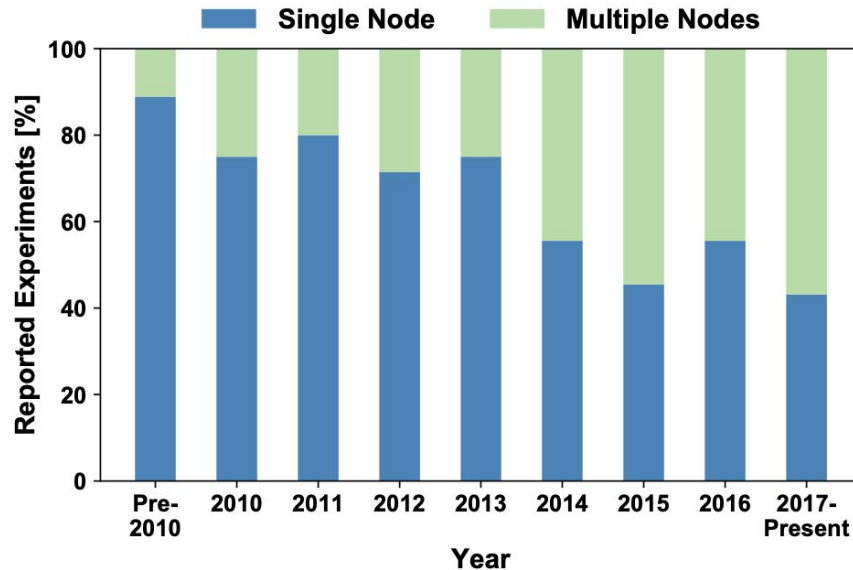


Распределенное обучение нейросетей

Богданов Александр, Пилипенко Сергей, Козлова Ольга

Что это?

Обучение, при котором рабочая нагрузка для обучения модели разделяется между несколькими рабочими узлами

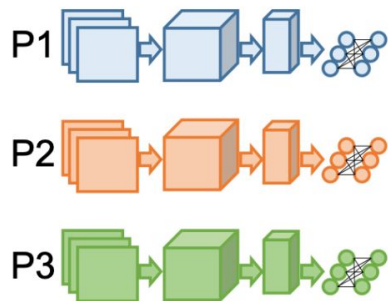


А чем не подходит обычное?

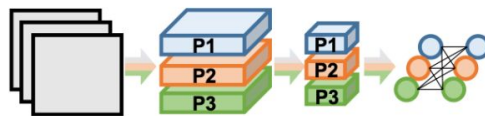
- Модель не помещается на GPU
- Данные не помещаются в RAM
- Слишком долго ждать



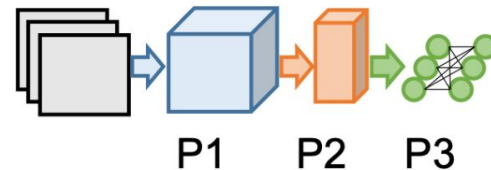
Какие есть подходы?



(a) Data Parallelism



(b) Model Parallelism



(c) Layer Pipelining

Что расскажем мы?

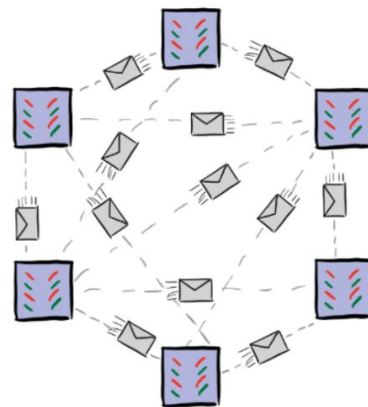
- Intro to Distributed Systems & Data Parallelism by me
- Model parallelism & Layer Pipelining by Alexander
- Practice by Sergey

Intro to Distributed Systems

полный курс смотрите у ребят с PC

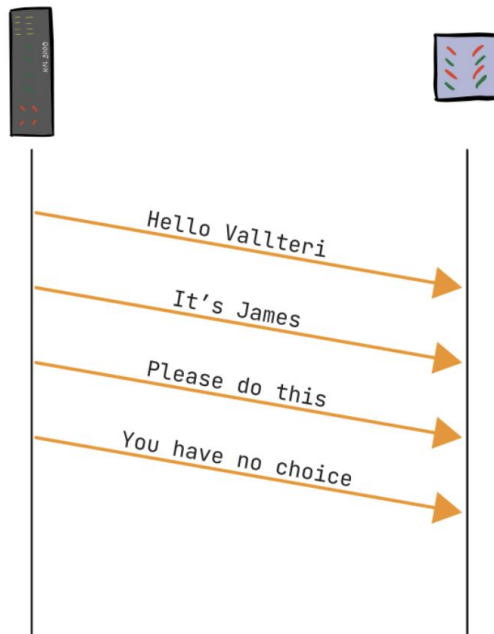
Распределенная система – это ...

С программной точки зрения: совокупность независимых процессов, взаимодействующих посредством передачи сообщений



ОСНОВЫ

- Процессы выполняются на различных узлах
- Каждый процесс имеет собственное состояние
- Процессы не имеют прямого доступа к состояниям других процессов



Data Parallelism

вернемся к сеткам

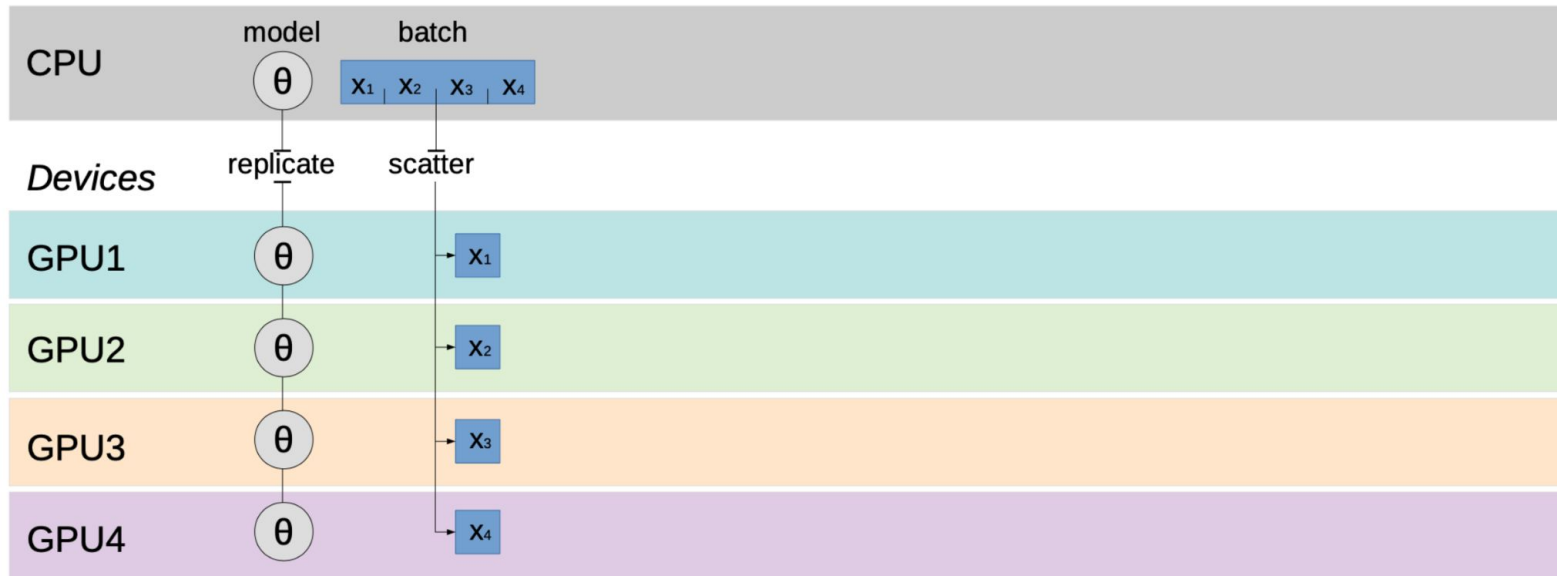
Проблемы?

- Слишком долго ждать
- Данные не помещаются в RAM

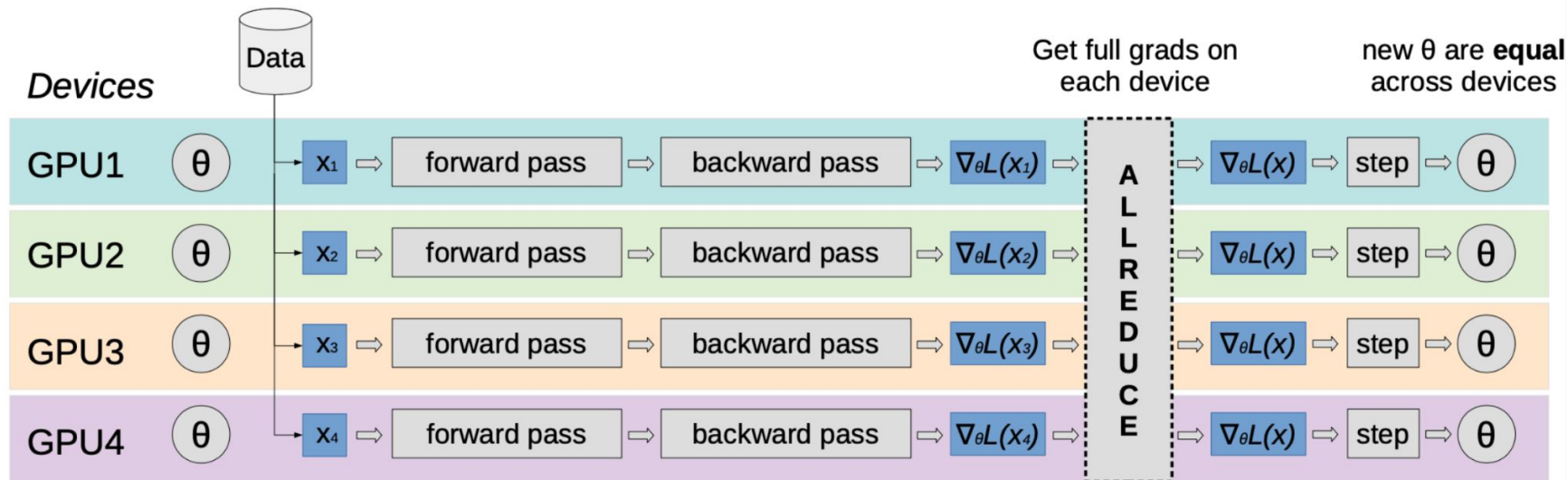


Что происходит?

- Знаем, что модель полностью помещается в GPU
- Данные разбиваются на куски и распределяются по процессам

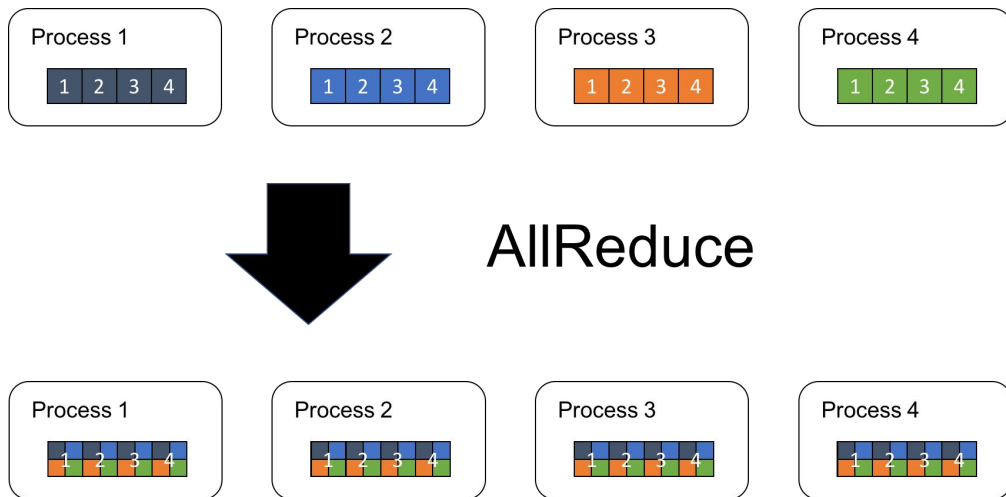


Процесс работы (ожидание)



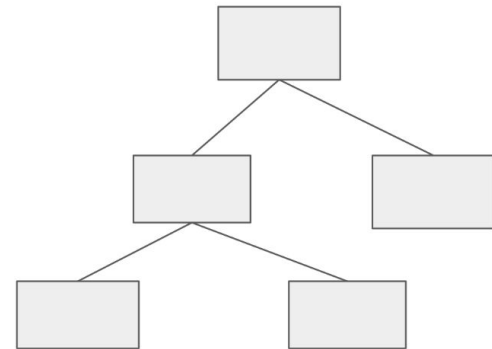
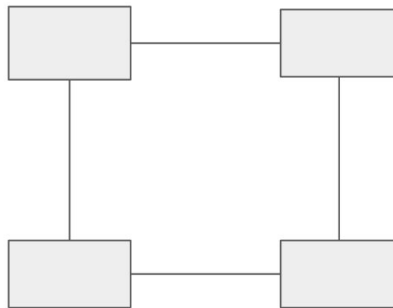
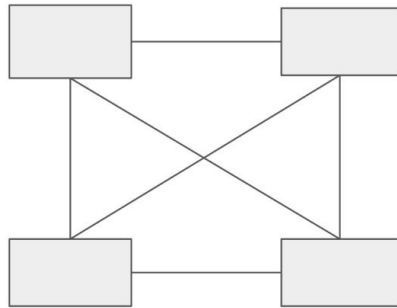
AllReduce

- Процесс обмена градиентами между процессами (агрегируем векторы)
- На всех узлах будут одинаковые данные (модели)



Как? – Способы коммуникации

Топология – это конфигурация графа, вершинам которого соответствуют конечные узлы сети



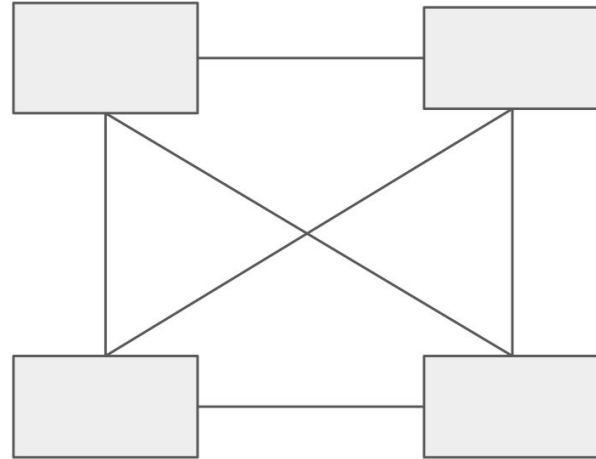
Каждый с каждым

Плюсы:

- Каждый отправляет данные каждому узлу, все очень просто

Минусы:

- Это точно будет долго $O(n^2)$



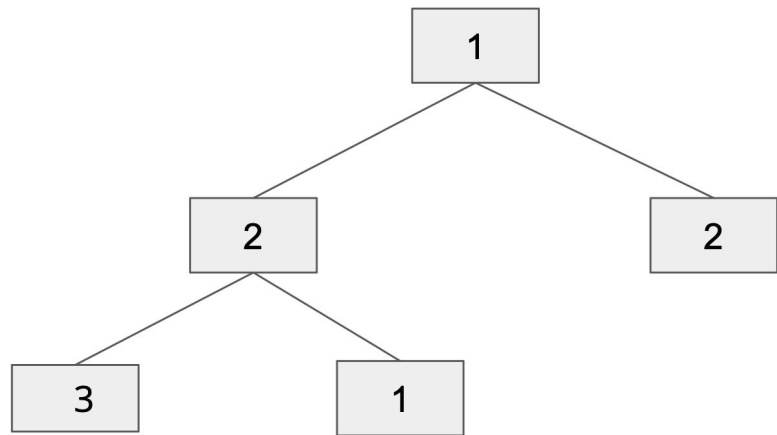
По дереву

Плюсы:

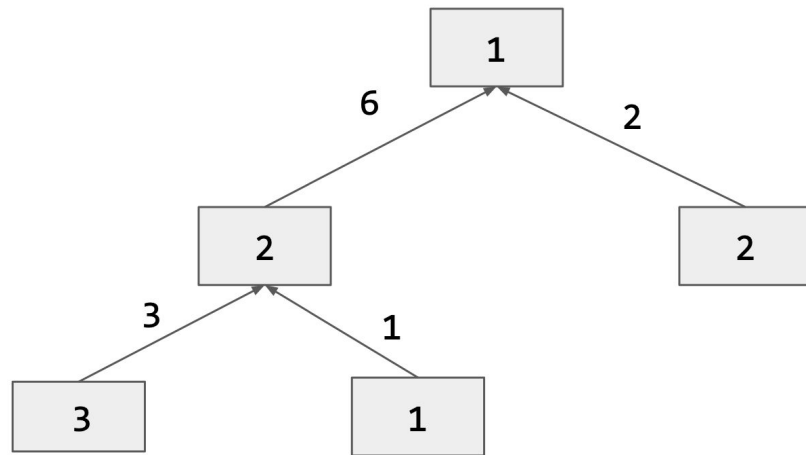
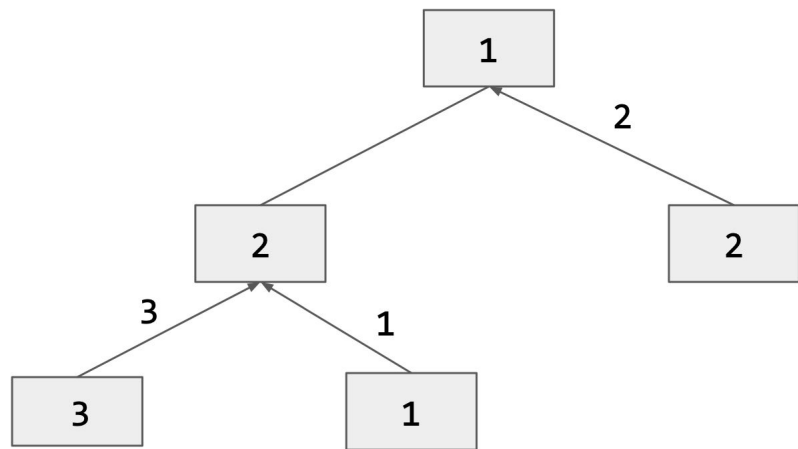
- Быстро $O(n)$

Минусы

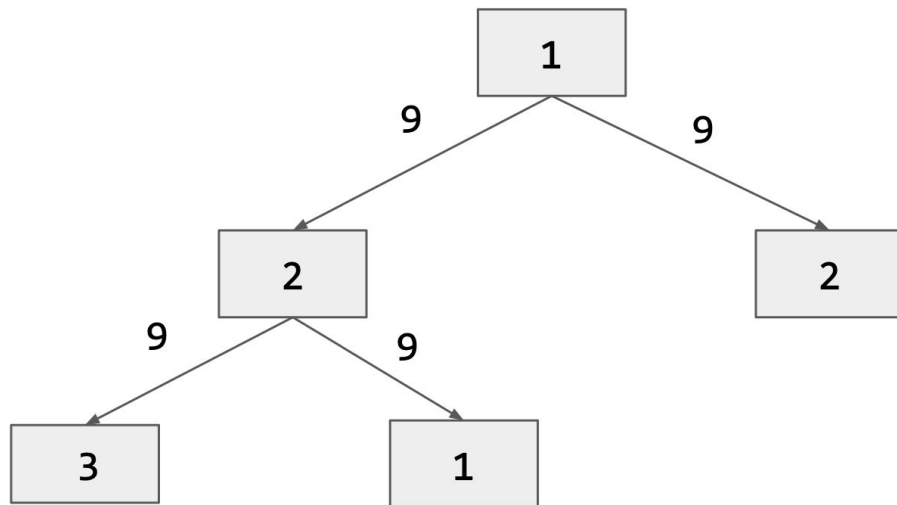
- Информация с листьев теряется если произошло отключение в сети



По дереву



По дереву



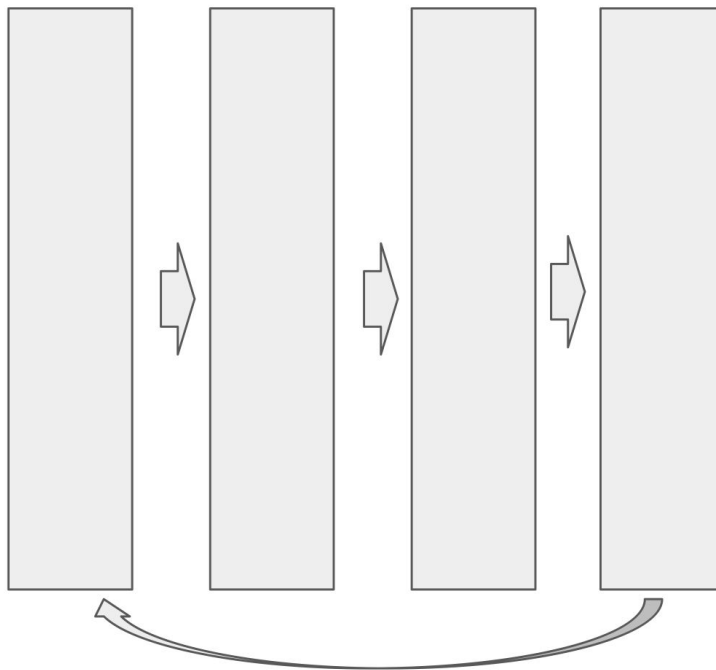
Кольцо

Плюсы:

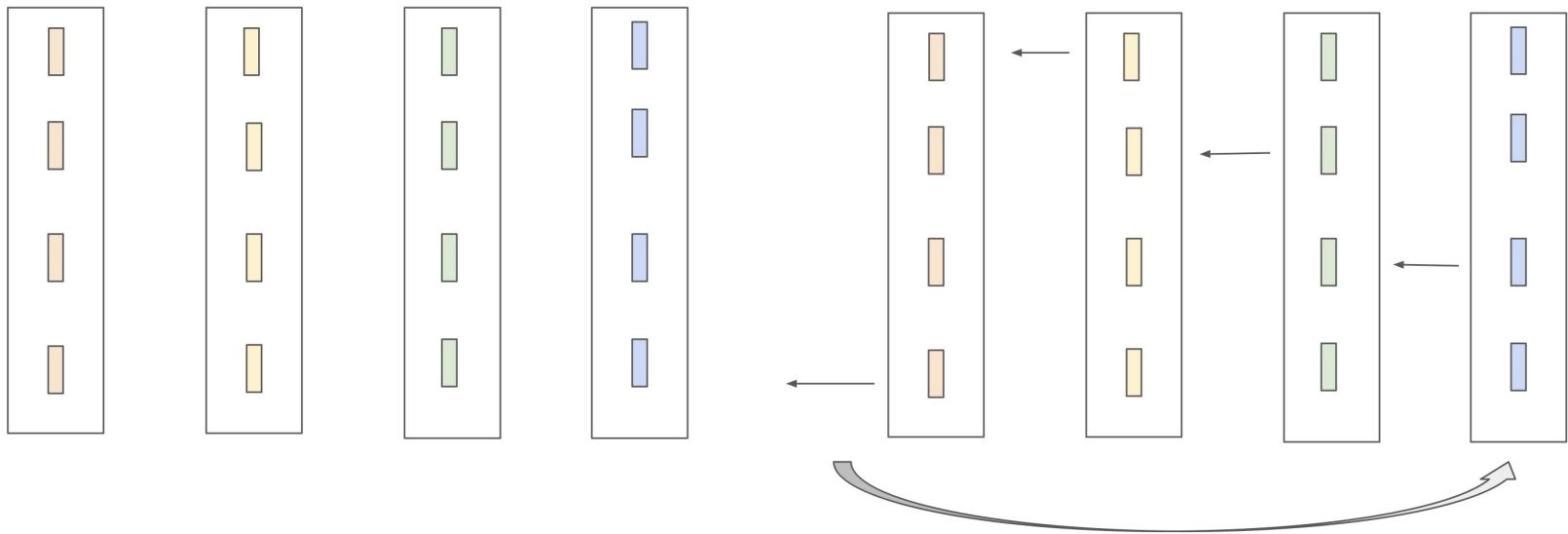
- Быстро $O(n)$

Минусы:

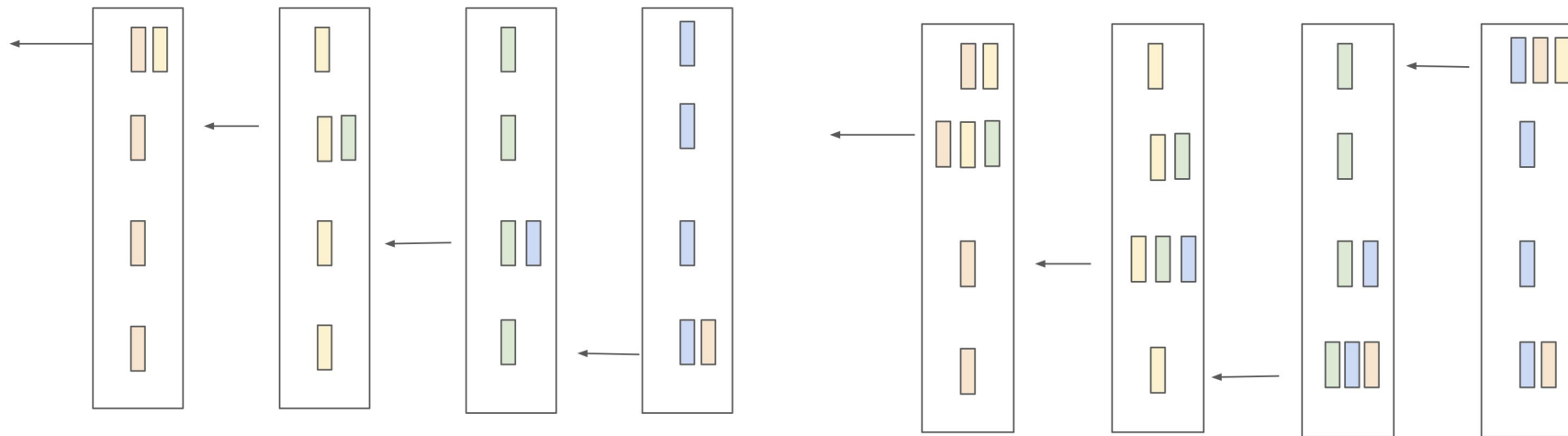
- Я не нашла



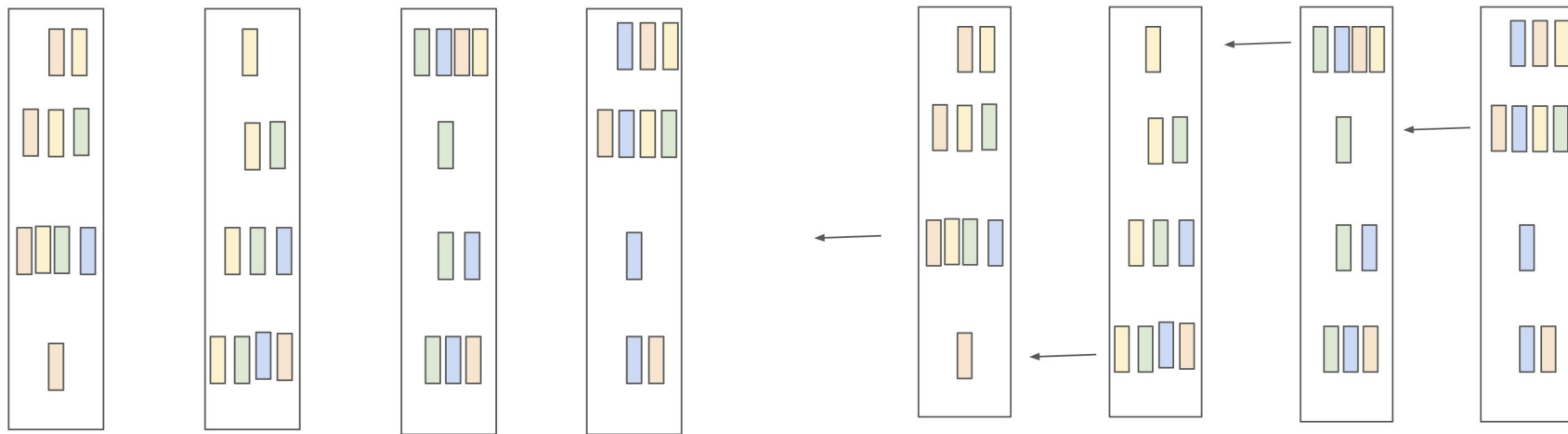
Кольцо



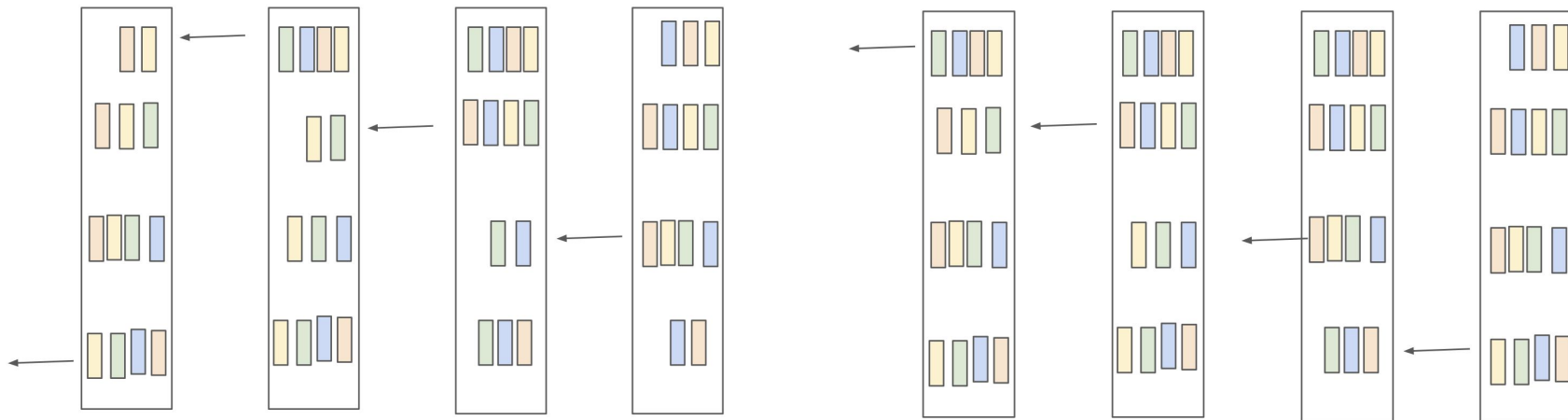
Кольцо



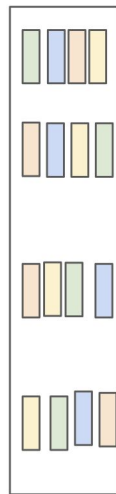
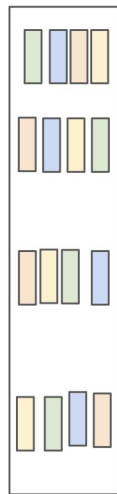
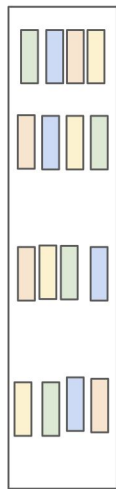
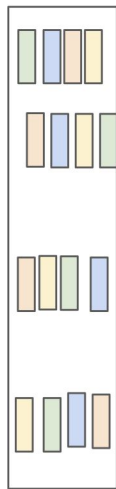
Кольцо



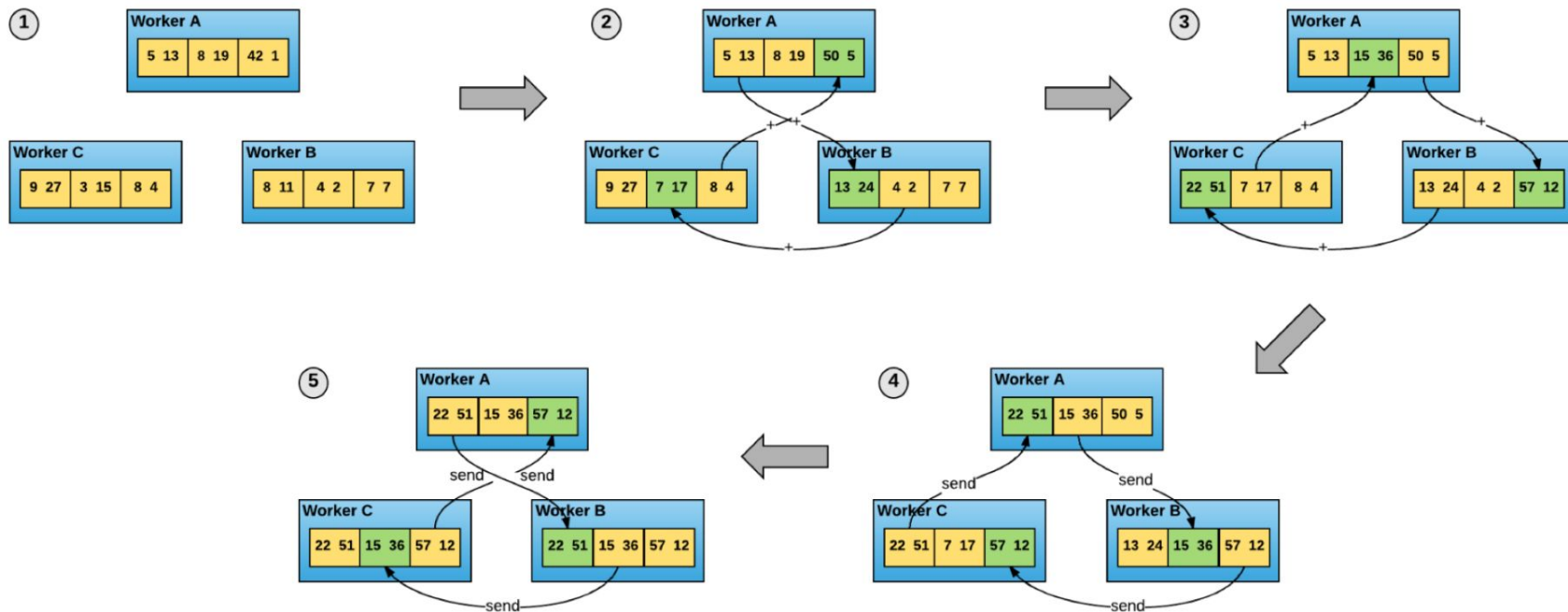
Кольцо



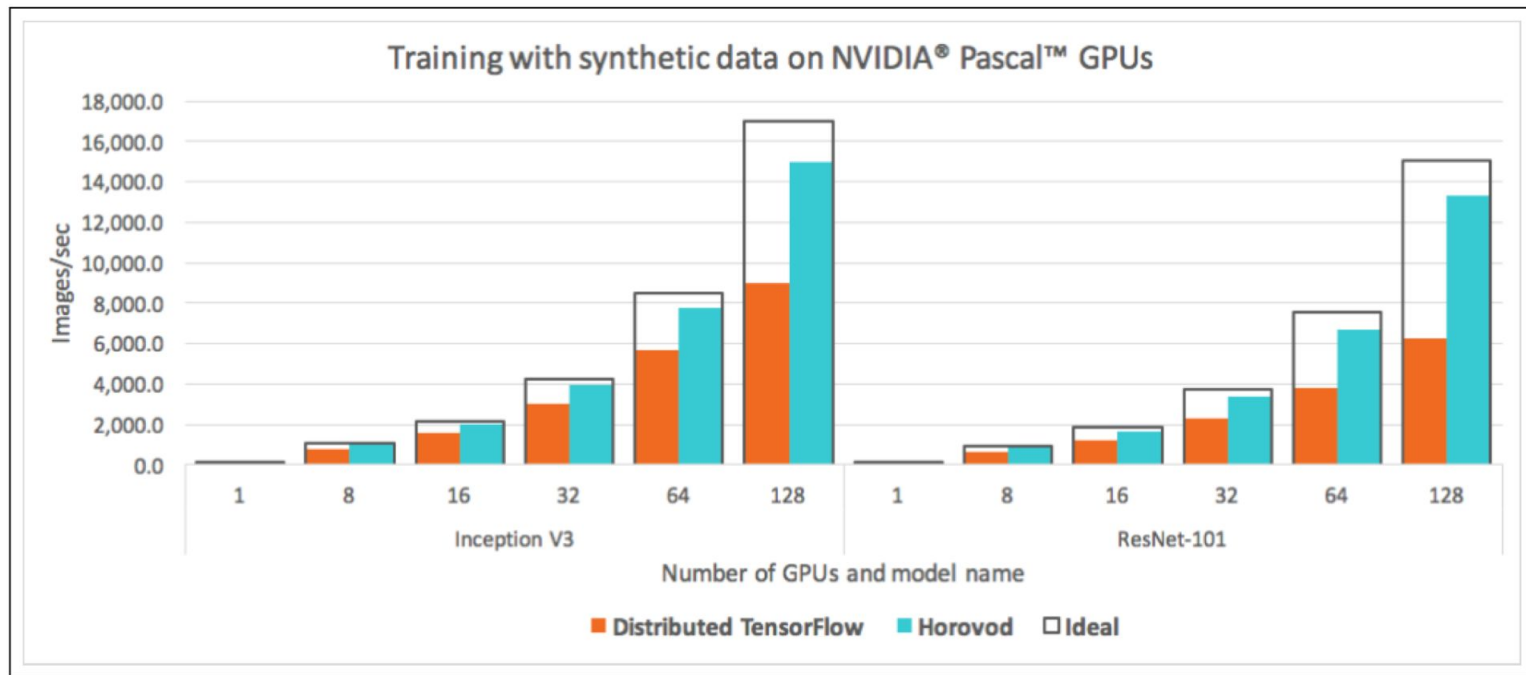
Кольцо



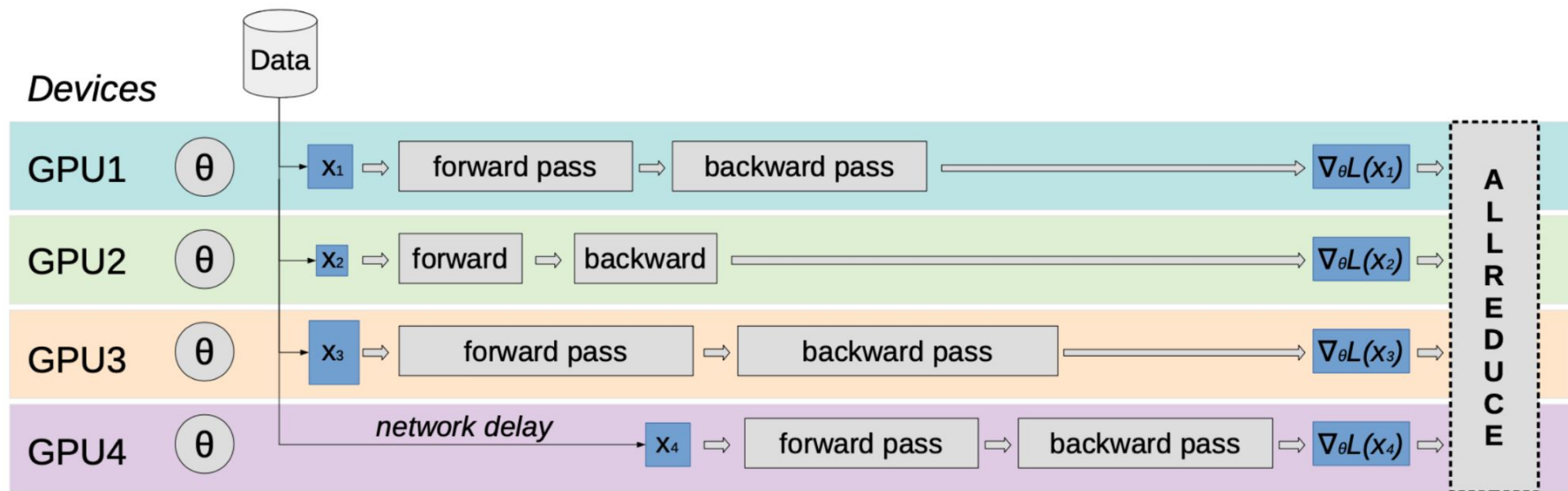
Horovod – Uber 2018



Horovod – Uber 2018



Процесс работы (реальность)

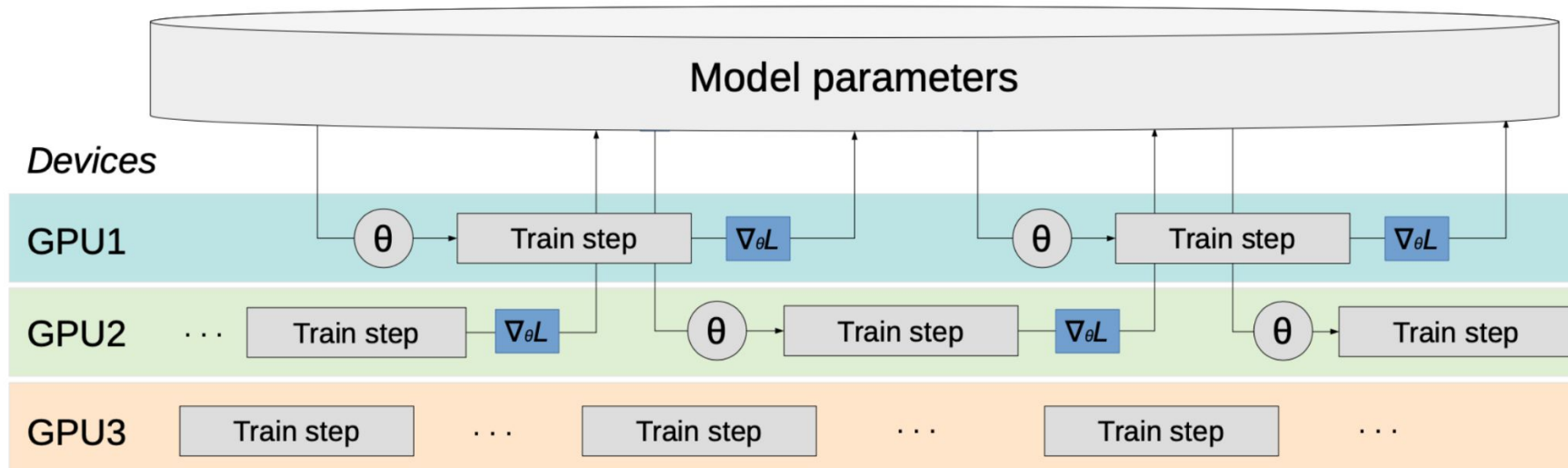


А что если не ждать?

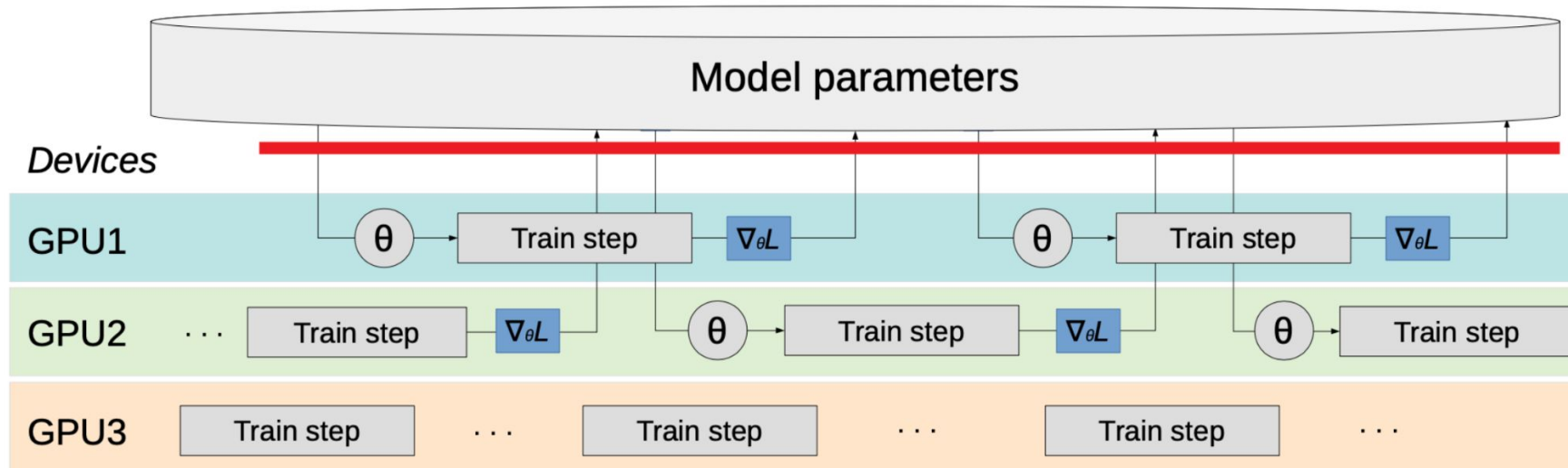
- Синхронная модель взаимодействия – ждем все узлы и только потом обновляем модель (обсуждали только что)
- Асинхронная модель – каждый узел получает обновленные значения градиента сразу и обновляет модель независимо (обсудим сейчас)

Parameter server

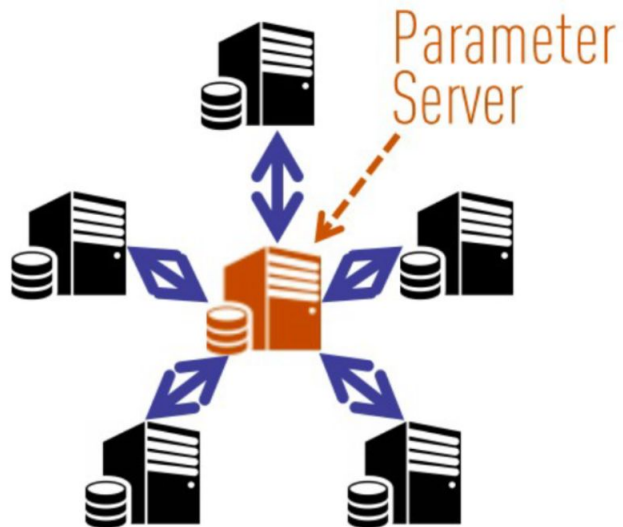
Idea: remove synchronization step altogether, use parameter server



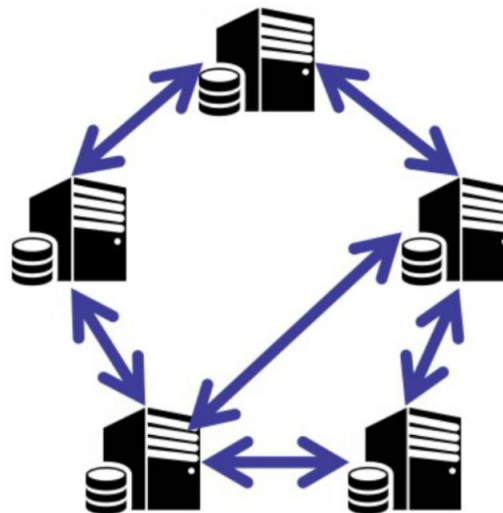
Problems? We have some



Итого



(a) Centralized Topology



(b) Decentralized Topology

Проблемы?



Источники

2018 Horovod Uber

- <https://arxiv.org/pdf/1802.05799.pdf>,
- <https://github.com/horovod/horovod>,
- <https://www.youtube.com/watch?v=SphfeTI70MI>

2018 An In-Depth Concurrency Analysis

- <https://arxiv.org/pdf/1802.09941.pdf>
- <https://www.youtube.com/watch?v=xtxxLWZznBI>

Курс ШАДа

- https://github.com/yandexdataschool/dlatscale_draft

Не источник, но посмотреть можно:

2019 TF replicator DeepMind

- <https://arxiv.org/pdf/1902.00465.pdf>

Me: *uses machine learning*

Machine: *learns*

Me:

