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 $\alpha \geq 1$. 

$n$ nodes partitioned into many equal size clusters
The Learning Variant

Perfect Partition: a partition with no inter-cluster edge, unknown to us.

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

• Algorithm must *learn*/recover the partition while edges arrive.
• First request, between $u$ and $v$.

• On which cluster we should collocates?

• Which node we should evict to make space?
• Next: between \( \mathbf{v} \) and \( \mathbf{x} \).

• On which cluster we should collocates?

• Which node we should evict to make space?
Next: between \textbf{x} and \textbf{y}.

On \textbf{which cluster} we should \textbf{collocates}?

\textbf{Which node} we should \textbf{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 initial partition.

• $O(n)$-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


