Advanced Topics in Distributed Systems

PRESENTED BY: ZAINAB ABBAS

Opponent: Sana Imtiaz

PAPERS

Device Placement Optimization with Reinforcement Learning (ICML 2017)

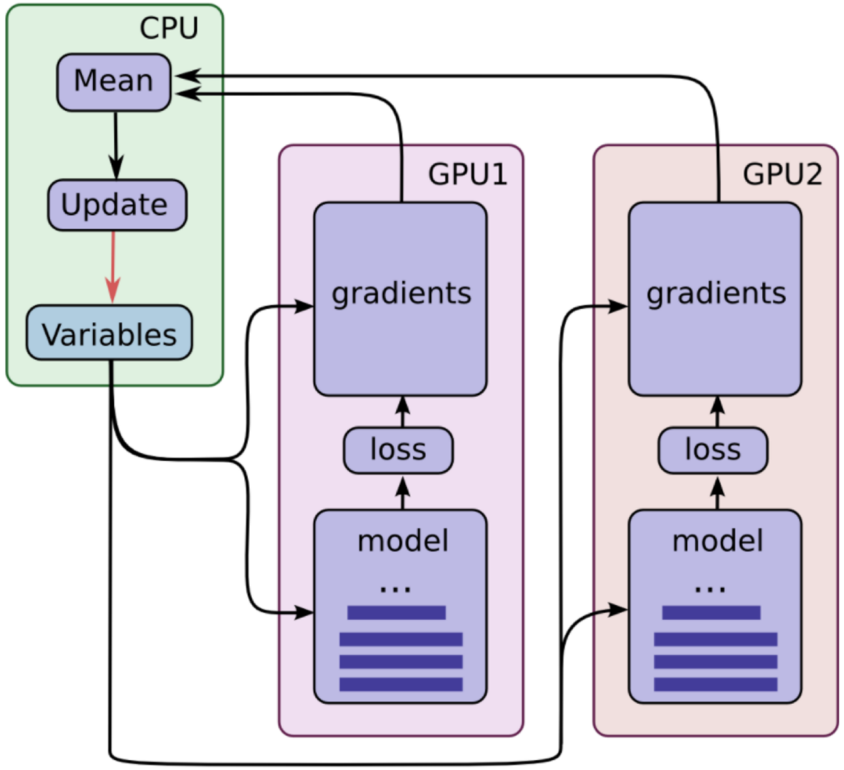
A Deep Learning Framework for Graph Partitioning (ICLR 2019)

Streaming Graph Partitioning for Large Distributed Graphs (KDD 2012)

DEVICE PLACEMENT 1

The decision of placing parts of the neural models on devices is often made by human experts based on simple heuristics and intuitions.

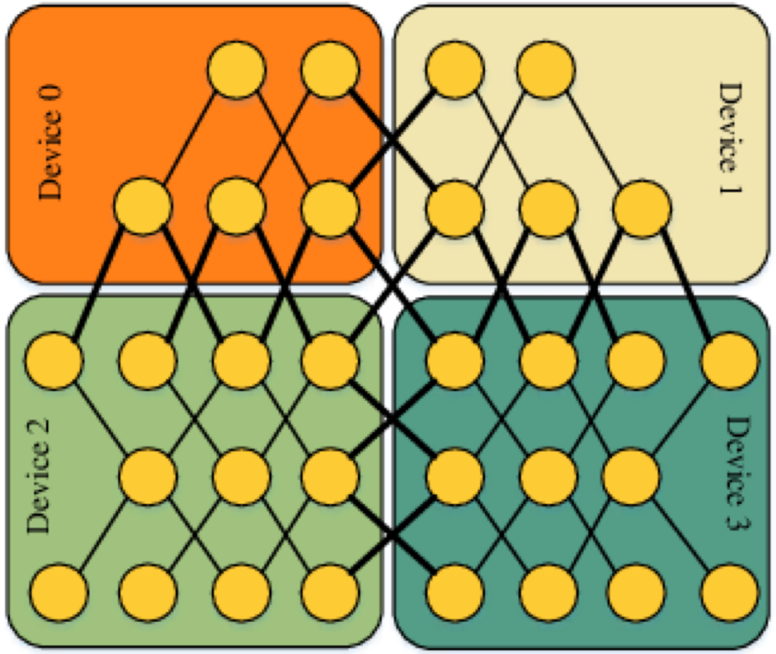
Can we optimize the execution time of the ML job if we partition smartly?



* *Image reference: https://jhui.github.io/2017/03/07/TensorFlow-GPU/*

DEVICE PLACEMENT 2

The decision of placing parts of the neural models on devices is often made by human experts based on simple heuristics and intuitions.

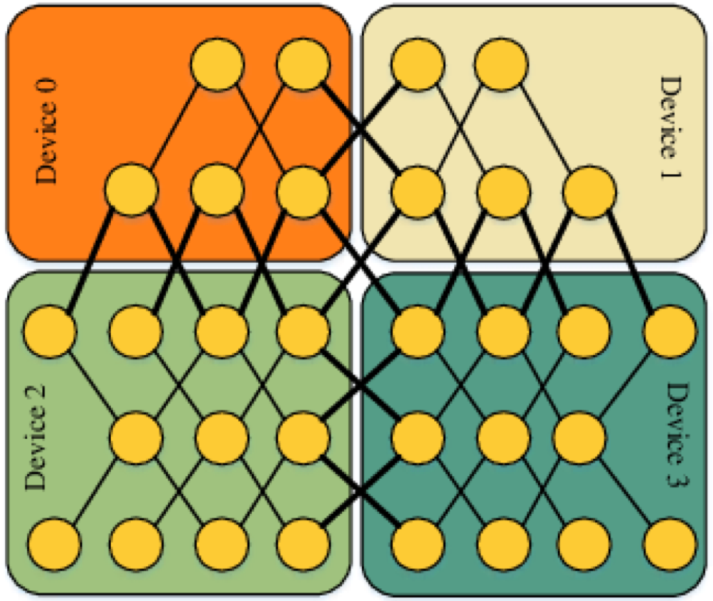


Can we optimize the execution time of the ML job if we partition smartly?

* *Image reference: Liu, Junnan et al. "An effective method for operations placement in Tensor Flow." HP3C '19 (2019).*

DEVICE PLACEMENT 3

A TensorFlow computational graph G, consisting of M operations {o1, o2, ..., oM }, and D available devices.



The goal of device placement optimization is to find a placement the execution time is minimized.

* *Image reference: Liu, Junnan et al. "An effective method for operations placement in Tensor Flow." HP3C '19 (2019).*

DEVICE PLACEMENT USNG REINFORCEMENT LEARNING

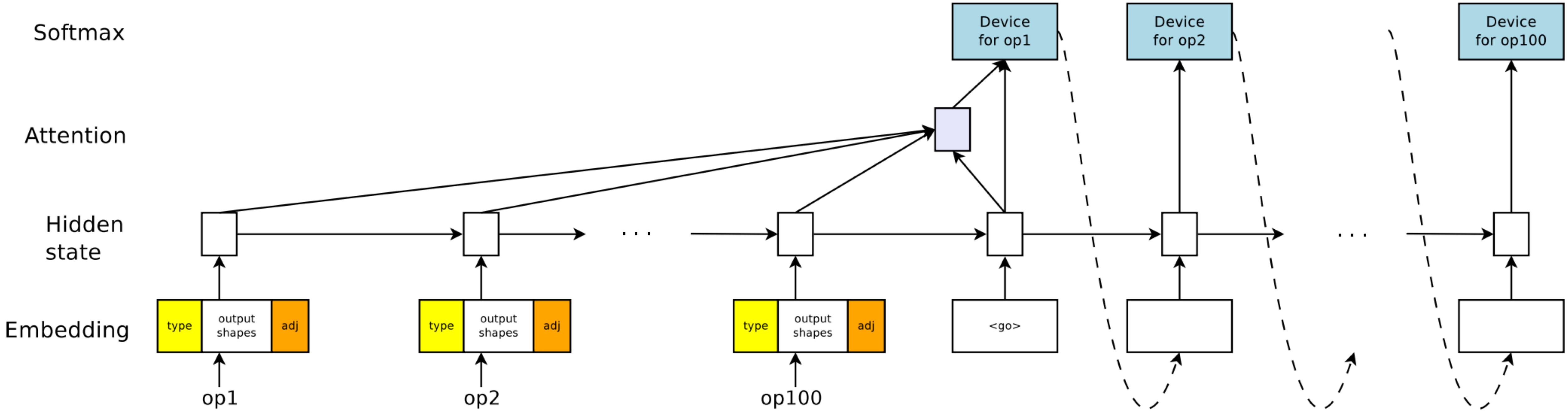
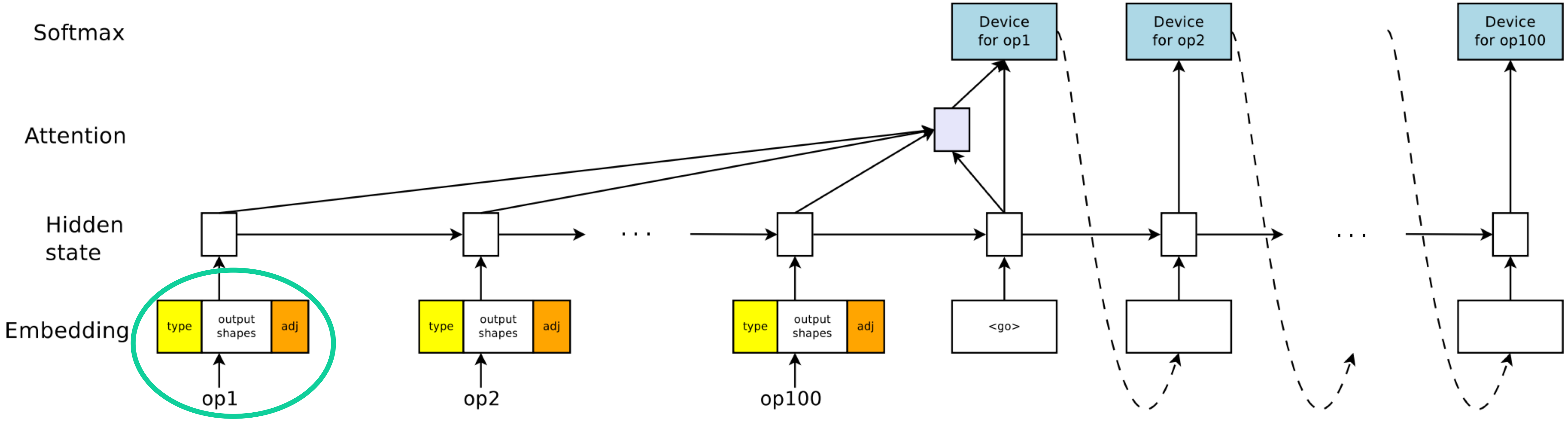


Image source: Azalia Mirhoseini et.al. 2017. Device placement optimization with reinforcement learning. ICML'17, Vol. 70. JMLR.org 2430-2439.

DEVICE PLACEMENT USNG REINFORCEMENT LEARNING



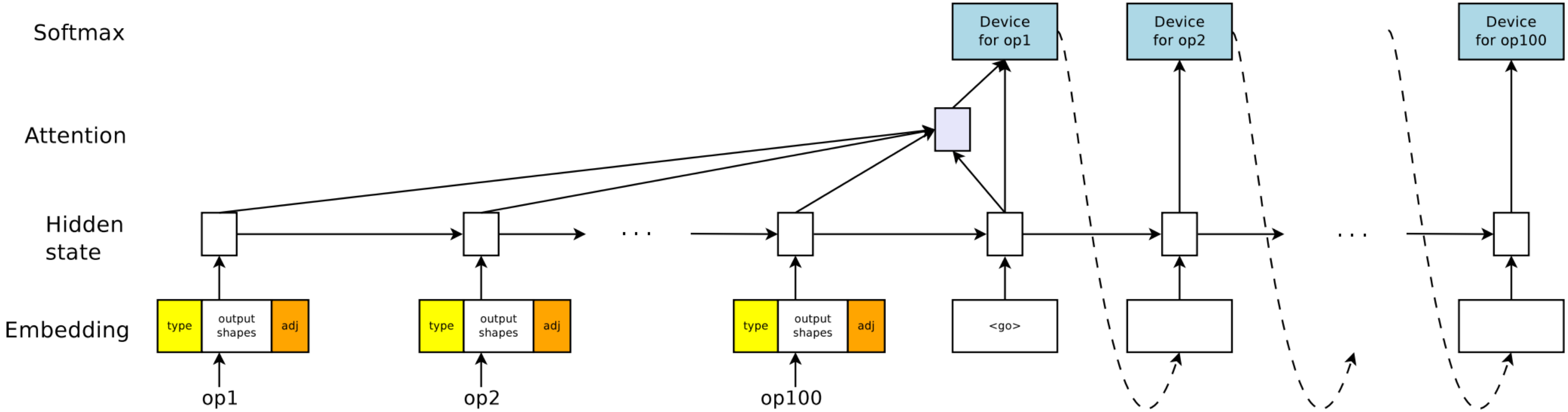
Operation’s type such as, MatMul or conv2d

Operation’s output shape

Adjacency matrix of operations being input output to other operations

*Image source: Azalia Mirhoseini et.al. 2017. Device placement optimization with reinforcement learning. ICML’17, Vol. 70. JMLR.org 2430-2439.

DEVICE PLACEMENT USNG REINFORCEMENT LEARNING

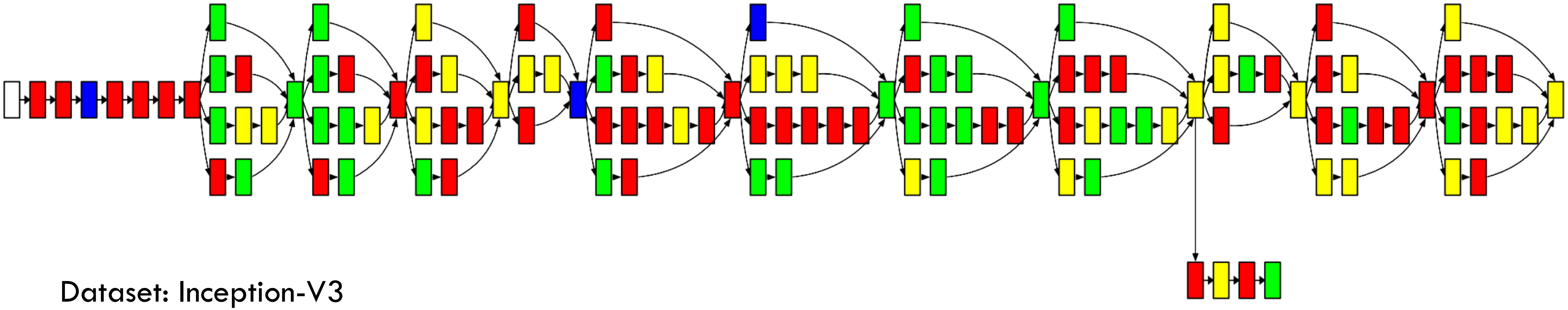


Encoder Decoder approach using LSTMs

At each step, the decoder outputs the device for the operation at the same encoder time step. Each device has its own tunable embedding, which is then fed as input to the next decoder time step.

Image source: Azalia Mirhoseini et.al. 2017. Device placement optimization with reinforcement learning. ICML'17, Vol. 70. JMLR.org 2430-2439.

RESULTS AND SHORTCOMINGS



Dataset: Inception-V3

19.3 % better execution time compared to the expert designed placement

Models were not general enough

Tensorflow operations grouping

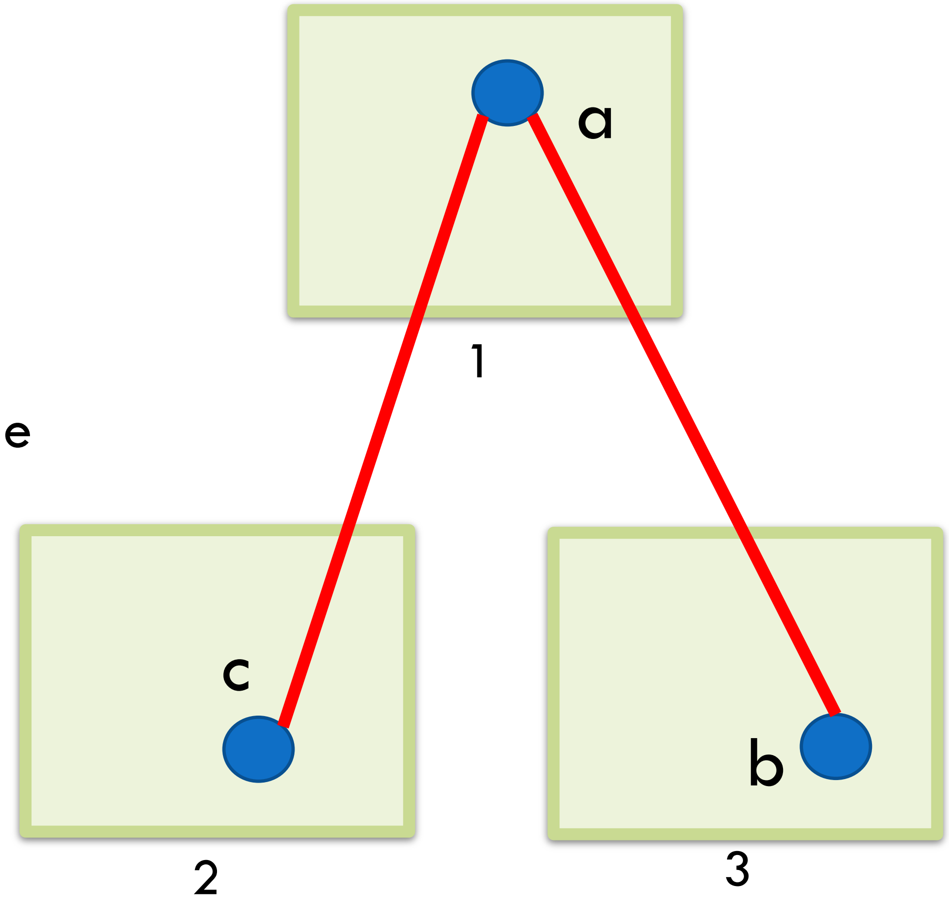
Image source: Azalia Mirhoseini et.al. 2017. Device placement optimization with reinforcement learning. ICML'17, Vol. 70. JMLR.org 2430-2439.

GAP

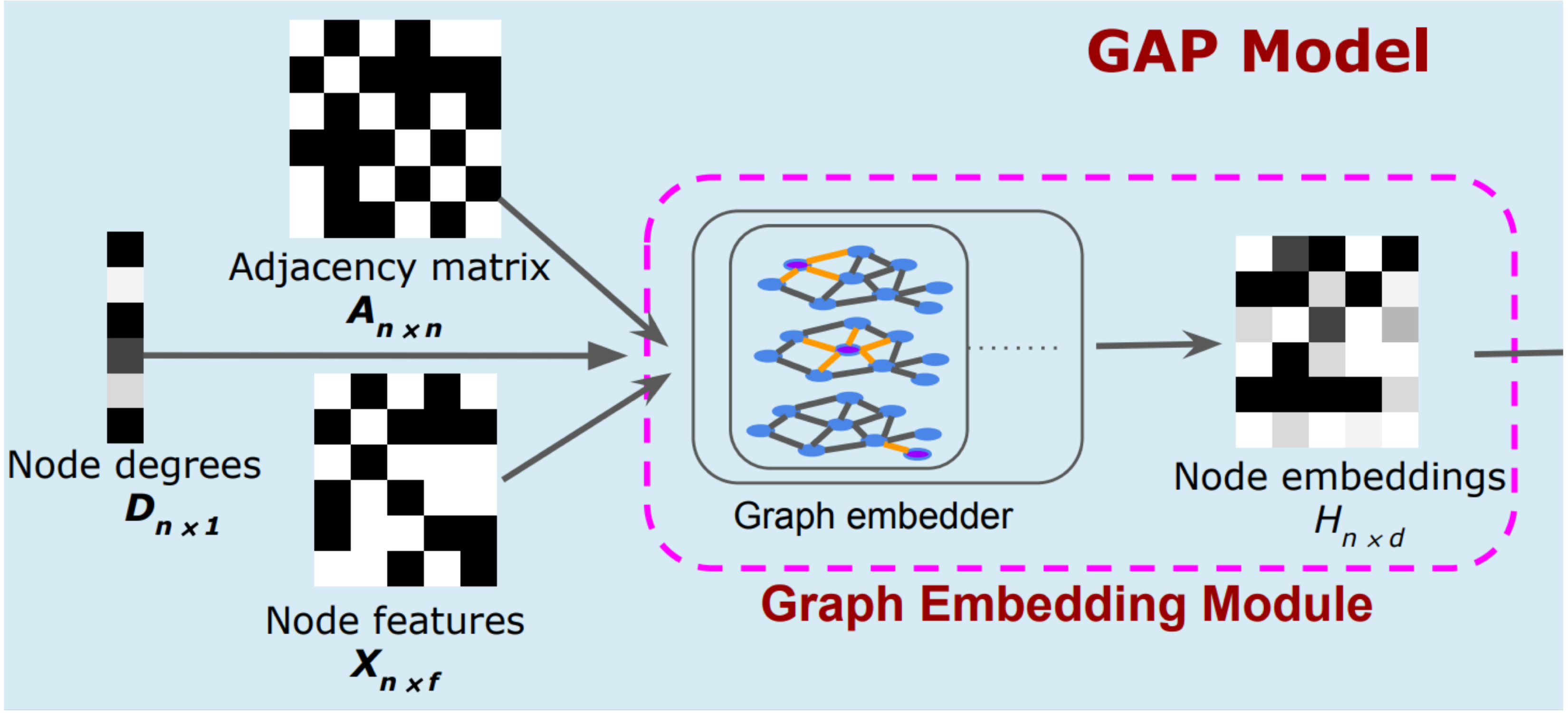
Deep learning based graph partitioning

Objective is partitioning a graph into balanced partitions while minimizing the number of edge cut across those partitions.

A deep model trained to optimize for this objective.



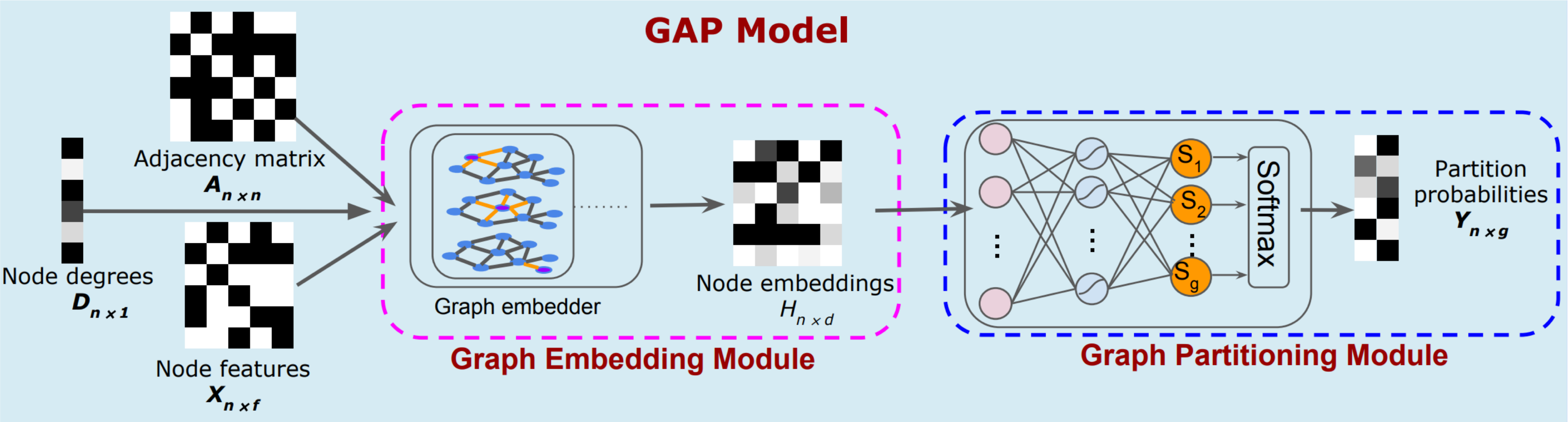
GAP



The graph structure and node features are used for embedding

*Image source: A Deep Learning Framework for Graph Partitioning. Azade Nazi, Will Hang, Anna Goldie, Sujith Ravi and Azalia Mirhoesini

GAP



Node embeddings used for partitioning

[Image source: A Deep Learning Framework for Graph Partitioning](). Azade Nazi, Will Hang, Anna Goldie, Sujith Ravi and Azalia Mirhoesini

GAP LIMITATIONS

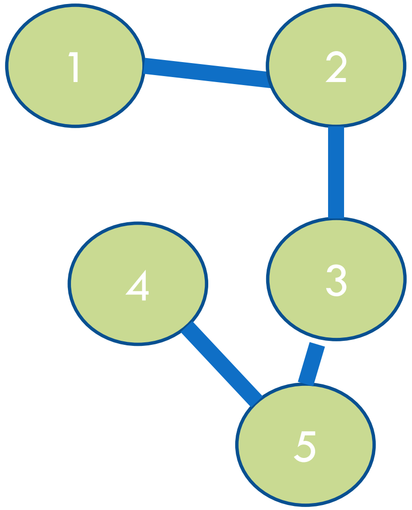
Claim: Capable of generalization, i.e, training a model on a set of graphs can be used at inference time on unseen graphs of varying sizes.

Baseline methods have to redo the entire optimization for each new graph.

- More experiments using large graph datasets
- Real-world data sets of social media network not used
- Missing performance experiments

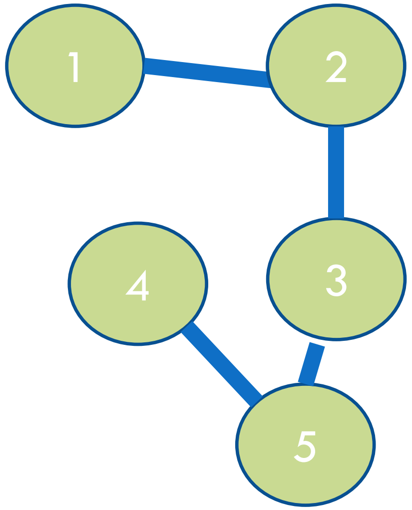
STREAMING GRAPH PARTITIONING

OFFLINE PARTITIONING METHOD

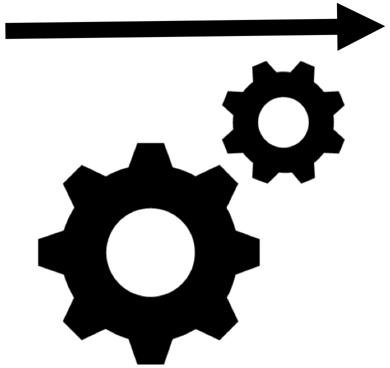


Input
graph

OFFLINE PARTITIONING METHOD



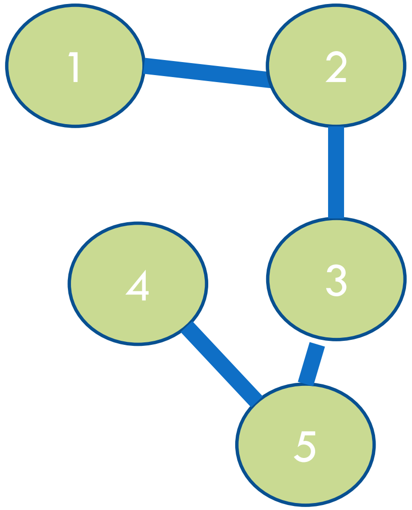
Input graph



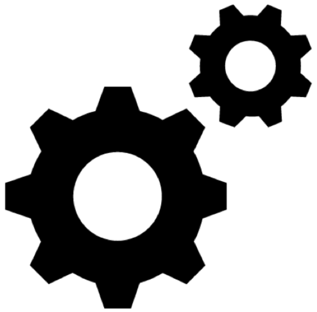
	1	2	3	4	5
1	0	1	0	0	0
2	1	0	1	0	0
3	0	1	0	0	1
4	0	0	0	0	1
5	0	0	1	1	0

Processing before partitioning

OFFLINE PARTITIONING METHOD



Input graph



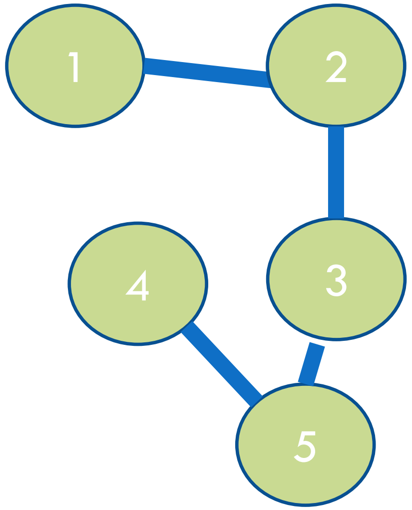
	1	2	3	4	5
1	0	1	0	0	0
2	1	0	1	0	0
3	0	1	0	0	1
4	0	0	0	0	1
5	0	0	1	1	0

Processing before partitioning

partitioning

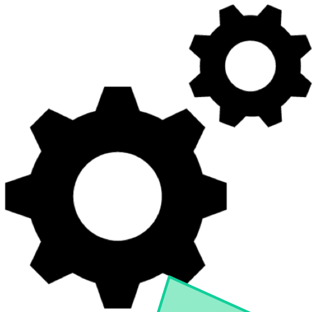
OFFLINE PARTITIONING METHOD

High Processing Cost



Inp

gr



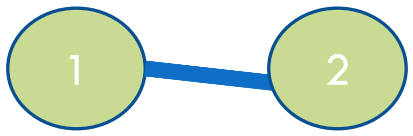
Need to observe the whole graph before partitioning

	1	2	3	4	5
1	0	1	0	0	0
2	1	0	1	0	0
3	0	1	0	0	1
4	0	0	0	0	1
5	0	0	1	1	0

Processing before partitioning

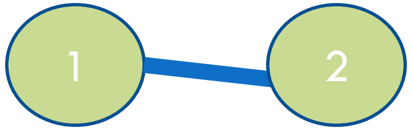
partitioning

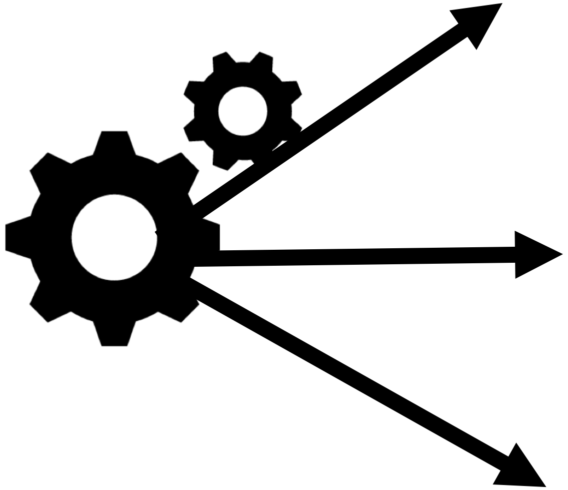
STREAMING GRAPH PARTITIONING



Input
stream

STREAMING GRAPH PARTITIONING

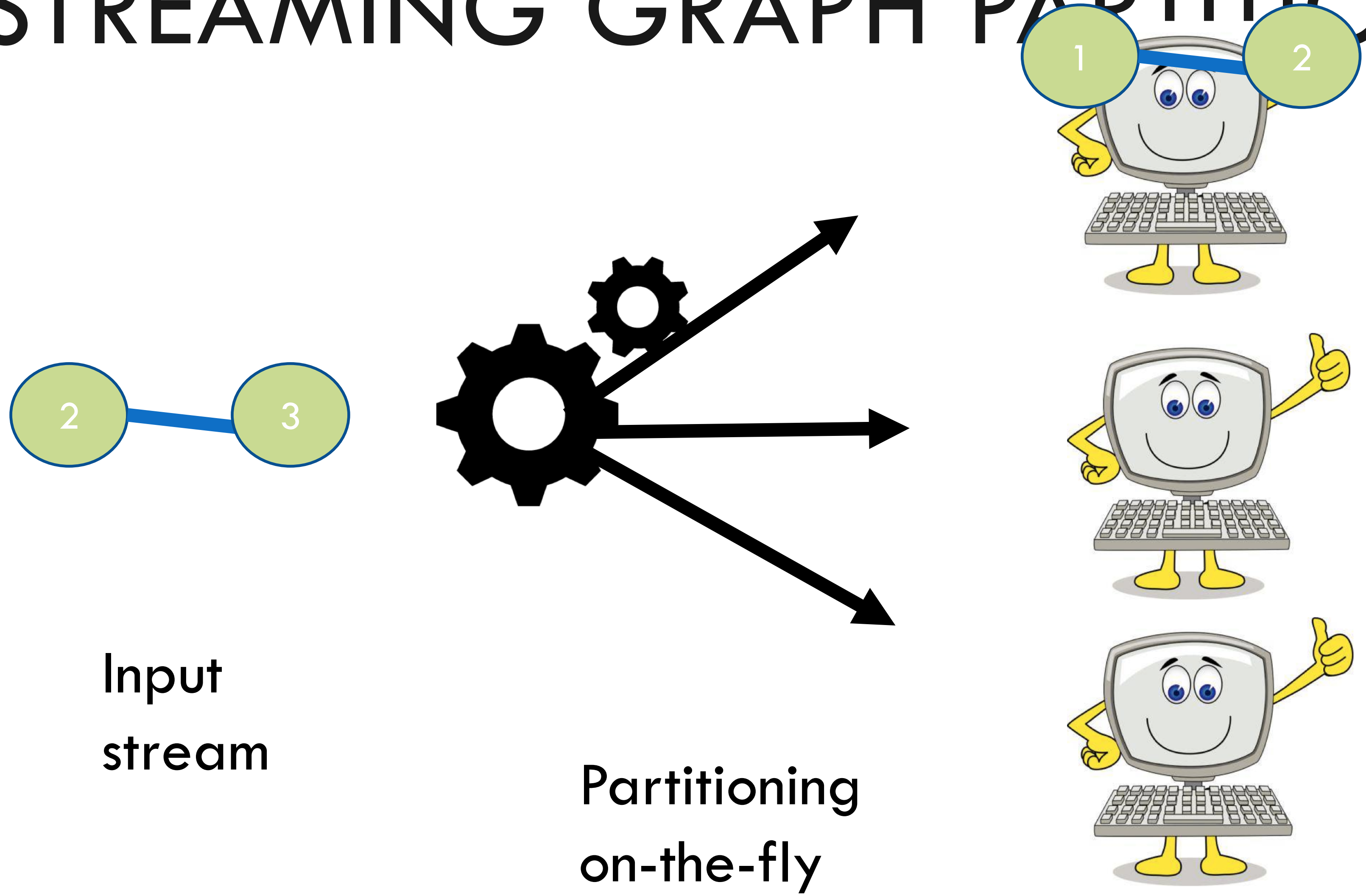




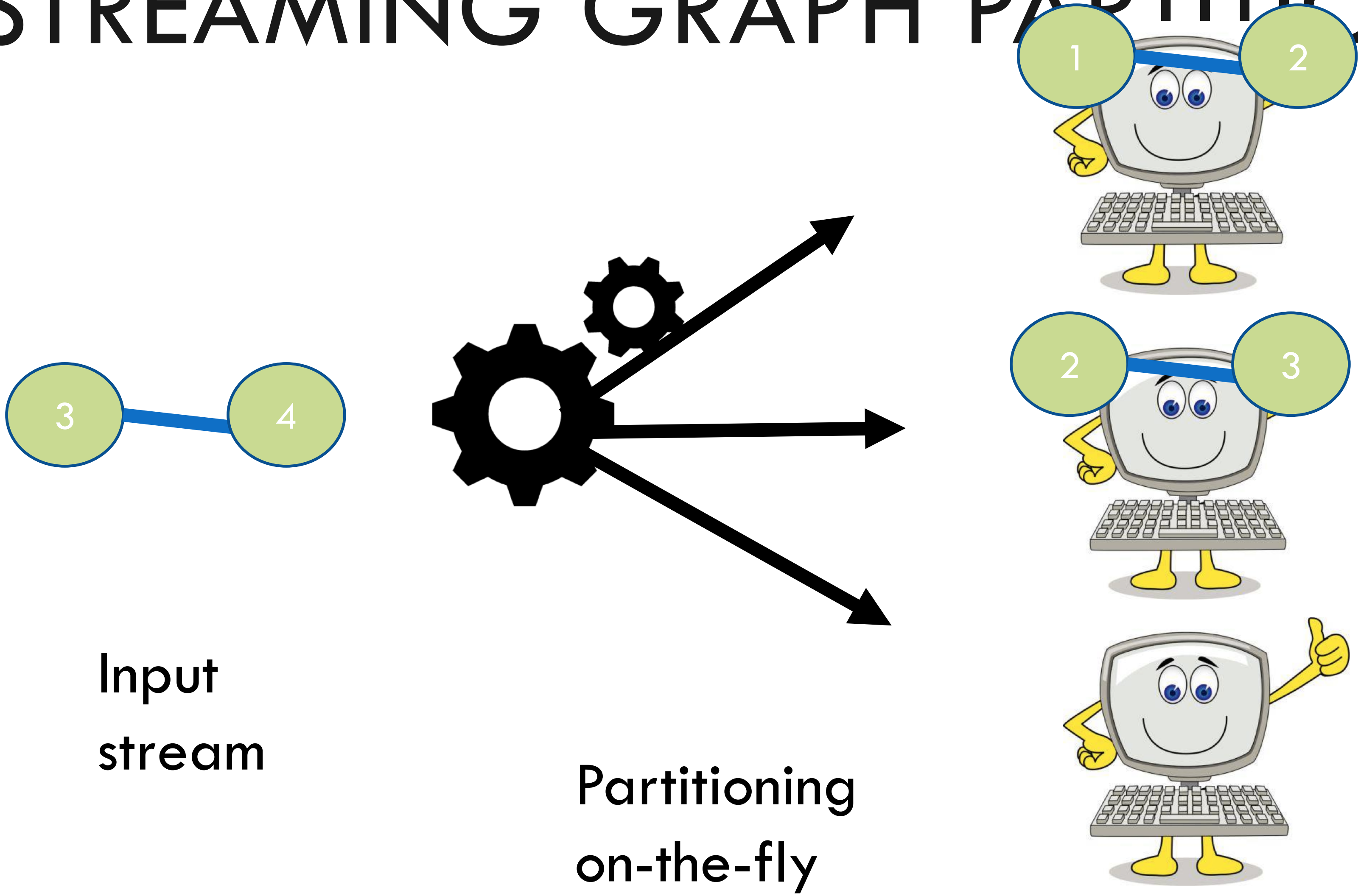
Input
stream

Partitioning
on-the-fly

STREAMING GRAPH PARTITIONING



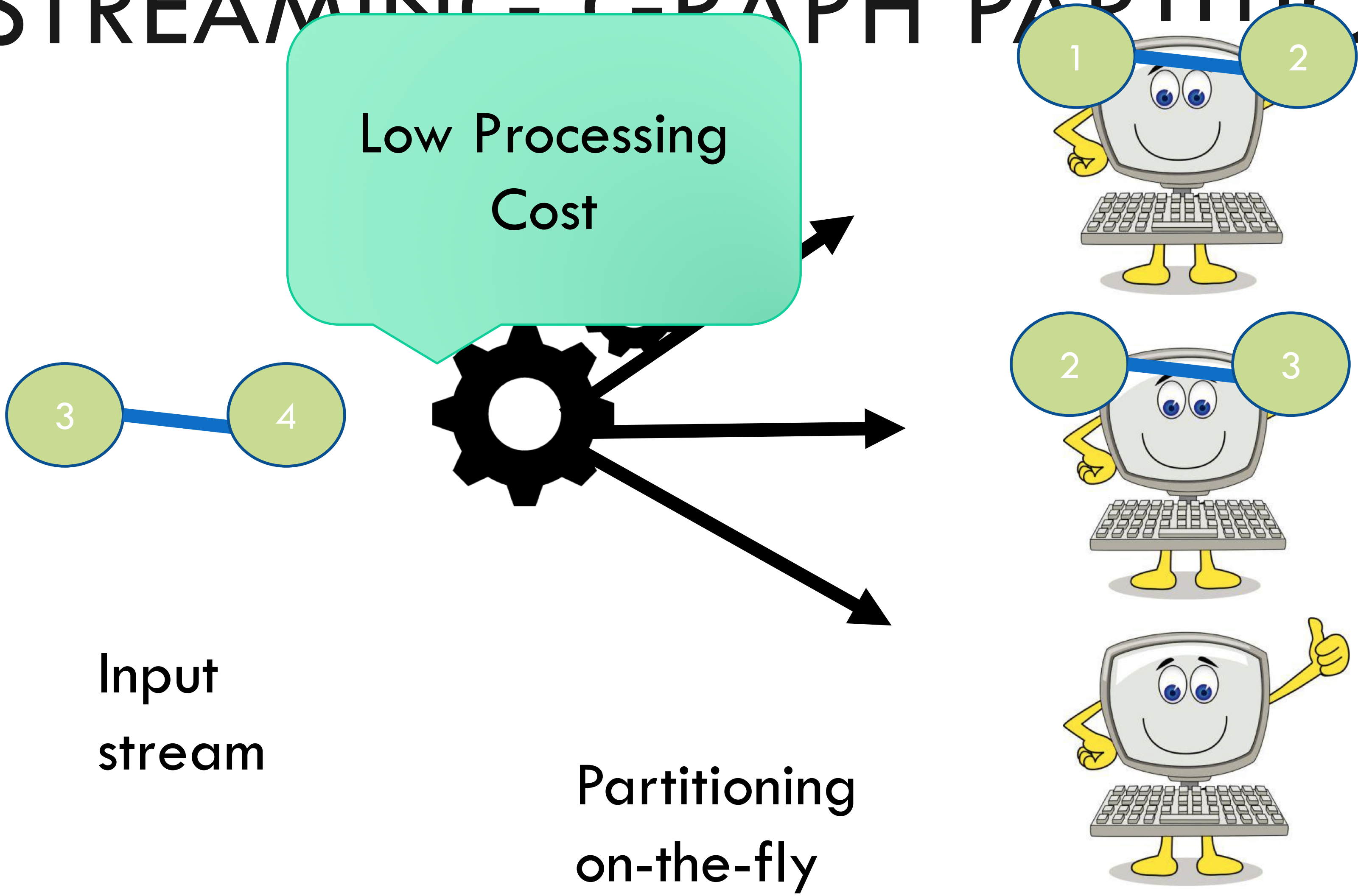
STREAMING GRAPH PARTITIONING



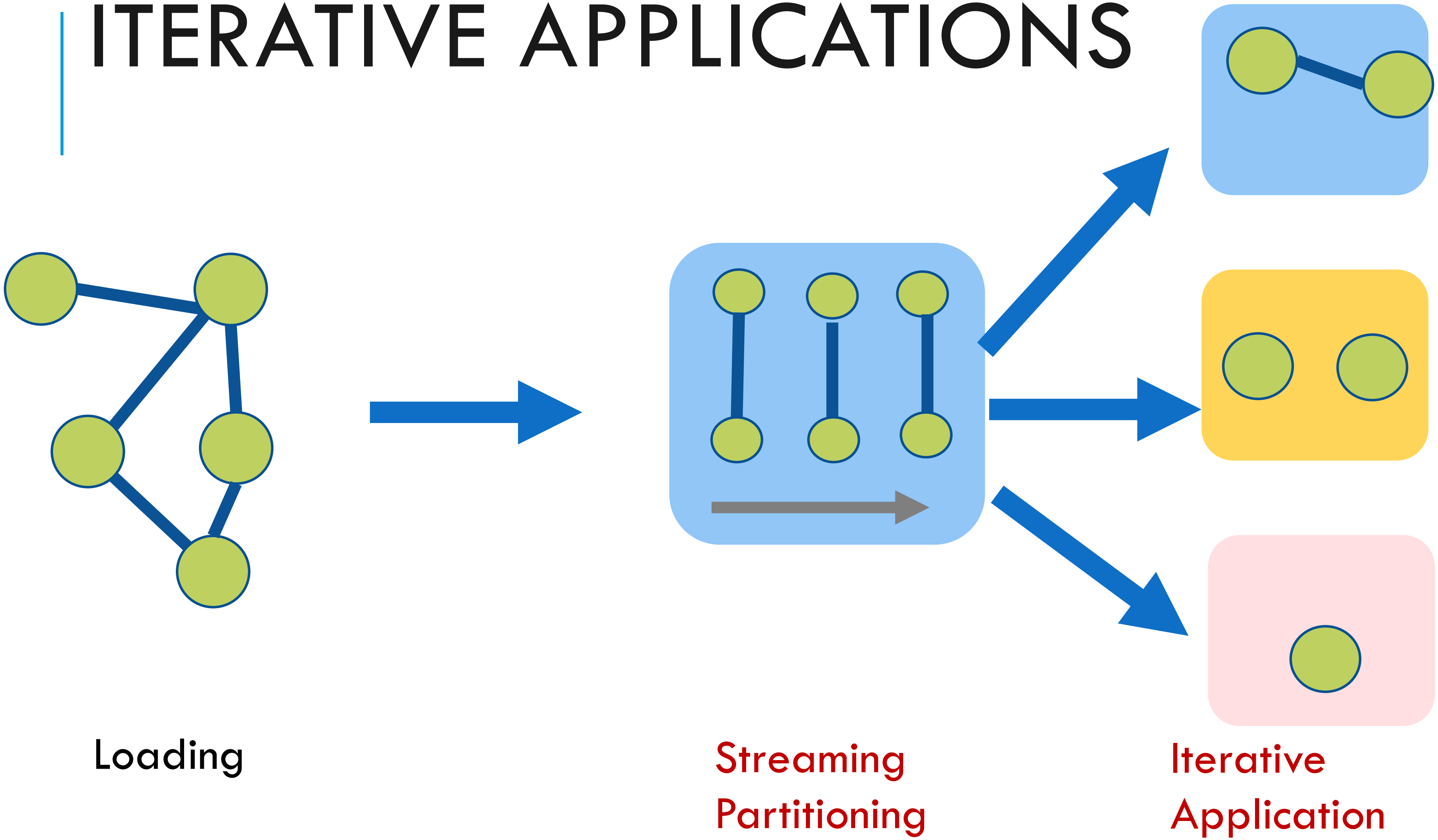
Input stream

Partitioning on-the-fly

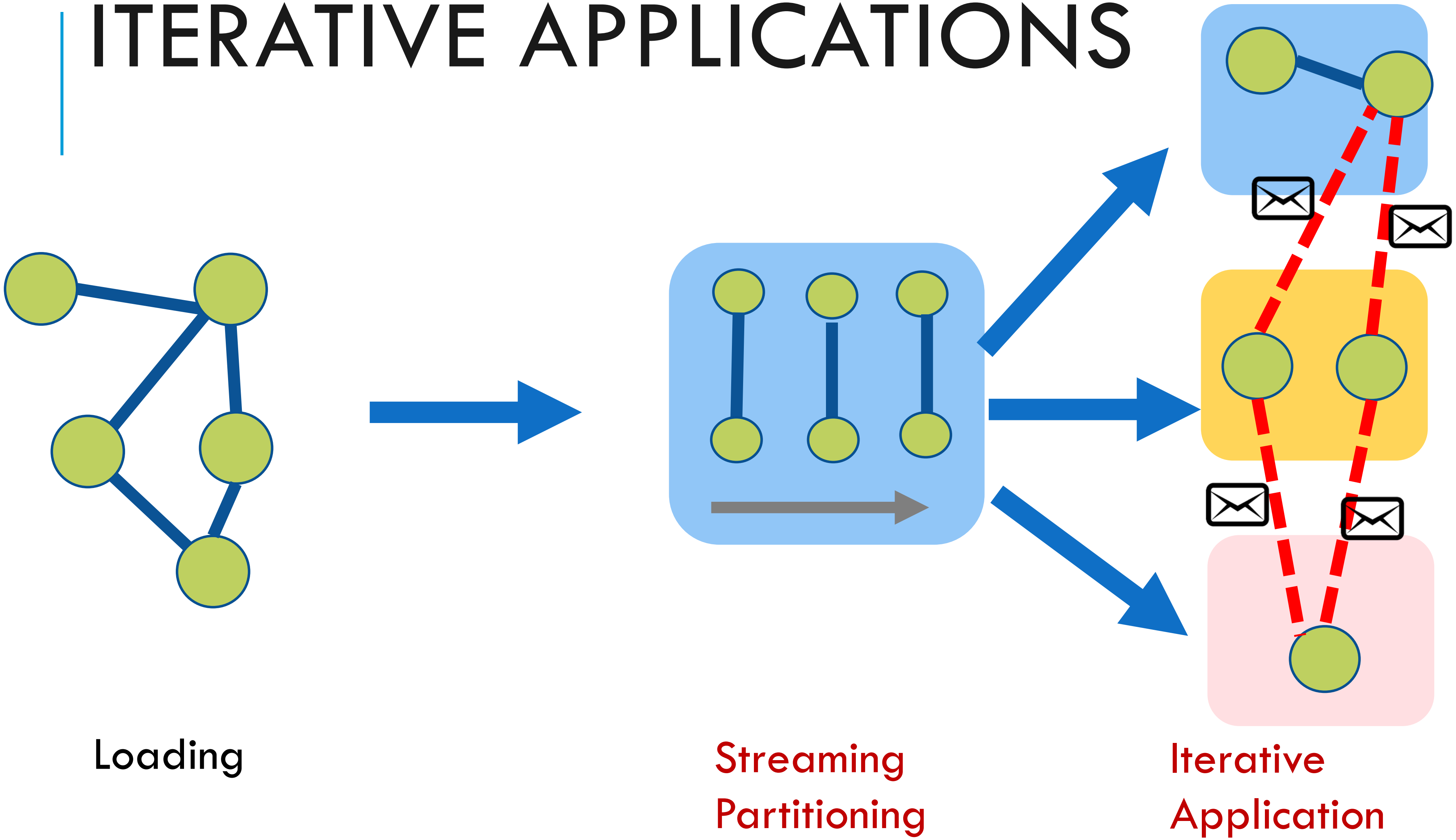
STREAMING GRAPH PARTITIONING



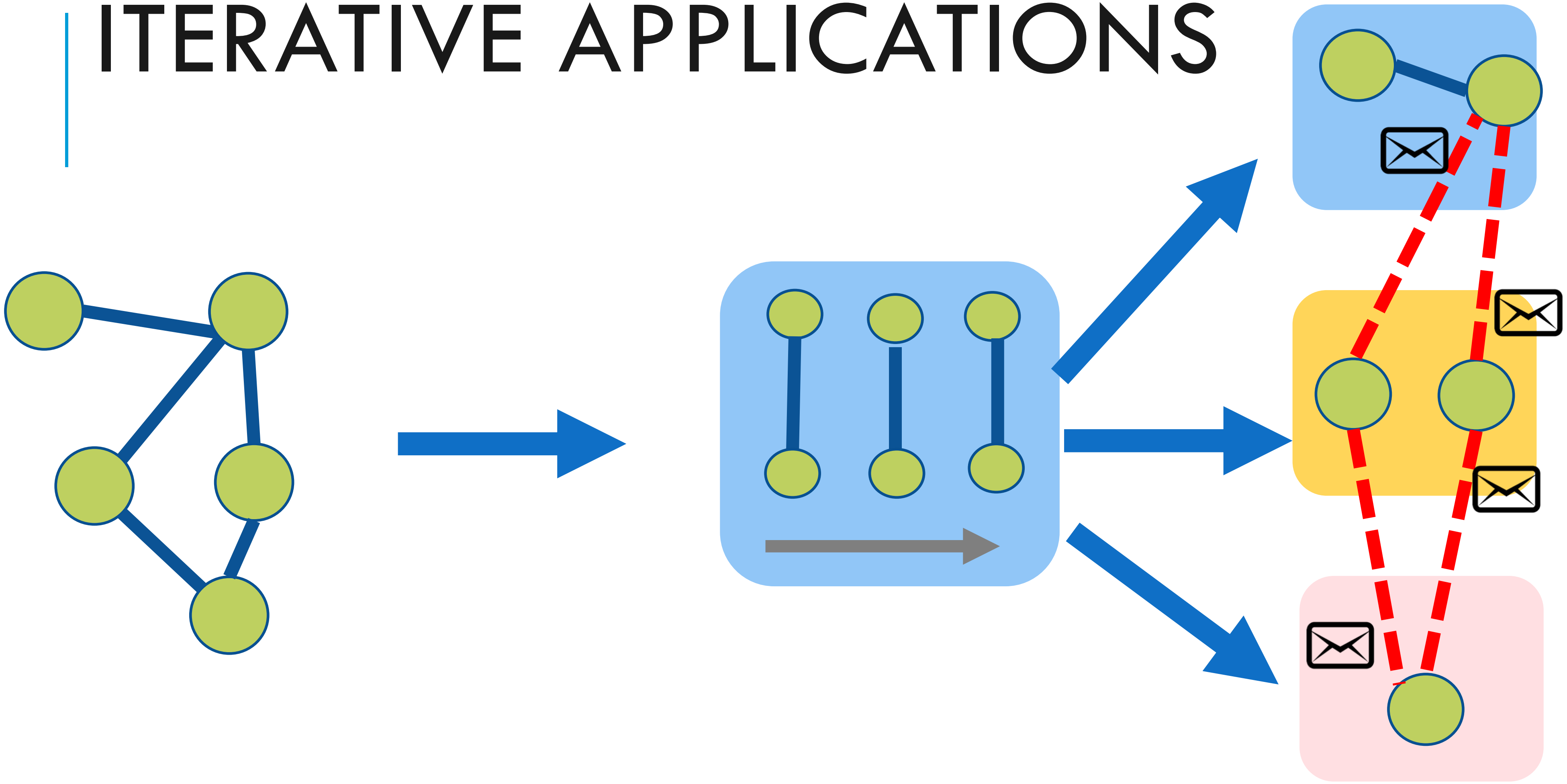
ITERATIVE APPLICATIONS



ITERATIVE APPLICATIONS



ITERATIVE APPLICATIONS

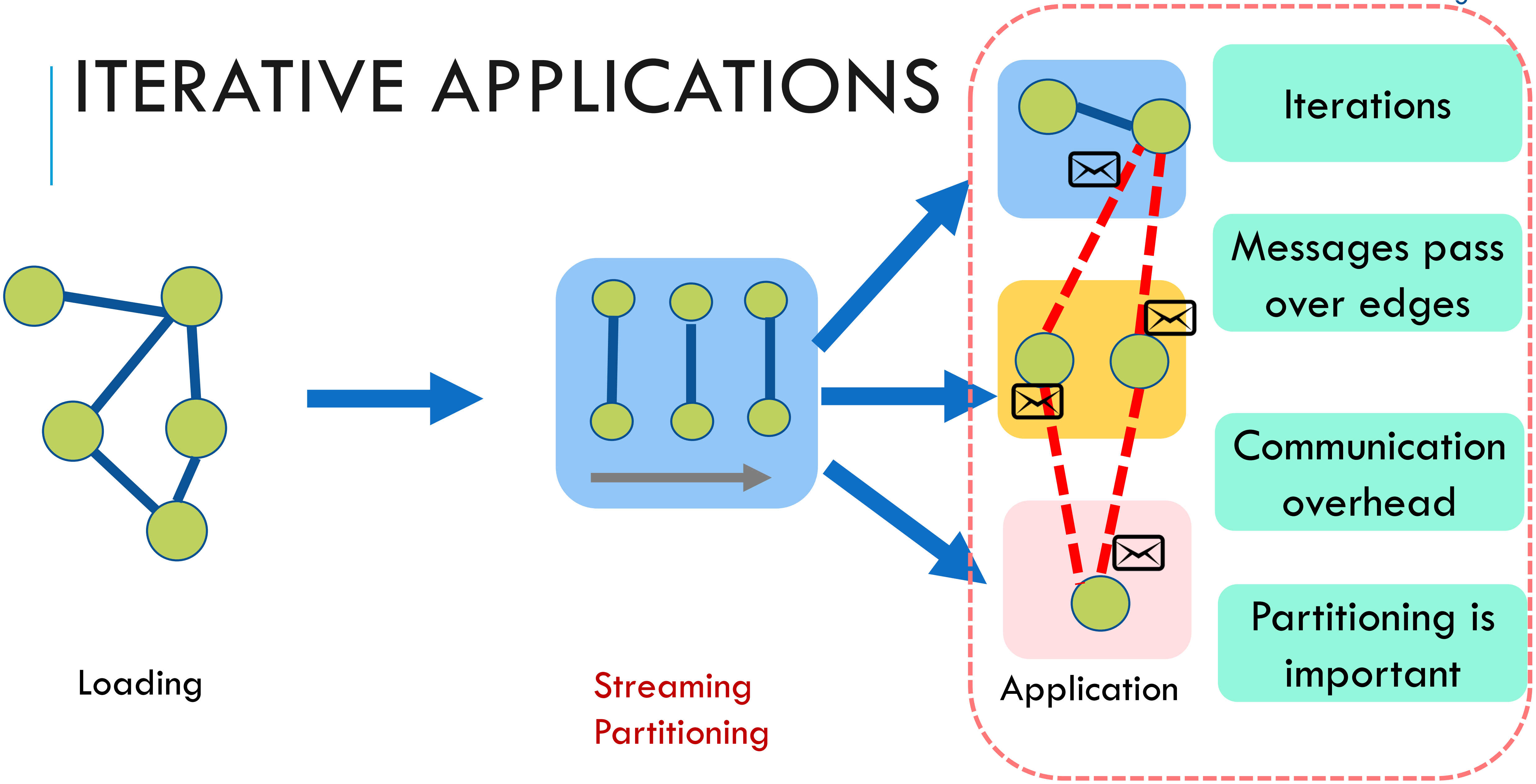


Loading

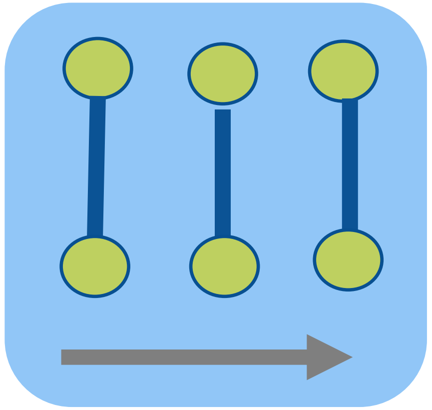
Streaming
Partitioning

Iterative
Application

ITERATIVE APPLICATIONS



STREAMING PARTITIONING



Streaming
Partitioning

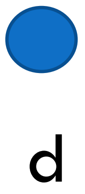
Ingest input as stream

Partitioning on the fly

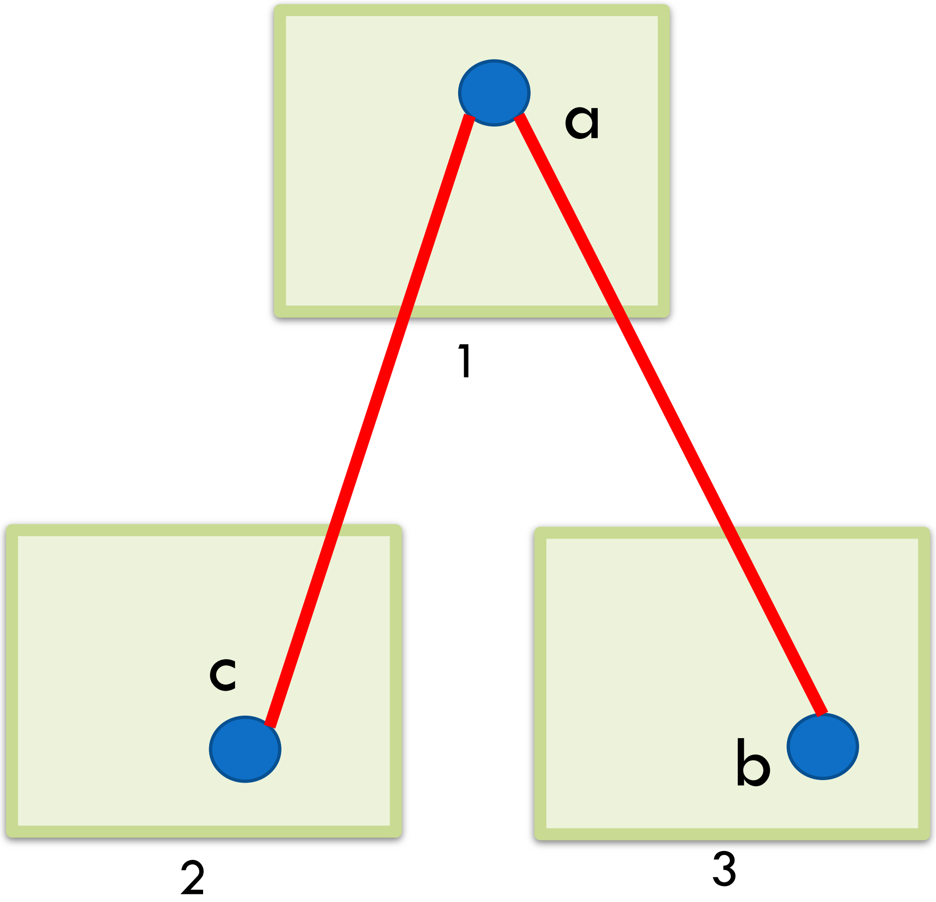
Single Pass

Partial graph knowledge

LINEAR DETERMINISTIC GREEDY



Neighbours : a, b



Assign vertex to a partition with maximum neighbours

While keeping some capacity constrains

RESULTS

Better cuts and load balance compared to METIS

Better performance of applications compared to HASH and METIS

Page Rank improved 18% to 39% on large input graphs

LIMITATIONS

Some prior graph information is required

Uses vertex partitioning only

COMPARISON

Papers	Task	Objective	Input	Offline/Online
Streaming graph partitioning	Partition streaming graphs on the fly	Less cuts and load balance => improve execution time of graph applications	Streaming graphs with nodes and edges	Online
Device placement with Reinforcement learning	Partition deep learning computational graphs	Improve execution time of the model	TensorFlow operations graphs	Offline – redo optimizations per graph
GAP	Partitions deep learning computational graphs	Less cuts and load balance & Generalize the model	TensorFlow operations graphs / Synthetic random graphs	Training offline / Inference online

COMPARISON

Papers	Task
Streaming graph partitioning	Partition streaming graphs on the fly
Device placement with Reinforcement learning	Partition deep learning computational graphs
GAP	Partitions deep learning computational graphs

COMPARISON

Papers	Task	Objective
Streaming graph partitioning	Partition streaming graphs on the fly	Less cuts and load balance => improve execution time of graph applications
Device placement with Reinforcement learning	Partition deep learning computational graphs	Improve execution time of the model
GAP	Partitions deep learning computational graphs	Less cuts and load balance & Generalize the model

COMPARISON

Papers	Task	Objective	Input
Streaming graph partitioning	Partition streaming graphs on the fly	Less cuts and load balance => improve execution time of graph applications	Streaming graphs with nodes and edges
Device placement with Reinforcement learning	Partition deep learning computational graphs	Improve execution time of the model	TensorFlow operations graphs
GAP	Partitions deep learning computational graphs	Less cuts and load balance & Generalize the model	TensorFlow operations graphs / Synthetic random graphs

COMPARISON

Papers	Task	Objective	Input	Offline/Online
Streaming graph partitioning	Partition streaming graphs on the fly	Less cuts and load balance => improve execution time of graph applications	Streaming graphs with nodes and edges	Online
Device placement with Reinforcement learning	Partition deep learning computational graphs	Improve execution time of the model	TensorFlow operations graphs	Offline – redo optimizations per graph
GAP	Partitions deep learning computational graphs	Less cuts and load balance & Generalize the model	TensorFlow operations graphs / Synthetic random graphs	Training offline / Inference online

FUTURE WORK

Graph partitioning and deep learning approaches to be explored more, both in the domain of graph partitioning and device placement.

Modern stream processing architectures to be kept in mind while developing partitioning algorithms for both the task of deep learning and graph processing applications.

THANK YOU ☺