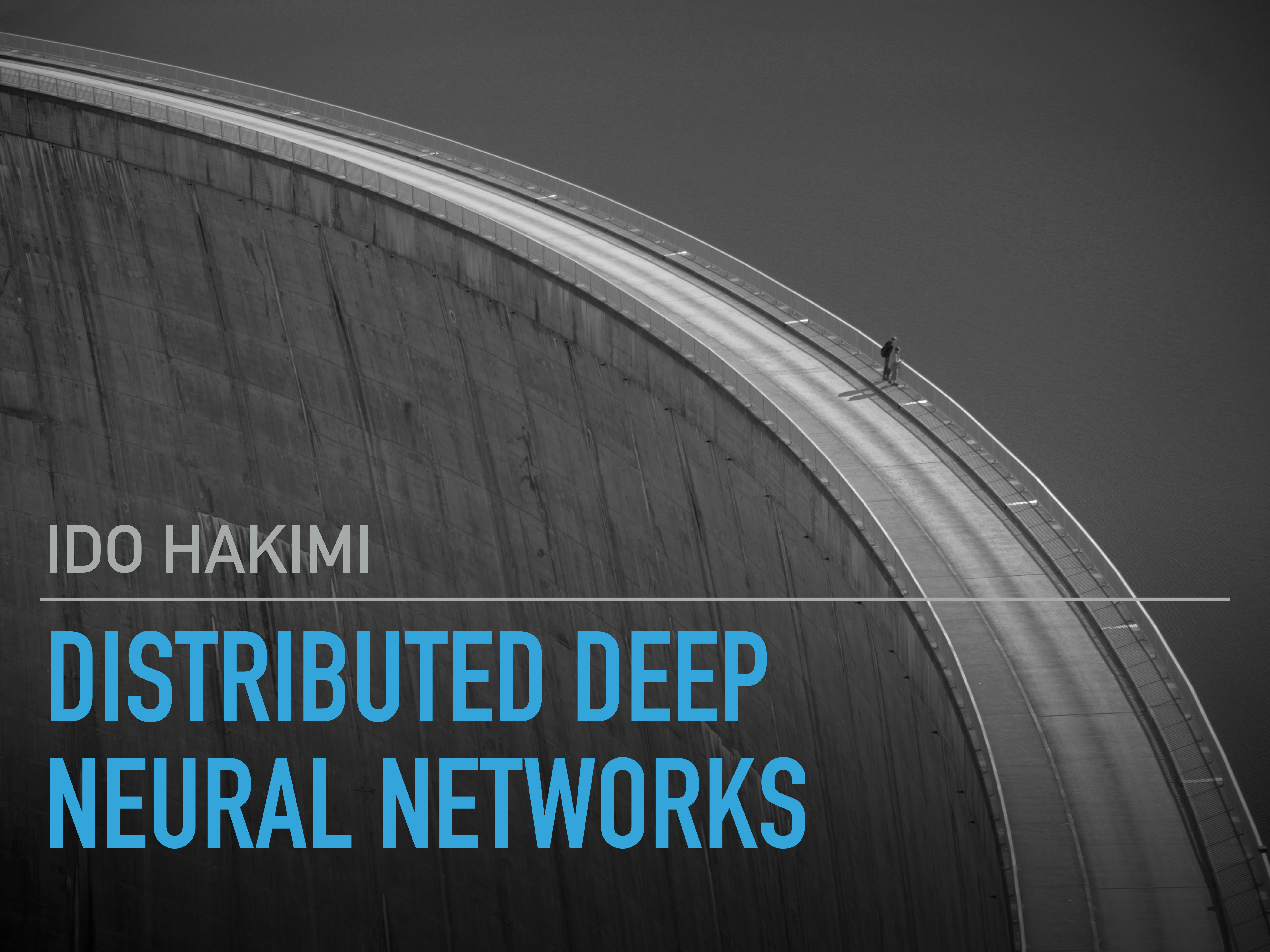


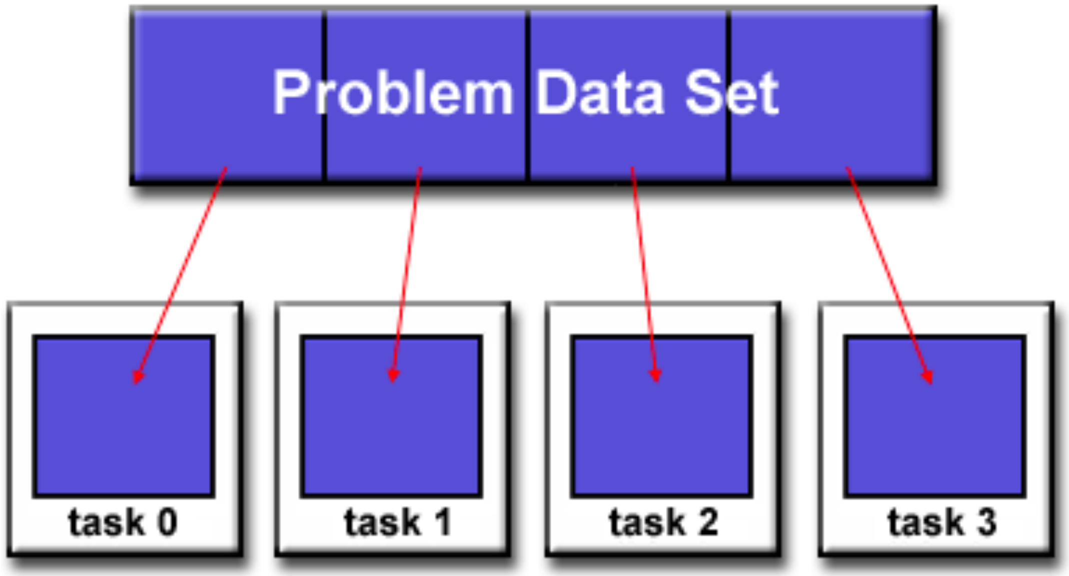
IDO HAKIMI

# DISTRIBUTED DEEP NEURAL NETWORKS

# PROBLEM DECOMPOSITION

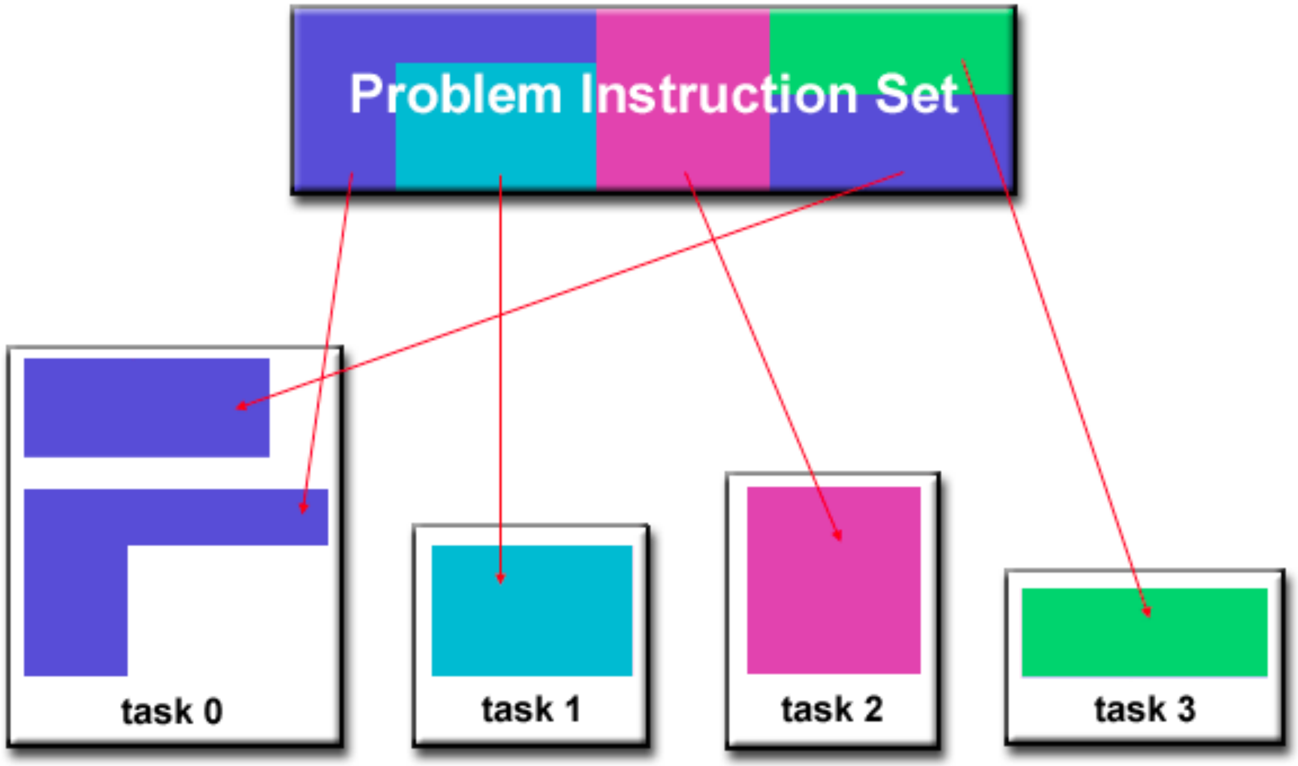
# DOMAIN DECOMPOSITION

- The data is partitioned across tasks and each task only works on its portion of the data.
- SPMD (single program, multiple data)



# FUNCTIONAL DECOMPOSITION

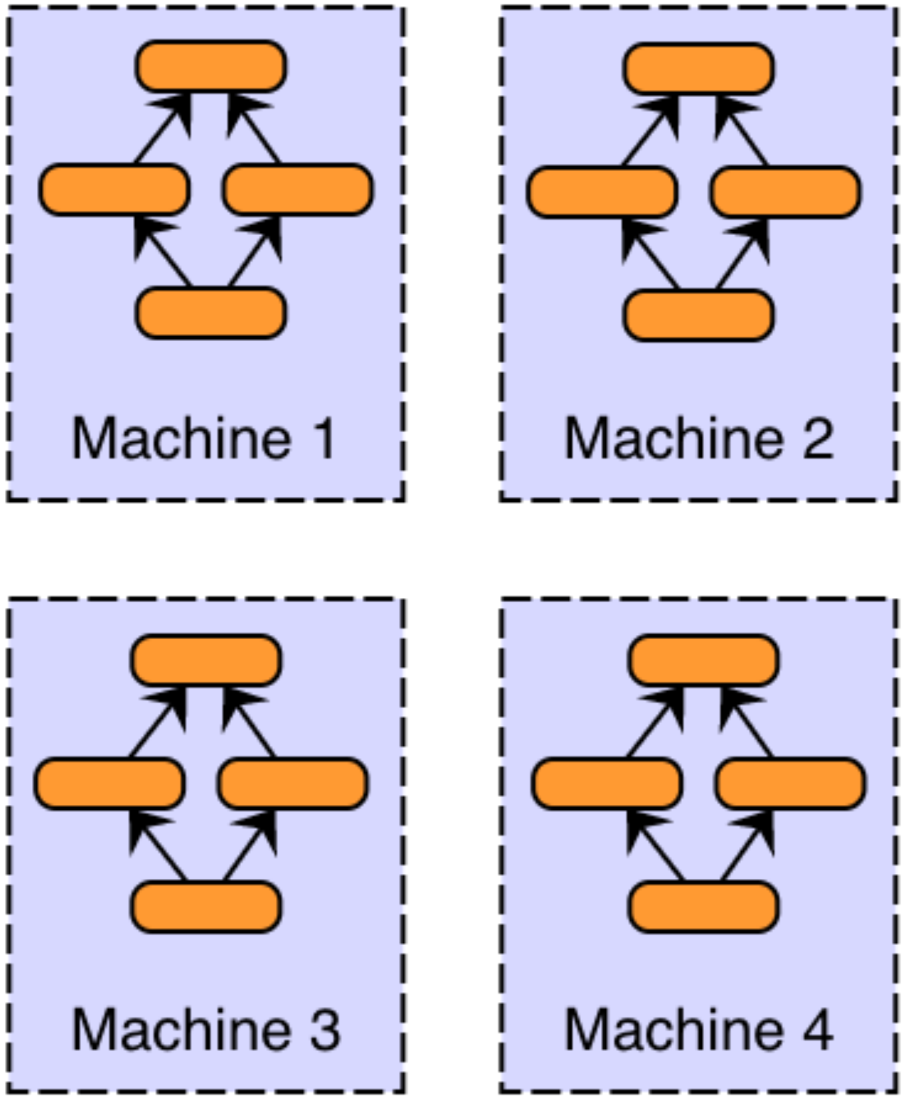
- The focus is on partitioning the computations rather than the data. The problem itself is decomposed and each task performs a portion of the overall work.
- Typically used when pieces of data require different processing times



# DISTRIBUTED DEEP NEURAL NETWORKS

# DOMAIN DECOMPOSITION (DATA PARALLELISM)

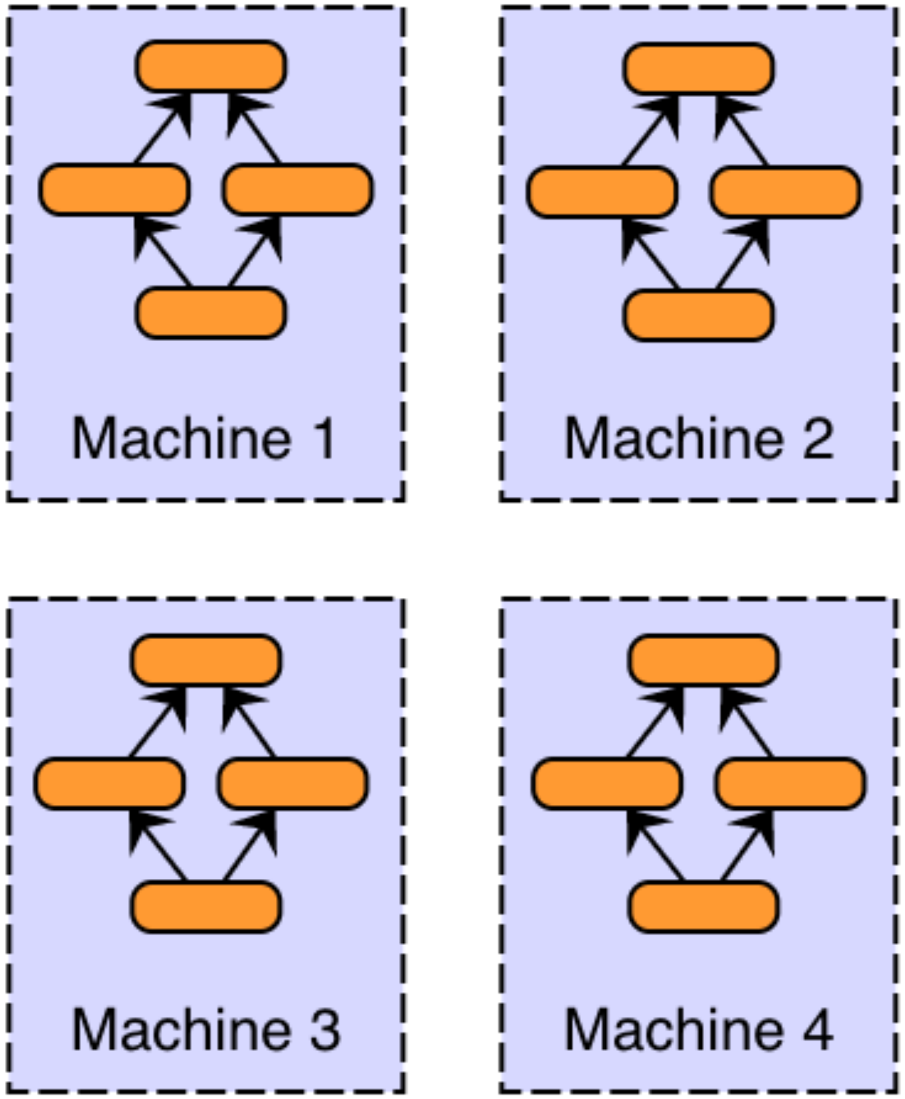
- This strategy is straightforward; partition the work of the batch across different machines.
- Different machines have a complete copy of the model.
- Each machine simply gets a different portion of the batch, and results from each are combined.



source: skymind

# DOMAIN DECOMPOSITION (DATA PARALLELISM)

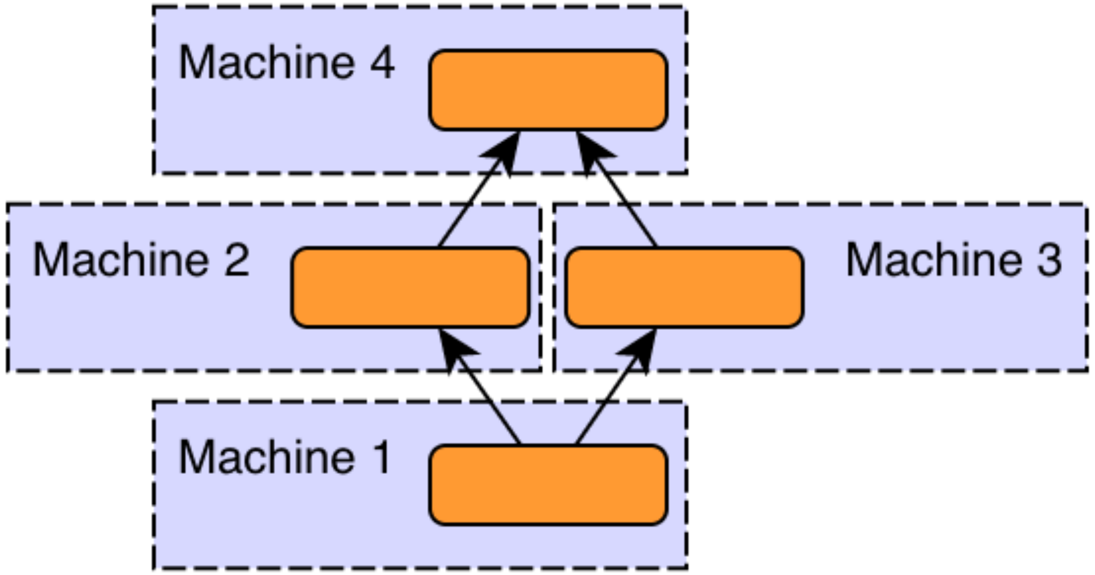
- Scaling the performance of data parallelism requires increasing the "effective batch size".
- The increased batch size can result in a decrease of the model's final accuracy.



source: skymind

# FUNCTIONAL DECOMPOSITION (MODEL PARALLELISM)

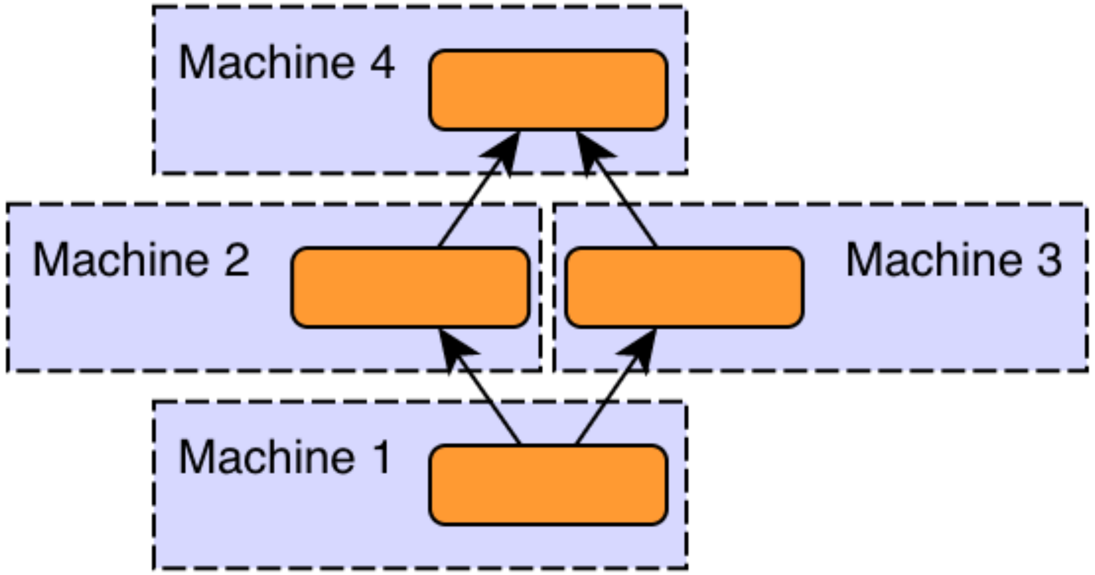
- This strategy divides the work according to the neurons in each layer.
- Different machines in the distributed system are responsible for the computations in different parts of a single network.
- For example, each layer in the neural network may be assigned to a different machine.



source: skymind

# FUNCTIONAL DECOMPOSITION (MODEL PARALLELISM)

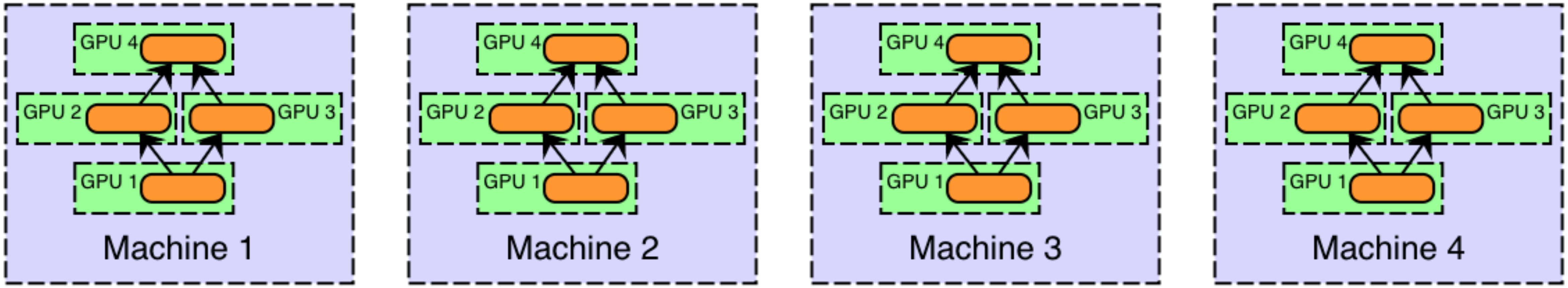
- While model parallelism can work well in practice, data parallelism is arguably the preferred approach for distributed systems and has been the focus of more research.
- For one thing, implementation, fault tolerance and good cluster utilization is easier for data parallelism than for model parallelism.
- Furthermore, model parallelism requires massive amount of communication.



source: skymind

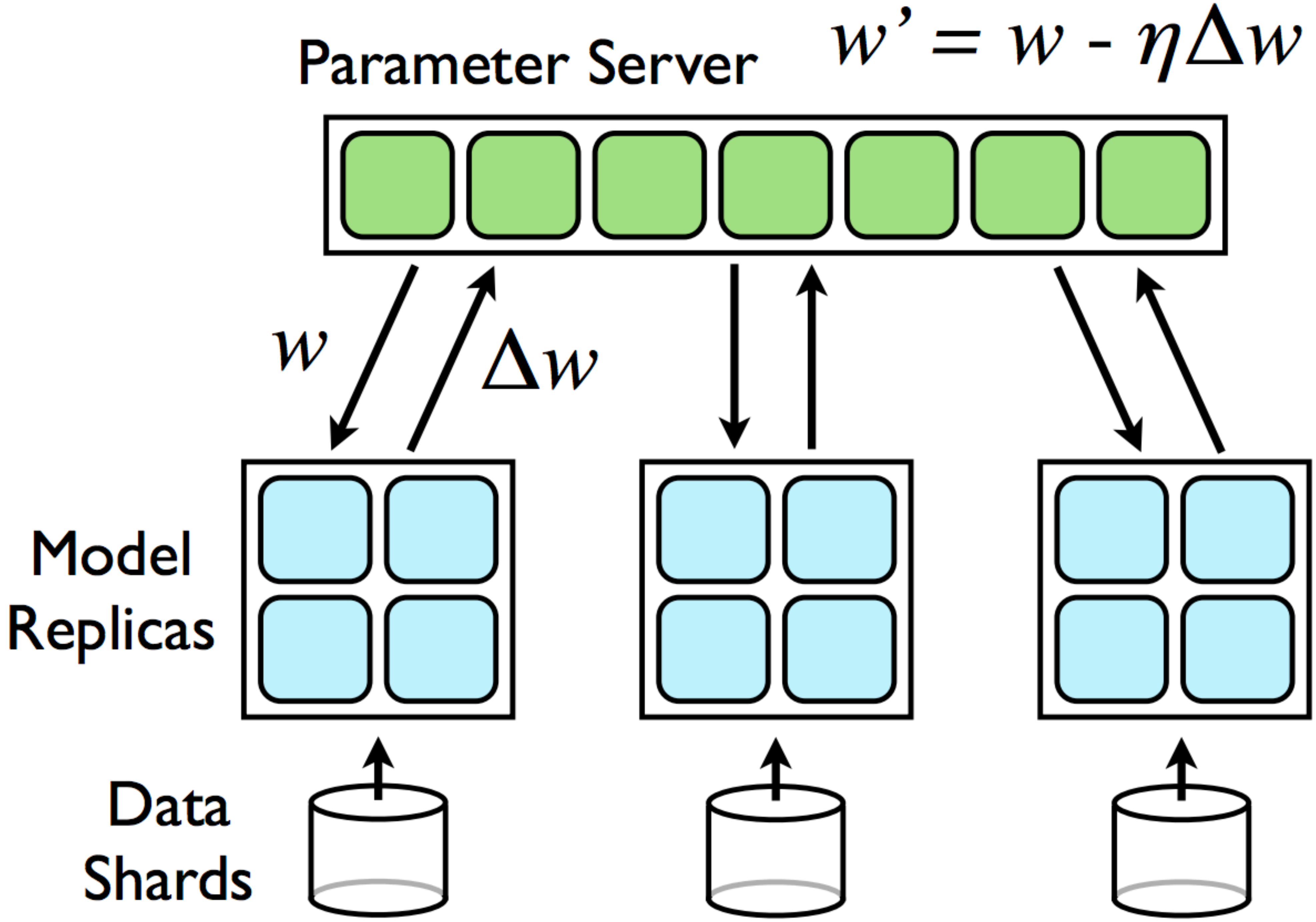
# HYBRID PARALLELISM

- The combination of multiple parallelism schemes can overcome the drawbacks of each scheme.



source: skymind

# DISTRIBUTED TRAINING

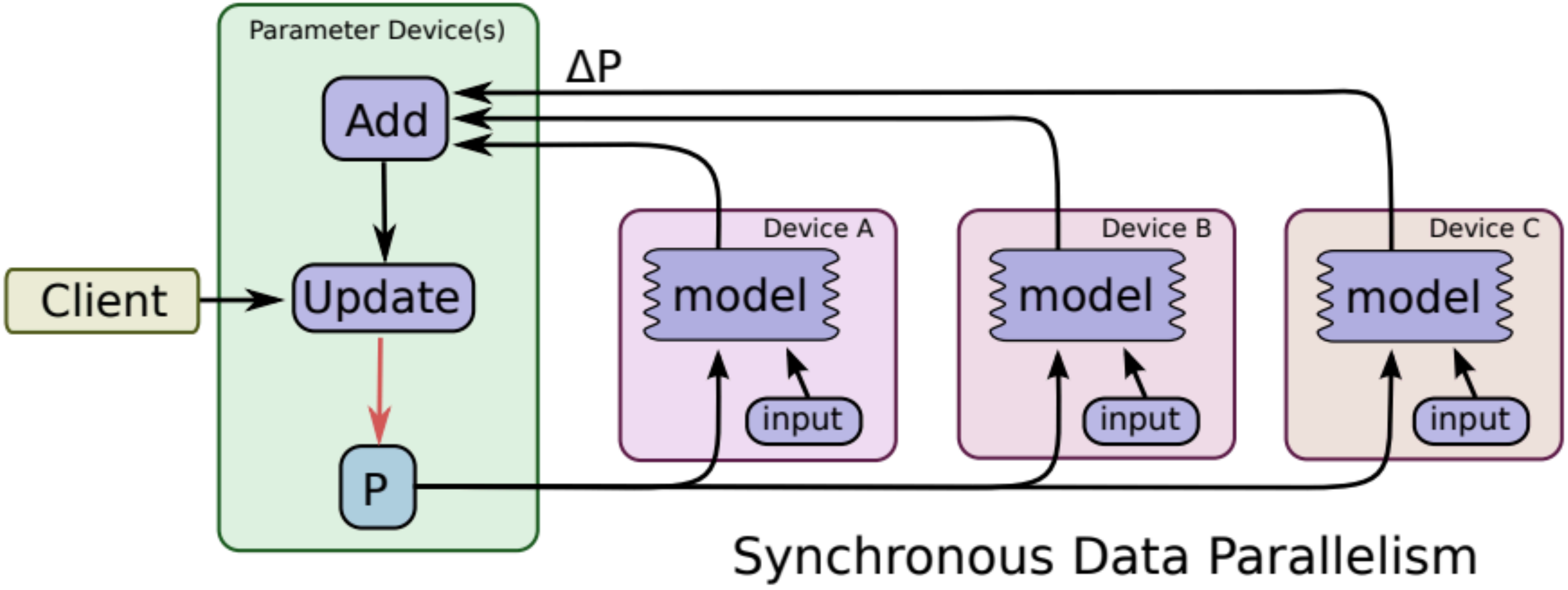


# SYNCHRONOUS TRAINING

# SYNCHRONOUS ALGORITHM

1. Each worker computes gradients on its part of the data.
2. Average gradients from all workers.
3. Update the model.

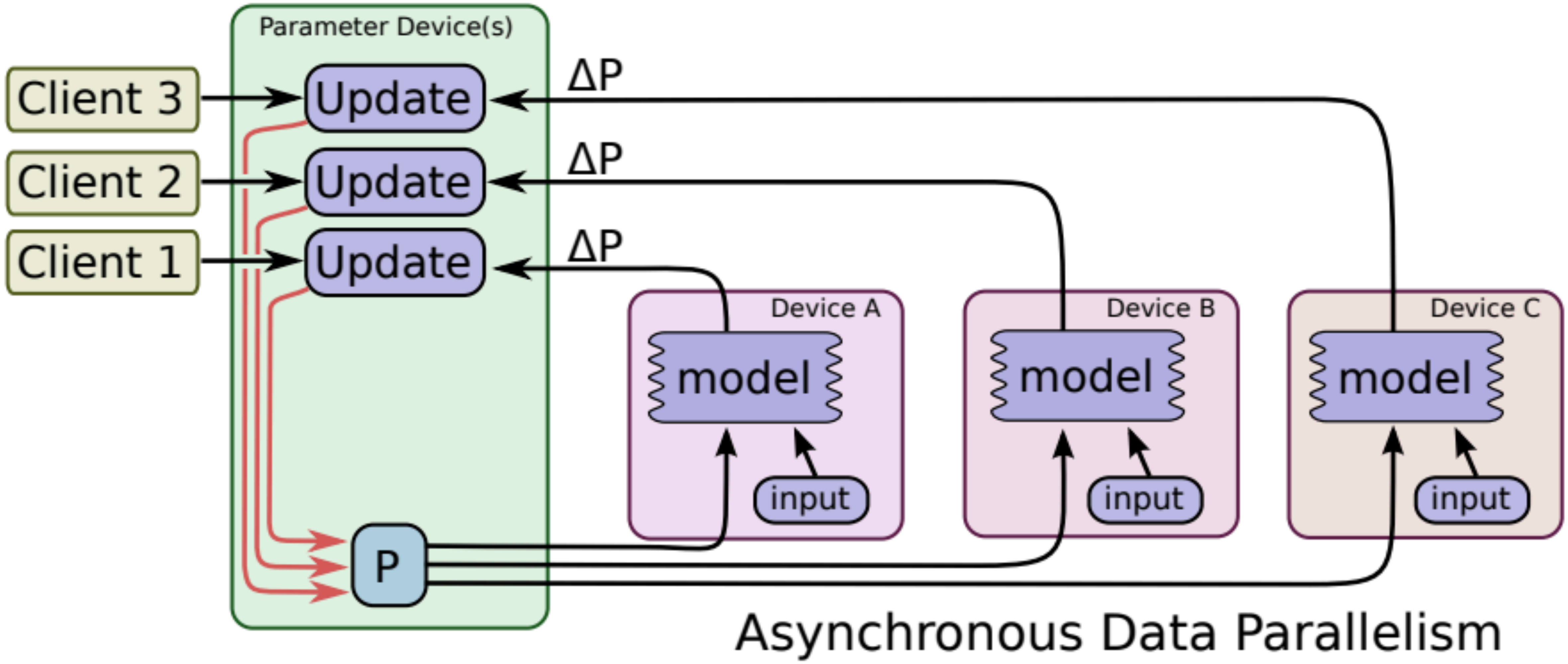
# SYNCHRONOUS SCHEME



source: “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems”

# ASYNCHRONOUS TRAINING

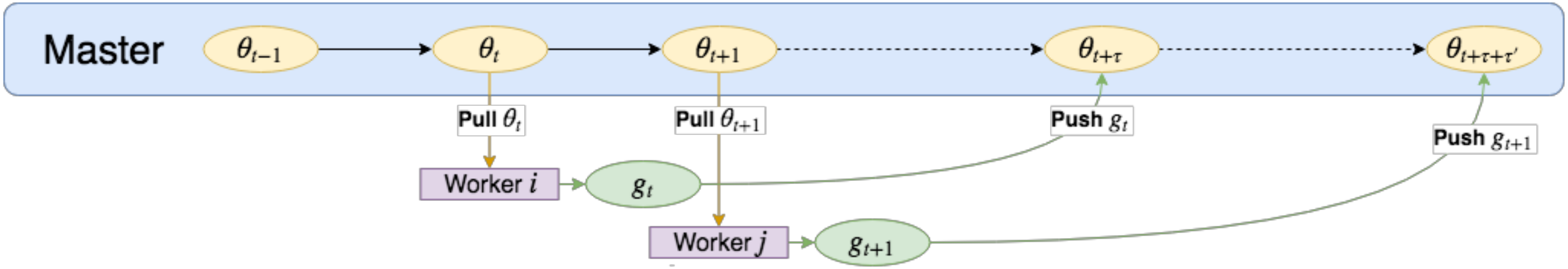
# ASYNCHRONOUS SCHEME



source: “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems”

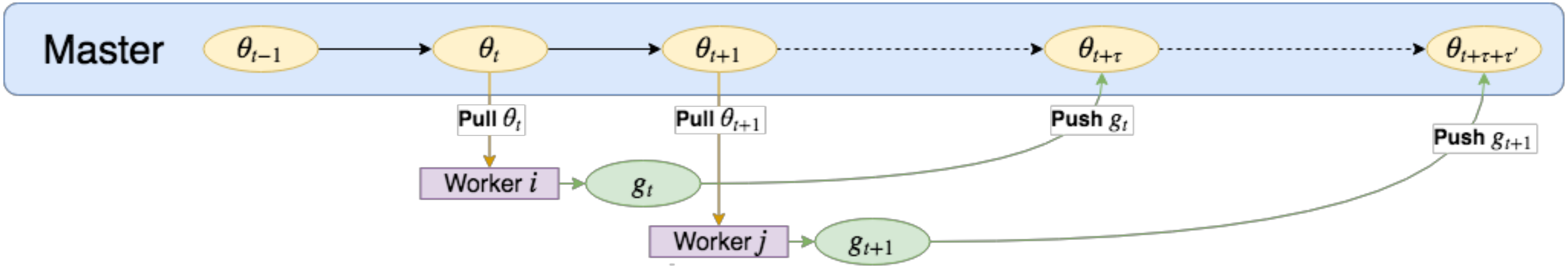
# GRADIENT STALENESS

- ASGD suffers from **gradient staleness**: gradients sent by workers are often based on parameters that are older than the master's (parameter server) current parameters.



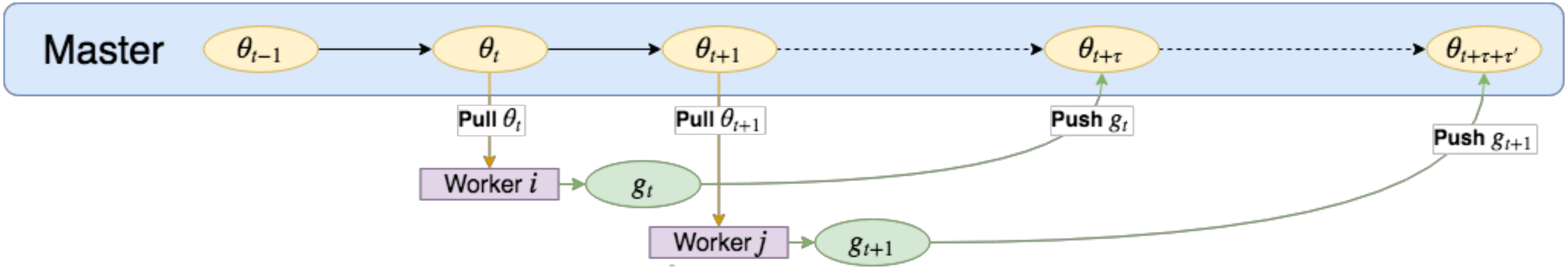
# GRADIENT STALENESS

- This gradient staleness is a major obstacle to scaling ASGD since the it grows as we increase the number of workers, which decreases gradient accuracy, and ultimately reduces the accuracy of the trained model.



# STALENESS AWARENESS

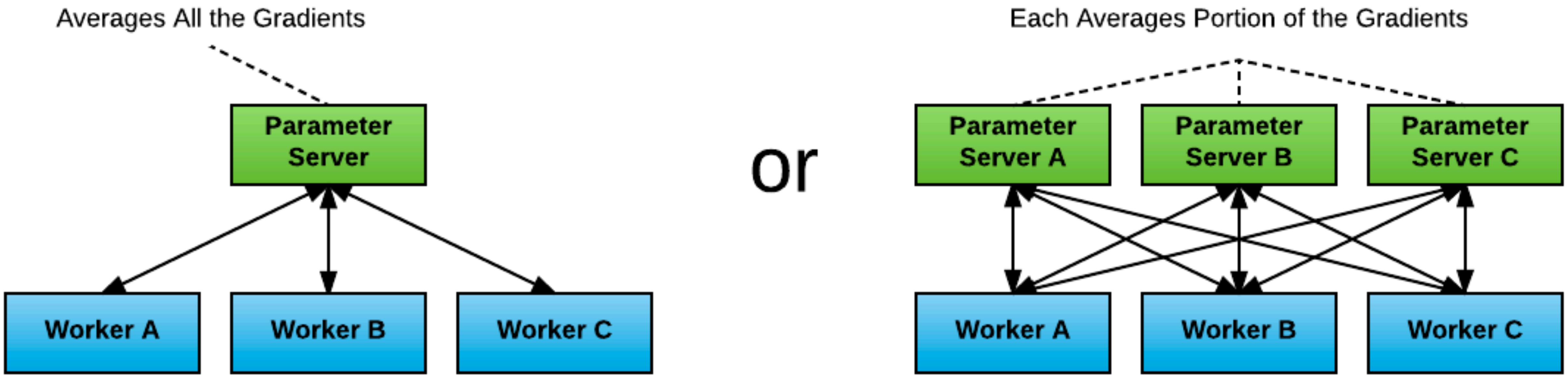
- Adjust the learning rate to the staleness magnitude.
- **Softsync** - Instead of updating the parameters immediately, the master waits to collect a number of updates from any of workers, and only then updates the parameters.



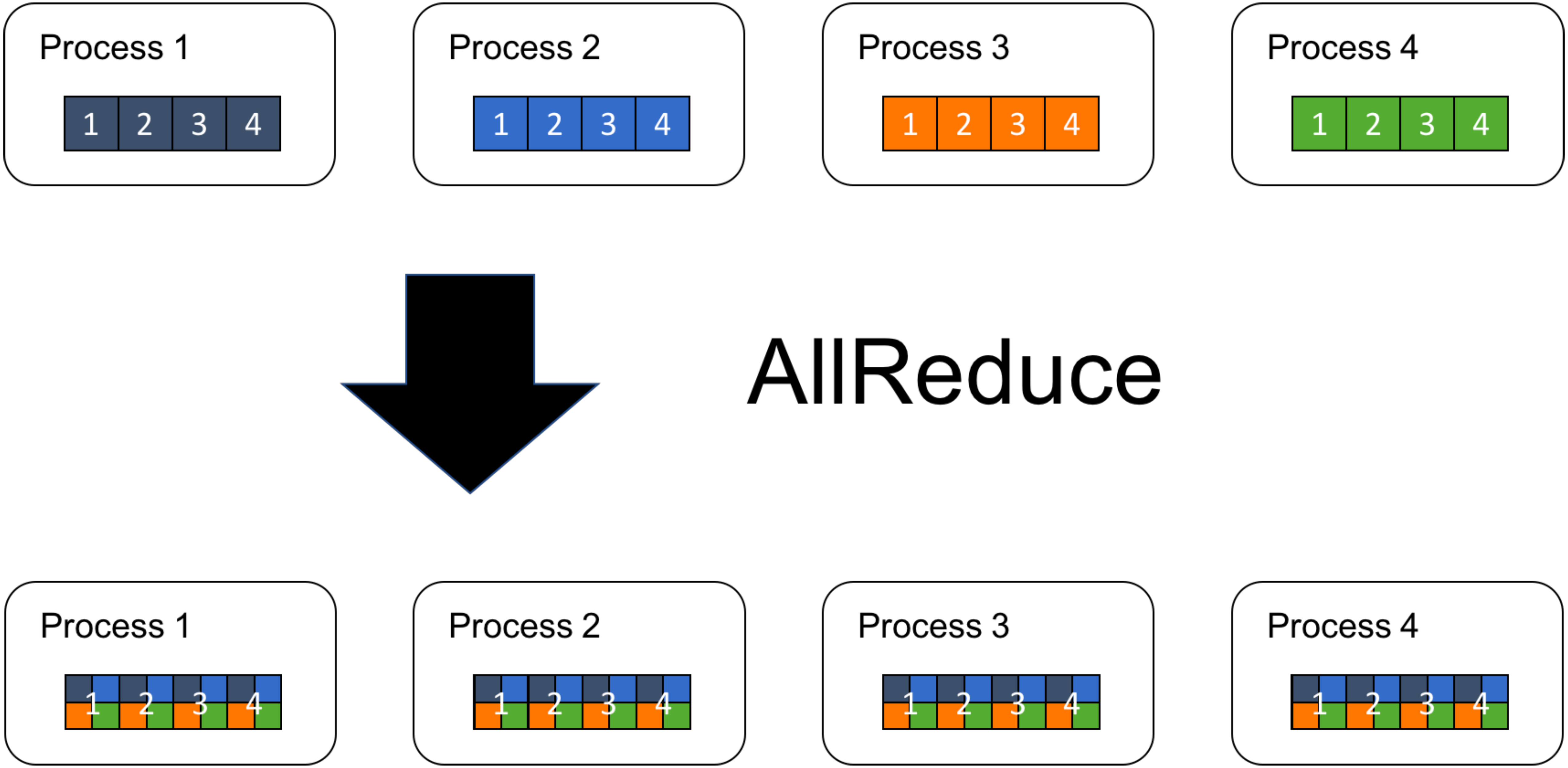
# PARAMETER SERVER

# SHARDING

- If one parameter server is used, it will likely become a networking or computational bottleneck.

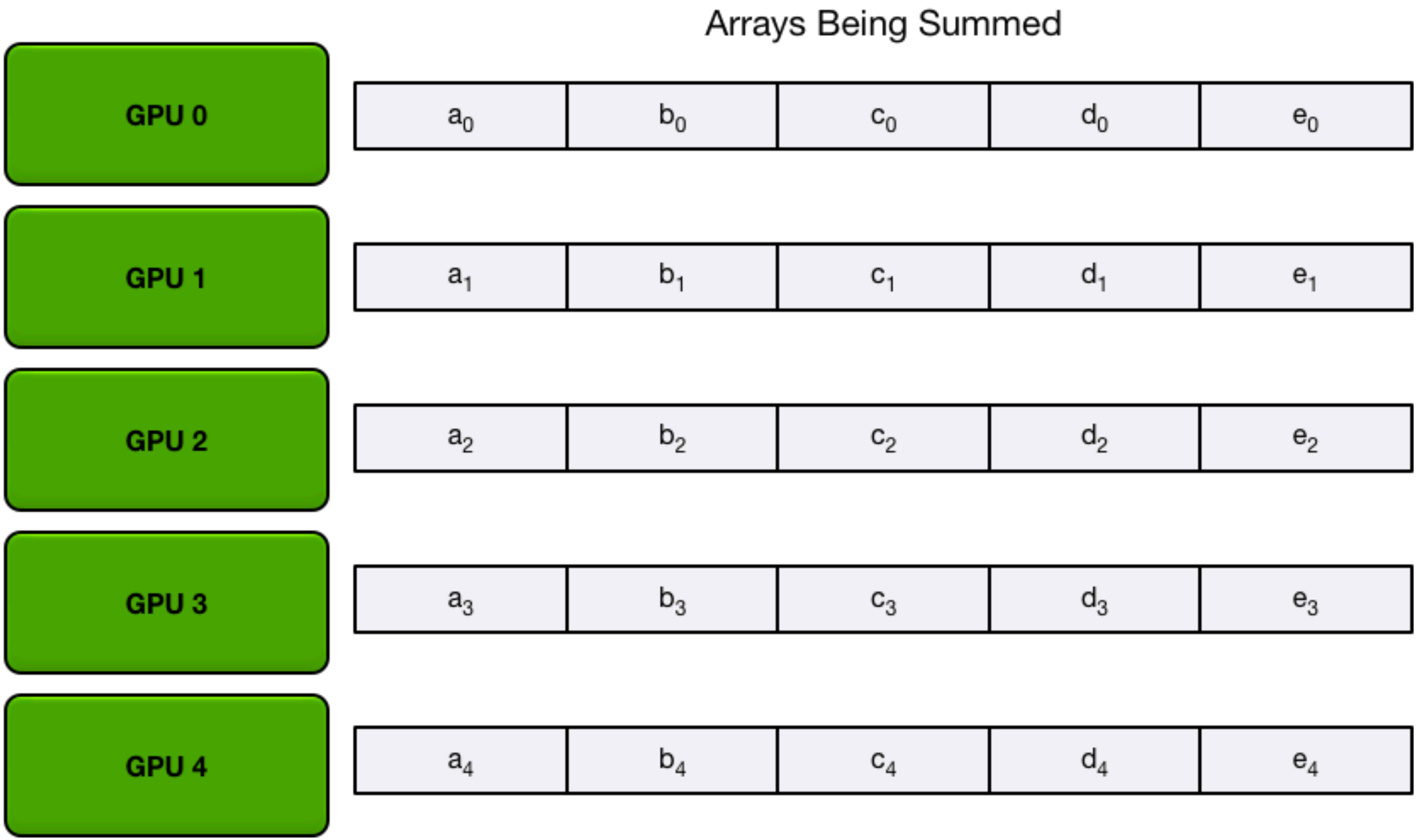


# ALLREDUCE



source: https://preferredresearch.jp/2018/07/10/technologies-behind-distributed-deep-learning-allreduce/

# RING-ALLREDUCE

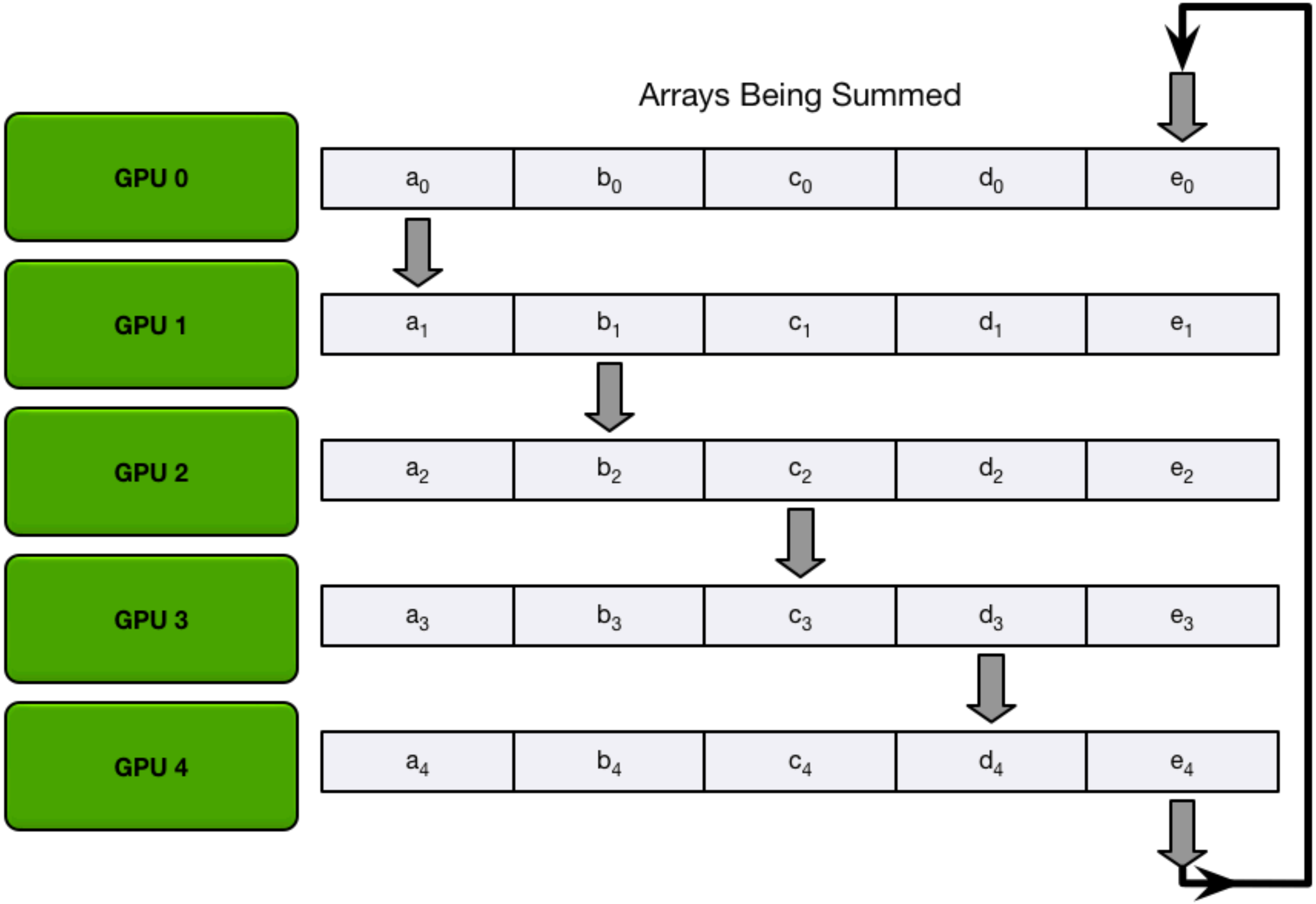


Arrays Being Summed

|  |  |  |  |  |  |
|---|---|---|---|---|---|
| GPU 0 | $a_0$ | $b_0$ | $c_0$ | $d_0$ | $e_0$ |
| GPU 1 | $a_1$ | $b_1$ | $c_1$ | $d_1$ | $e_1$ |
| GPU 2 | $a_2$ | $b_2$ | $c_2$ | $d_2$ | $e_2$ |
| GPU 3 | $a_3$ | $b_3$ | $c_3$ | $d_3$ | $e_3$ |
| GPU 4 | $a_4$ | $b_4$ | $c_4$ | $d_4$ | $e_4$ |

source: http://andrew.gibiansky.com/

# RING-ALLREDUCE

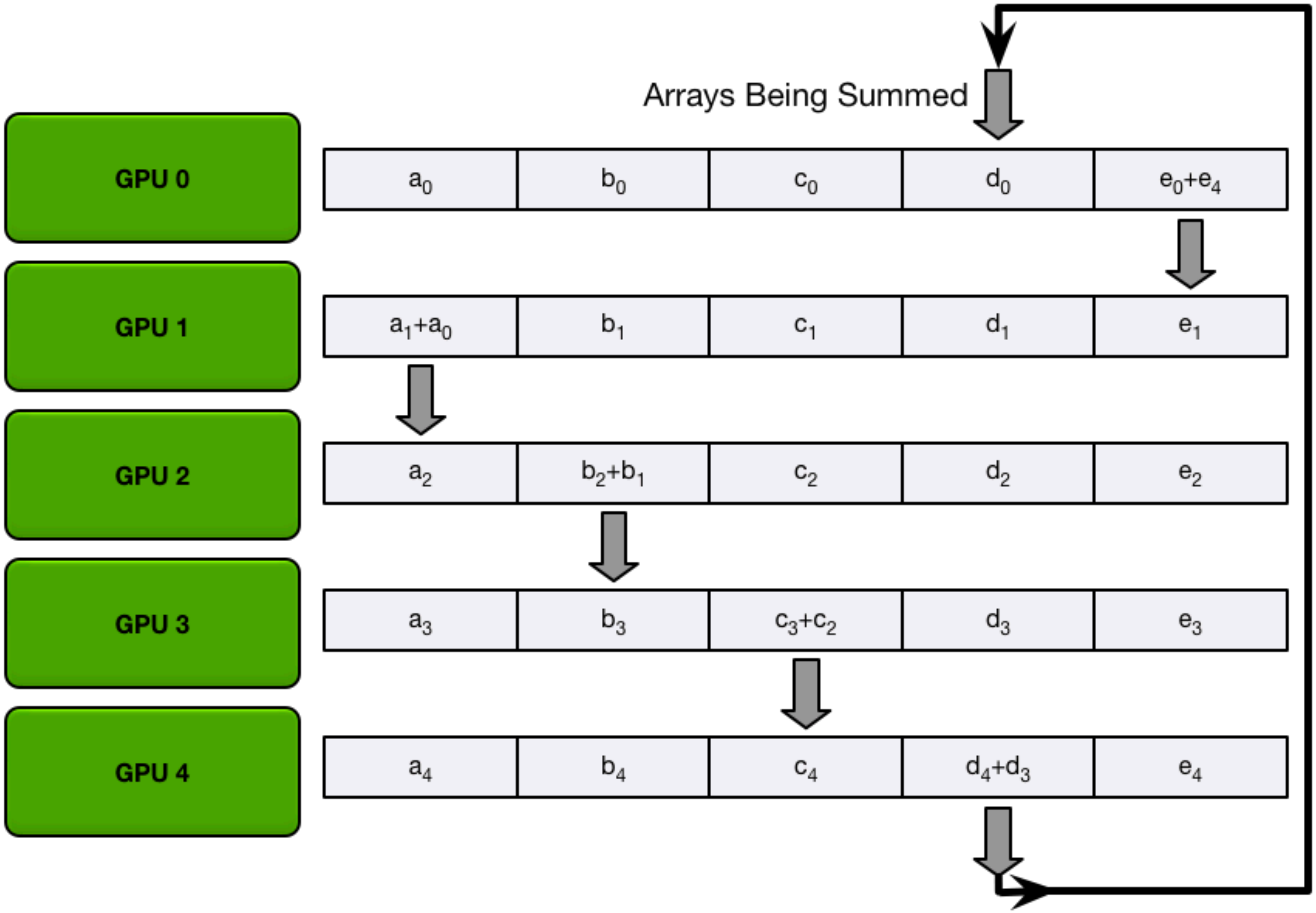


Arrays Being Summed

| | | | | | |
|---|---|---|---|---|---|
| GPU 0 | $a_0$ | $b_0$ | $c_0$ | $d_0$ | $e_0$ |
| GPU 1 | $a_1$ | $b_1$ | $c_1$ | $d_1$ | $e_1$ |
| GPU 2 | $a_2$ | $b_2$ | $c_2$ | $d_2$ | $e_2$ |
| GPU 3 | $a_3$ | $b_3$ | $c_3$ | $d_3$ | $e_3$ |
| GPU 4 | $a_4$ | $b_4$ | $c_4$ | $d_4$ | $e_4$ |

source: http://andrew.gibiansky.com/

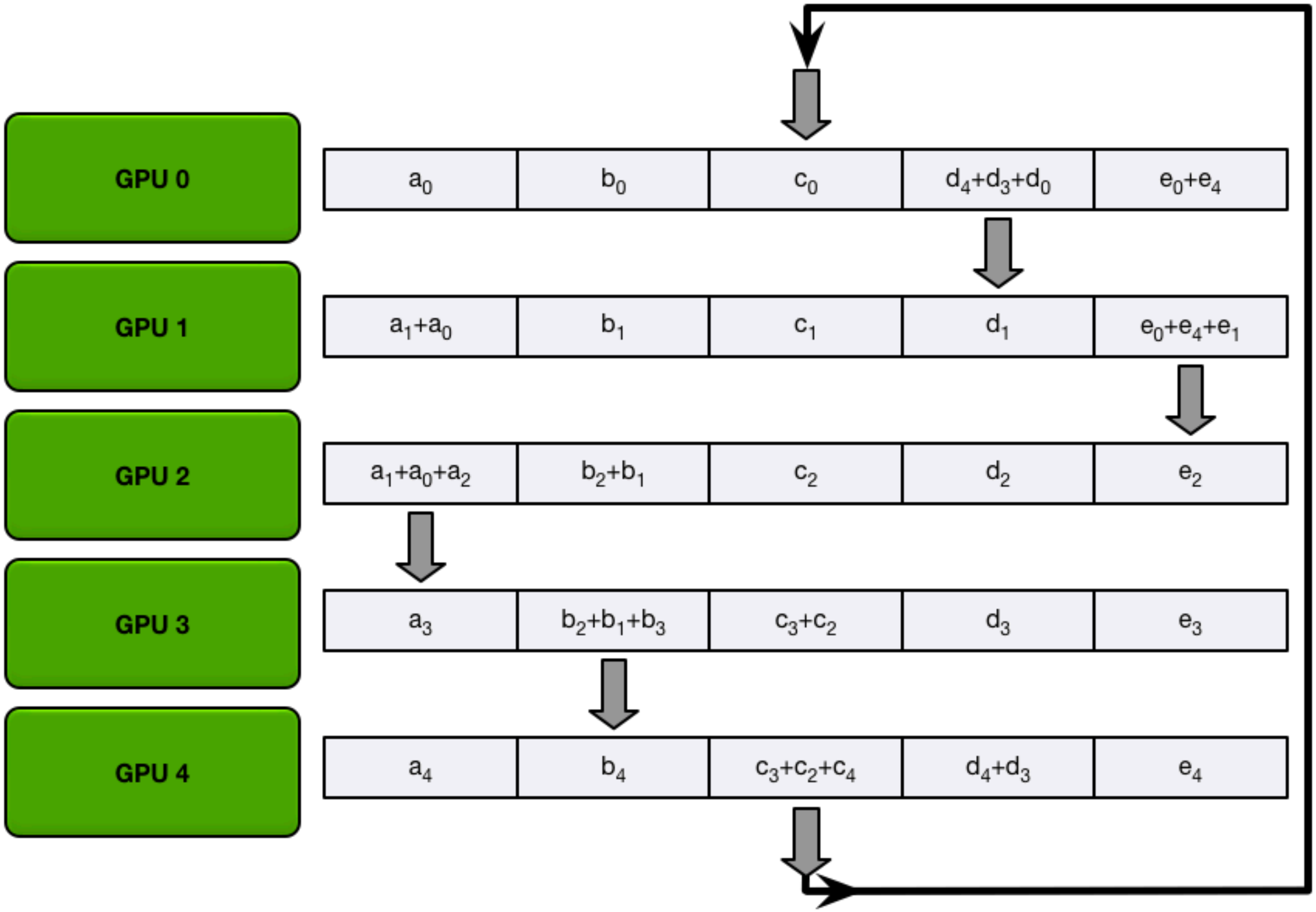
# RING-ALLREDUCE



| | | | | | |
|---|---|---|---|---|---|
| GPU 0 | $a_0$ | $b_0$ | $c_0$ | $d_0$ | $e_0+e_4$ |
| GPU 1 | $a_1+a_0$ | $b_1$ | $c_1$ | $d_1$ | $e_1$ |
| GPU 2 | $a_2$ | $b_2+b_1$ | $c_2$ | $d_2$ | $e_2$ |
| GPU 3 | $a_3$ | $b_3$ | $c_3+c_2$ | $d_3$ | $e_3$ |
| GPU 4 | $a_4$ | $b_4$ | $c_4$ | $d_4+d_3$ | $e_4$ |

source: http://andrew.gibiansky.com/

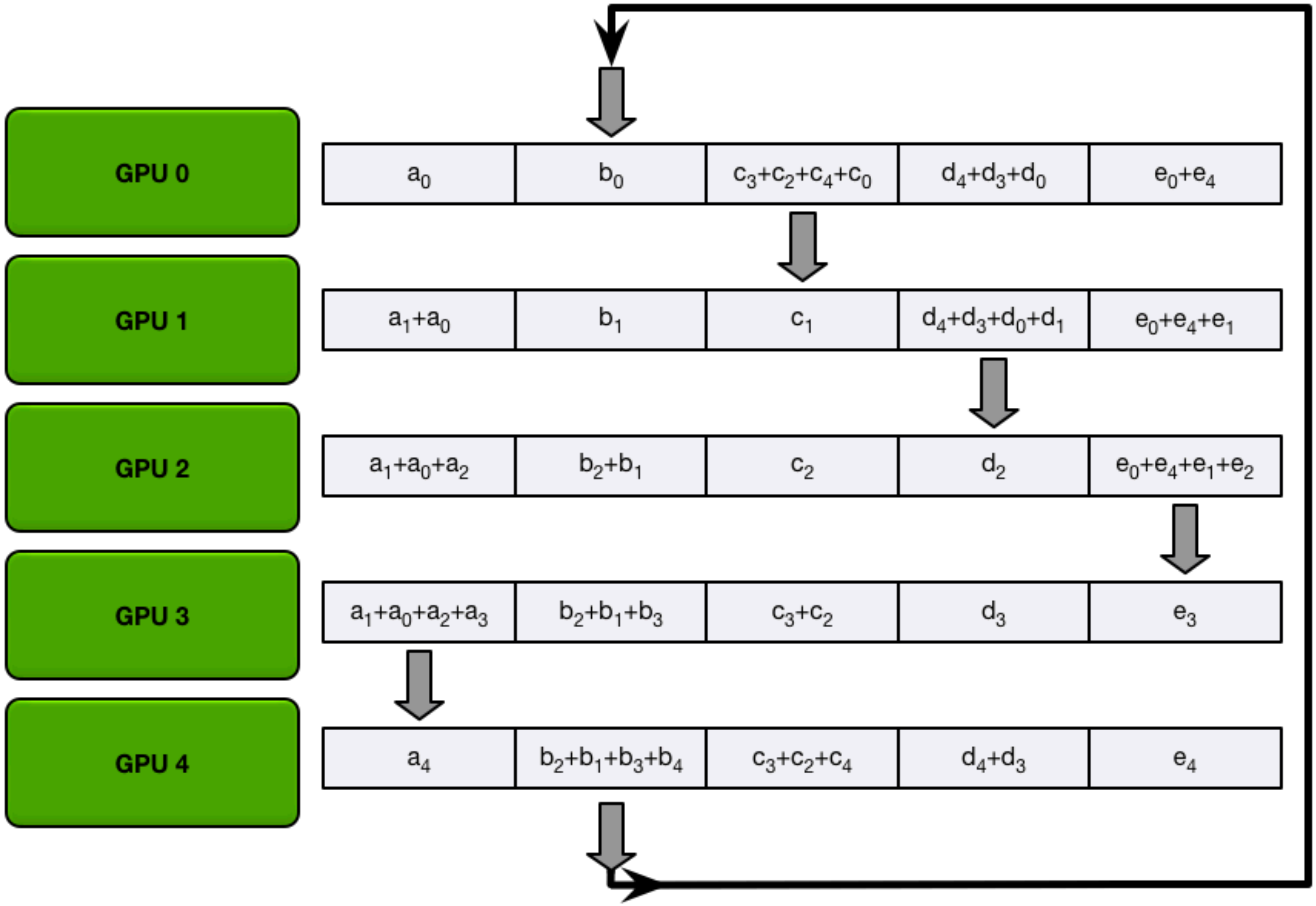
# RING-ALLREDUCE



| | | | | | |
|---|---|---|---|---|---|
| GPU 0 | $a_0$ | $b_0$ | $c_0$ | $d_4+d_3+d_0$ | $e_0+e_4$ |
| GPU 1 | $a_1+a_0$ | $b_1$ | $c_1$ | $d_1$ | $e_0+e_4+e_1$ |
| GPU 2 | $a_1+a_0+a_2$ | $b_2+b_1$ | $c_2$ | $d_2$ | $e_2$ |
| GPU 3 | $a_3$ | $b_2+b_1+b_3$ | $c_3+c_2$ | $d_3$ | $e_3$ |
| GPU 4 | $a_4$ | $b_4$ | $c_3+c_2+c_4$ | $d_4+d_3$ | $e_4$ |

source: http://andrew.gibiansky.com/

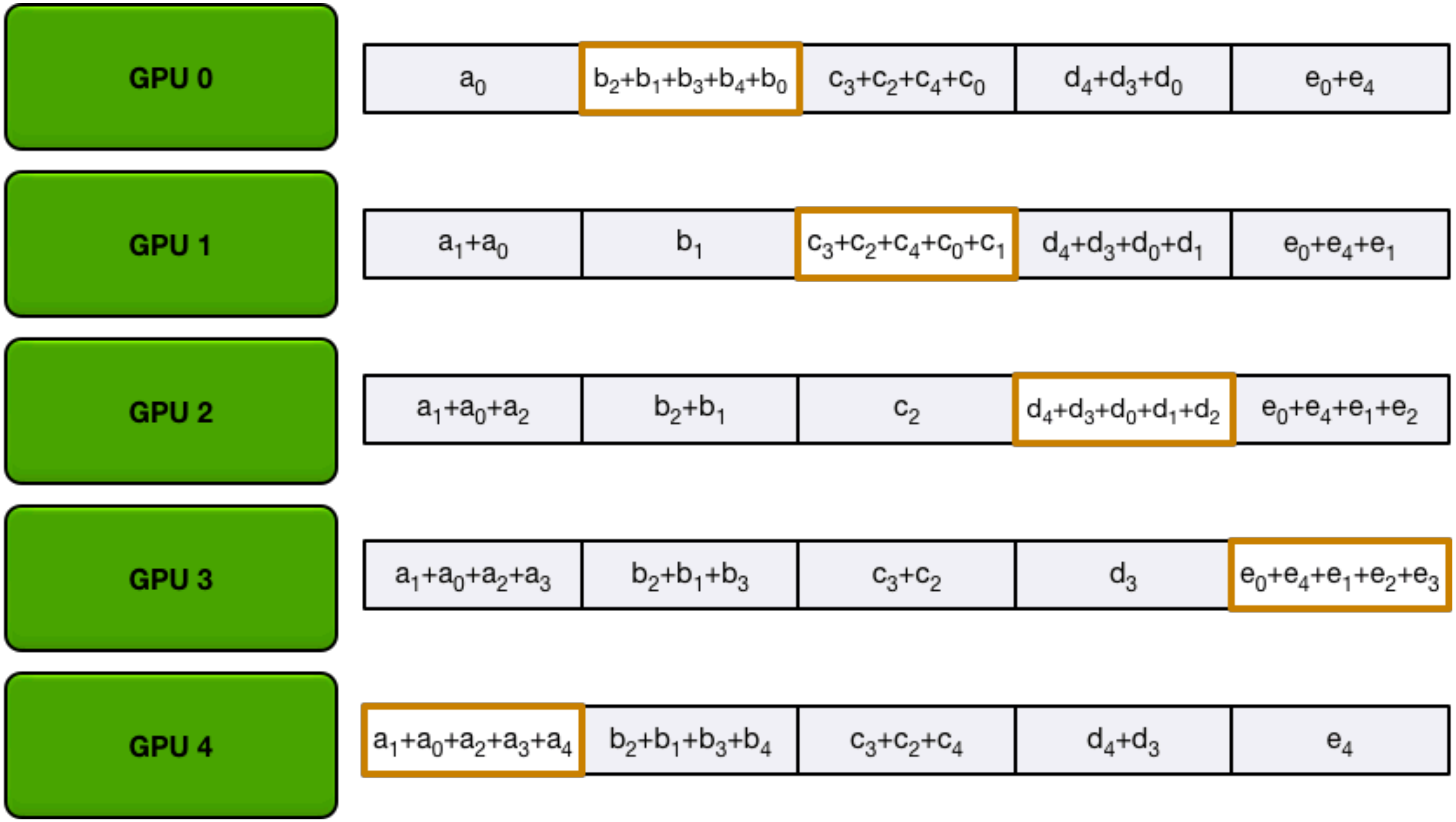
# RING-ALLREDUCE



| | | | | | |
|---|---|---|---|---|---|
| GPU 0 | $a_0$ | $b_0$ | $c_3+c_2+c_4+c_0$ | $d_4+d_3+d_0$ | $e_0+e_4$ |
| GPU 1 | $a_1+a_0$ | $b_1$ | $c_1$ | $d_4+d_3+d_0+d_1$ | $e_0+e_4+e_1$ |
| GPU 2 | $a_1+a_0+a_2$ | $b_2+b_1$ | $c_2$ | $d_2$ | $e_0+e_4+e_1+e_2$ |
| GPU 3 | $a_1+a_0+a_2+a_3$ | $b_2+b_1+b_3$ | $c_3+c_2$ | $d_3$ | $e_3$ |
| GPU 4 | $a_4$ | $b_2+b_1+b_3+b_4$ | $c_3+c_2+c_4$ | $d_4+d_3$ | $e_4$ |

source: http://andrew.gibiansky.com/

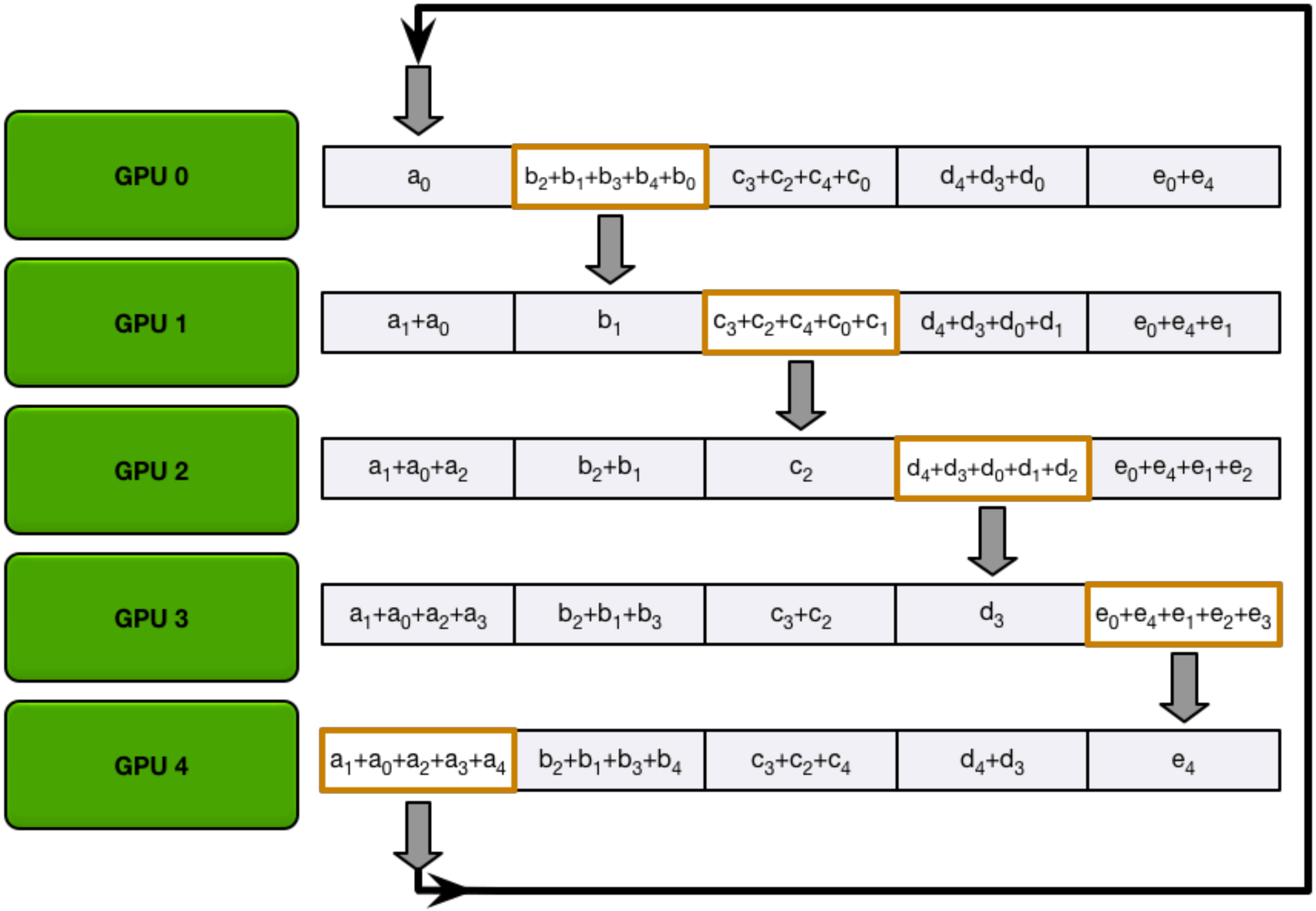
# RING-ALLREDUCE



| | | | | | |
|---|---|---|---|---|---|
| GPU 0 | $a_0$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0$ | $d_4+d_3+d_0$ | $e_0+e_4$ |
| GPU 1 | $a_1+a_0$ | $b_1$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1$ | $e_0+e_4+e_1$ |
| GPU 2 | $a_1+a_0+a_2$ | $b_2+b_1$ | $c_2$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2$ |
| GPU 3 | $a_1+a_0+a_2+a_3$ | $b_2+b_1+b_3$ | $c_3+c_2$ | $d_3$ | $e_0+e_4+e_1+e_2+e_3$ |
| GPU 4 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4$ | $c_3+c_2+c_4$ | $d_4+d_3$ | $e_4$ |

source: http://andrew.gibiansky.com/

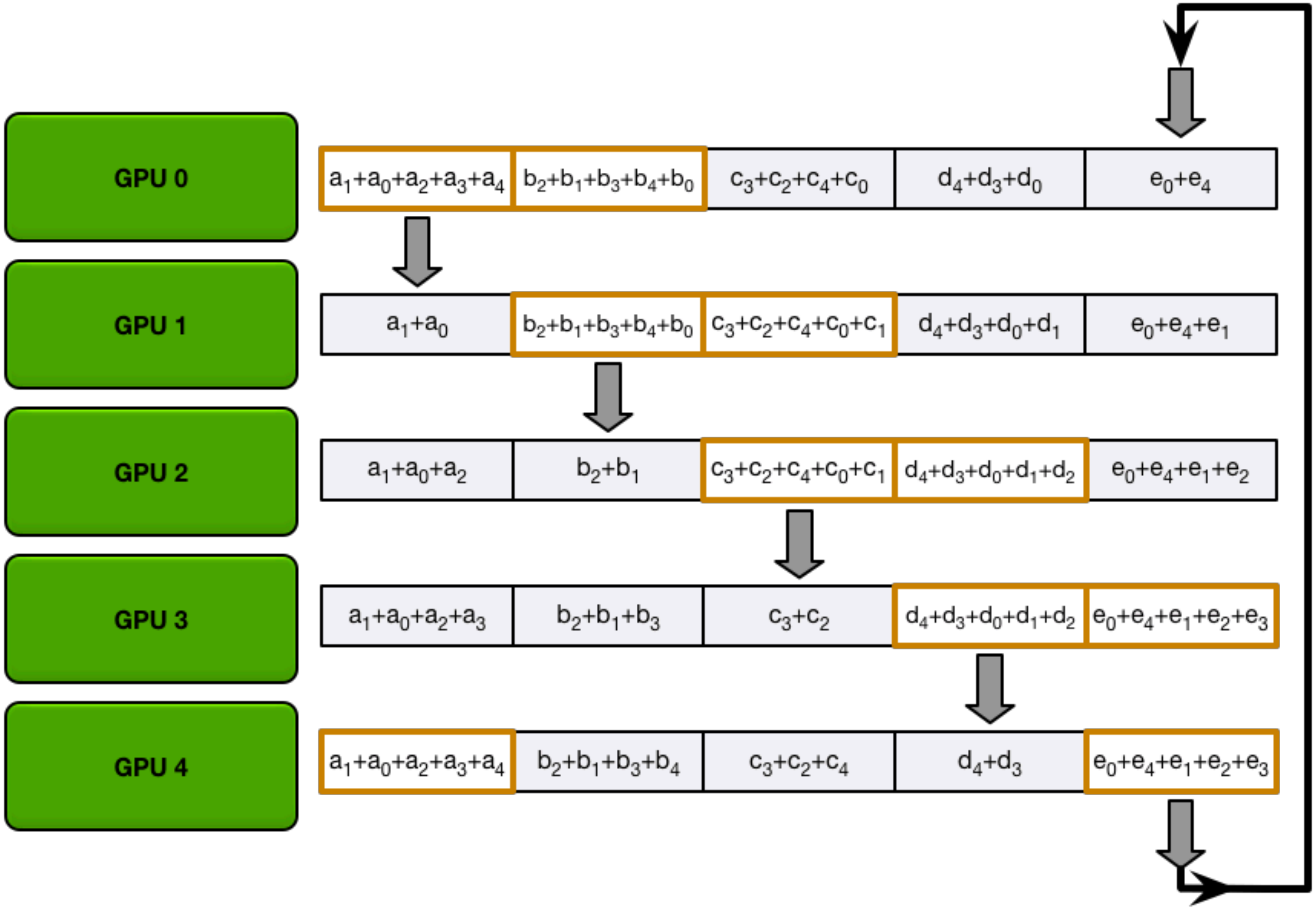
# RING-ALLREDUCE



| | | | | | |
|---|---|---|---|---|---|
| GPU 0 | $a_0$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0$ | $d_4+d_3+d_0$ | $e_0+e_4$ |
| GPU 1 | $a_1+a_0$ | $b_1$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1$ | $e_0+e_4+e_1$ |
| GPU 2 | $a_1+a_0+a_2$ | $b_2+b_1$ | $c_2$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2$ |
| GPU 3 | $a_1+a_0+a_2+a_3$ | $b_2+b_1+b_3$ | $c_3+c_2$ | $d_3$ | $e_0+e_4+e_1+e_2+e_3$ |
| GPU 4 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4$ | $c_3+c_2+c_4$ | $d_4+d_3$ | $e_4$ |

source: http://andrew.gibiansky.com/

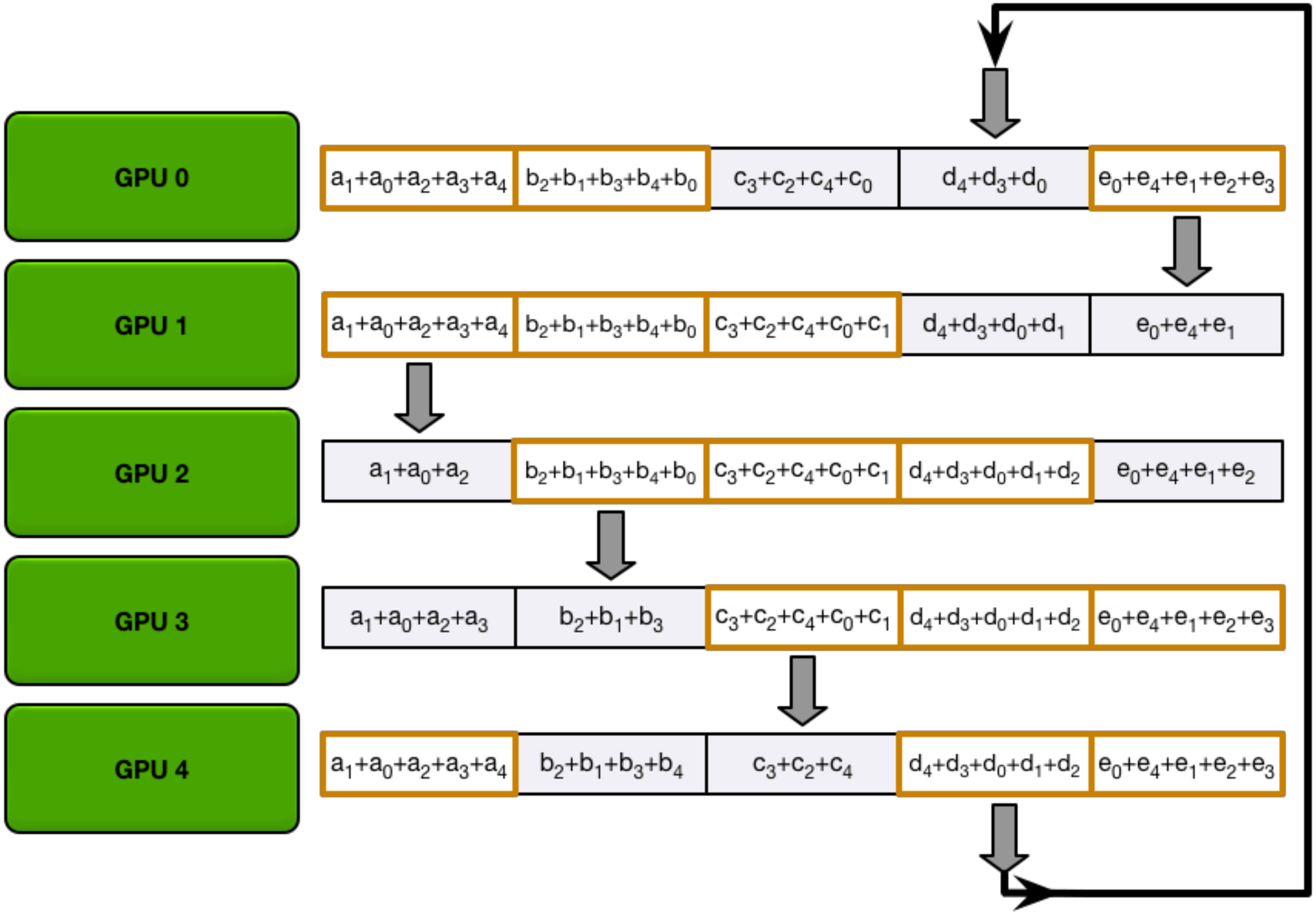
# RING-ALLREDUCE



| | | | | | |
|---|---|---|---|---|---|
| GPU 0 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0$ | $d_4+d_3+d_0$ | $e_0+e_4$ |
| GPU 1 | $a_1+a_0$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1$ | $e_0+e_4+e_1$ |
| GPU 2 | $a_1+a_0+a_2$ | $b_2+b_1$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2$ |
| GPU 3 | $a_1+a_0+a_2+a_3$ | $b_2+b_1+b_3$ | $c_3+c_2$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2+e_3$ |
| GPU 4 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4$ | $c_3+c_2+c_4$ | $d_4+d_3$ | $e_0+e_4+e_1+e_2+e_3$ |

source: http://andrew.gibiansky.com/

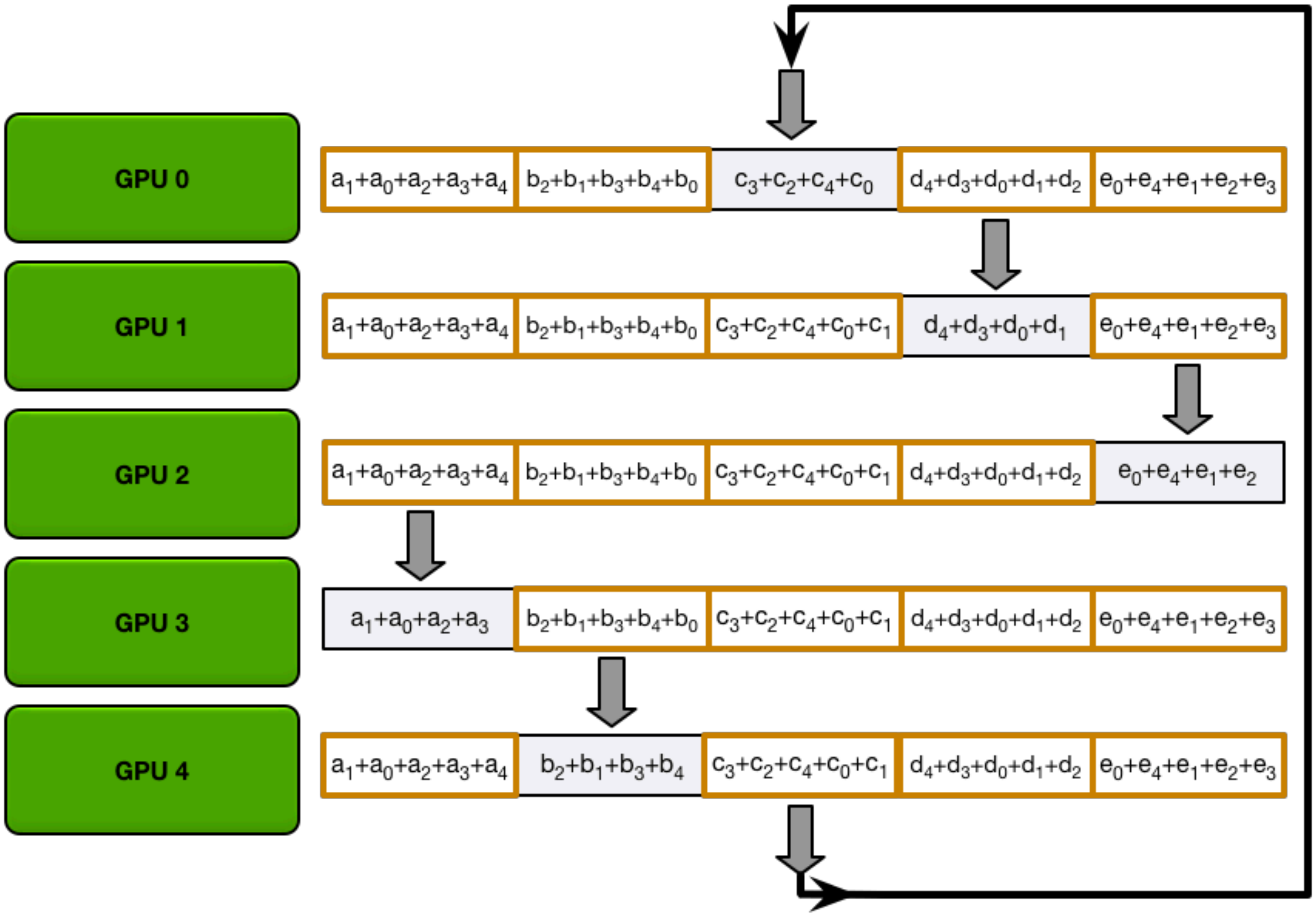
# RING-ALLREDUCE



| | | | | | |
|---|---|---|---|---|---|
| GPU 0 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0$ | $d_4+d_3+d_0$ | $e_0+e_4+e_1+e_2+e_3$ |
| GPU 1 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1$ | $e_0+e_4+e_1$ |
| GPU 2 | $a_1+a_0+a_2$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2$ |
| GPU 3 | $a_1+a_0+a_2+a_3$ | $b_2+b_1+b_3$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2+e_3$ |
| GPU 4 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4$ | $c_3+c_2+c_4$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2+e_3$ |

source: http://andrew.gibiansky.com/

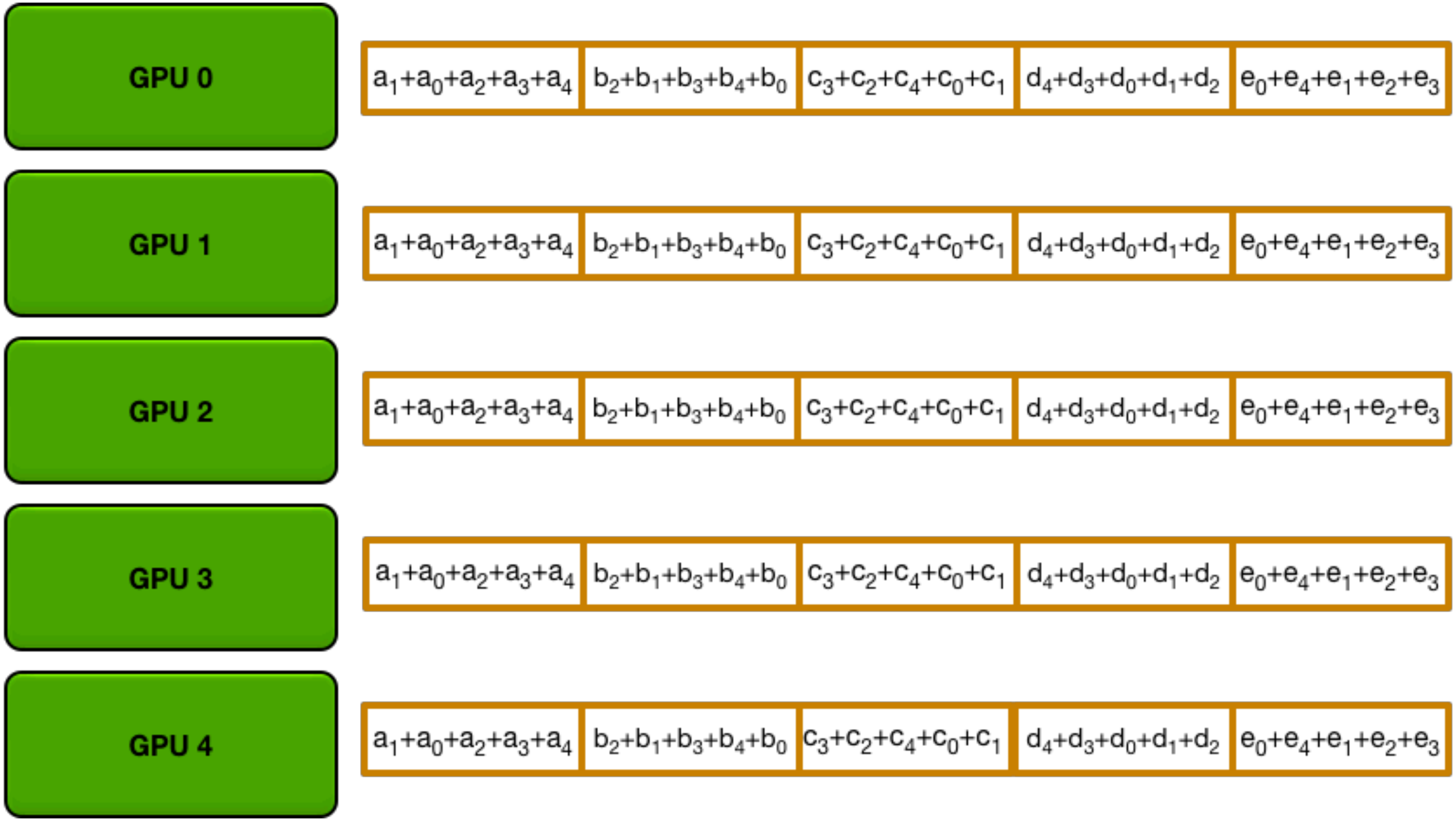
# RING-ALLREDUCE



| | | | | | |
|---|---|---|---|---|---|
| GPU 0 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2+e_3$ |
| GPU 1 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1$ | $e_0+e_4+e_1+e_2+e_3$ |
| GPU 2 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2$ |
| GPU 3 | $a_1+a_0+a_2+a_3$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2+e_3$ |
| GPU 4 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2+e_3$ |

source: http://andrew.gibiansky.com/

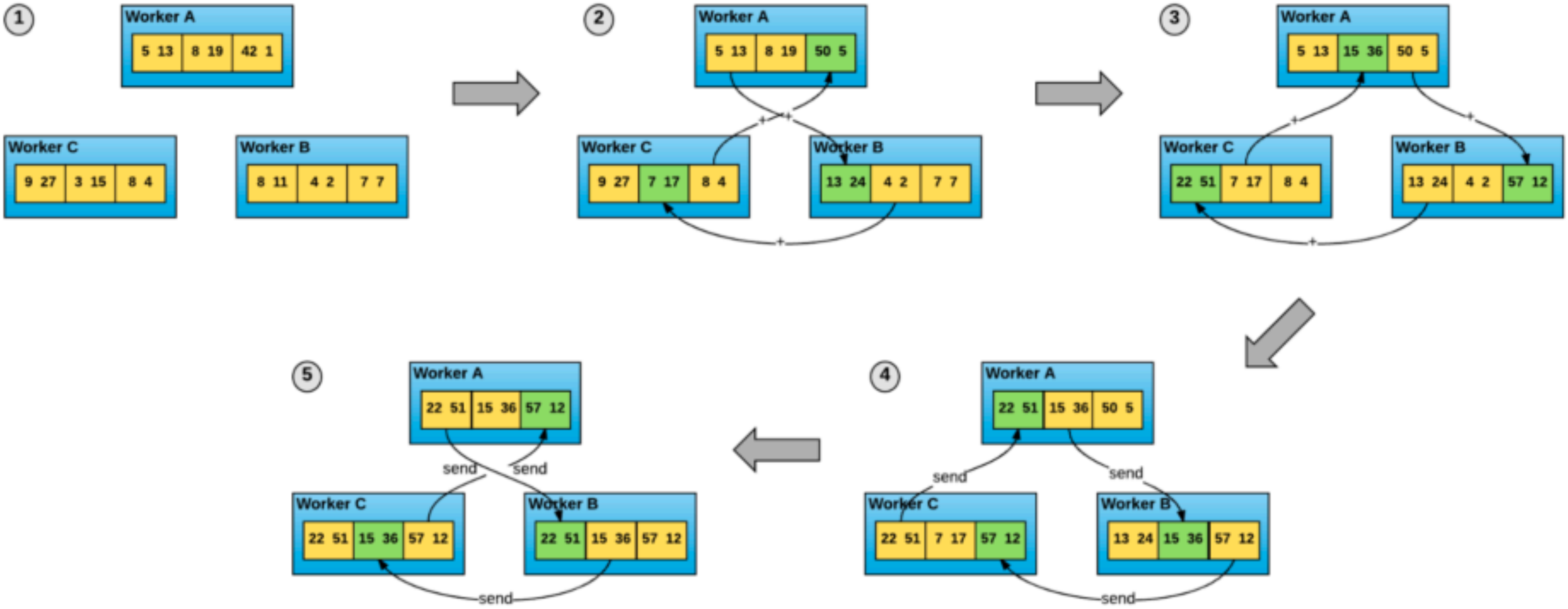
# RING-ALLREDUCE



| | | | | | |
|---|---|---|---|---|---|
| GPU 0 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2+e_3$ |
| GPU 1 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2+e_3$ |
| GPU 2 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2+e_3$ |
| GPU 3 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2+e_3$ |
| GPU 4 | $a_1+a_0+a_2+a_3+a_4$ | $b_2+b_1+b_3+b_4+b_0$ | $c_3+c_2+c_4+c_0+c_1$ | $d_4+d_3+d_0+d_1+d_2$ | $e_0+e_4+e_1+e_2+e_3$ |

source: http://andrew.gibiansky.com/

# RING-ALLREDUCE EXAMPLE

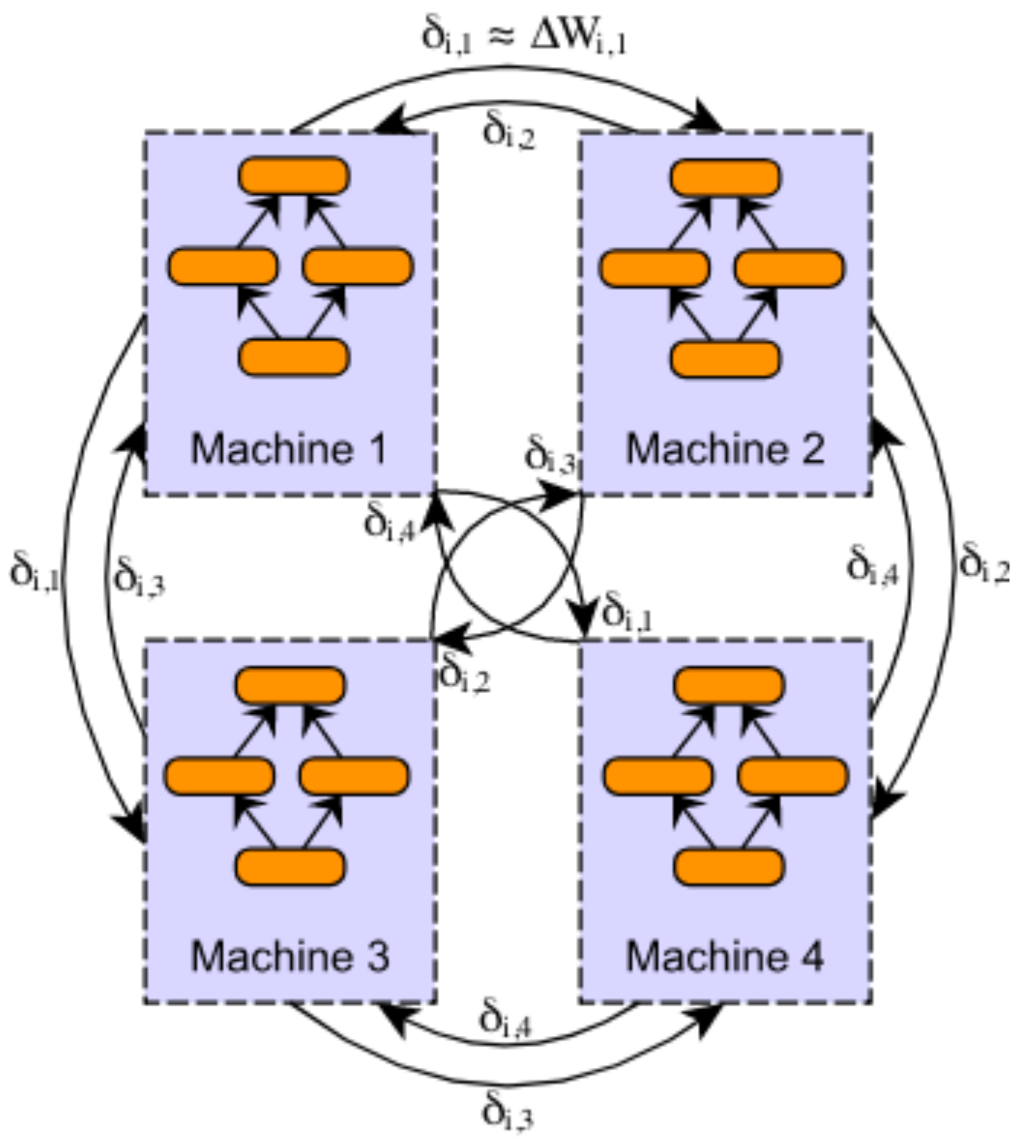


source: https://eng.uber.com/horovod/

# DECENTRALIZED TRAINING

# DECENTRALIZED SCHEME

- One bottleneck of centralized algorithms lies on high communication cost to the parameter server.
- No centralized parameter server is present in the system. Instead a peer to peer communication is used to transmit model updates between workers.


$\delta_{i,1} \approx \Delta W_{i,1}$
$\delta_{i,2}$
Machine 1
Machine 2
$\delta_{i,3}$
$\delta_{i,4}$
$\delta_{i,1}$
$\delta_{i,3}$
$\delta_{i,1}$
$\delta_{i,2}$
$\delta_{i,4}$
$\delta_{i,2}$
Machine 3
Machine 4
$\delta_{i,4}$
$\delta_{i,3}$