

Распределенное обучение нейросетей

Ваньков Тимур, БПМИ-181

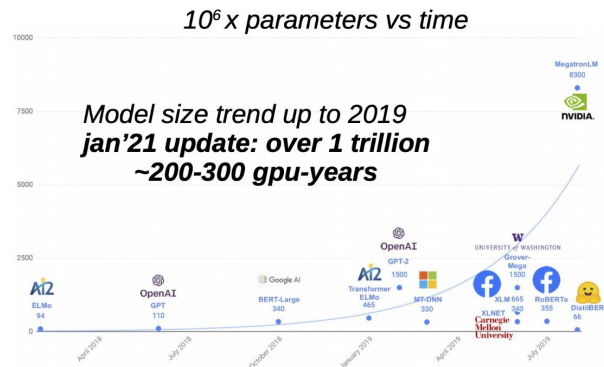
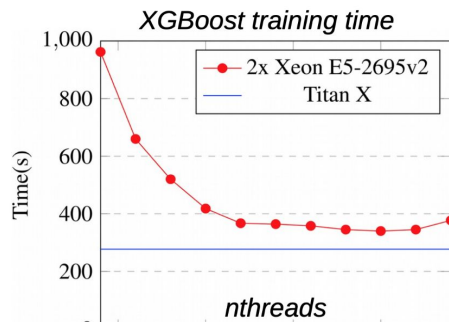
План презентации

1. Распределенное обучение
 - 1.1. Что это и зачем?
 - 1.2. Основы
2. Data parallelism
 - 2.1. Что это?
 - 2.2. Применения
 - 2.3. Реальность
 - 2.4. Оптимизации
 - 2.5. Проблема
 - 2.6. Решение
3. Model parallelism
 - 3.1. Коротко
 - 3.2. PipeDream
4. Federated training
 - 4.1. Что это?
 - 4.2. Итерации обучения модели
 - 4.3. Применения

Распределенное обучение

Что это и зачем?

1. Метод обучения моделей
2. Для ускорения обучения
3. Когда данные не помещаются на 1 сервере
4. Когда модель не помещается на 1 сервере



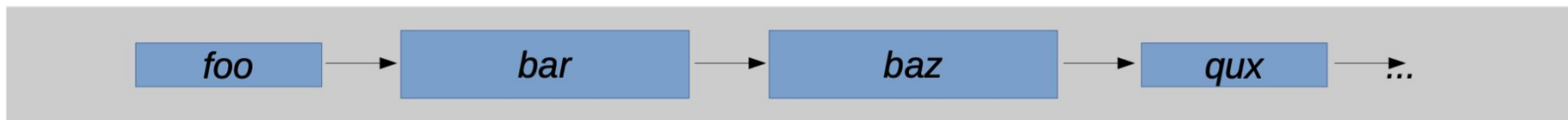
BERT-Large Training Times on GPUs

Time	System	Number of Nodes	Number of V100 GPUs
47 min	DGX SuperPOD	92 x DGX-2H	1,472
67 min	DGX SuperPOD	64 x DGX-2H	1,024
236 min	DGX SuperPOD	16 x DGX-2H	256

(single V100 – over 2 weeks)

ОСНОВЫ

Rules: Process



Process:

- Runs some code
- Has some memory

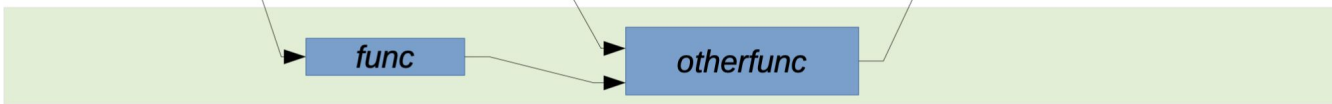
ОСНОВЫ

Rules: Channel / Pipe

Process A:



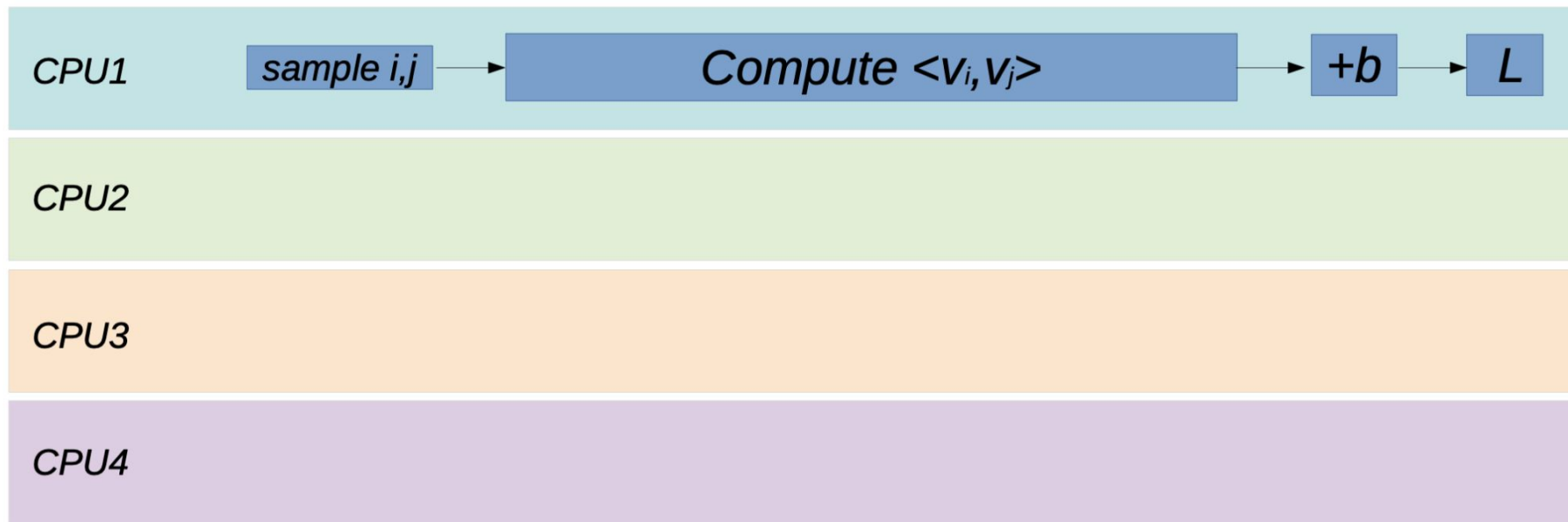
Process B:



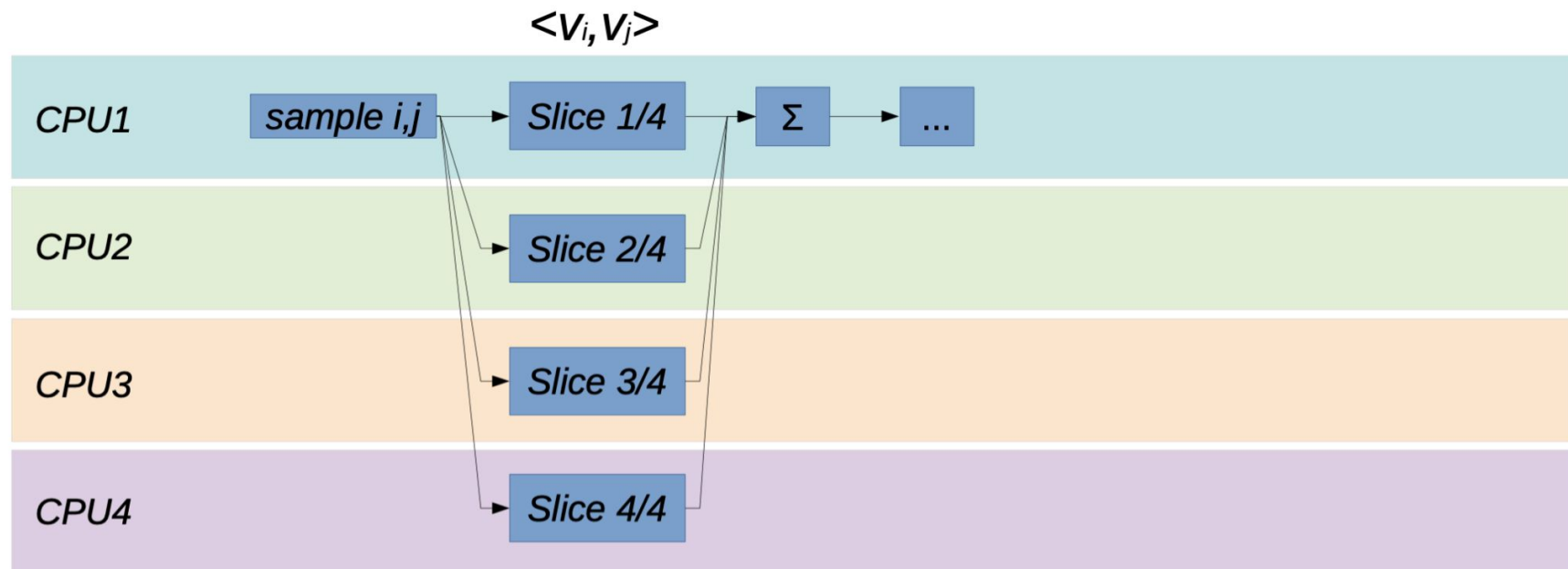
Channel (pipe):

- Communication in $O(\text{message size})$
- Asynchronous read/write

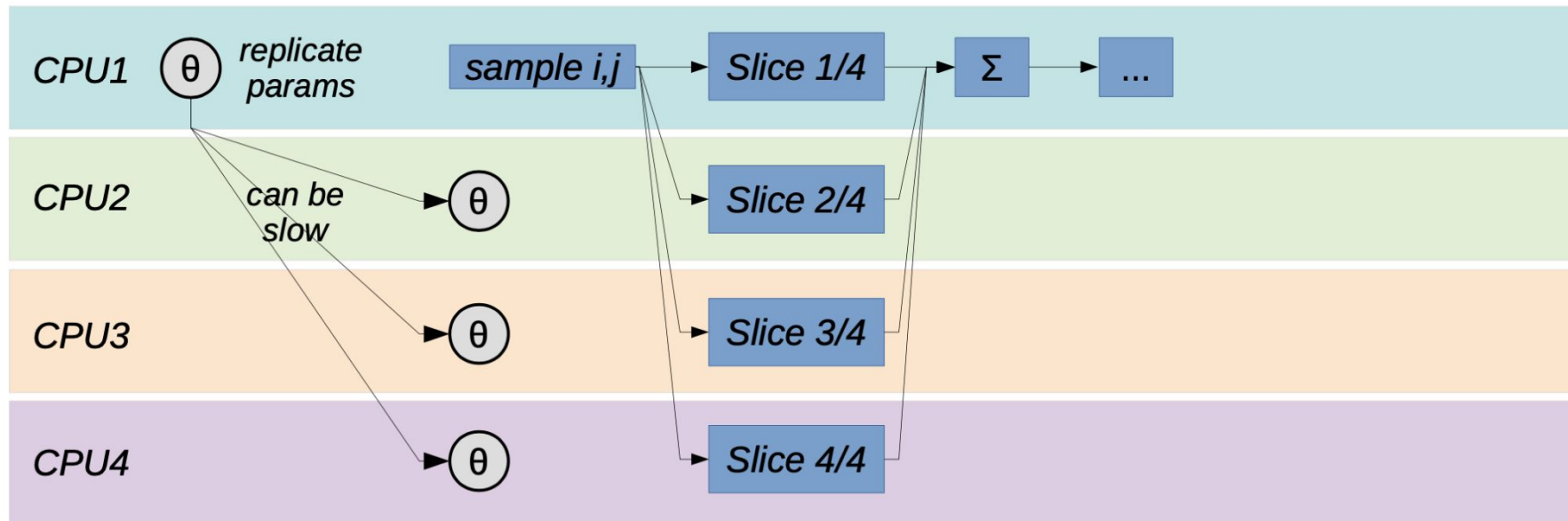
ОСНОВЫ



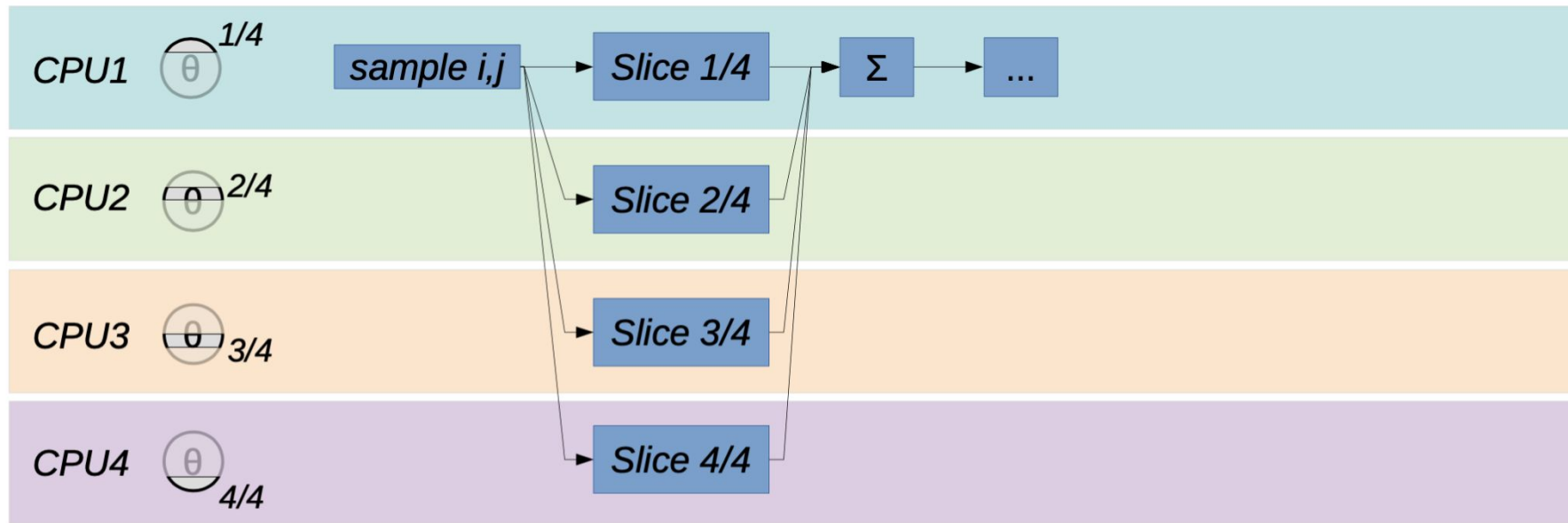
ОСНОВЫ



ОСНОВЫ



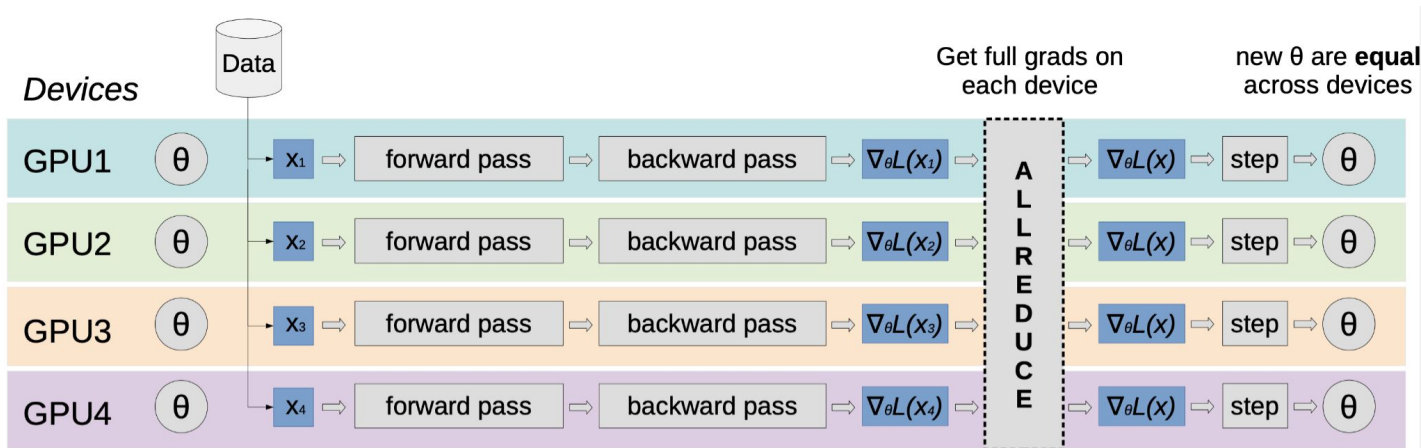
ОСНОВЫ



Data parallelism

Что это?

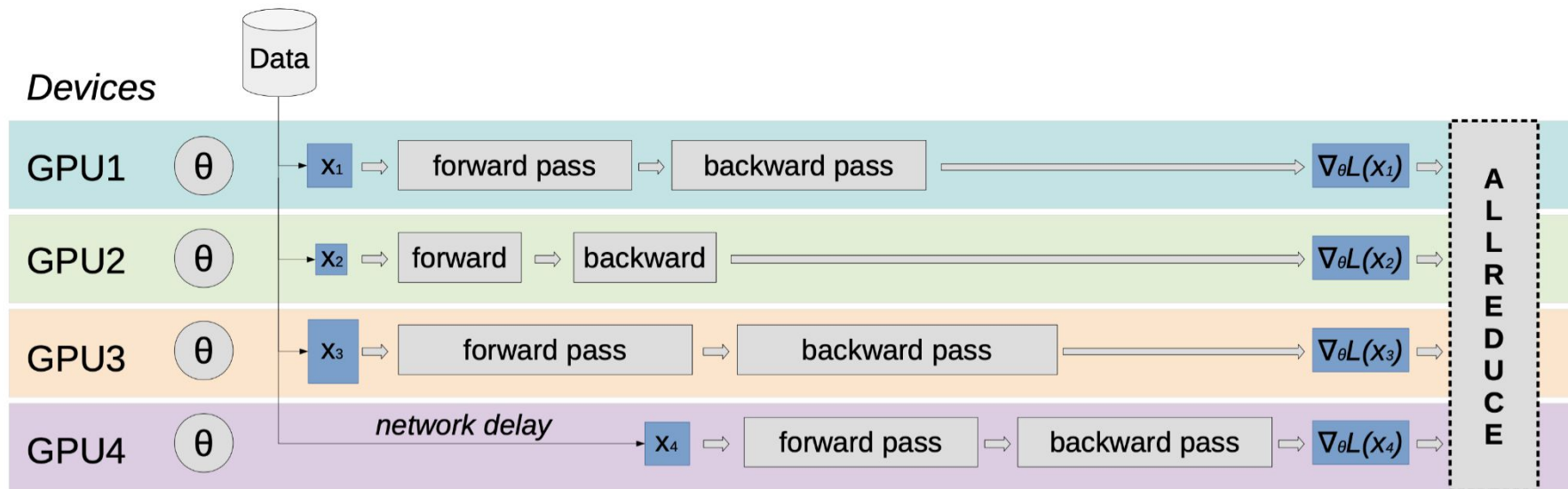
1. Метод распределенного машинного обучения
2. Модель одна, данные одни
3. Данные разбиваются на куски и распределяются по разным запущенным процессам



Применения

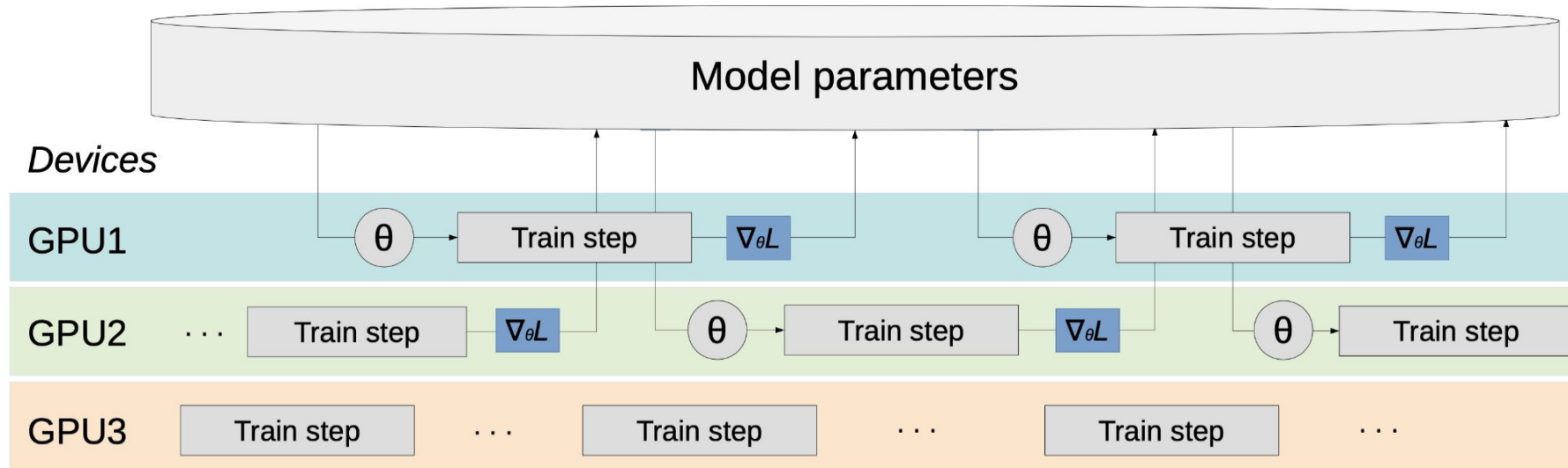
1. Данных очень много для 1 машины
2. Обучение очень долгое
3. Для балансировки нагрузки в распределенных системах

Реальность

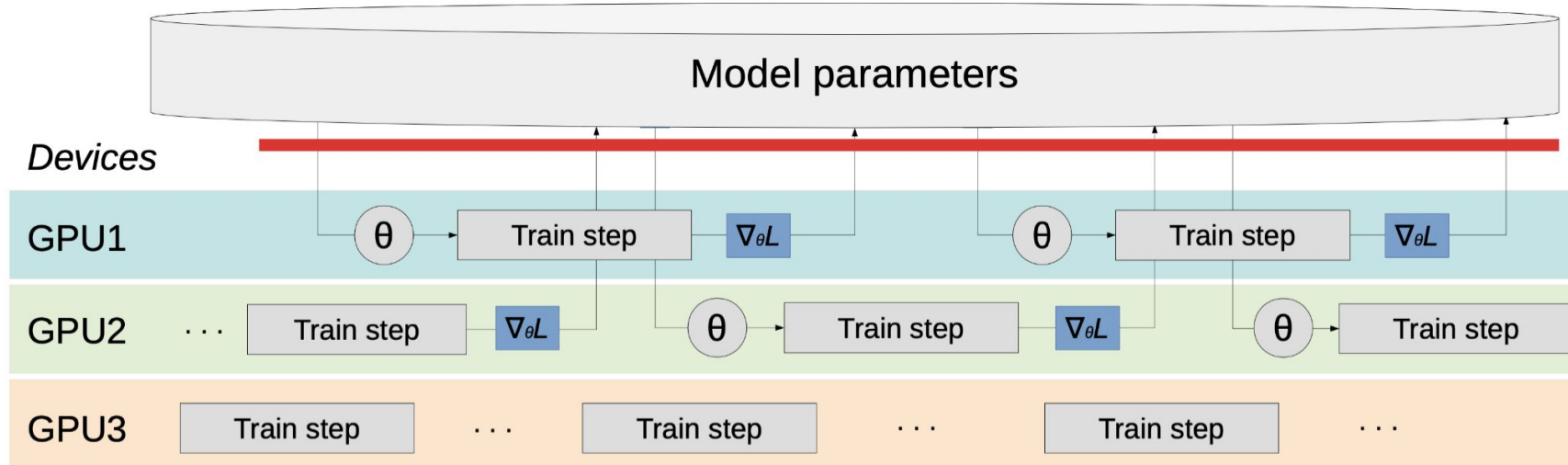


Оптимизации

Idea: remove synchronization step altogether, use parameter server

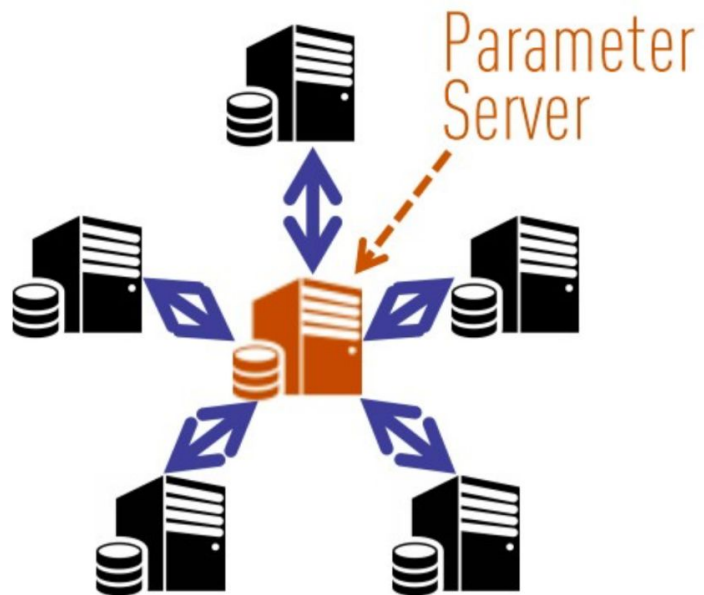


Проблема

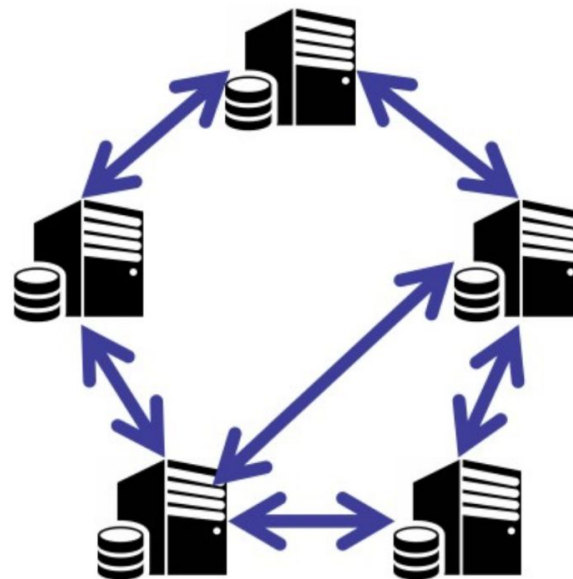


Problem: parameter servers need to ingest tons of data over training

Решение



(a) Centralized Topology

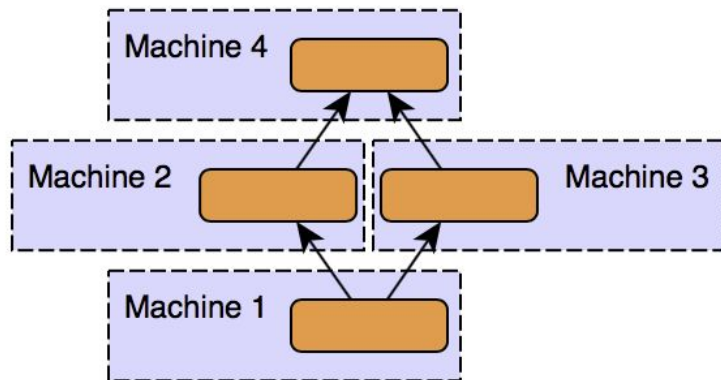


(b) Decentralized Topology

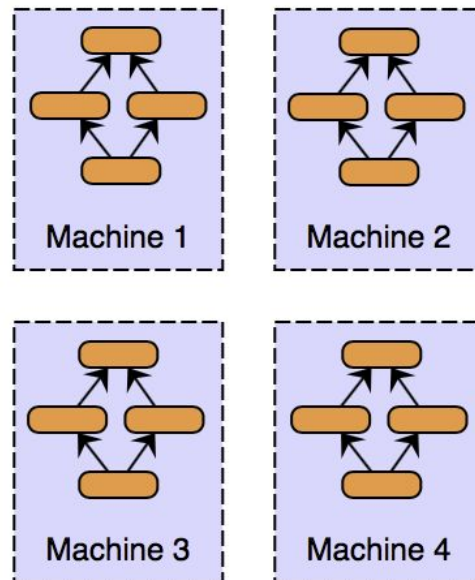
Model parallelism

Коротко

Model Parallelism



Data Parallelism



Коротко

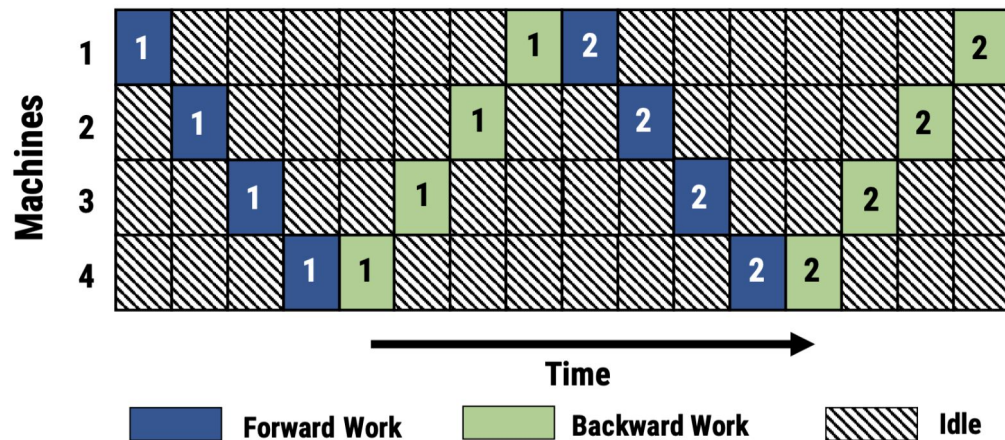


Figure 3: Model parallel training with 4 machines. Numbers indicate minibatch ID. For simplicity, here we assume that forward and backward work in every stage takes one time unit, and communicating activations across machines has no overhead.

PipeDream

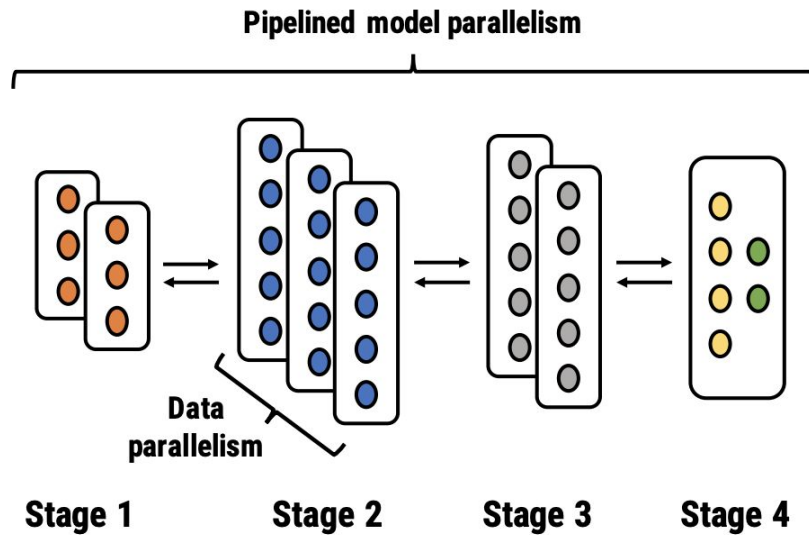


Figure 6: Pipeline Parallel training in PipeDream combines pipelining, model- and data-parallel training.

PipeDream

1. Профилировка:

- a. Для каждого слоя считаются 3 значения:
 - i. TI - Общее время для forward и backward проходов
 - ii. al - размер выхода функции активации
 - iii. wl - размер параметров для слоя

2. Разделение:

- a. Разделение слоев на этапы
- b. Коэффициент репликация для каждого этапа
- c. Оптимальное количество минибатчей для обеспечения занятости конвейера

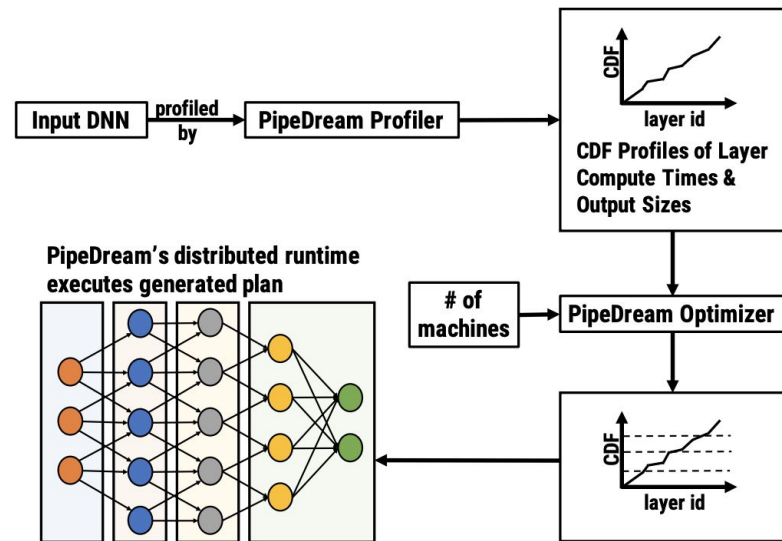


Figure 7: PipeDream's automated mechanism to partition DNN layers into stages. PipeDream first profiles the input DNN, to get estimates for each layer's compute time and output size. Using these estimates, PipeDream's optimizer partitions layers across available machines.

PipeDream

1. Алгоритм разделения пытается минимизировать общее время обучения конвейера ~ минимизация времени, затрачиваемого на самую медленную стадию конвейера.
2. Конвейер состоит из маленьких конвейеров - для каждого решаем задачу - динамическое программирование на помощь.

Пусть $A(j, m)$ - время затрачиваемое самым медленным этапом в оптимальном конвейере между 1 и j - ом слоем используя m машин. Цель алгоритма найти $A(N, M)$ и соответствующее распределение. Пусть $T(i \rightarrow j, m)$ - время затрачиваемое этапом состоящим из слоев от i до j реплицированным на m машин:

$$T(i \rightarrow j, m) = \frac{1}{m} \max \left(\sum_{l=i}^j T_l, \sum_{l=i}^j W_l^m \right)$$

PipeDream

1. Case 1: оптимальный конвейер состоит из одного этапа реплицированного на m машин, тогда: $A(j, m) = T(1 \rightarrow j, m)$
2. Case 2: оптимальный конвейер состоит из более одного этапа. Тогда он может быть разбит на под-конвейеры проходящие от 1 к i ому слою реплицированный на $m - m'$ машин, за которым следует один этап состоящий из слоев от $i + 1$ до j , тогда мы получим:

$$A(j, m) = \min_{1 \leq i < j} \min_{1 \leq m' < m} \max \begin{cases} A(i, m - m') \\ 2 \cdot C_i \\ T(i + 1 \rightarrow j, m') \end{cases}$$

PipeDream

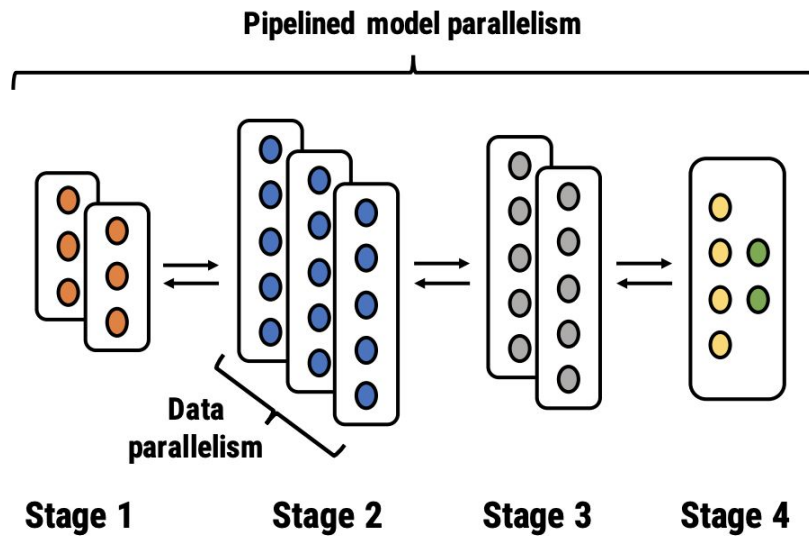


Figure 6: Pipeline Parallel training in PipeDream combines pipelining, model- and data-parallel training.

PipeDream

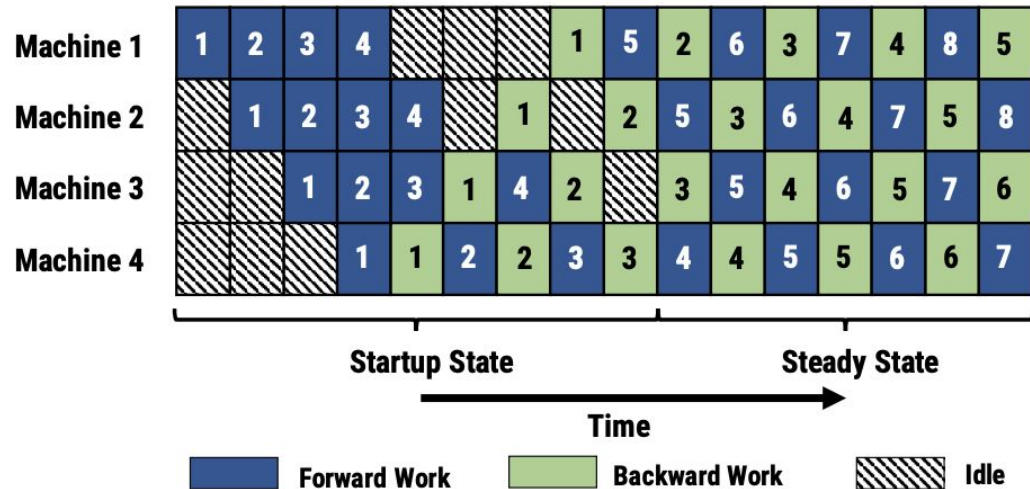


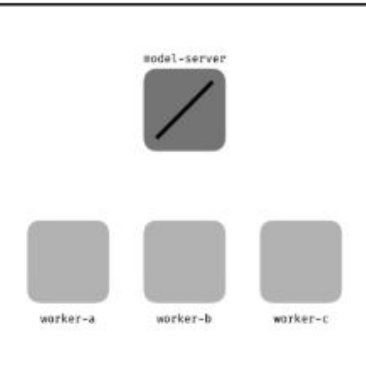
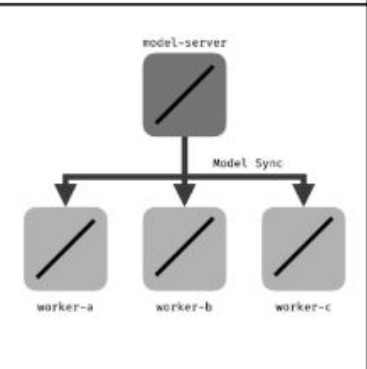
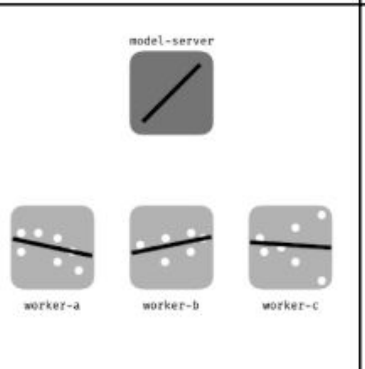
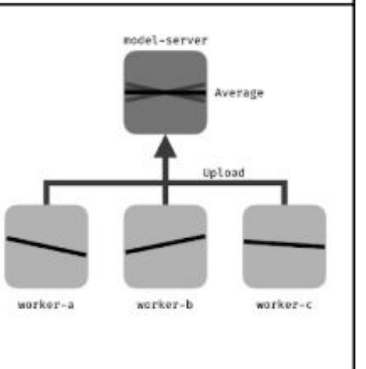
Figure 8: An example pipeline with 4 machines, showing startup and steady states.

Federated training

Что это?

1. Централизованное федеративное обучение
2. Децентрализованное федеративное
3. Гетерогенное федеративное обучение

Итерации обучения модели

Step 1	Step 2	Step 3	Step 4
			
Central server chooses a statistical model to be trained	Central server transmits the initial model to several nodes	Nodes train the model locally with their own data	Central server pools model results and generate one global mode without accessing any data

Применения

1. Беспилотные автомобили
2. Медицина: нода - данные из клиники
3. FLoC - Federated Learning of Cohorts
4. С появлением Apple M1 - огромный простор

ИСТОЧНИКИ

1. <https://arxiv.org/pdf/1806.03377.pdf>
2. https://github.com/yandexdataschool/dlatscale_draft
3. <https://arxiv.org/pdf/1802.05799.pdf>
4. <https://arxiv.org/pdf/1811.06965.pdf>