Brief Announcement: Deterministic Lower Bound for Dynamic Balanced Graph Partitioning

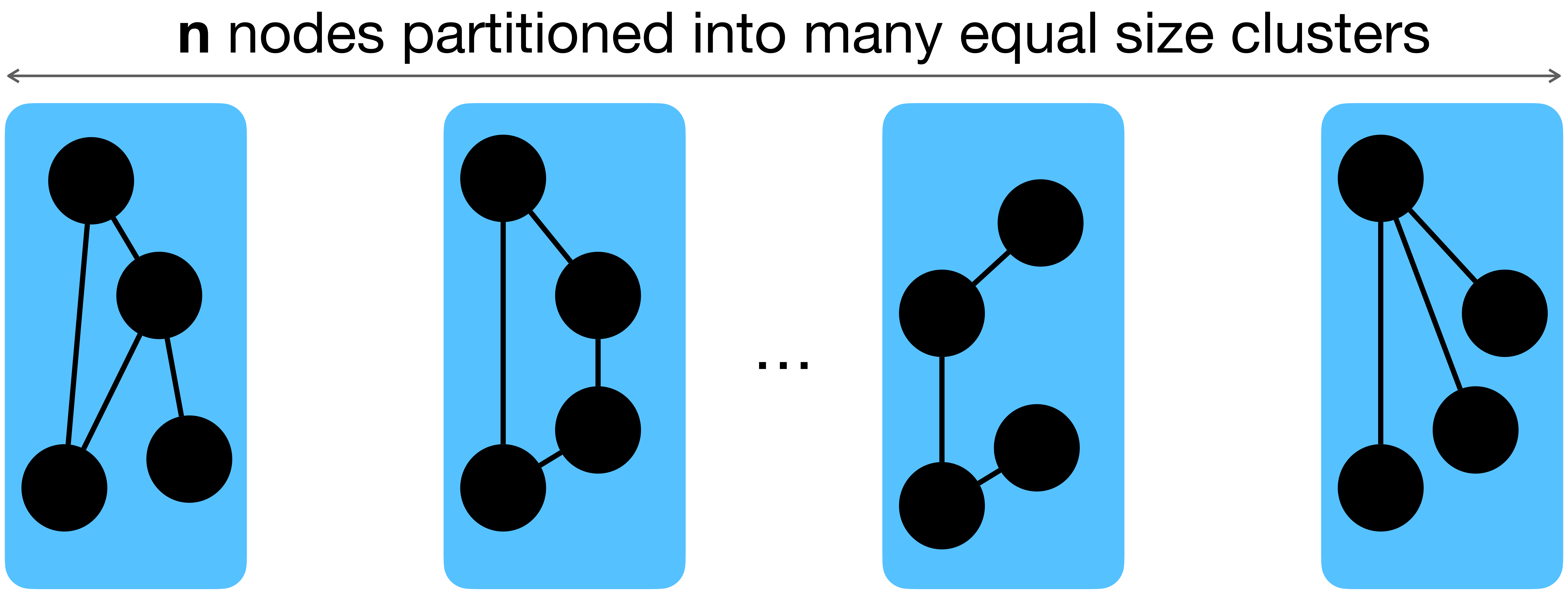
Maciej Pacut
maciej.pacut@univie.ac.at
Faculty of Computer Science,
University of Vienna
Austria

Mahmoud Parham
mahmoud.parham@univie.ac.at
Faculty of Computer Science,
University of Vienna
Austria

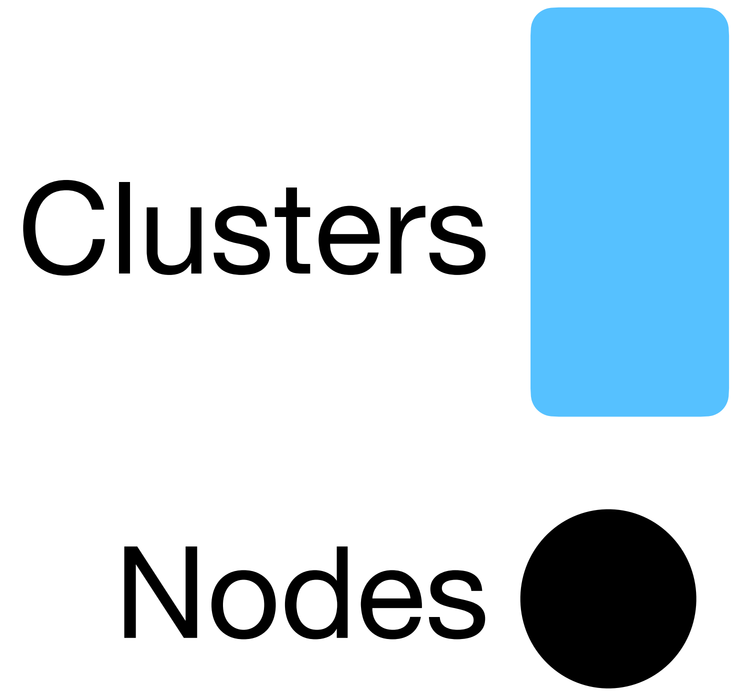
Stefan Schmid
stefan_schmid@univie.ac.at
Faculty of Computer Science,
University of Vienna
Austria

PODC 2020, Brief Announcements.

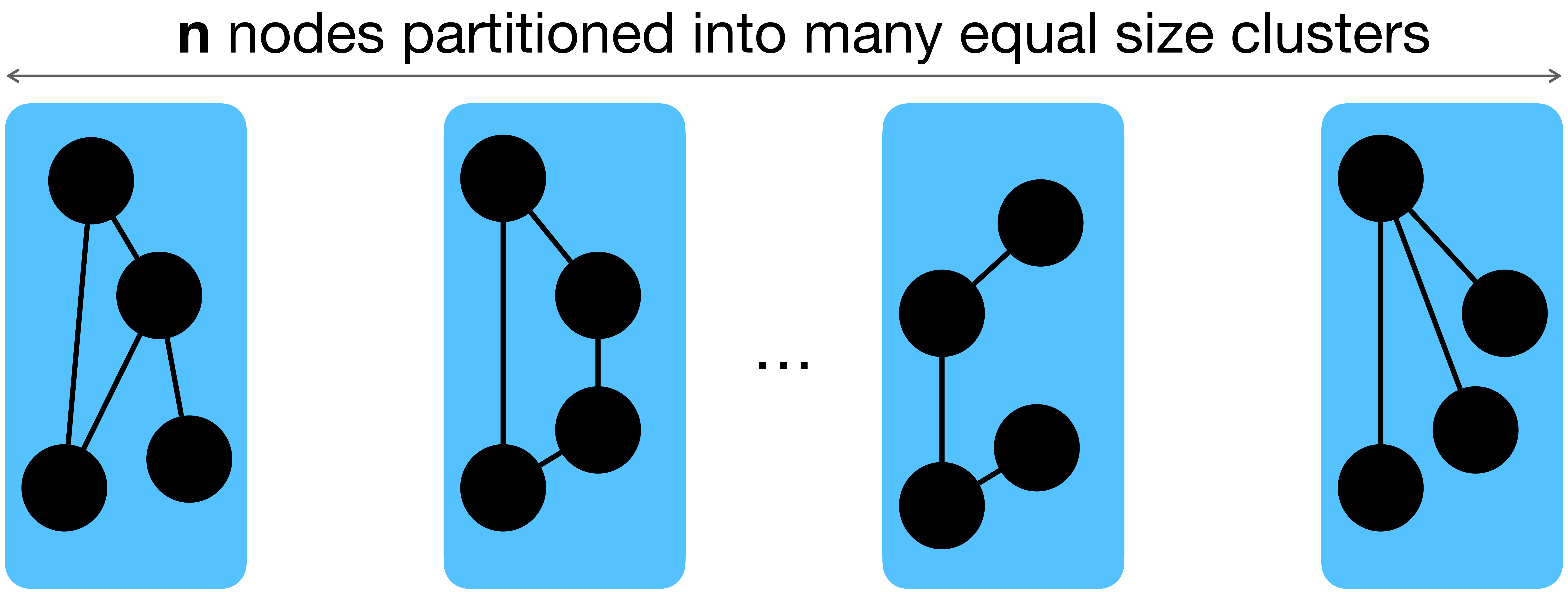
Balanced Partitioning



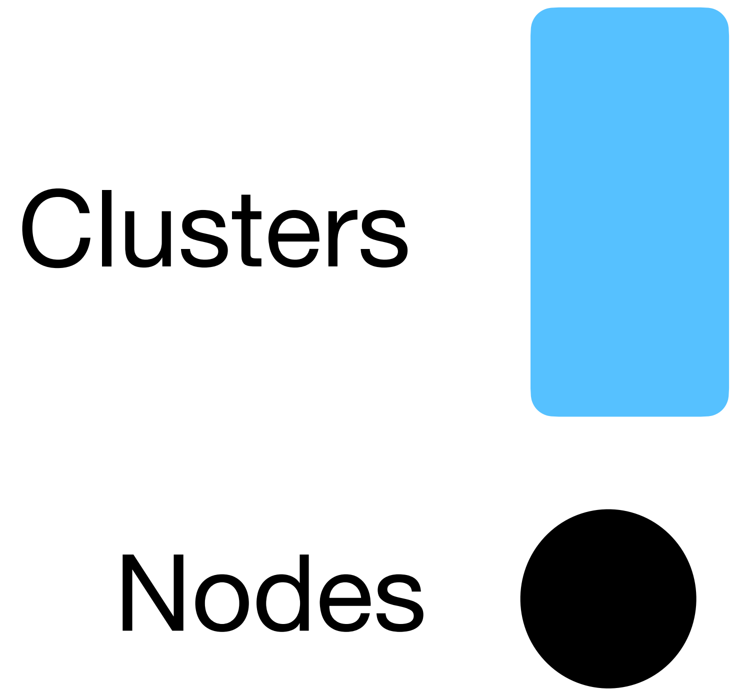
- **Nodes** represent virtual machines in a datacenter.
- Each **cluster** is a physical server machine.



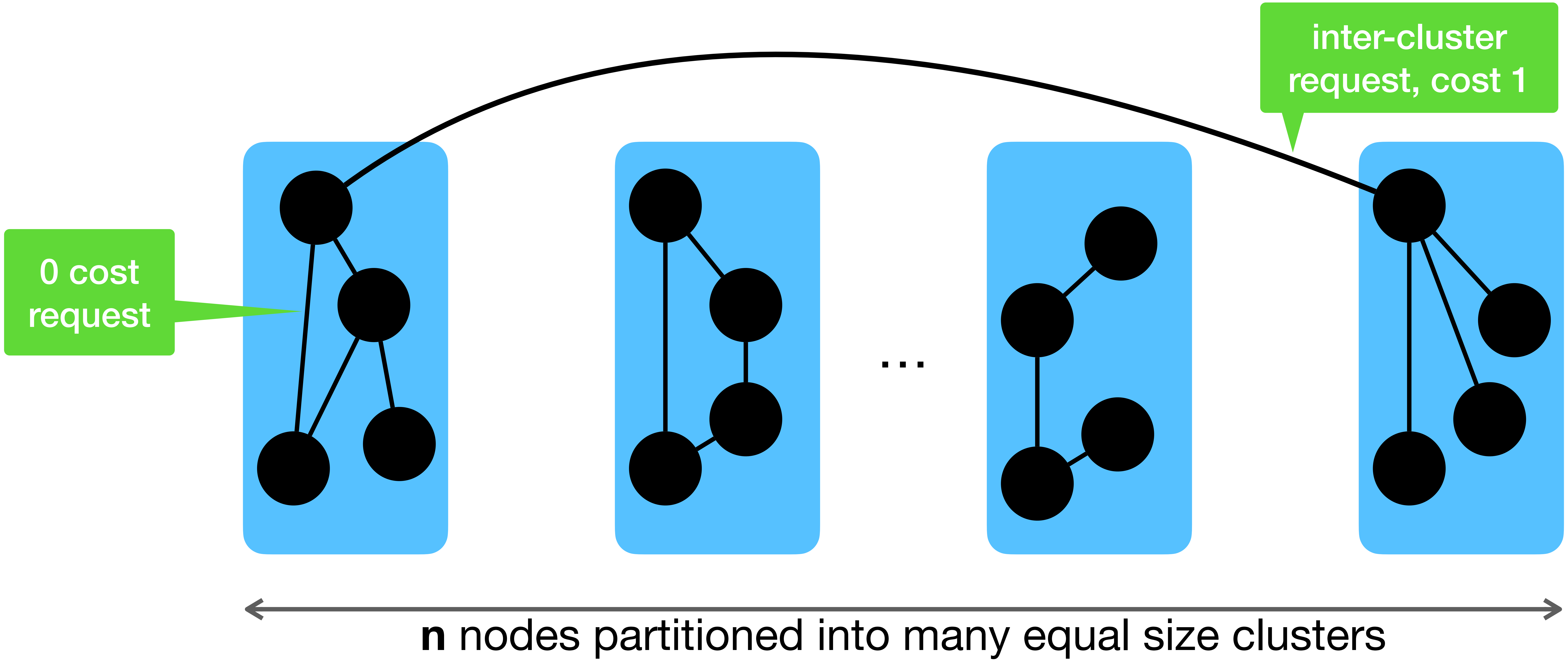
Balanced Partitioning: Definitions



- **Input:** pairwise communication **requests**, shown as **edges**.
- Requests arrive one at a time, in an **online** manner.
- Algorithms **serve** each request as soon as it arrives.
- Before the next request, it may **migrate/move** nodes.

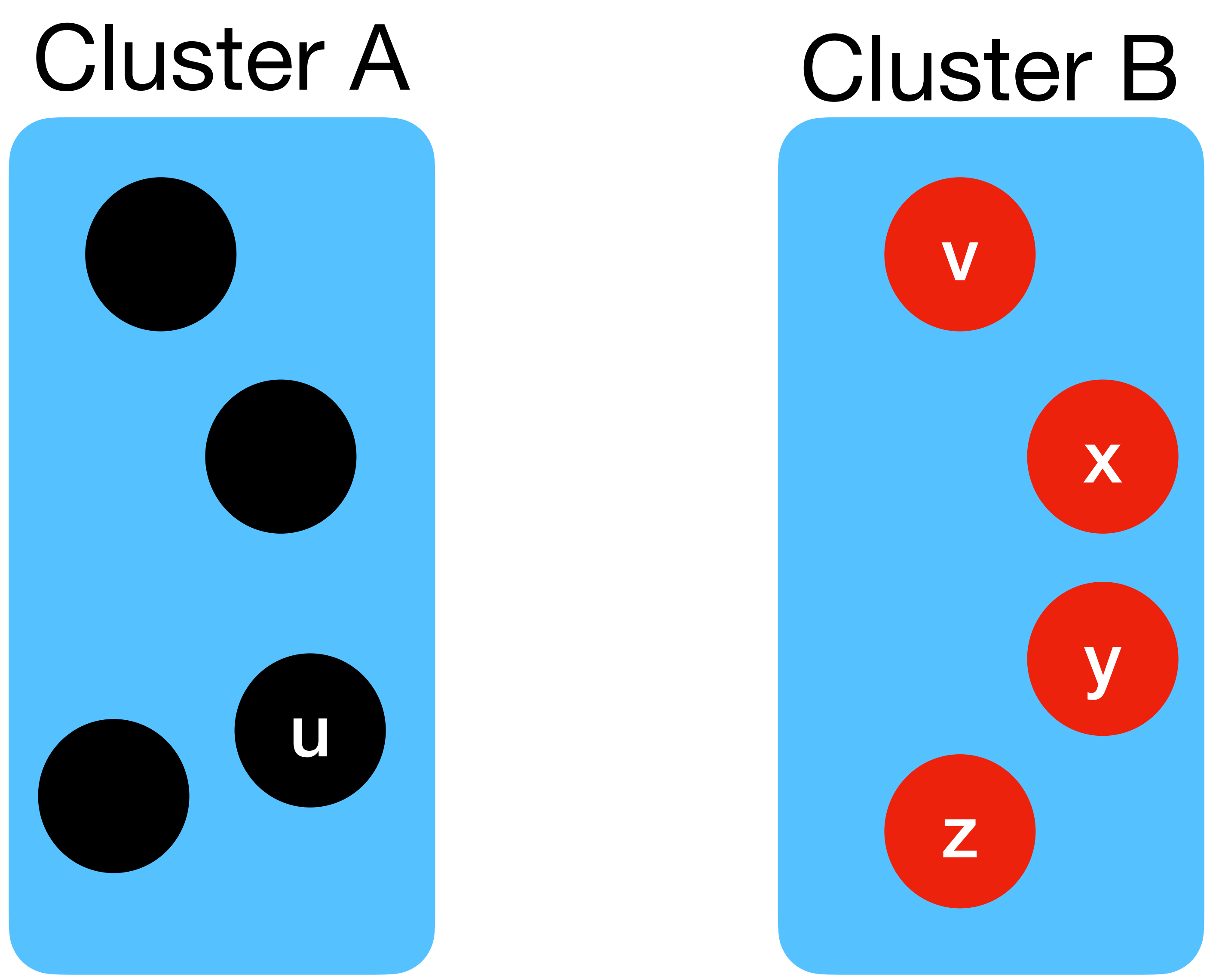


Balanced Partitioning: Definitions (1)



- Requests within a cluster are served with **no cost.**
- Requests between clusters cost **1.**
- Moving nodes costs $\alpha \geq 1$.

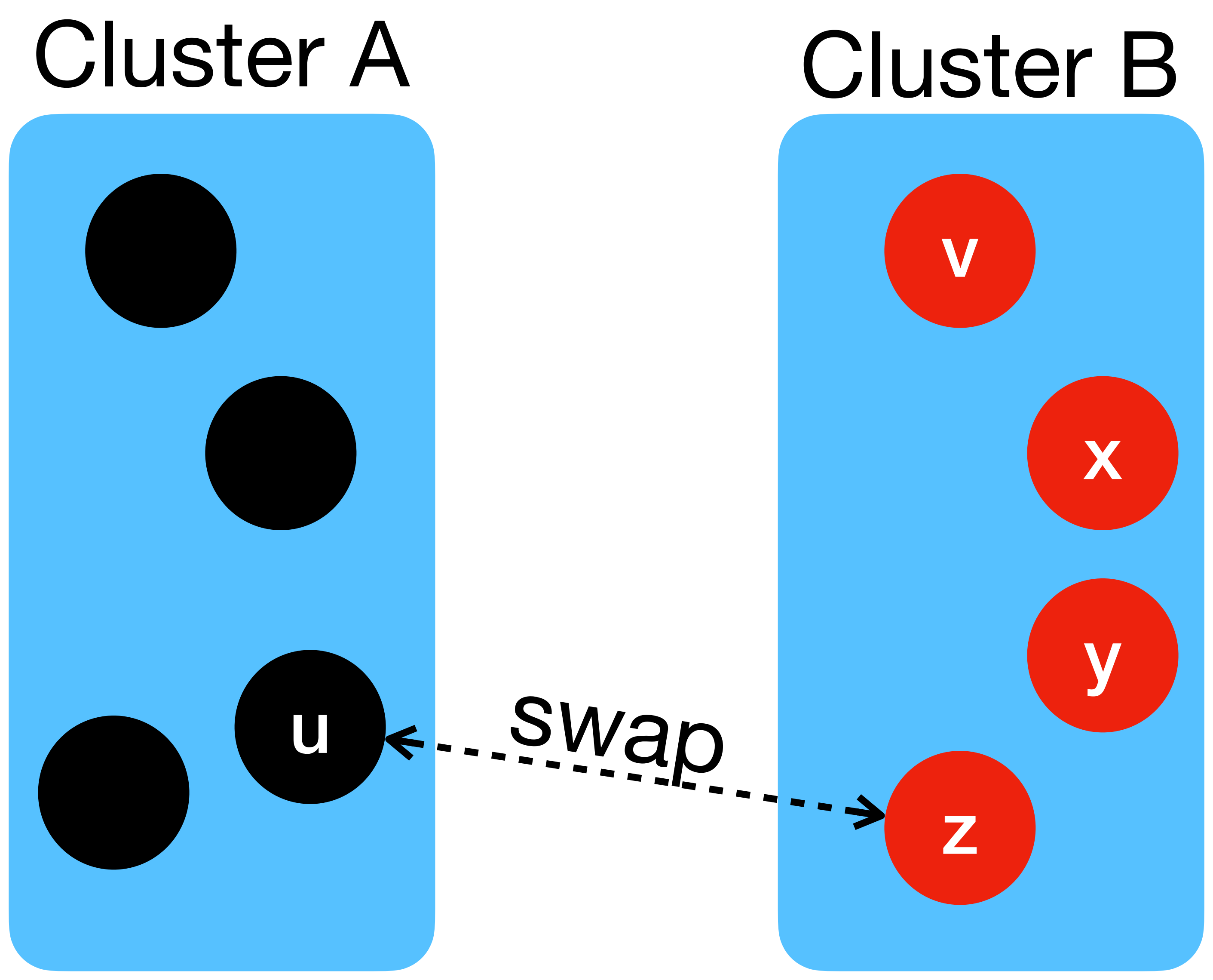
The Learning Variant



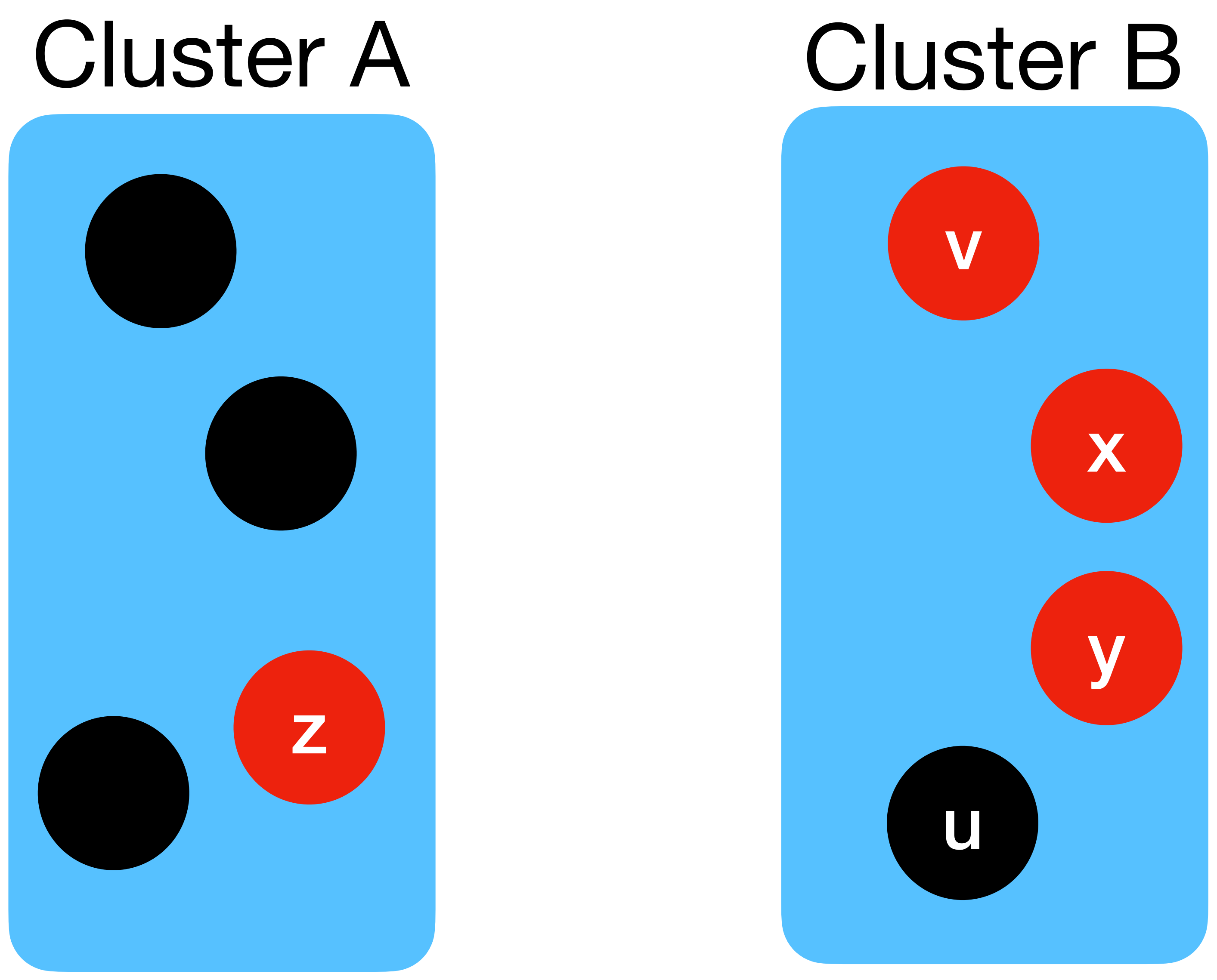
Initial partition

- **Perfect Partition**: a partition with **no inter-cluster** edge, unknown to us.
- Algorithm must **learn**/recover the partition while edges arrive.

The Learning Variant



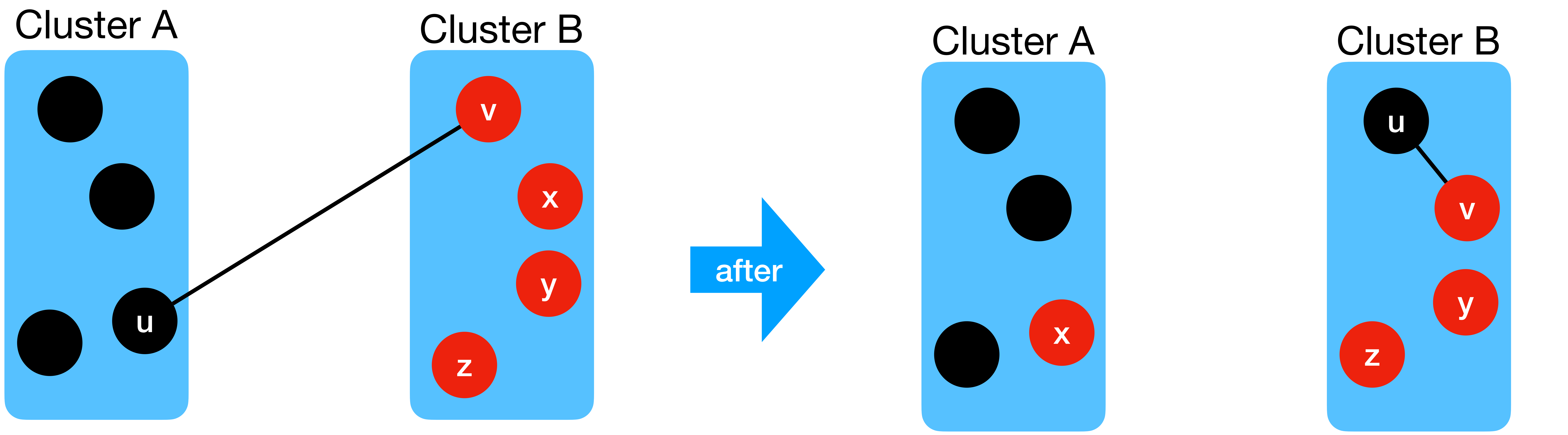
Initial partition



Hidden Perfect partition

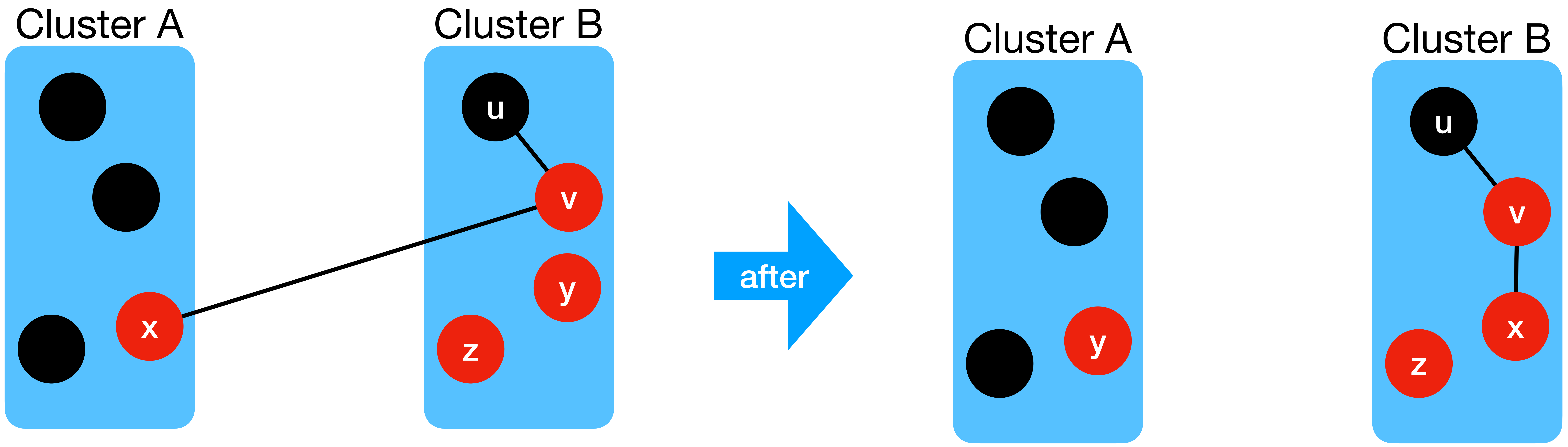
- **Perfect Partition**: a partition with **no inter-cluster** edge, initially unknown.
- Algorithm must **learn**/recover the partition while edges arrive.

The Learning Variant: Example



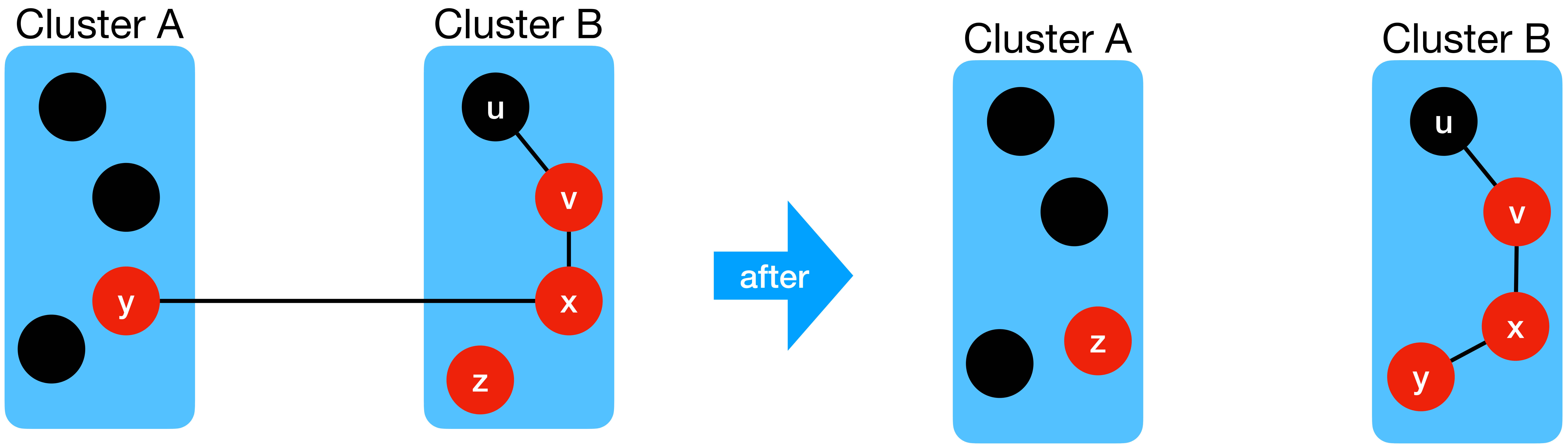
- First request, between **u** and **v**.
- On <u>which cluster</u> we should **collocates**?
- <u>Which node</u> we should **evict** to make space?

The Learning Variant: Example



- Next: between **v** and **x.**
- On <u>which cluster</u> we should **collocates**?
- <u>Which node</u> we should **evict** to make space?

The Learning Variant: Example



- Next: between **x** and **y**.
- On which cluster we should **collocates**?
- Which node we should **evict** to make space?

The Learning Variant: Lower Bound

- We assume a **perfect partition** always exists.
- Any deterministic algorithm is doomed to make wrong swaps.
- **Optimal offline** makes the right swaps.
- Any deterministic algorithm is $\Omega(n)$-competitive against the optimal.
- Can we get a matching upper bound?

The Learning Variant: Upper Bound

- We assume a **perfect partition** always exists.
- Competitive algorithms repartition per inter-cluster request.
- There can be many partition collocating the endpoints.
- Arbitrary choice of partition can be as bad as $O(n^2)$-competitive.
- The one closest to the initial partition is an optimal choice! $O(n)$

The End

Algorithm: PPL

- Idea: maintain connected **components**.
- Once an edge arrives, join the two endpoints into a single component.
- Move to a partition **closest** to the <u>initial partition</u>.
- O(n)-approximation or **O(n)-competitive**.

Lower Bound

- Cluster size at least 3.
- No algorithm can approximate the optimal within **o(n)**.
- Meaning, PPL is asymptotically optimal.
- For clusters of 2 nodes the LB is 3.
- A 6-approximation for the size **2** case.

Co-authored Publications

1. K.-T. Foerster, M. Parham, and S. Schmid. Path restoration on product graphs: Resilient tori, generalized hypercubes, and beyond. Submitted to SRDS 2019, May 2019
2. K.-T. Foerster, M. Parham, S. Schmid, and T. Wen. Local fast segment rerouting on hypercubes. In *22nd International Conference on Principles of Distributed Systems (OPODIS)*, December 2018
3. S. A. Amiri, S. Dudycz, M. Parham, S. Schmid, and S. Wiederrecht. On polynomial-time congestion-free software-defined network updates. In *18th IFIP Networking Conference (IFIP Networking)*, May 2019
4. S. A. Amiri, K.-T. Foerster, R. Jacob, M. Parham, and S. Schmid. Waypoint routing in special networks. In *17th IFIP Networking Conference (IFIP Networking 2018)*, May 2018
5. C. Avin, L. Cohen, M. Parham, and S. Schmid. Competitive clustering of stochastic communication patterns on a ring. *Computing*, Sep 2018
6. K. Foerster, M. Parham, M. Chiesa, and S. Schmid. TI-MFA: keep calm and reroute segments fast. In *Global Internet Symposium (GI)*, April 2018
7. K.-T. Foerster, M. Parham, and S. Schmid. A walk in the clouds: Routing through vnfs on bidirected networks. In *ALGOCLOUD*, Lecture Notes in Computer Science, pages 11–26. Springer, September 2017