - Minimal Rewiring solver for fine-grained expansion of Clos
    - Reduces average number of expansion stages by 3.1X

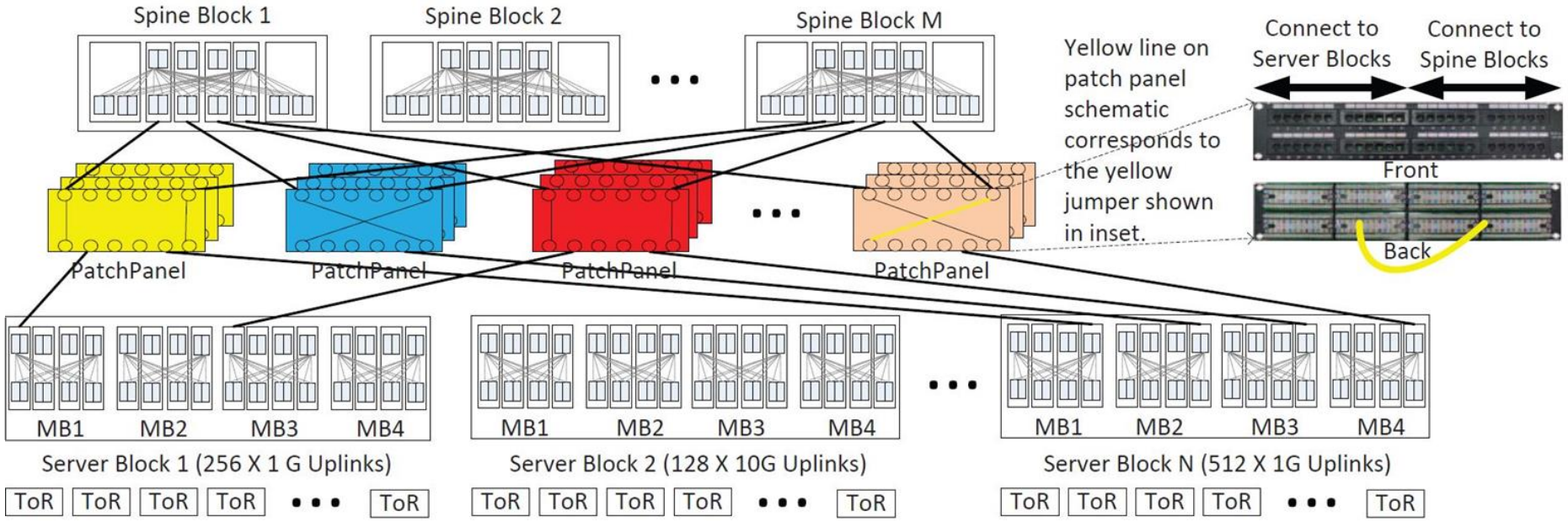
Physical Architecture



Physical Architecture

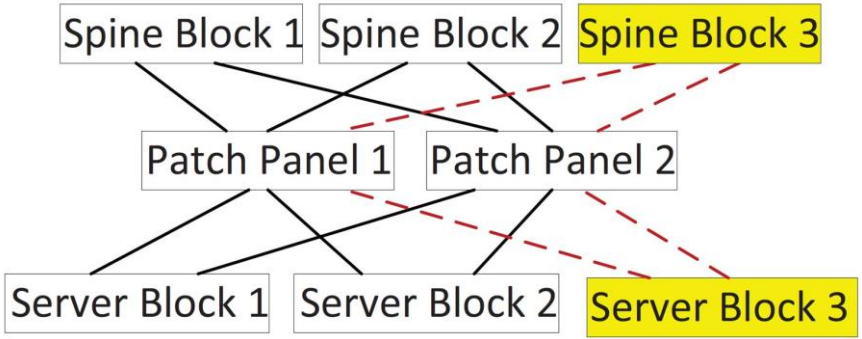


Physical Architecture



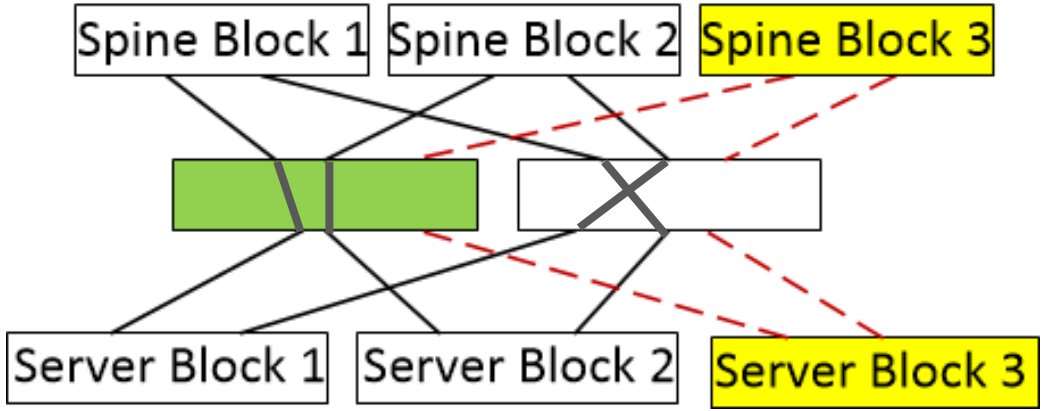
Patch-Panel based Expansion

1. Connect new server blocks and new spine blocks to the patch panel layer
2. Compute a new topology (To be discussed later)
3. Change topology in stages, to ensure sufficient capacity



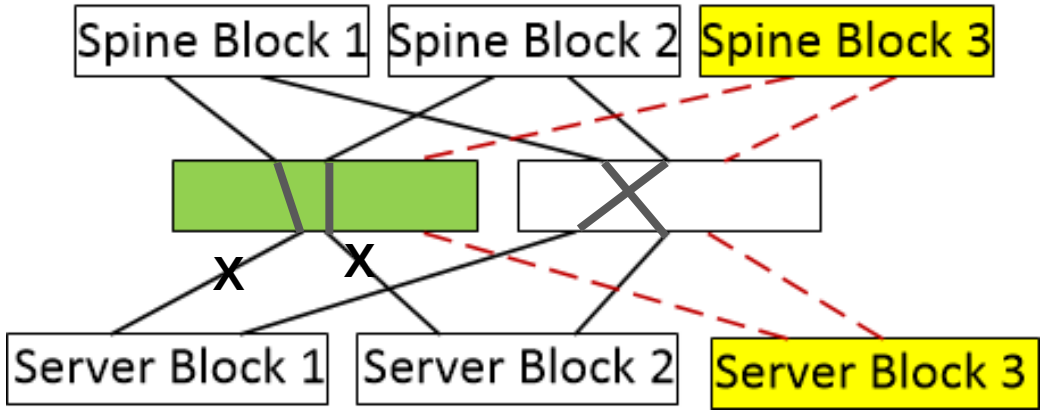
Each Stage Requires Careful Sequencing

1. Route traffic around the patch panels to be touched
2. Rewire patch panels (***labor intensive, takes a few hours***)
3. Test topology correctness and link quality
4. Update topology configuration in SDN controllers
5. Enable routing through these patch panels again



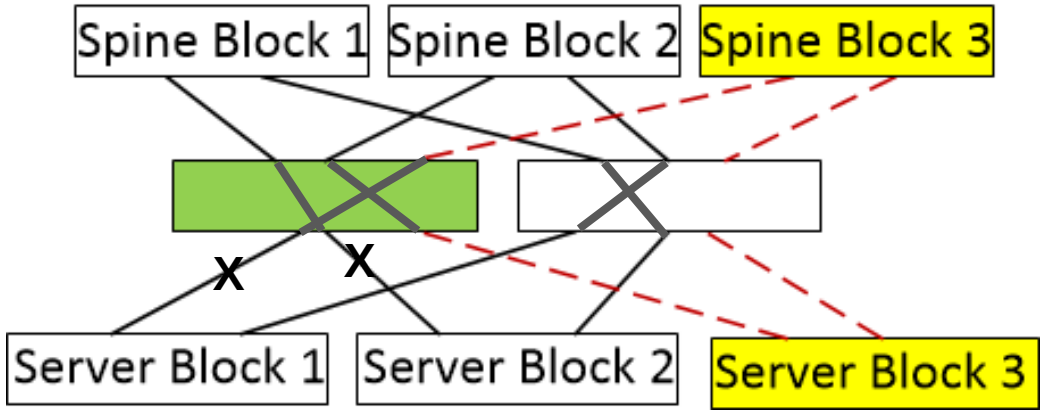
Each Stage Requires Careful Sequencing

1. **Route traffic around the patch panels to be touched**
2. Rewire patch panels (***labor intensive, takes a few hours***)
3. Test topology correctness and link quality
4. Update topology configuration in SDN controllers
5. Enable routing through these patch panels again



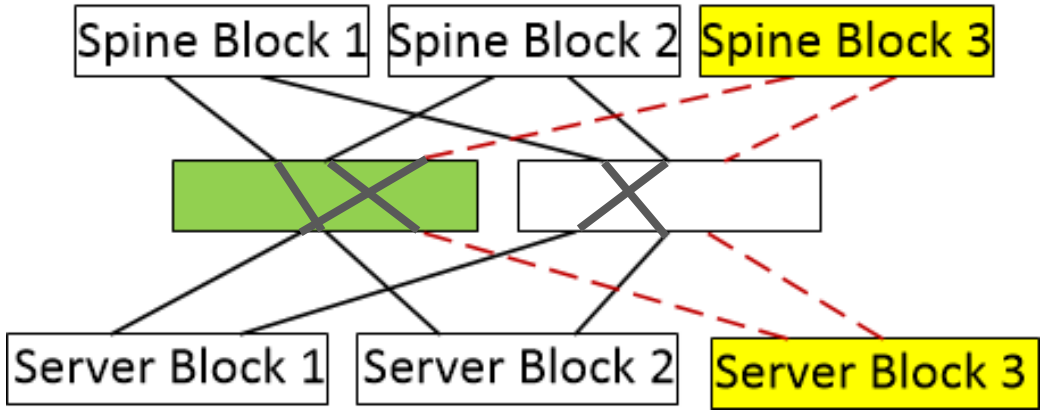
Each Stage Requires Careful Sequencing

1. Route traffic around the patch panels to be touched
2. **Rewire patch panels** (***labor intensive, takes a few hours***)
3. **Test topology correctness and link quality**
4. Update topology configuration in SDN controllers
5. Enable routing through these patch panels again



Each Stage Requires Careful Sequencing

1. Route traffic around the patch panels to be touched
2. Rewire patch panels (***labor intensive, takes a few hours***)
3. Test topology correctness and link quality
4. **Update topology configuration in SDN controllers**
5. **Enable routing through these patch panels again**



Current Expansion Takes Too Much Time!

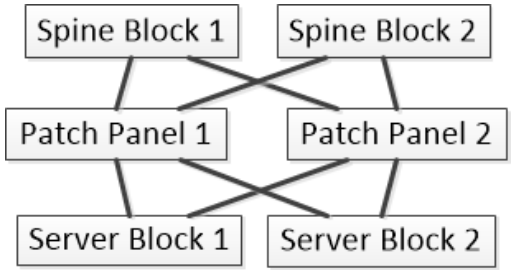
- Each Stage takes considerable amount of time
  - Manual rewiring could take a few hours
- Need multiple stages to guarantee sufficient residual capacity
  - The higher the traffic, the larger the number of stages
- Previous topology solver rewires almost all the links
  - If max link utilization is 90%, then at least 10 stages are required
- Our solution:
  - ***Minimizes number of rewires during expansion.***

How to Minimize Rewires During Expansion?

- Naive Solution:
  - Break a few links and let new blocks in.
  - Problem: Highly imbalanced new topology, leading to poor performance
- Optimization-based Approaches: e.g., LEGUP [3], REWIRE [4]
  - Did not consider patch panels
  - High computational complexity!
    - Branch and Bound / Simulated Annealing have poor convergence!

[3] A. R. Curtis, *et. al.*, “LEGUP: Using Heterogeneity to Reduce the Cost of Data Center Network Upgrades,” in ACM CoNEXT 2010.
[4] A. R. Curtis, *et. al.*, “REWIRE: An optimization-based framework for unstructured data center network design,” in Infocom 2012.

Minimal Rewiring: an ILP-based Solver



- Output: DCN Topology d^k_{mn}
  - m: server block, n: spine block, k: patch panel
- Objective: Minimize # of links drained

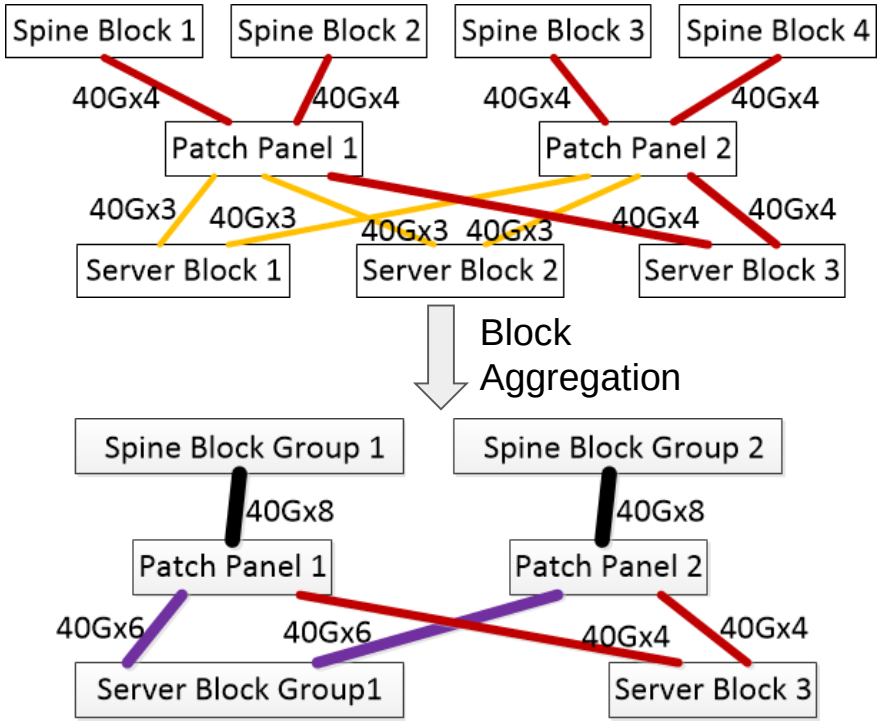
  $\sum_{m,n,k} \max\{b^k_{mn} - d^k_{mn}, 0\}$ where b^k_{mn} Is the existing DCN topology.
- Constraints:
  - In each patch panel, the total number of logical links from a server block cannot exceed the total number physical links from this server block. Same for Spine blocks: $\sum_n b^k_{mn} \le L^k_m, \sum_m b^k_{mn} \le R^k_n,$
  - (*Topology Balance Constraints*) Links from a server block should be evenly distributed among spine blocks: $p \le \sum_k b^k_{mn} \le p+1$

Complexity Challenge of Minimal Rewiring

- Scale of Minimal Rewiring in Our Data Centers:
  - # of server blocks O(100) X # of spine blocks O(100) X # of patch panels O(100)
- Consequence:
  - ~70% of 4500 benchmark cases cannot be solved!

Block Aggregation to Reduce Complexity

- Motivation
  - Homogeneous components exists.
- Problem Size Reduction
  - # of server block groups (1~10) X
    # of spine block groups (1~10) X
    # of patch panel groups (1~10)
- Guaranteed Decomposable
  - ILP Approach
  - Min-Cost-Flow Approach
    (polynomial but less optimal)



Experiment Setup

- 2250 Base Configurations:
  - 256 Patch Panels
  - A mix of server blocks with 256/512/1024 uplinks
  - A mix of spine blocks with 128/512 downlinks
  - Up to 80 server blocks
- 4500 Expansion Cases:
  - Add one server block with 256 uplinks
  - Upgrade two server blocks from 256 uplinks to 512 uplinks

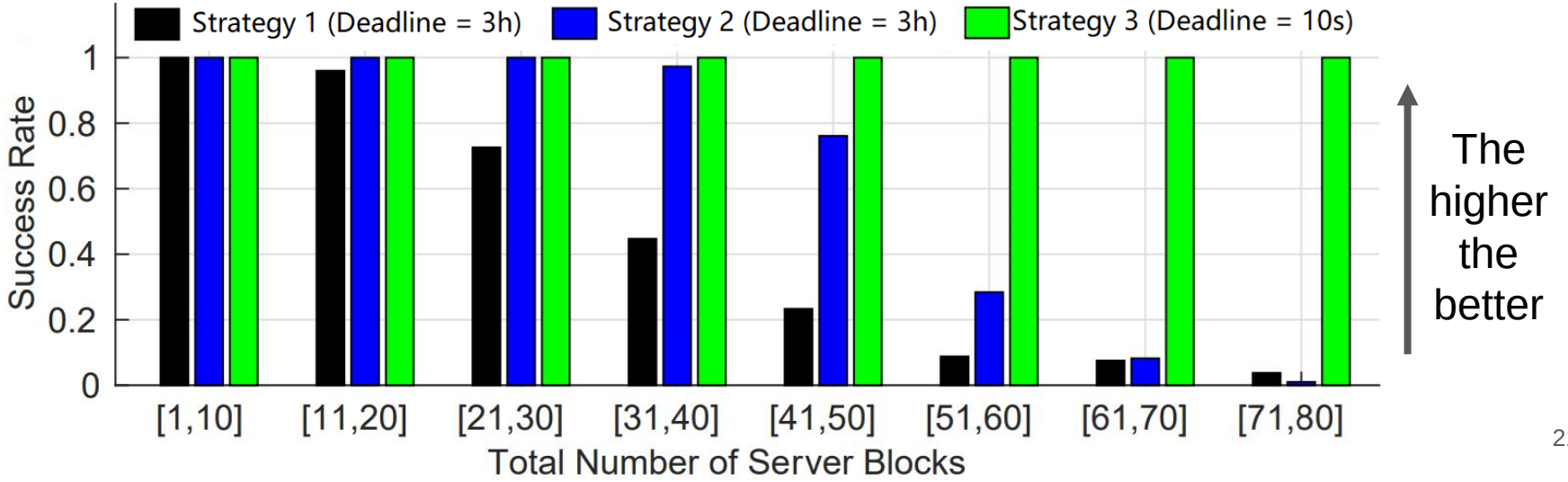
Real data centers have mixed block sizes!

Metrics

- Success Rate, within A Deadline:
  - Directly determines if our algorithm is usable or not in production.
- Rewiring Ratio:
  - The performance of minimal rewiring solver
  - An indirect measure on the speedup of data center expansion.
- Number of Expansion Stages:
  - A direct measure on the speedup of data center expansion.

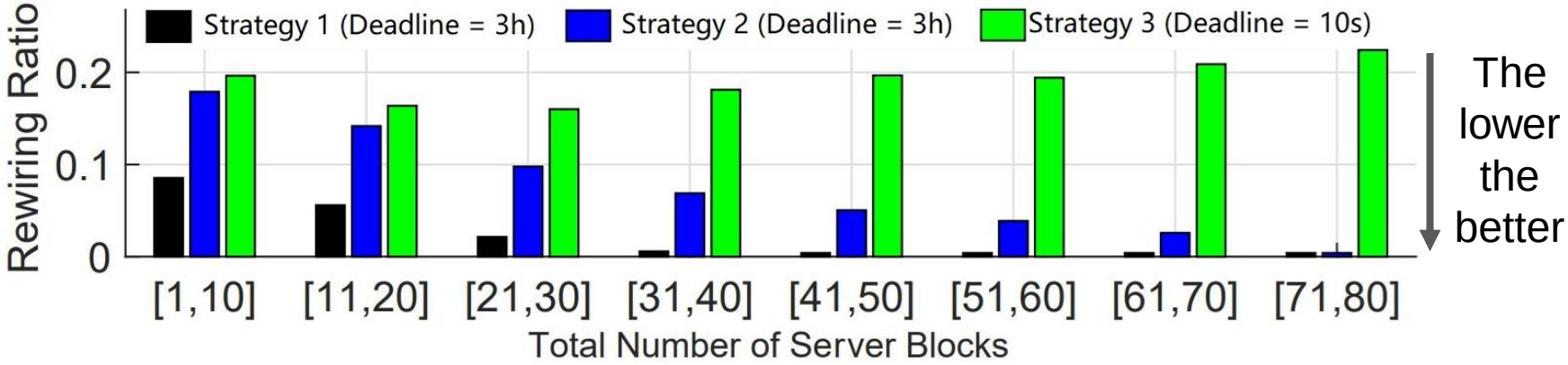
Success Rate within Certain Deadline

- Strategy 1: No Aggregation
- Strategy 2: Aggregate server blocks/spine blocks/patch panels. Decompose server blocks/spine blocks using ILP, and patch panels using min-cost-flow
- Strategy 3: Aggregate server blocks/spine blocks/patch panels. Decompose server blocks/spine blocks/patch panels using min-cost-flow



Rewiring Ratio

- Strategy 1: No Aggregation
- Strategy 2: Aggregate server blocks/spine blocks/patch panels. Decompose server blocks/spine blocks using ILP, and patch panels using min-cost-flow
- Strategy 3: Aggregate server blocks/spine blocks/patch panels. Decompose server blocks/spine blocks/patch panels using min-cost-flow

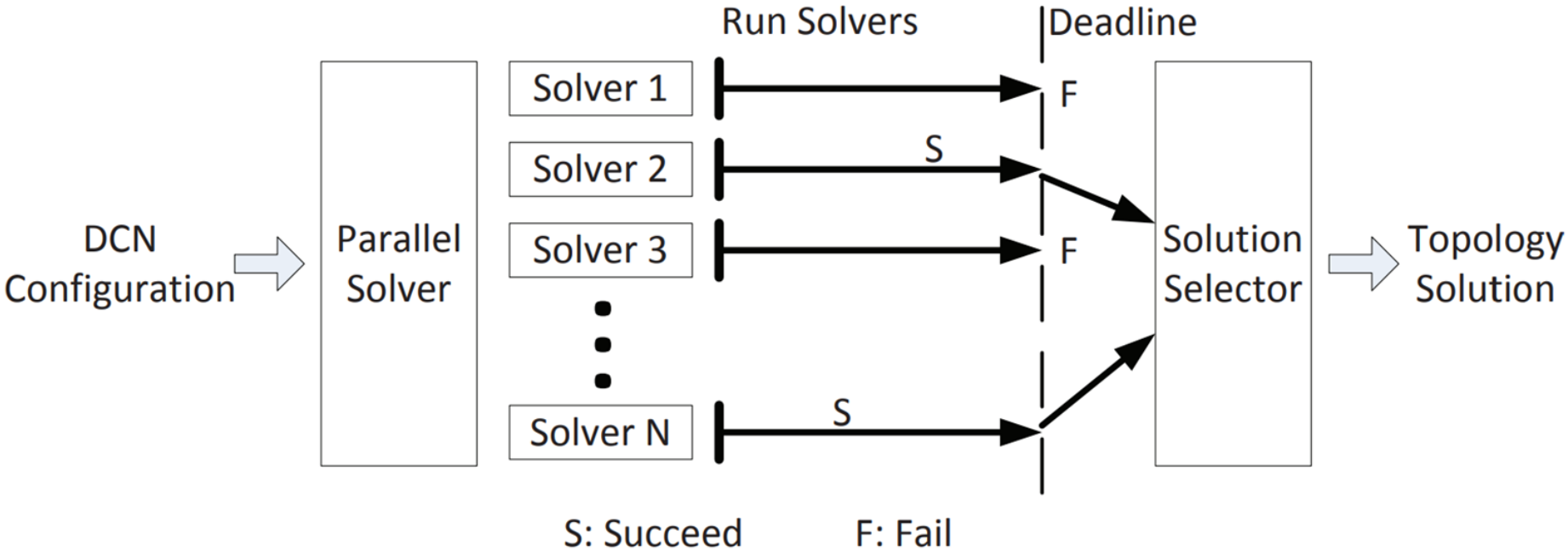


Take Away Message

- Block aggregation can significantly reduce algorithmic complexity!
- Block aggregation may incur suboptimality in terms of rewiring
- Decomposing using ILP is also expensive.
- Min-cost-flow decomposition algorithm incurs additional optimality loss

There is a tradeoff!

Parallel Solver



Savings of Expansion Stages

- Prior to Minimal Rewiring
  - Assuming expansion takes 4 stages, ~70% bisection bandwidth can be maintained
- With Minimal Rewiring
  - On average 1.29 stages are required to ensure 70% bisection bandwidth

Number of expansion stages:	1	2	4
Aggregation Strategy (1)	1598	34	2868
Aggregation Strategy (2)	2668	416	1416
Aggregation Strategy (3)	1582	2914	4
Parallel Solver	3176	1324	0

Cells show # of test cases that need given # of stages.

Conclusion

- We demonstrated the importance of using Patch Panels in data centers, which has been generally overlooked in the literature
- We proposed, implemented and tested Minimal Rewiring:
  - Deals with patch panel constraints
  - Scales to large scale data centers with algorithmic optimization
- We reduced the average number of stages required for expansion from 4 to 1.29, which ***reduces expansion time and labor cost by 3.1X*** on average. Note that data center is more vulnerable to congestion during expansion.