Alibaba Cloud | Worldwide Cloud Services Partner

© Copyright by Alibaba Cloud All rights reserved

Whale: Efficient Giant Model Training over Heterogeneous GPUs

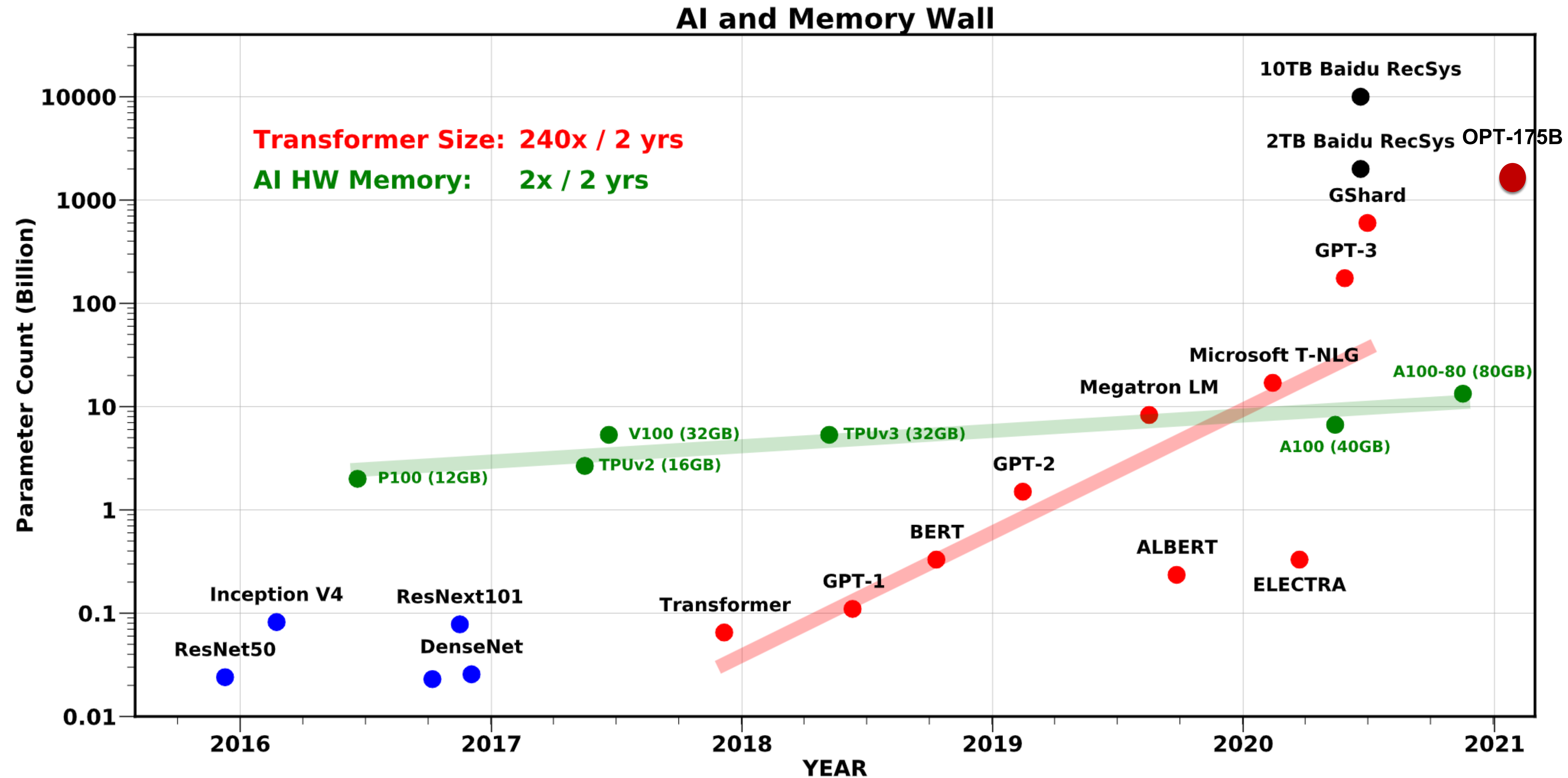
Xianyan Jia, Le Jiang, Ang Wang, Wencong Xiao, Ziji Shi, Jie Zhang, Xinyuan Li, Langshi Chen, Yong Li, Zhen Zheng, Xiaoyong Liu, Wei Lin

Alibaba Group

07/12/2022

Model-Size Increasing

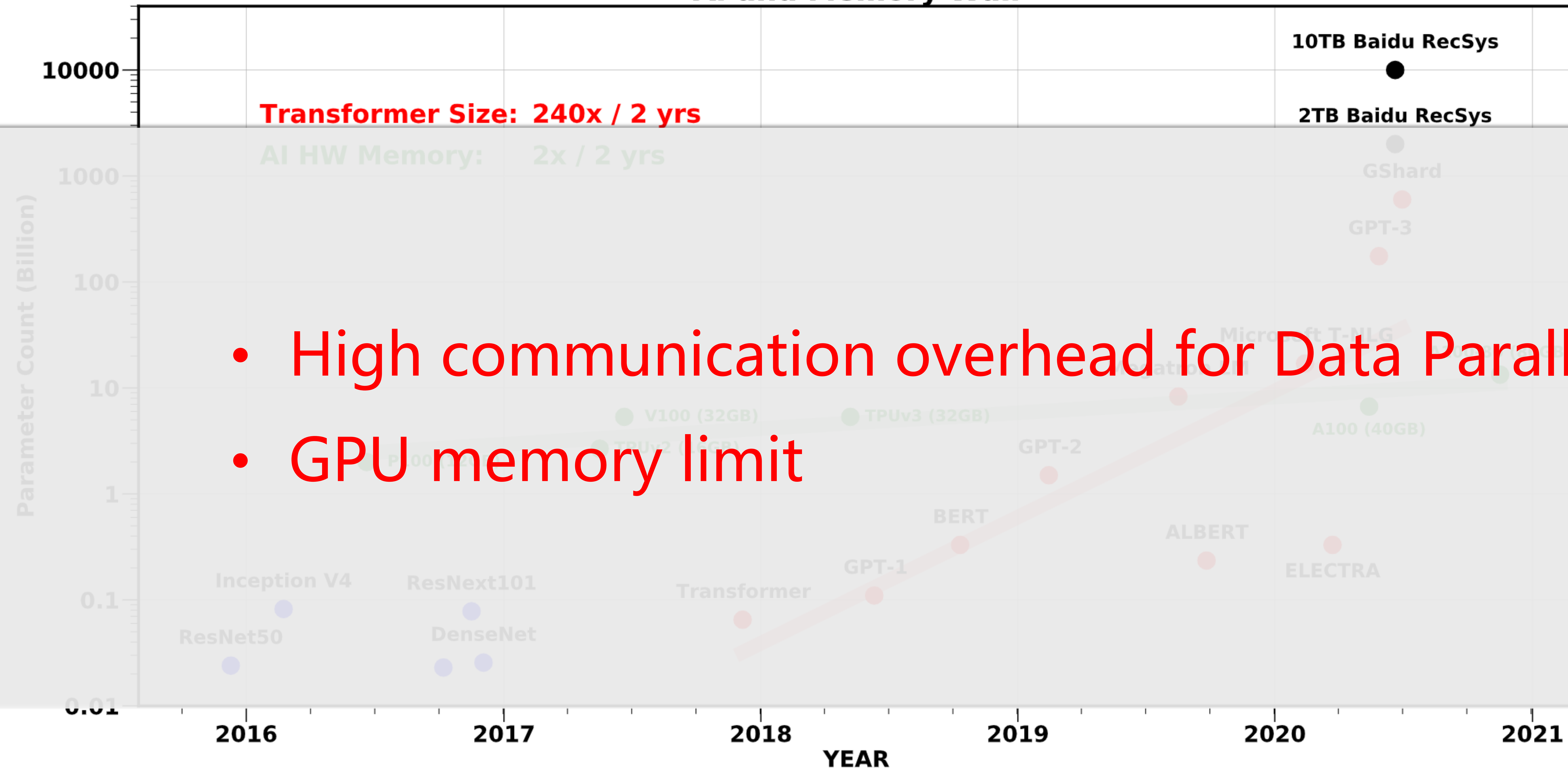
Alibaba Cloud
Worldwide Cloud Services Partner



AI and Memory Wall: https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

Memory & Bandwidth Wall

Alibaba Cloud
Worldwide Cloud Services Partner

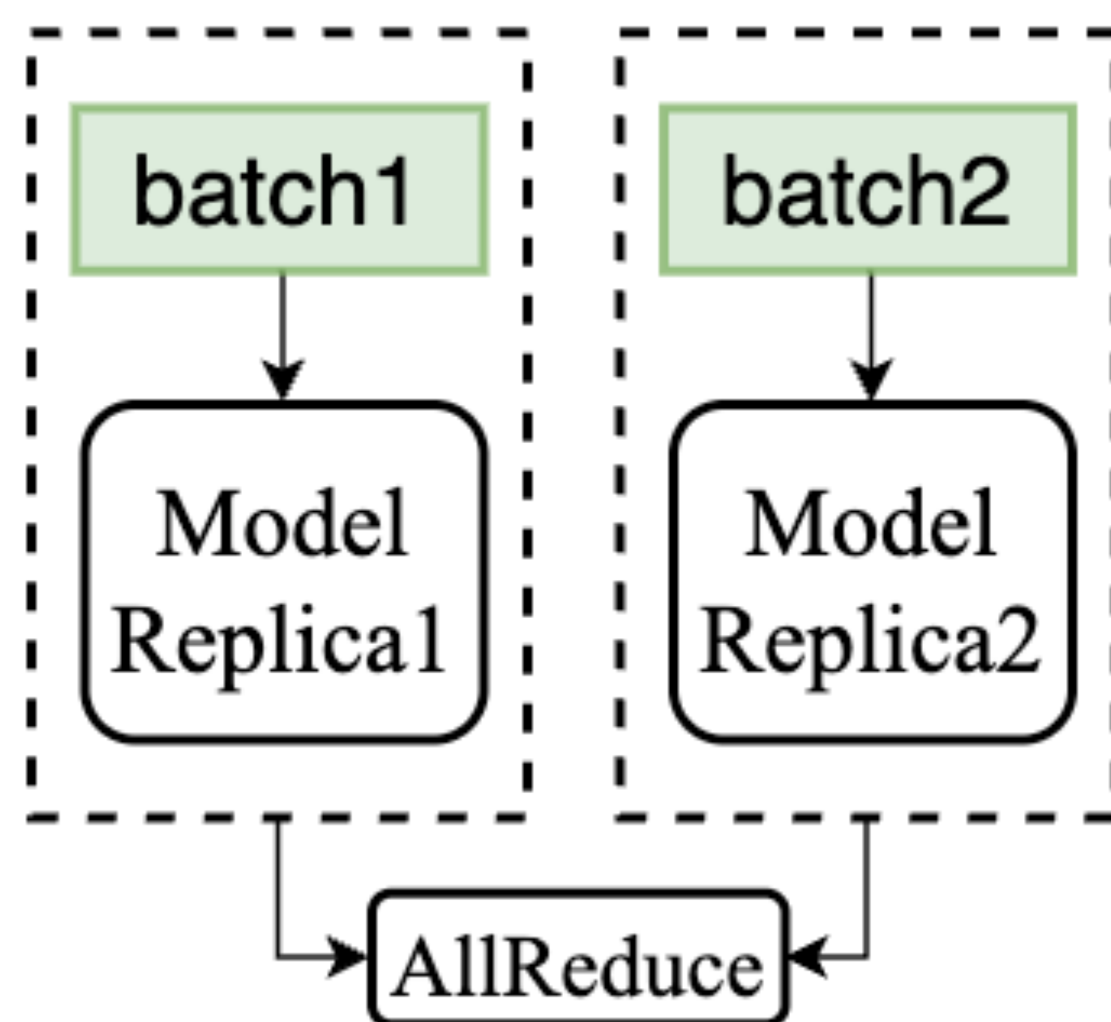


- High communication overhead for Data Parallel
- GPU memory limit

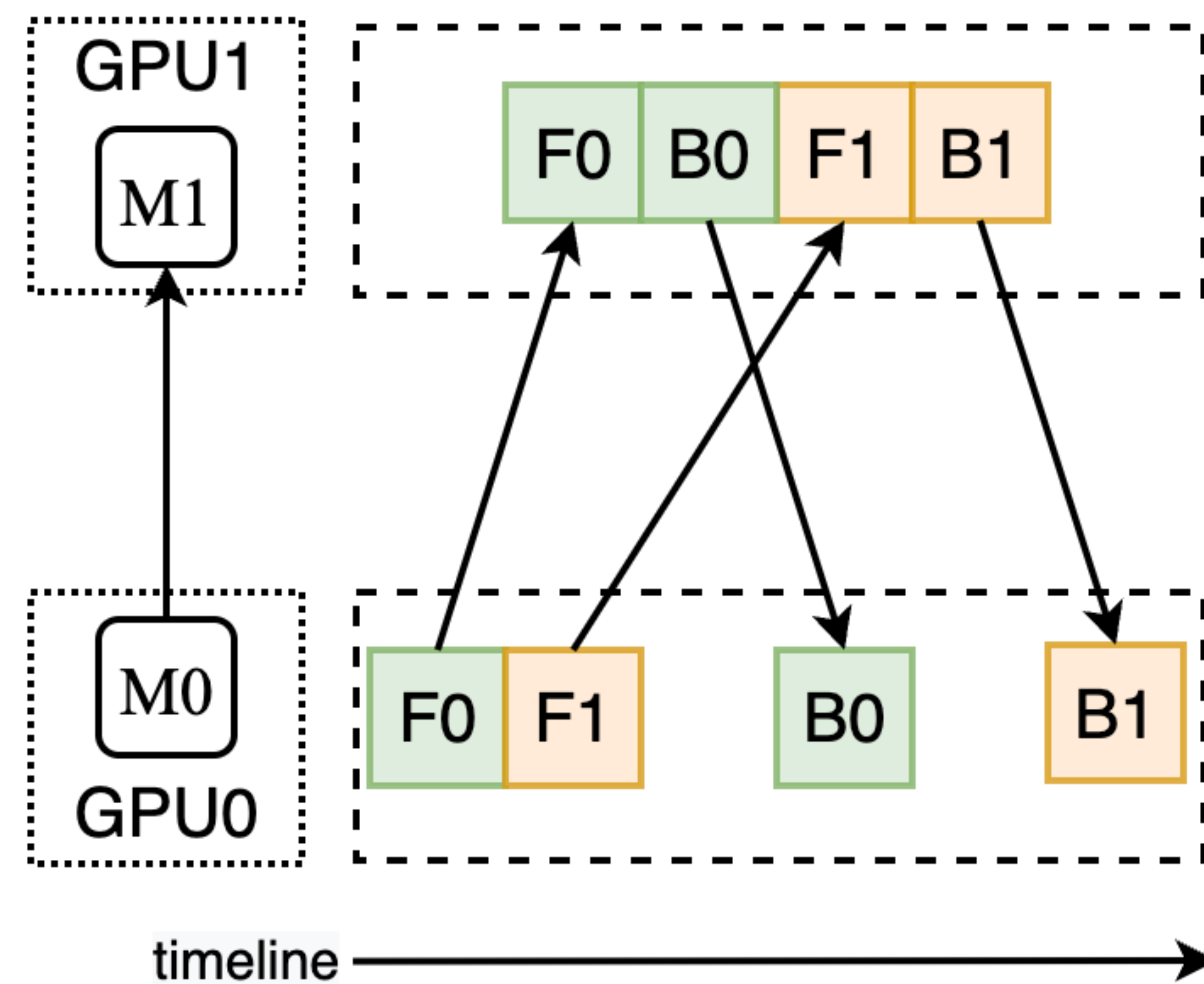
AI and Memory Wall: https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8

Distributed Training Strategies

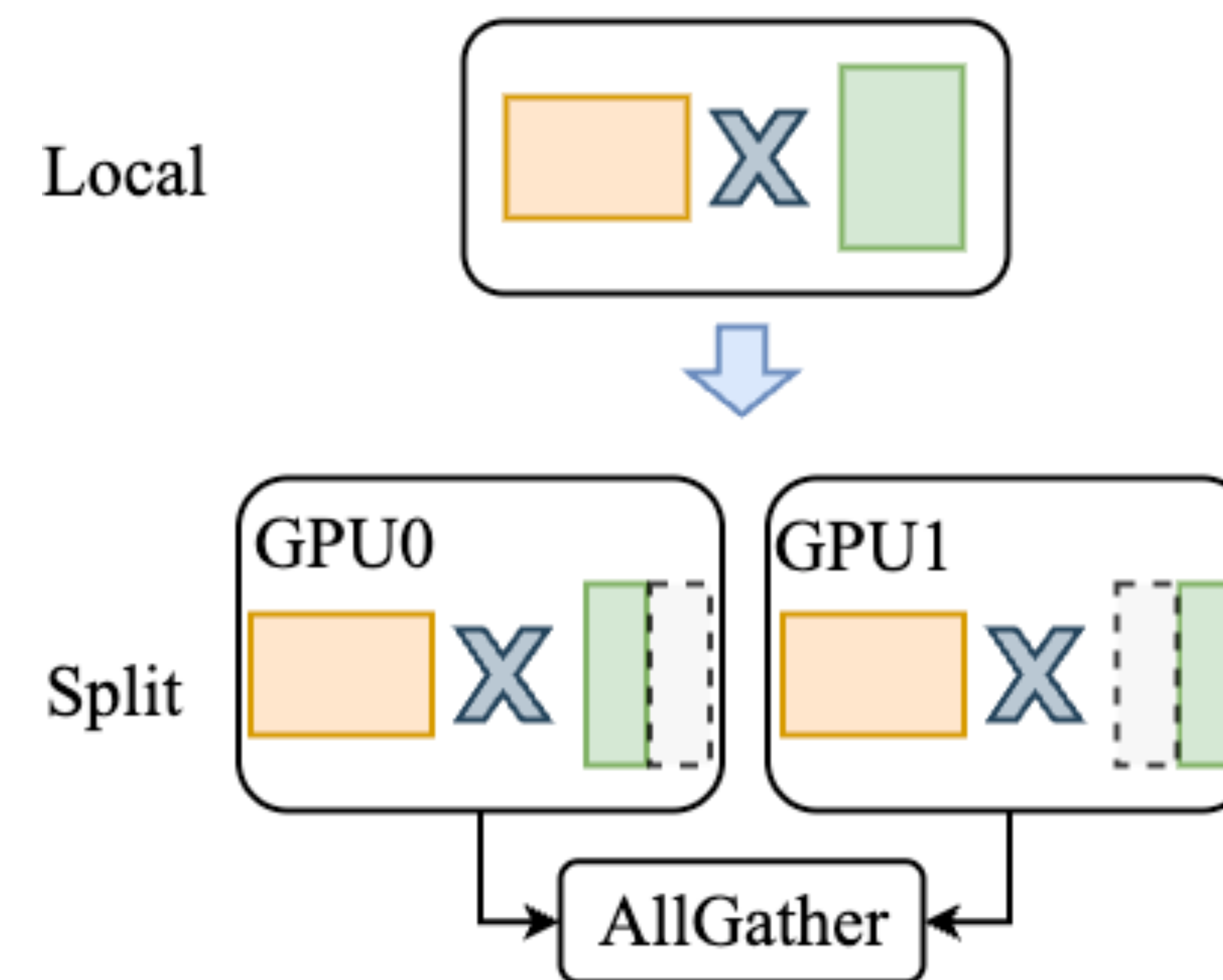
Alibaba Cloud
Worldwide Cloud Services Partner



Data Parallelism



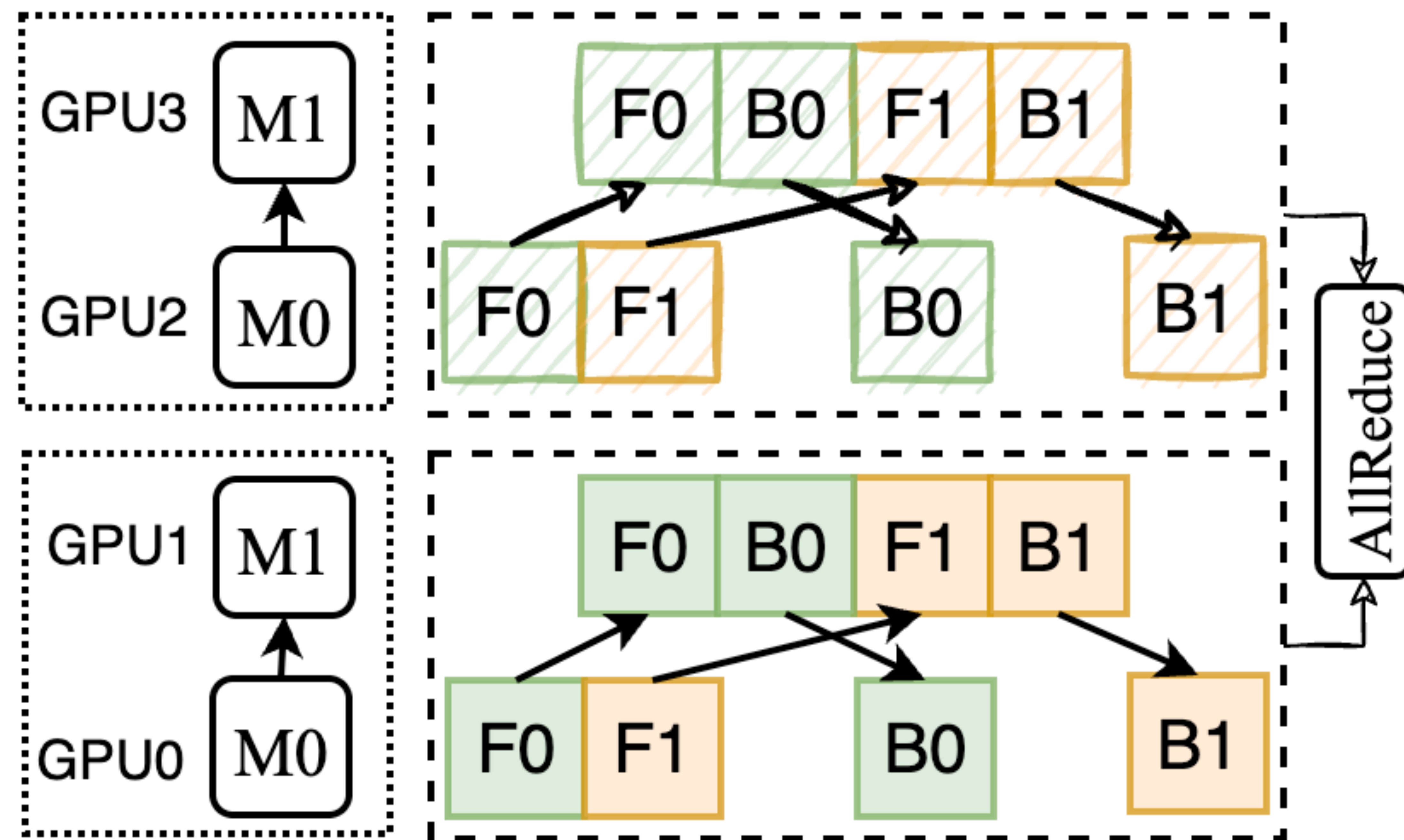
Pipeline Parallelism

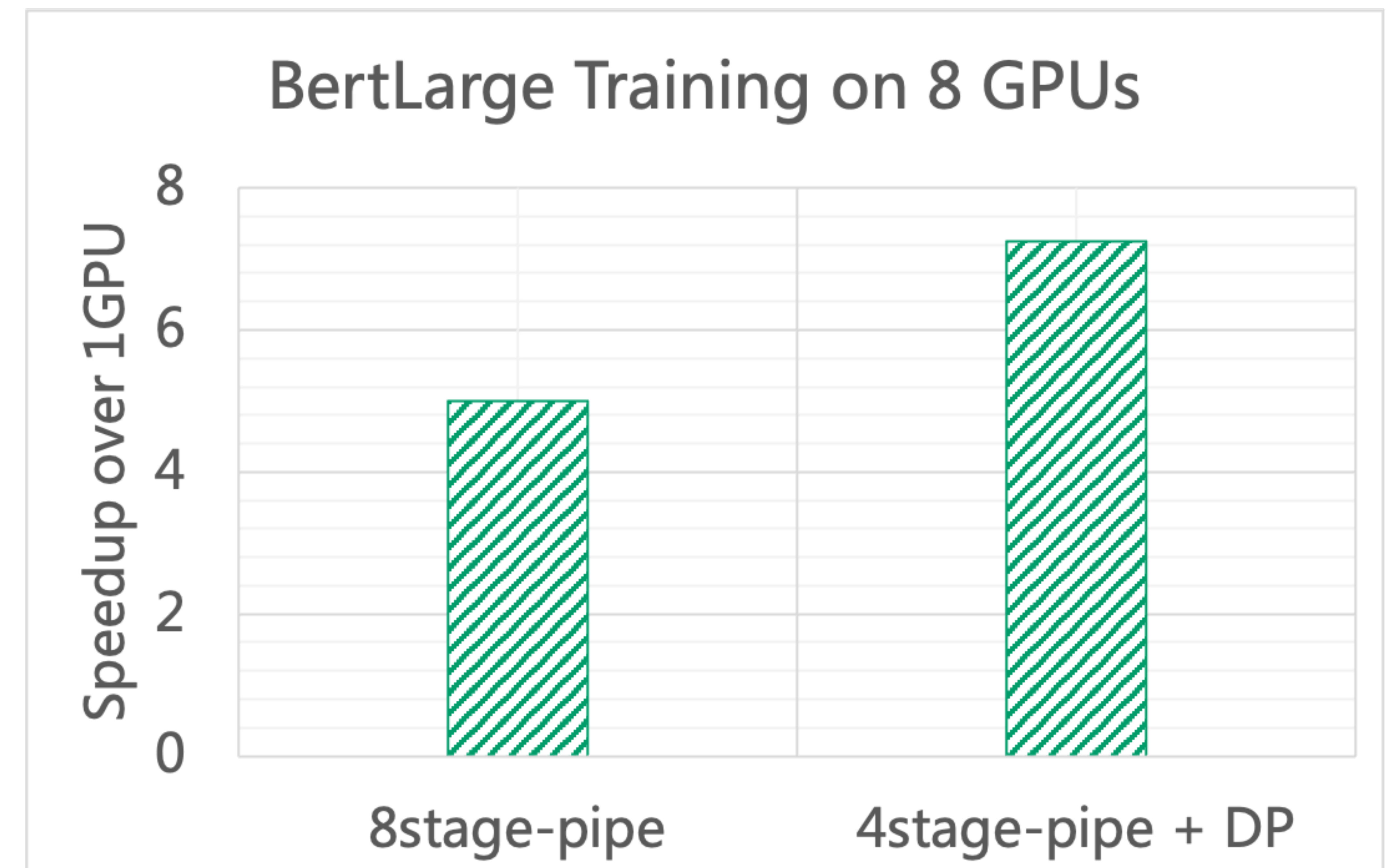


Tensor Model Parallelism

Data + Pipeline

Alibaba Cloud
Worldwide Cloud Services Partner

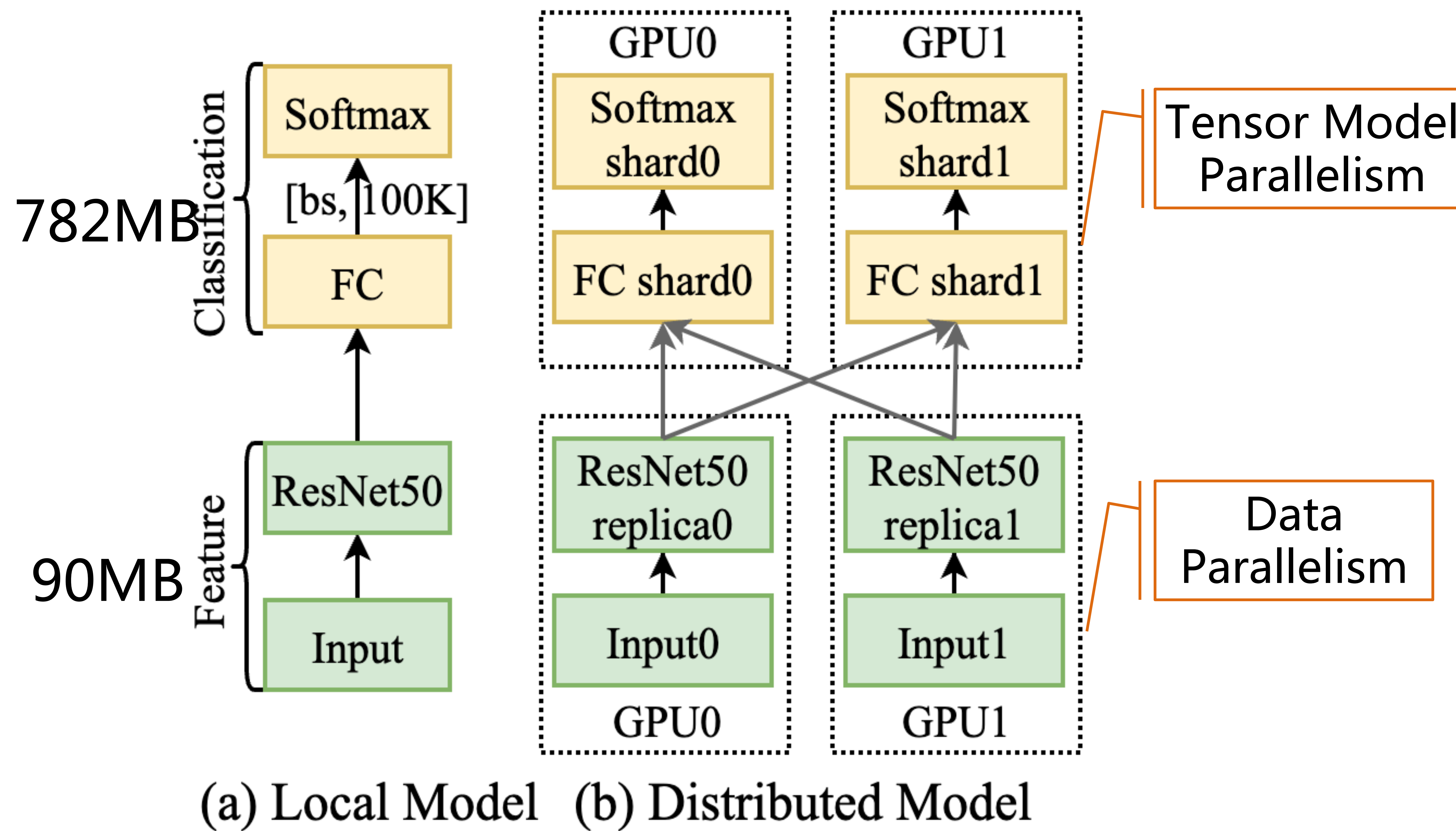




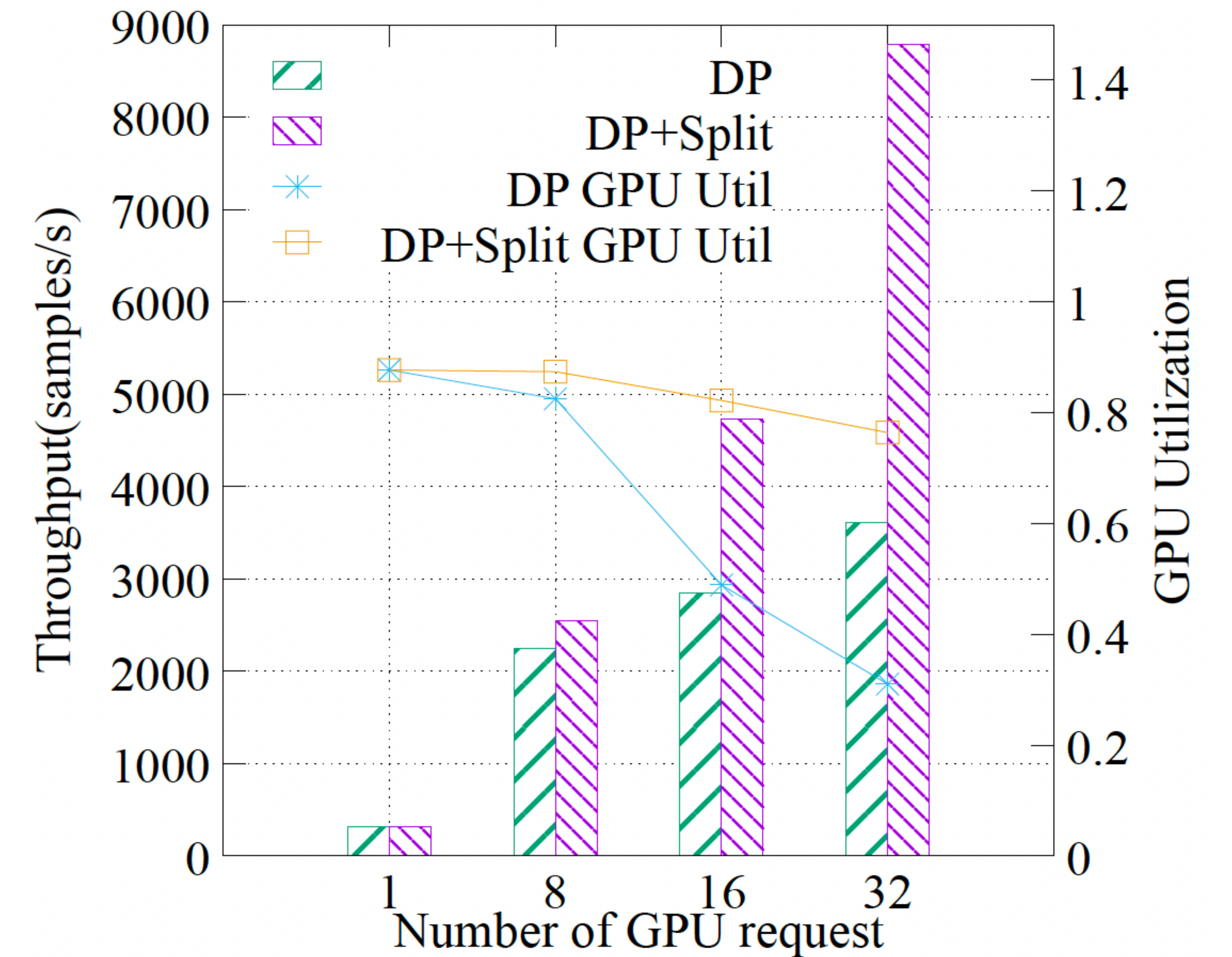
- Pure pipeline parallelism does not scale well with more GPUs
- Nested DP with pipeline

Data + Pipeline + Tensor

Alibaba Cloud | Worldwide Cloud Services Partner



(a) Local Model (b) Distributed Model

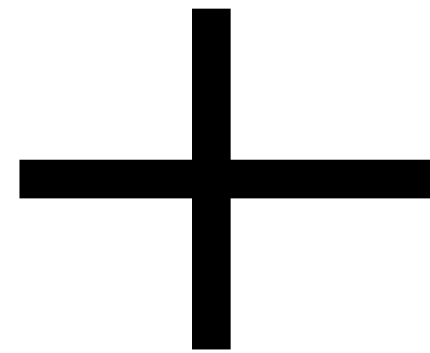


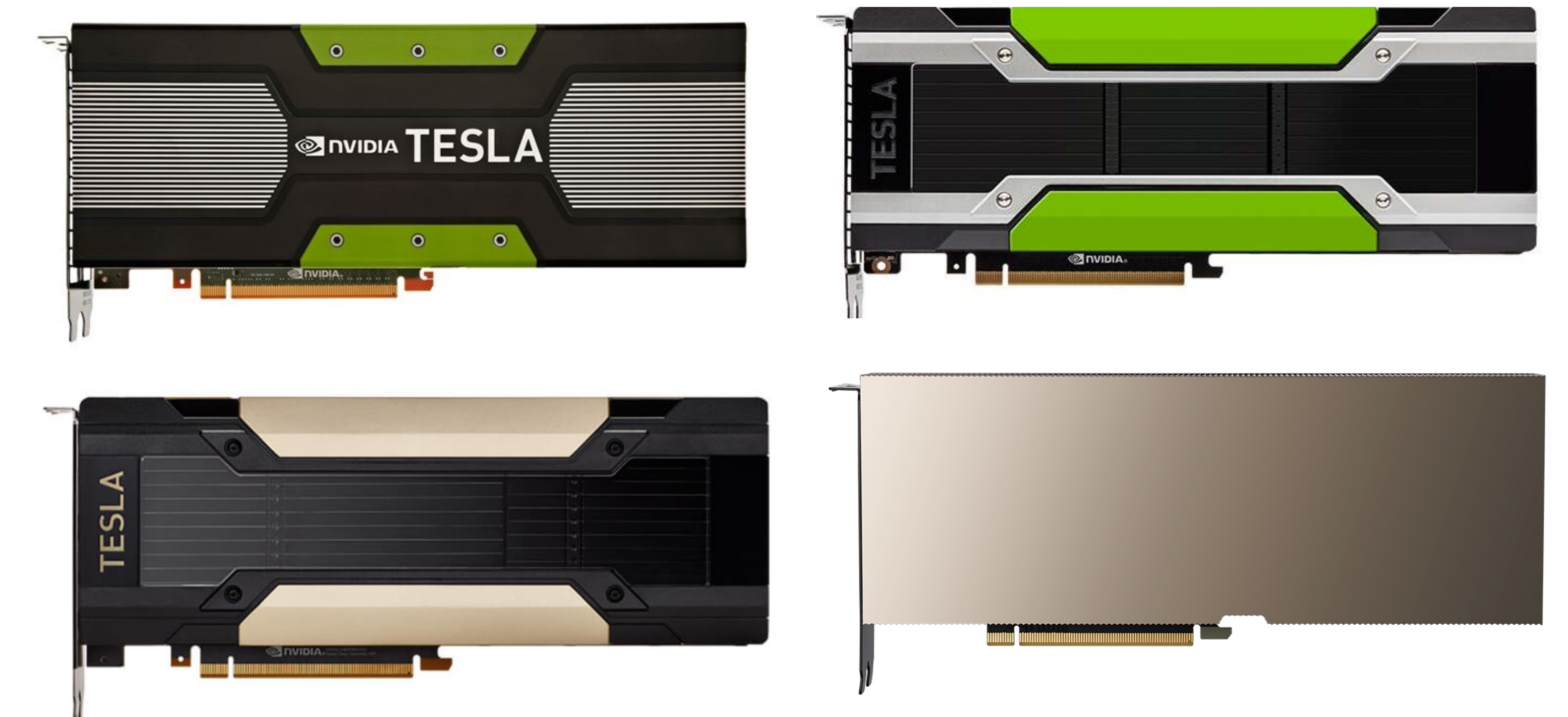
ResNet50 #class=100K
DP vs Hybrid

Apply different strategies to different model parts.

Heterogeneity in GPU Clusters

- Gang Schedule





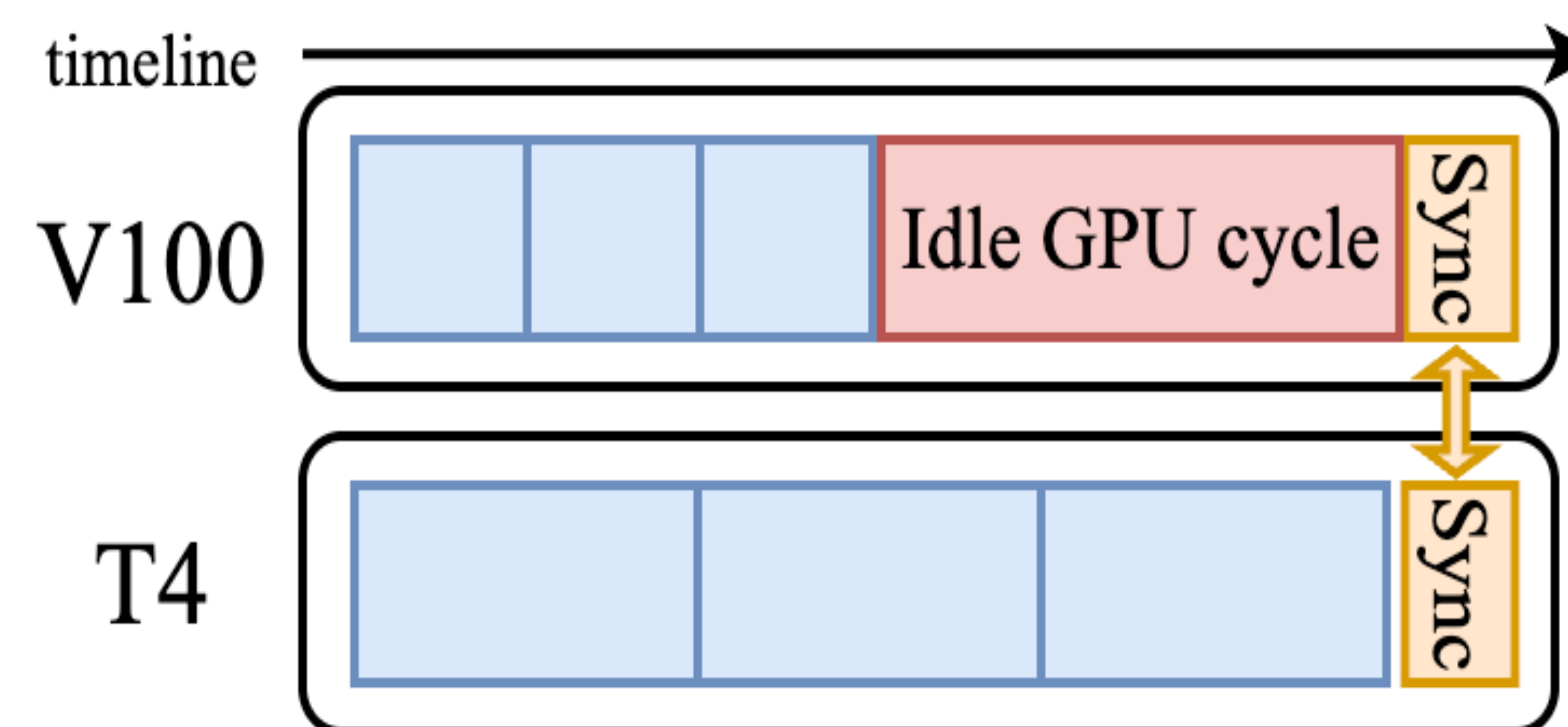
- Heterogeneous GPUs as a resource
- (*e.g.*, Computing Pool with GPUs: P100, V100, A100 and etc)

Challenges: Heterogeneous GPU Training

Alibaba Cloud | Worldwide Cloud Services Partner

Inefficiency in utilizing heterogeneous GPUs

- different GPU types: different computing/memory/network capacity
- imbalance in computing time -> low utilization
- gap between model development and the hardware environment



(a) Naïve DP with identical batch size

Gaps and Opportunities

Alibaba Cloud | Worldwide Cloud Services Partner

- Lack of unified abstraction to support all of the parallel strategies and the hybrids
- Fully automatic parallel strategy has high cost for giant models
- Inefficiency in utilizing heterogeneous GPUs
- Require significant model code refactoring

Gaps and Opportunities

Alibaba Cloud | Worldwide Cloud Services Partner

- Lack unified abstraction to support all of the parallel strategies and the hybrids

✓ Unified abstraction for strategy expression

- Fully automatic parallel strategy has high cost for giant models

✓ Incorporate user hints

- Inefficiency in utilizing heterogeneous GPUs

✓ Parallel strategies should be used adaptively and dynamically

- Require significant model code refactoring

✓ Minimize code change, switch among strategies easily

Whale: Efficient Giant Model Training over Heterogeneous GPUs

Alibaba Cloud | Worldwide Cloud Services Partner

- Two new high-level primitives for unified expression

- Transform distributed models efficiently and automatically

- Hardware-aware load balancing algorithm

- Train the largest multi-modality model M6 with ten trillion model with only 4-lines of code change

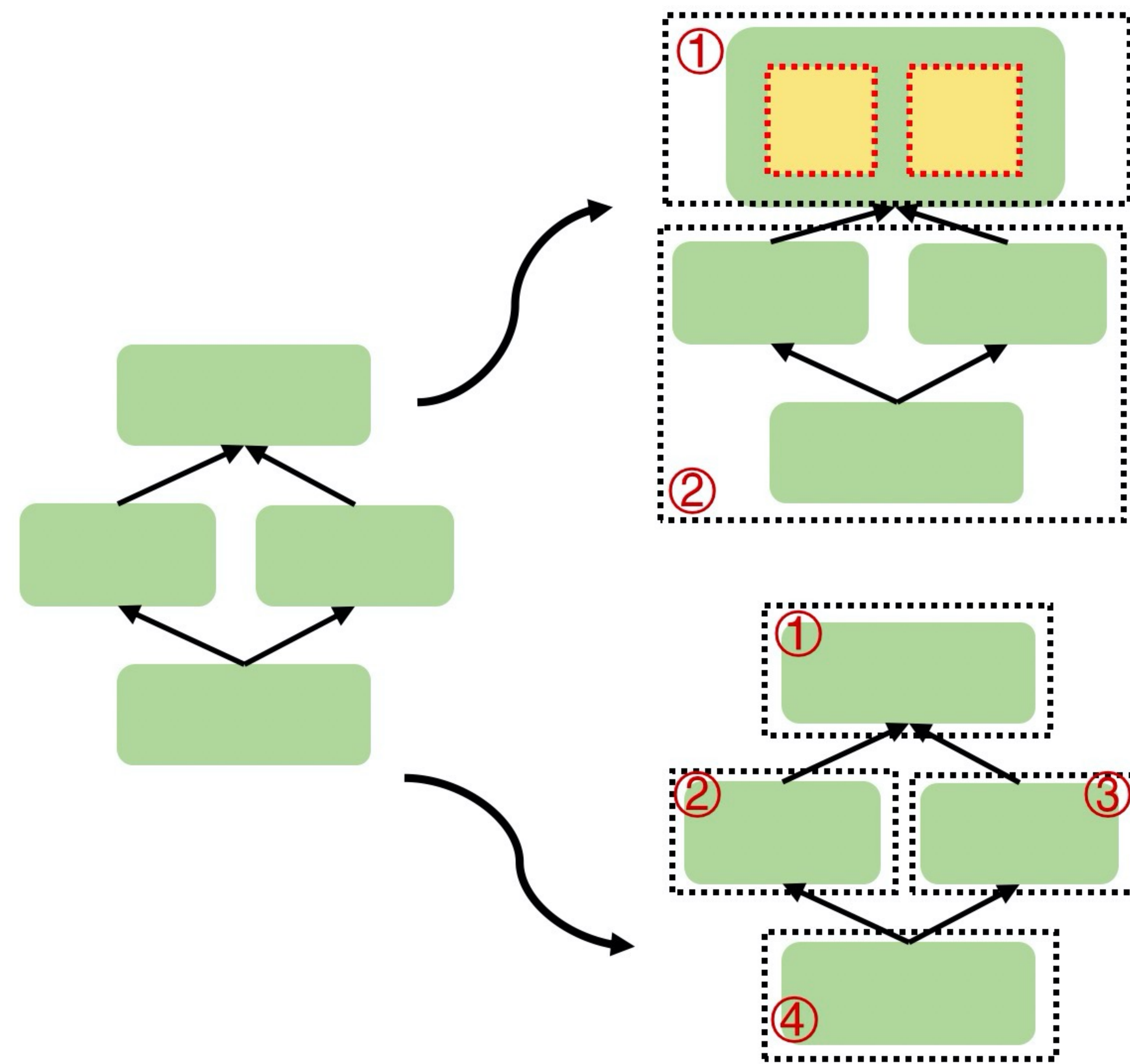
Outline

Alibaba Cloud | Worldwide Cloud Services Partner

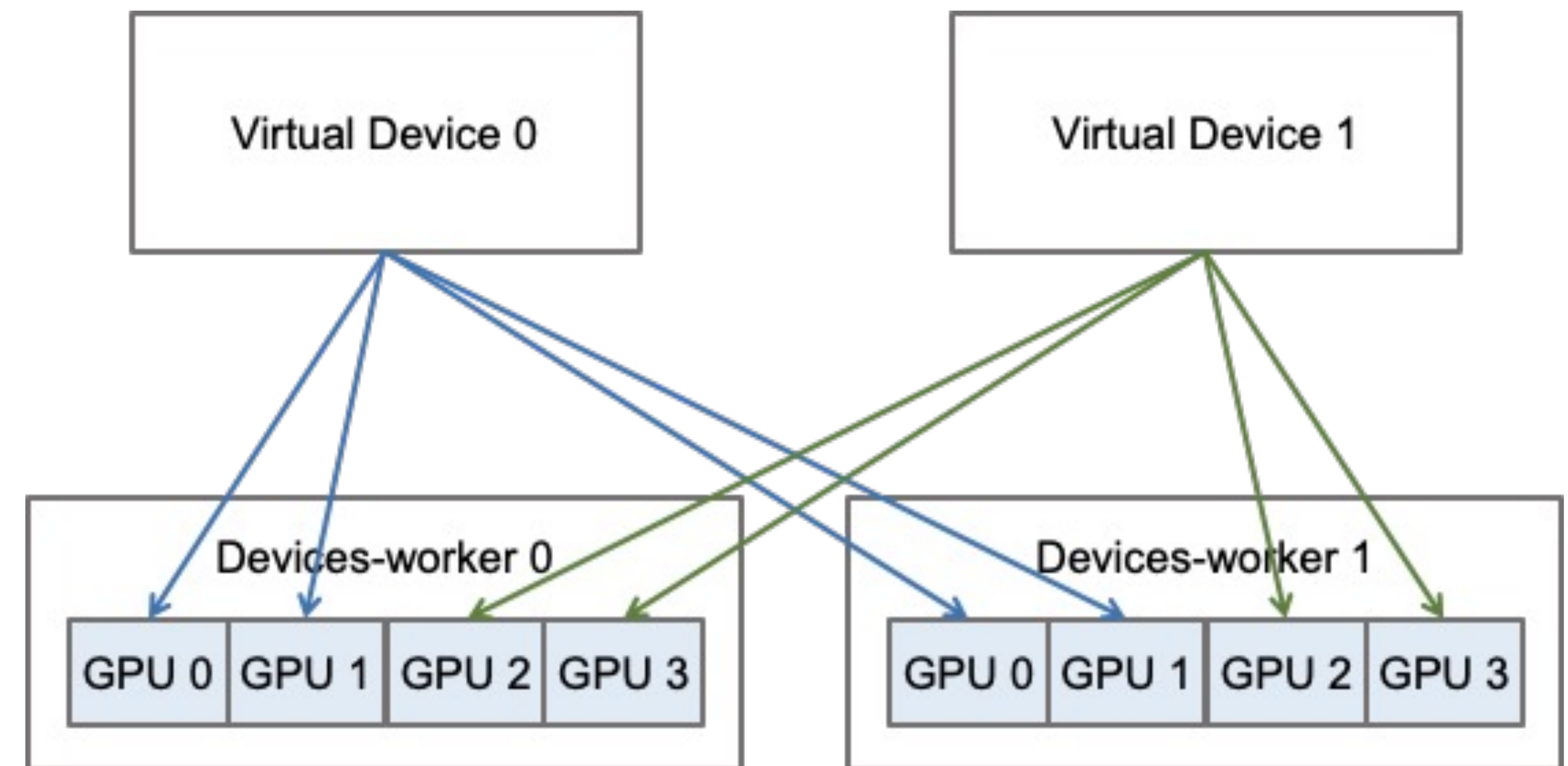
- Introduction
- **Whale: design abstraction**
- **Whale: parallel planner**
- **Whale: hardware-aware load balance**
- Evaluation
- Conclusion

Abstraction: Internal Key Concepts

Alibaba Cloud | Worldwide Cloud Services Partner



TaskGraph



VirtualDevice

Parallel Primitives

Parallel primitive is a Python context manager.
Operations defined under form one TaskGraph (TG)

replicate(device_count) annotates a TG to be replicated.

- device_count: #devices for TG replicas

split(device_count) annotates a TG to apply intra-tensor sharding.

- device_count: #devices for sharded partitions

Parallel Examples

Alibaba Cloud | Worldwide Cloud Services Partner

```
import whale as wh
wh.init(wh.Config({
  "num_micro_batch": 8}))
with wh.replicate(1):
  model_stage1()
with wh.replicate(1):
  model_stage2()
```

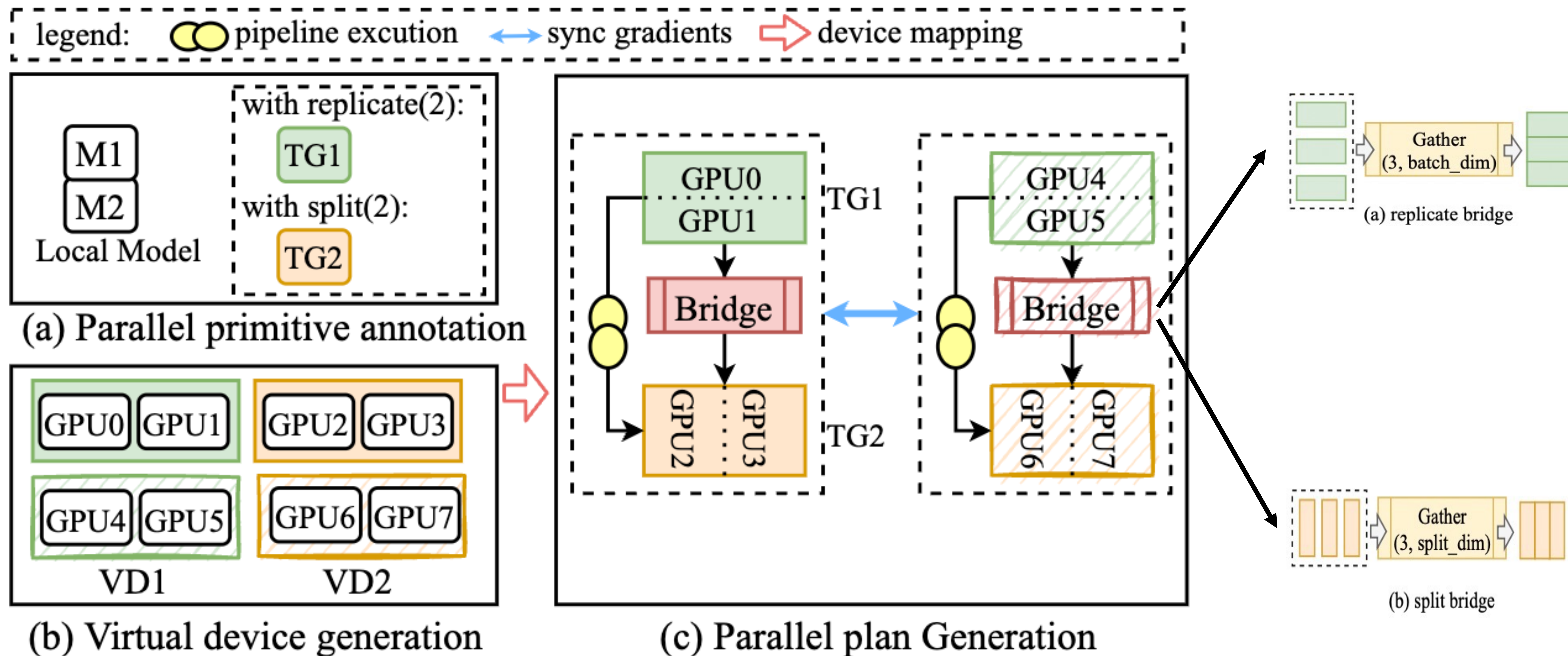
Pipeline with 2 TaskGraphs

```
import whale as wh
wh.init()
with wh.replicate(total_gpu):
  features = ResNet50(inputs)
with wh.split(total_gpu):
  logits = FC(features)
  predictions = Softmax(logits)
```

Hybrid of replicate and split

Parallel Planner

Alibaba Cloud | Worldwide Cloud Services Partner



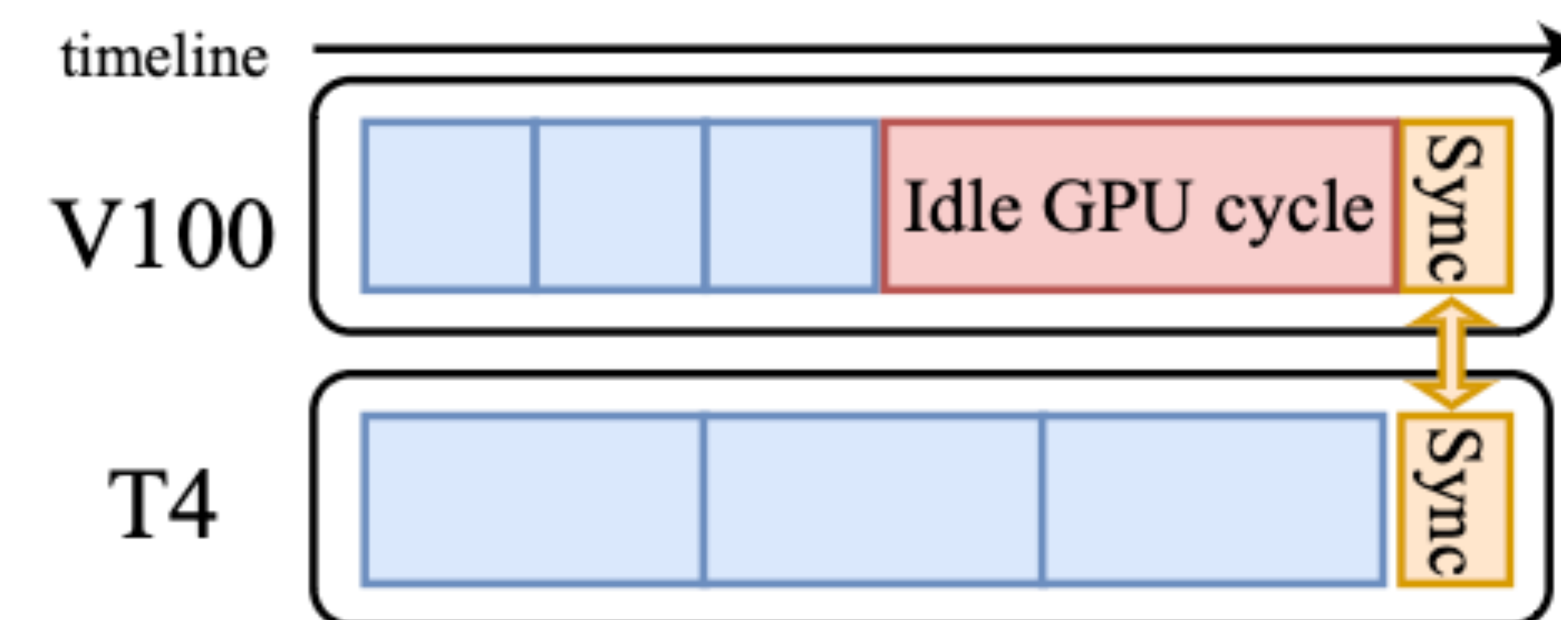
Hardware-aware Load Balancer

Alibaba Cloud | Worldwide Cloud Services Partner

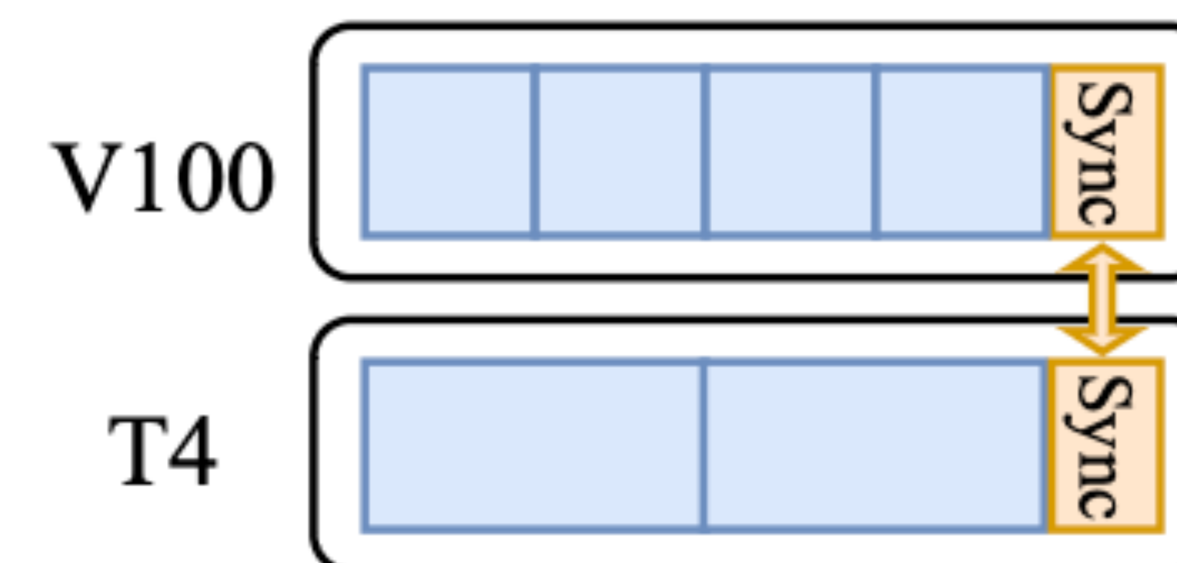
Balance the computing load proportional to the device computing capacity, s.t. memory constraints.

Data parallelism: balance the FLOP by adjusting local batch while keeps the mini-batch unchanged.

Tensor Model Parallelism: balance the FLOP of partitioned operations through uneven sharding.



(a) Naïve DP with identical batch size



(b) Hardware-aware DP with load balance

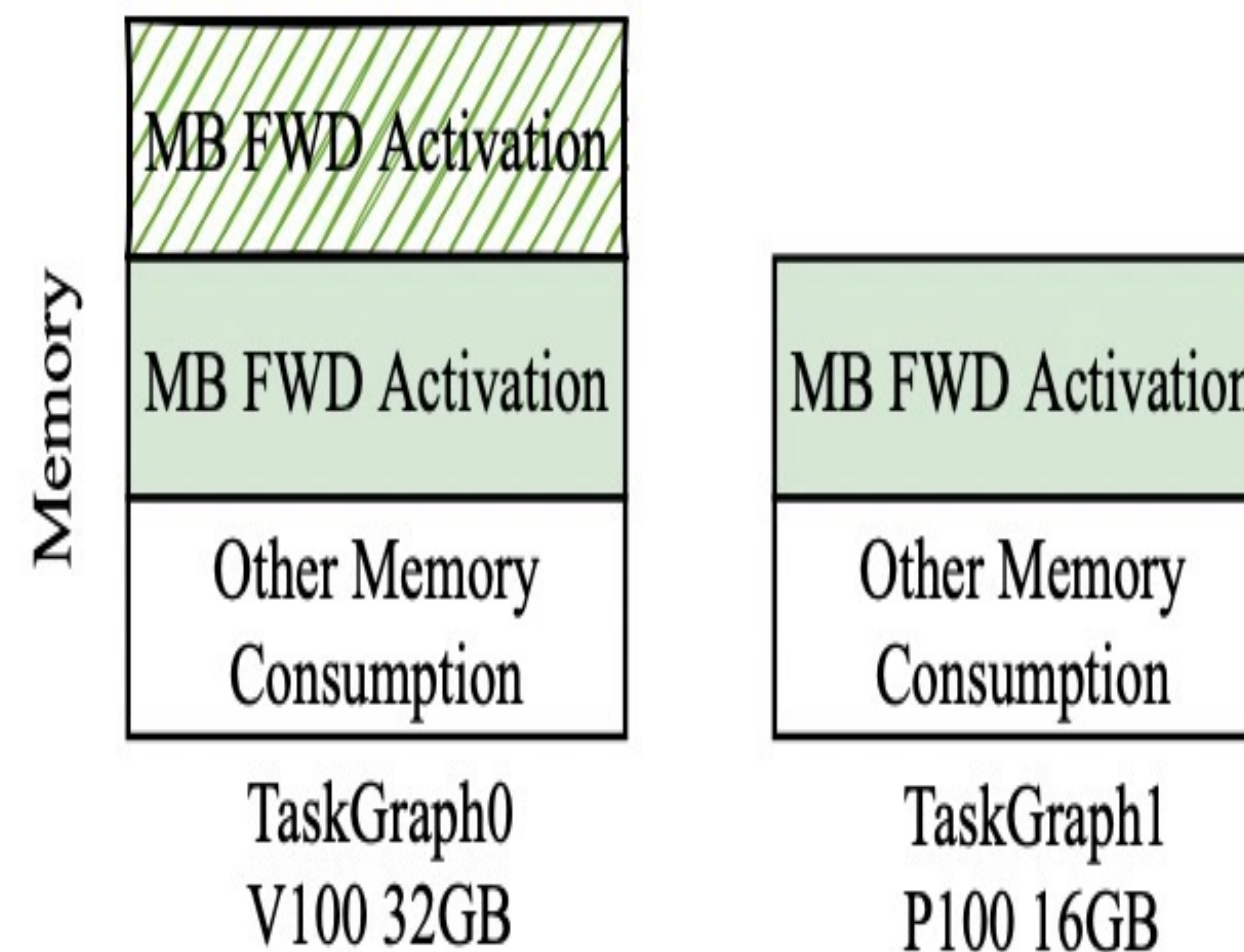
Memory-Constraint Load Balancing

Alibaba Cloud | Worldwide Cloud Services Partner

Algorithm 1: Memory-Constraint Load Balancing

Input: $TaskGraph\ TG, VirtualDevice(N)$

1 $load_ratios = \emptyset$; $mem_utils = \emptyset$; $flop_utils = \emptyset$
2 $oom_devices = \emptyset$; $free_devices = \emptyset$
3 **foreach** $i \in 0...N$ **do**
4 $\quad load_ratios[i] = \frac{DF_i}{\sum_{i=0}^{N} DF_i}$
5 $\quad mem_utils[i] = \frac{load_ratios[i] * TG_{mem}}{DM_i}$
6 $\quad flop_utils[i] = \frac{load_ratios[i] * TG_{flop}}{DF_i}$
7 $\quad$ **if** $mem_utils[i] > 1$ **then**
8 $\quad\quad oom_devices.append(i)$
9 $\quad$ **else**
10 $\quad\quad free_devices.append(i)$
11 **while** $oom_devices \neq \emptyset\ \&\ free_devices \neq \emptyset$ **do**
12 $\quad peak_device = argmax(oom_devices, key = mem_utils)$
13 $\quad valley_device = argmin(free_devices, key = (flop_utils, mem_utils))$
14 $\quad$ **if** $shift_load(peak_device, valley_device) == success$ **then**
15 $\quad\quad update_profile(mem_utils, flop_utils)$
16 $\quad\quad oom_devices.pop(peak_device)$
17 $\quad$ **else**
18 $\quad\quad free_devices.pop(valley_device)$



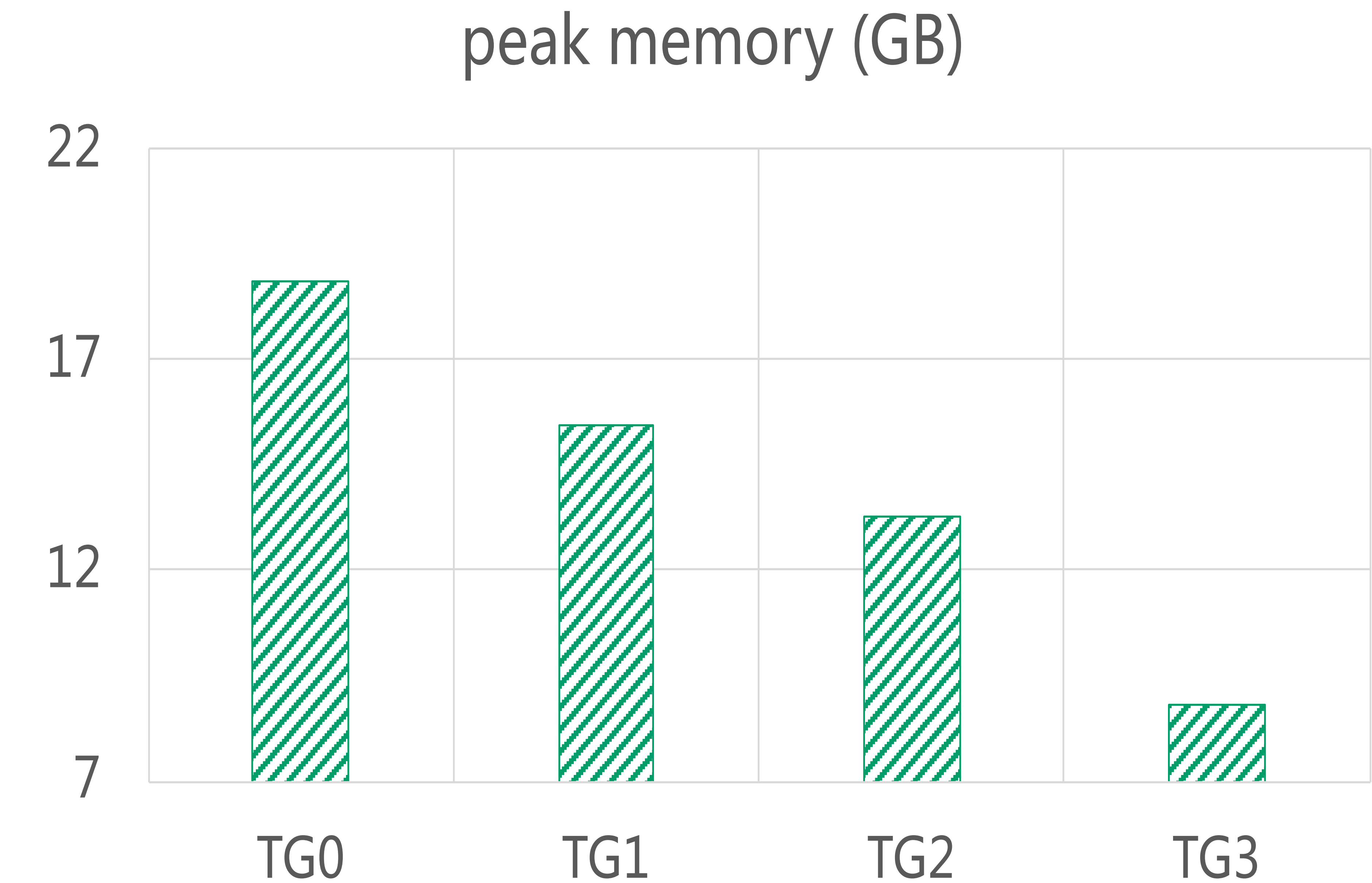
Load Balancer Example

Alibaba Cloud | Worldwide Cloud Services Partner

- Earlier TaskGraph has higher peak memory than later TaskGraph (e.g. BertLarge)

- Place earlier TaskGraphs on devices with higher memory capacity.

- Partition the model operations to TaskGraphs in a topological sort, balance the TaskGraphs computing FLOP proportional to device capacity.



Peak memory for TaskGraphs (BertLarge, micro-bs=6)

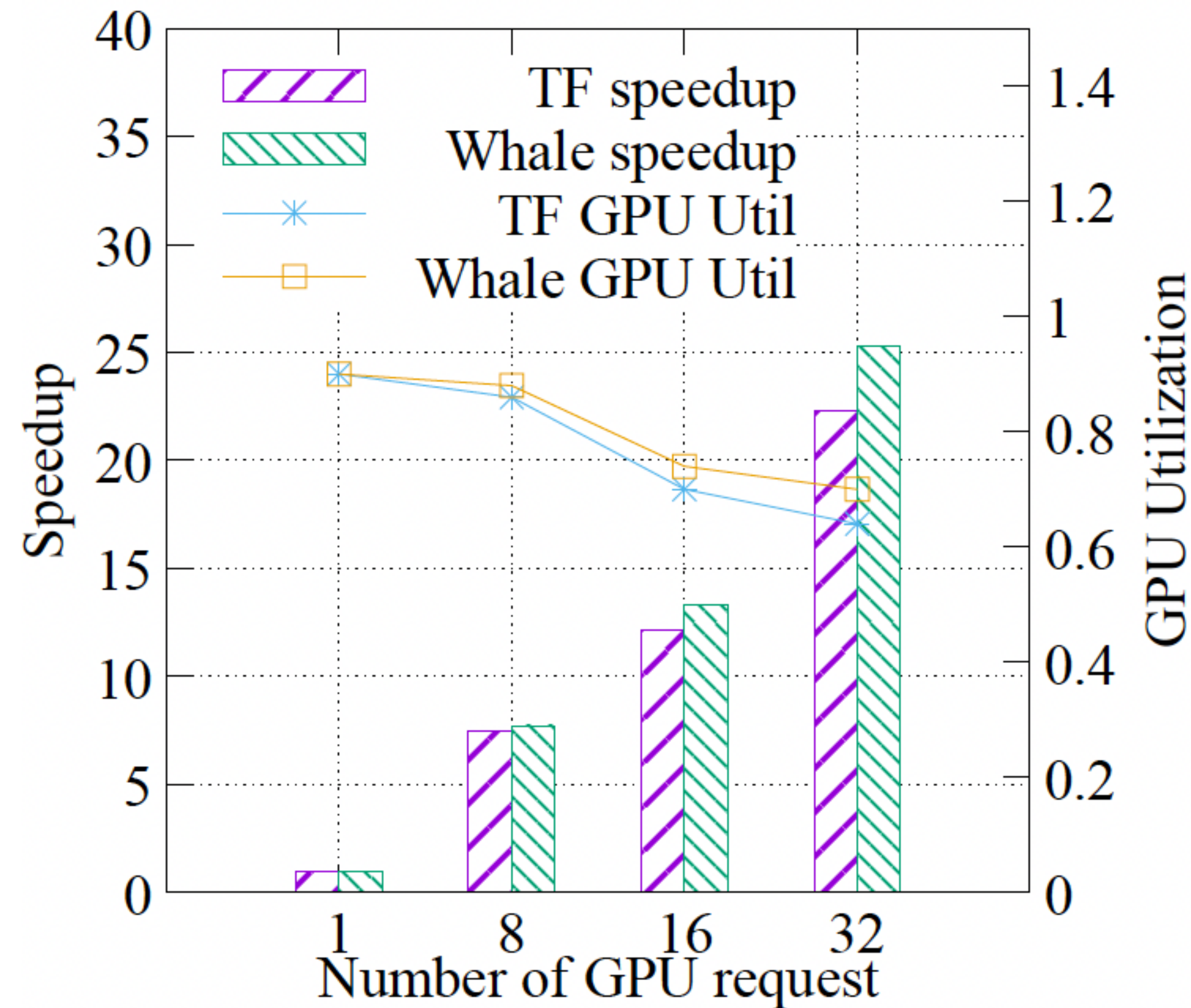
Outline

Alibaba Cloud | Worldwide Cloud Services Partner

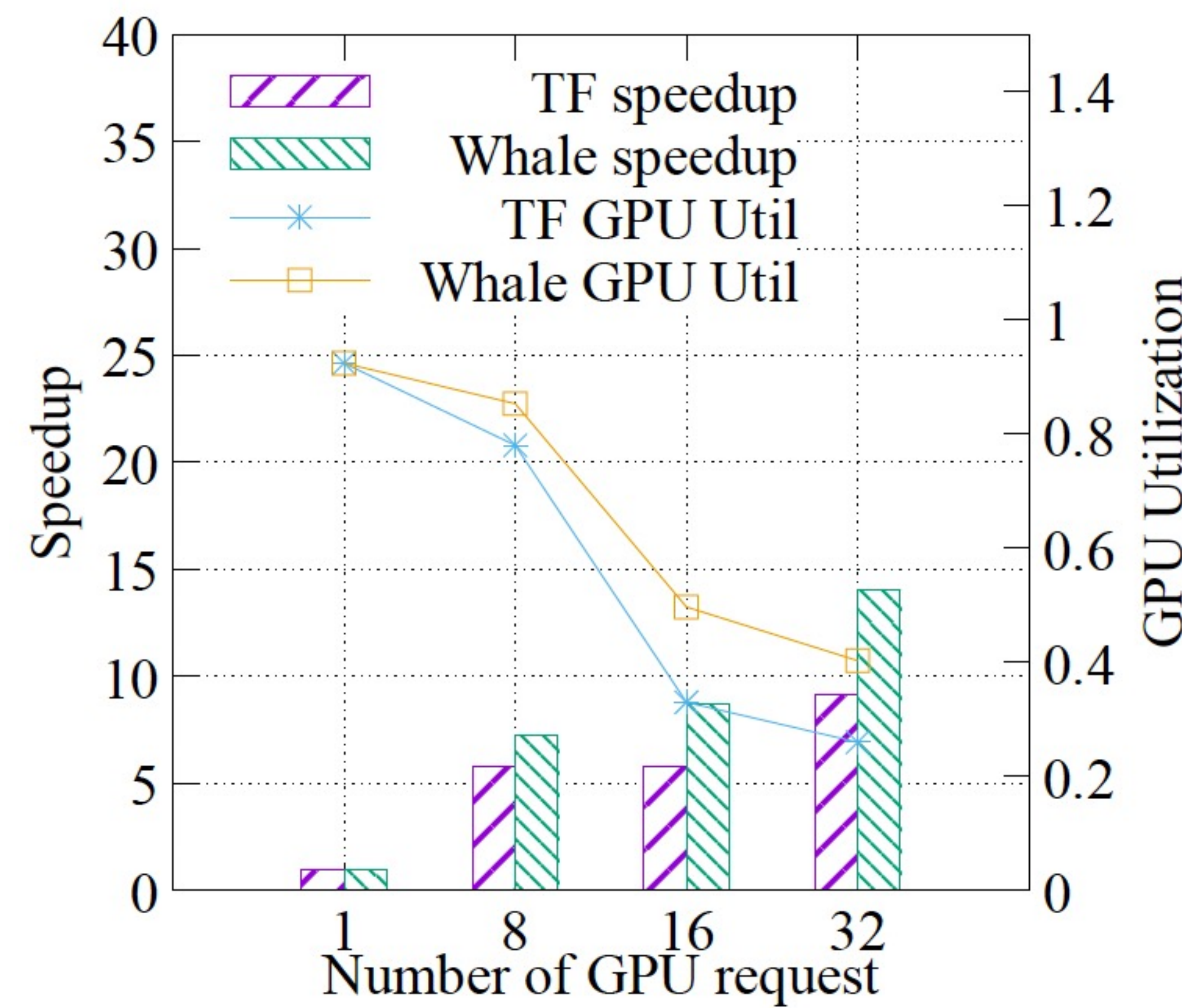
- Introduction
- Whale: design abstraction
- Whale: parallel planner
- Whale: hardware-aware load balance
- Evaluation
- Conclusion

Micro-benchmark: Single Parallel Strategy

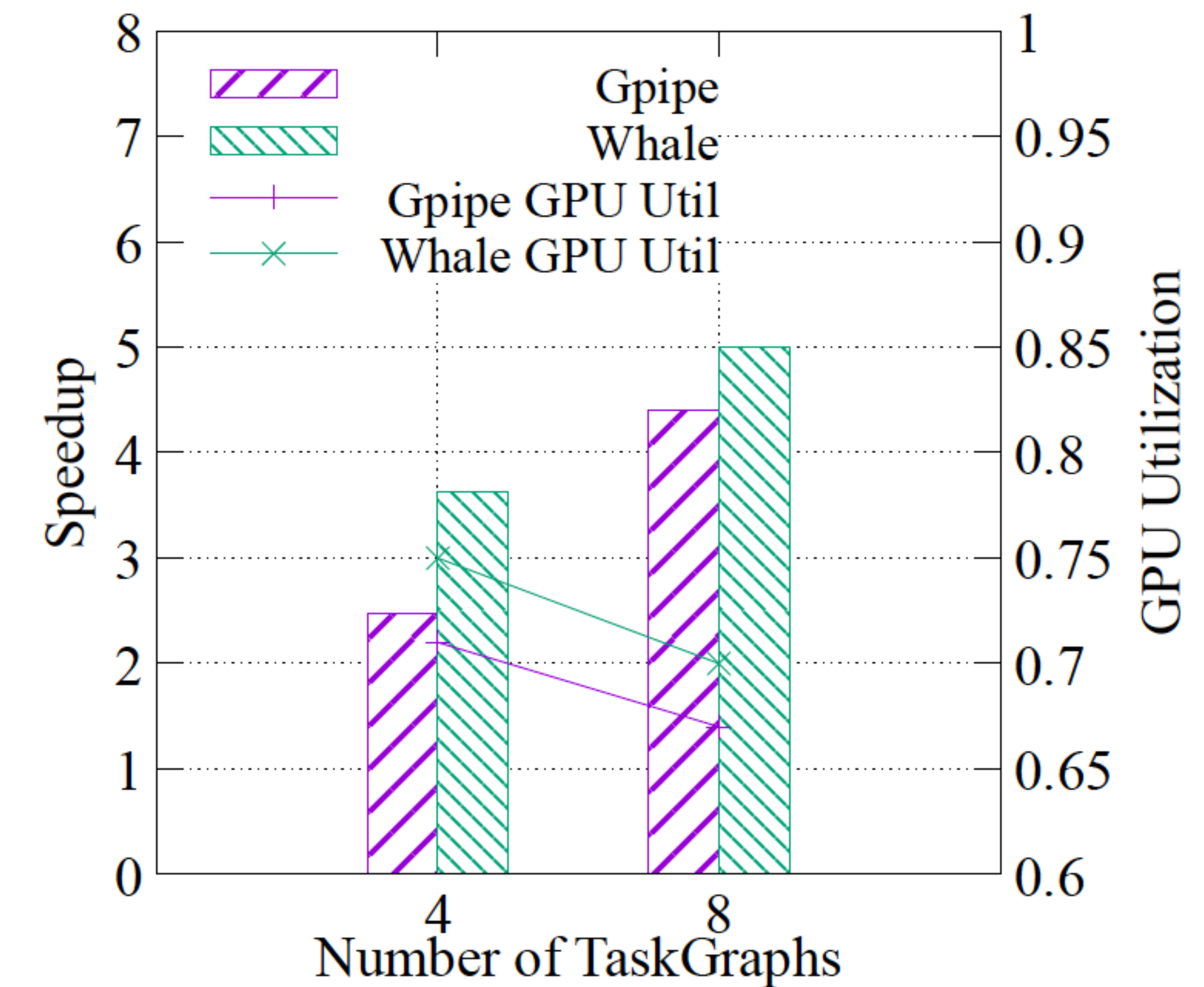
Alibaba Cloud | Worldwide Cloud Services Partner



(a) Whale DP vs TF DP on ResNet



(b) Whale DP vs TF DP on BertLarge



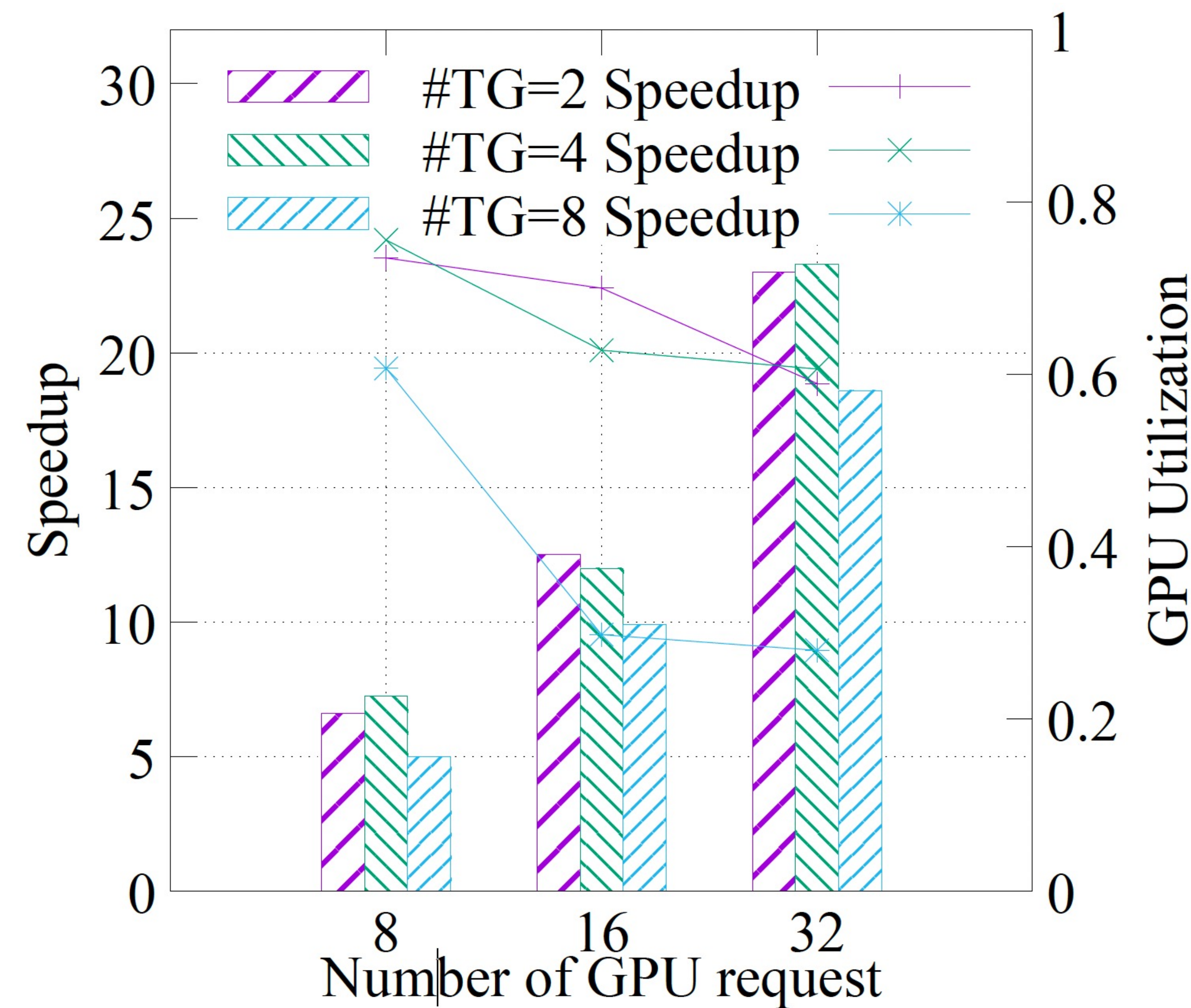
(c) Whale Pipeline vs Gpipe on BertLarge

(a, b) Whale DP obtained better performance than TF Estimator DP on ResNet and BertLarge

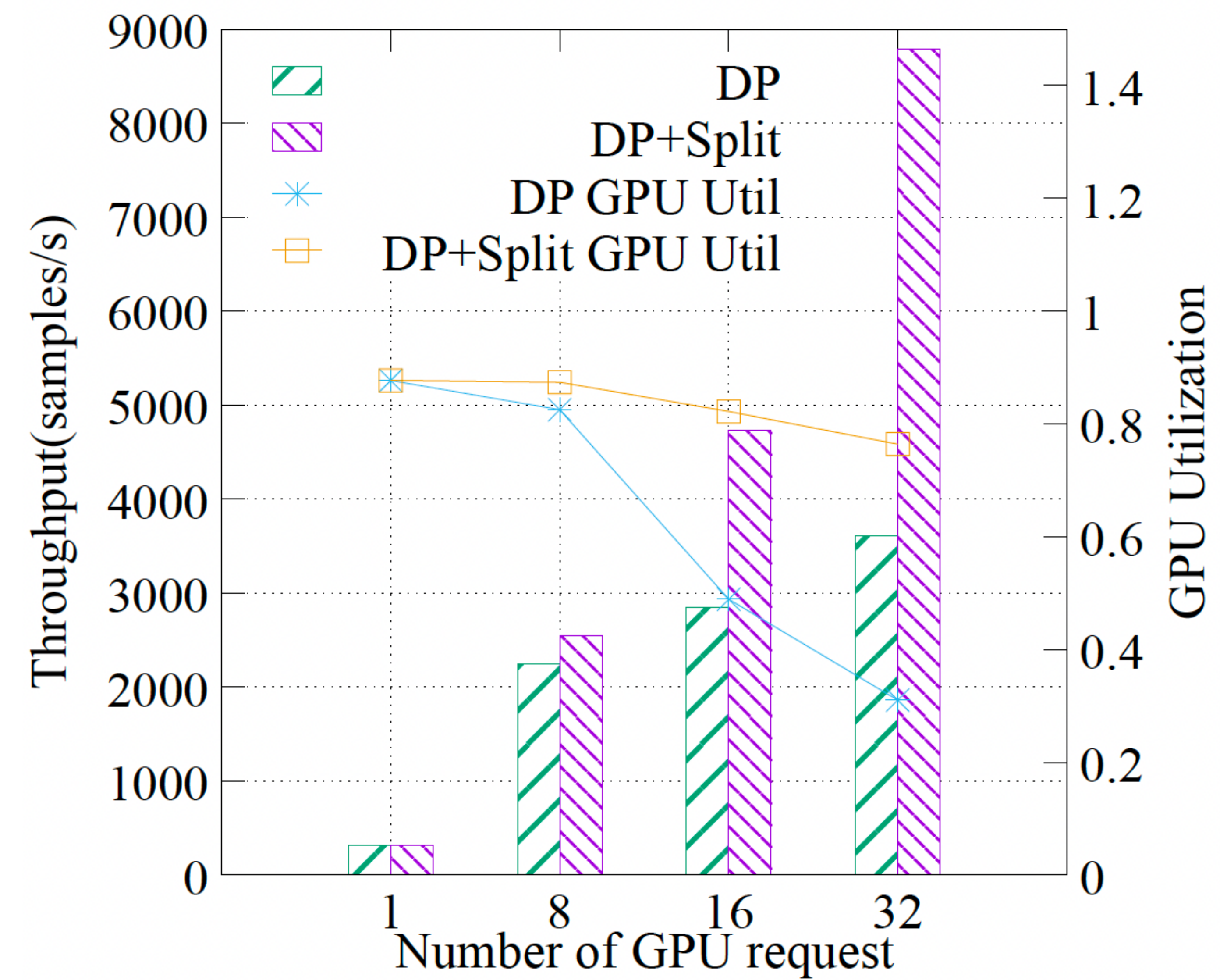
(c) Whale pipeline on BertLarge outperforms Gpipe

* 4-stages 1.45X ↗, and 8 stages 1.14X ↗

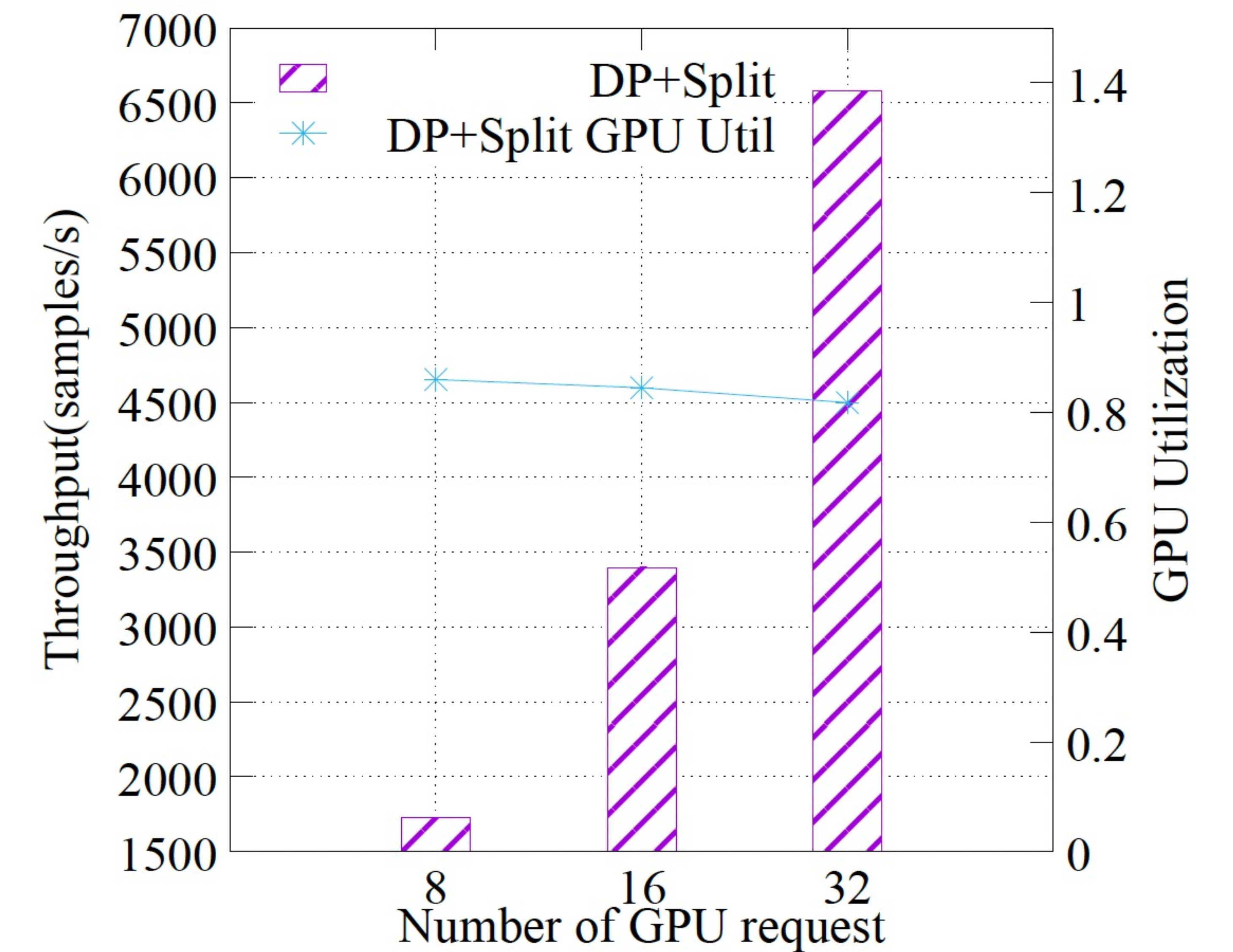
Micro-benchmark: Hybrid Strategy



(a) BertLarge
Hybrid strategy



(b) ResNet50 #class=100K
DP vs Hybrid



(c) ResNet50 #class=1M
Hybrid strategy

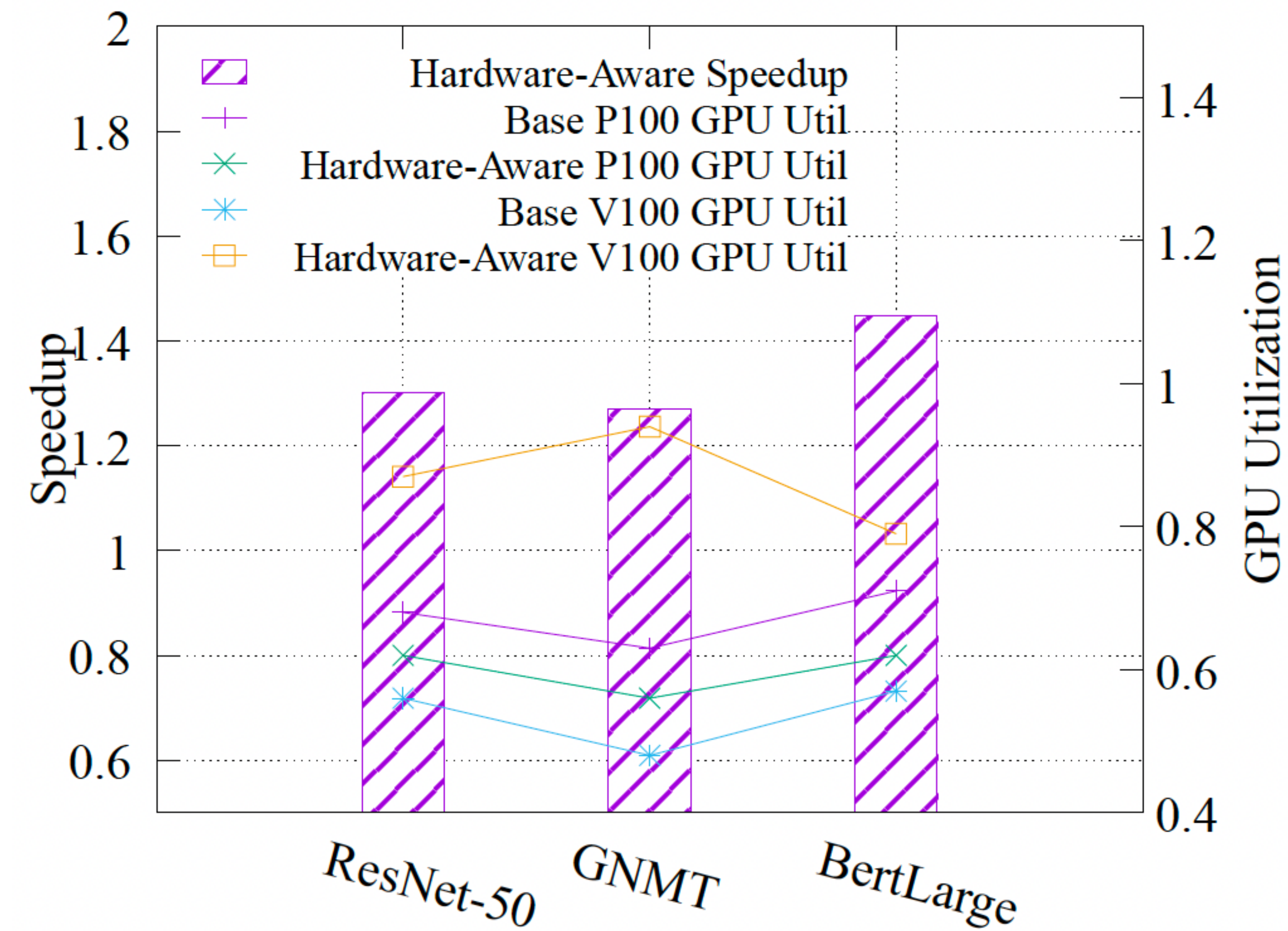
(a) Hybrid pipe+DP (TG=2 and TG=4) got better performance than pure pipe (TG=8) on 8 GPUs

(b) #class=100K, Hybrid split+DP got better performance than pure DP, 1.13~2.43X

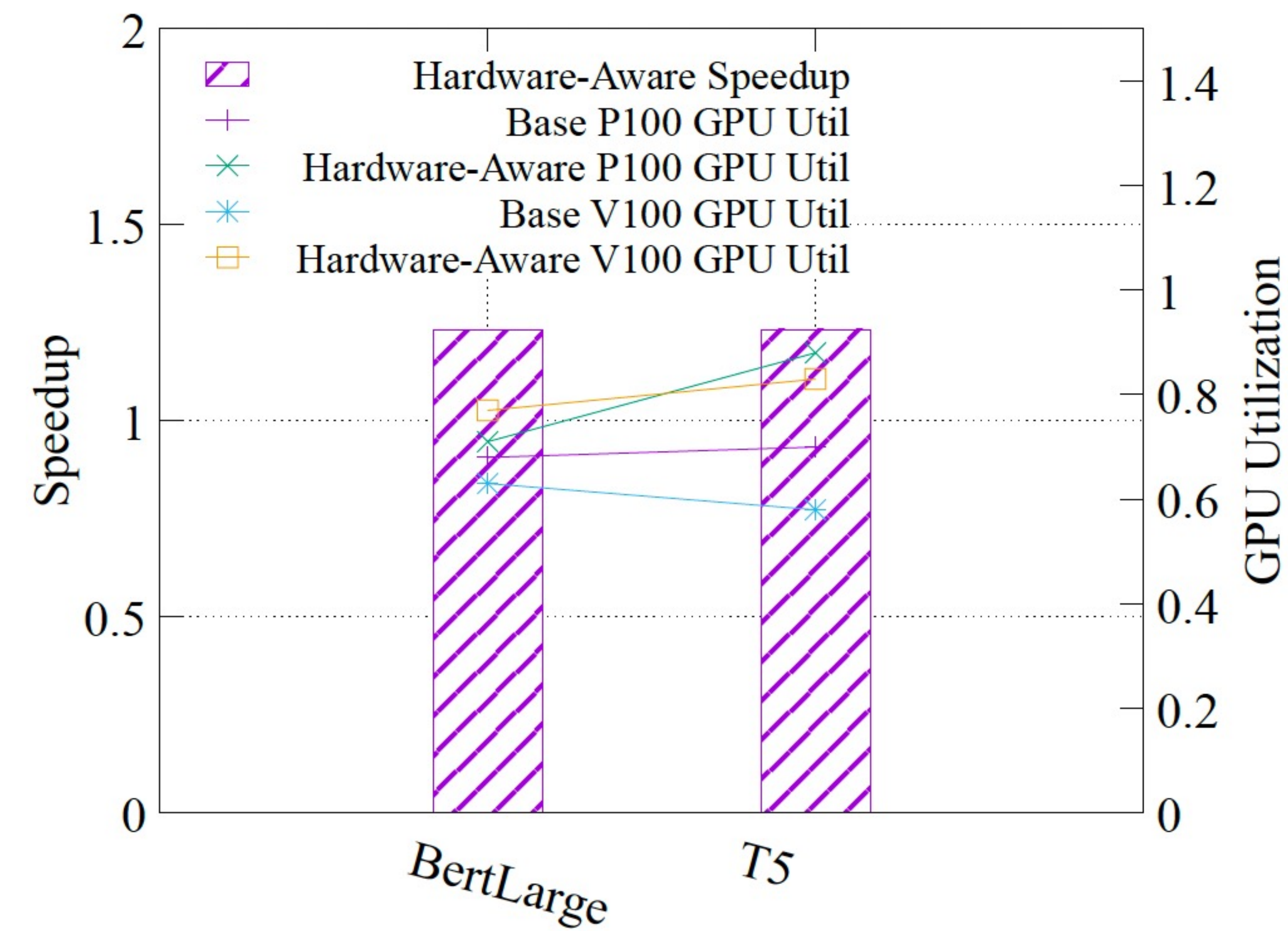
(c) #class=1M, DP fails due to OOM. Hybrid achieved 95% scaling from 8~32GPUs

Micro-benchmark: Hardware-aware

Alibaba Cloud | Worldwide Cloud Services Partner



(a) Hardware-Aware DP



(b) Hardware-Aware Pipeline

Setup: 8 32GB V100 GPUs and 8 16GB P100 GPUs

(a) Hardware-aware DP got 1.3X to 1.4X

(b) Hardware-aware Pipeline got 1.2X

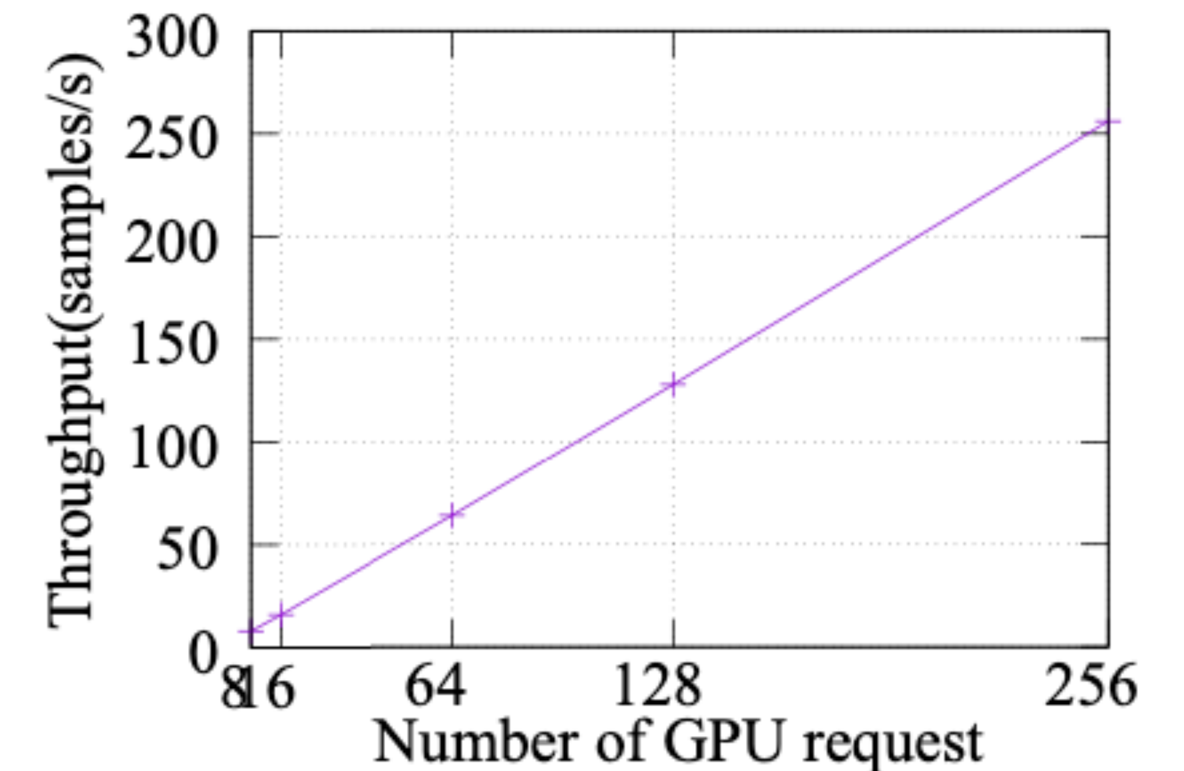
Industry-Scale Giant Model Training

Alibaba Cloud | Worldwide Cloud Services Partner

- M6-10B: 91% throughput scalability from 8 to 256 GPUs
- M6-MoE-10T: A few lines to switch from pipeline to tensor model parallelism (MoE). Train on 512 NVIDIA V100 GPUs.

```
import whale as wh
wh.init(wh.Config({
  "num_micro_batch": 35,
  "num_task_graph": 8}))
# Define M6 model.
m6_model_def()
```

(a) M6-10B with Pipeline and DP



(b) M6-10B throughput

```
import whale as wh
wh.init()
wh.set_default_strategy(wh.replicate(total_gpus))
combined_weights,dispatch_inputs=gating_dispatch()
with wh.split(total_gpus):
  outputs = MoE(combined_weights, dispatch_inputs)
```

(c) M6-MoE-10T

Conclusion

Whale: Efficient Giant Model Training over Heterogeneous GPUs

- Efficiency, programmability, and adaptability
- Supports various parallel strategies using a unified abstraction
- Adapts to heterogeneous GPUs with automatic graph optimizations
- Deployed DL infrastructure at Alibaba for real giant model training

[Code] https://github.com/alibaba/EasyParallelLibrary

Alibaba Cloud

Worldwide Cloud Services Partner

Thanks

Q&A