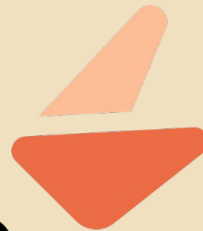


The background of the slide is a light beige color, decorated with numerous monarch butterflies in various sizes and orientations. The butterflies are primarily orange with black veins and white spots on their wings. They are scattered across the entire slide, with some appearing larger and more detailed than others.

torch.monarch

PyTorch Conference North America 2026



Monarch is an API for your supercomputer

Machines are Python **host** objects.

Host methods create **processes** and **actors**.

Pass **RDMA buffers** between actors to move data at line rate.

Imperative API

Say directly where and what to run.

Write distributed ML applications as if they were simple Python scripts.

```
class HostMesh:  
    def spawn_procs(  
        self, per_host=None  
    ) -> ProcMesh: ...
```

First-class objects

Pass hosts, processes, and actors in messages to delegate resources.

Manage compute with hosts, procs, actors

```
job = ProcessJob({"trainers": 2, "dataloaders": 1, "buffer": 1})  
#or SlurmJob, or KubernetesJob  
state = job.state(cached_path=None)  
  
trainer_procs = state.trainers.spawn_procs(per_host={"gpu": 8})  
loader_procs  = state.dataloaders.spawn_procs(per_host={"gpu": 1})  
replay_buffer_procs = state.buffer.spawn_procs()  
  
buffer = replay_buffer_procs.spawn("replay", ReplayBuffer)  
loaders = loader_procs.spawn("loader", DataLoader, "/data/imagenet", buffer)  
trainers = trainer_procs.spawn("trainer", Trainer, buffer)  
  
trainers.train.broadcast(epochs=10)
```

Data movement with RDMA buffers

```
class DataLoader(Actor):  
    @endpoint  
    def next_batch(self, step: int) -> None:  
        batch = torch.rand(256, 3, 224, 224, device="cuda")  
        sample = RDMABuffer(batch.view(torch.uint8).flatten())  
        self.replay_buffer.post.choose(sample).get()  
  
class Trainer(Actor):  
    # train(epochs) repeats step(); only step is shown here.  
    @endpoint  
    def step(self) -> None:  
        sample = self.replay_buffer.sample.choose().get()  
        sample.read_into(self.batch.view(torch.uint8).flatten()).get()  
        ...
```


Monarch for Agentic Development



As inference speeds up, tool waits stand out

 Inference

 Tool wait

Today's Agent Loop

Inference and tools alternate.



Future agents get faster

Same two tool calls.



Tool waits dominate

TWO LEVERS

Faster tools

Shorter wait per call



Fewer calls

One tool wait instead of two



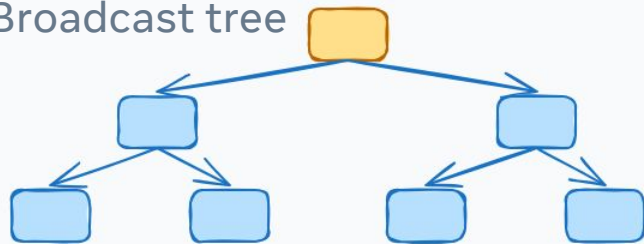
Faster tools with Monarch

Monarch-built tools can...

Start and message large
groups of processes quickly

Transfer data
at line rate

Broadcast tree

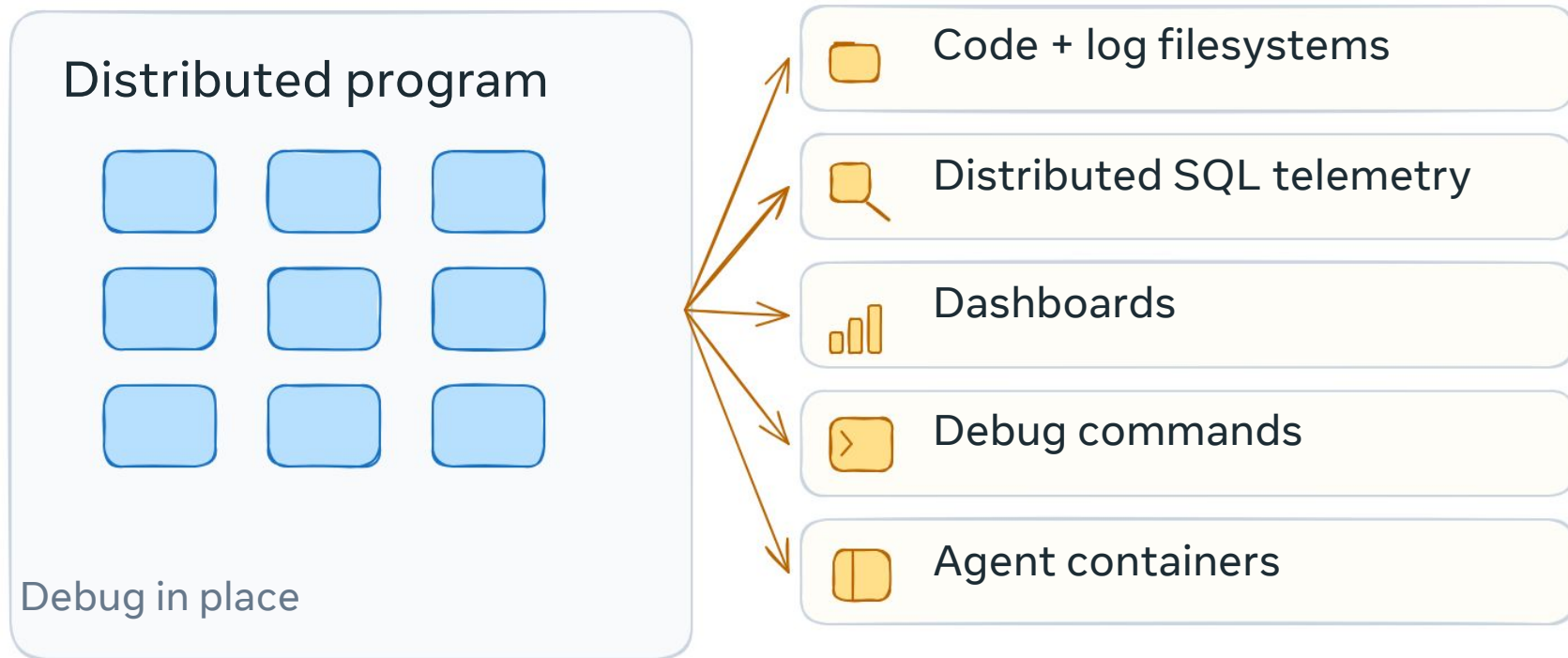


RDMA data bus

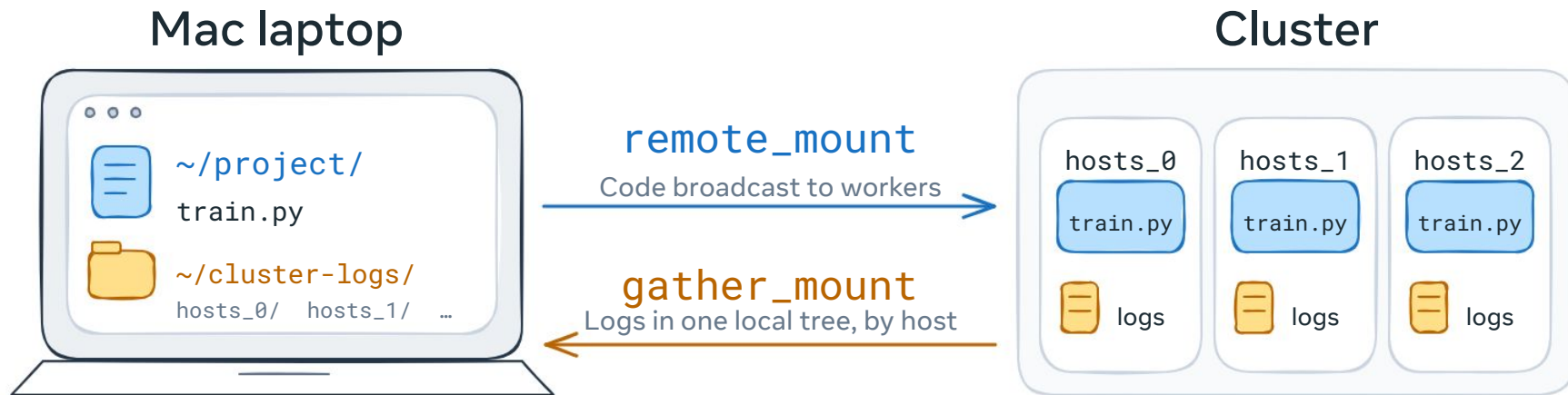


Fewer calls with monarch

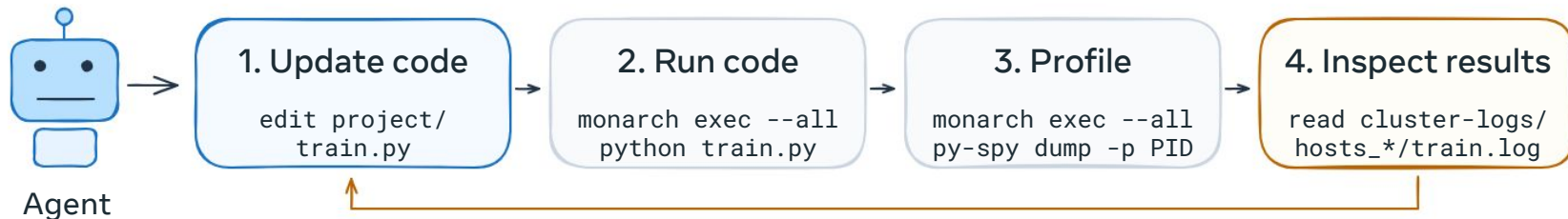
Build services for your distributed programs, agents take fewer turns to debug.



Fewer calls, example: remote_mount / gather_mount



Agent's debug loop — local files + Monarch CLI

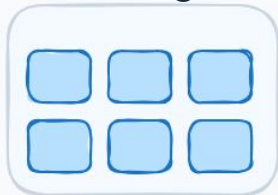


Bring your own service

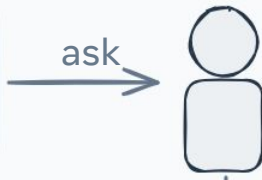
Traditional: ask a network admin. Monarch: bring what you need.

Traditional setup

Training run



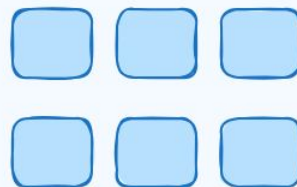
Network admin



Admin sets up the services

Monarch job

Your program



Filesystem

Telemetry

Agent env

You launch what you need

Fewer calls, example: WeightBridge

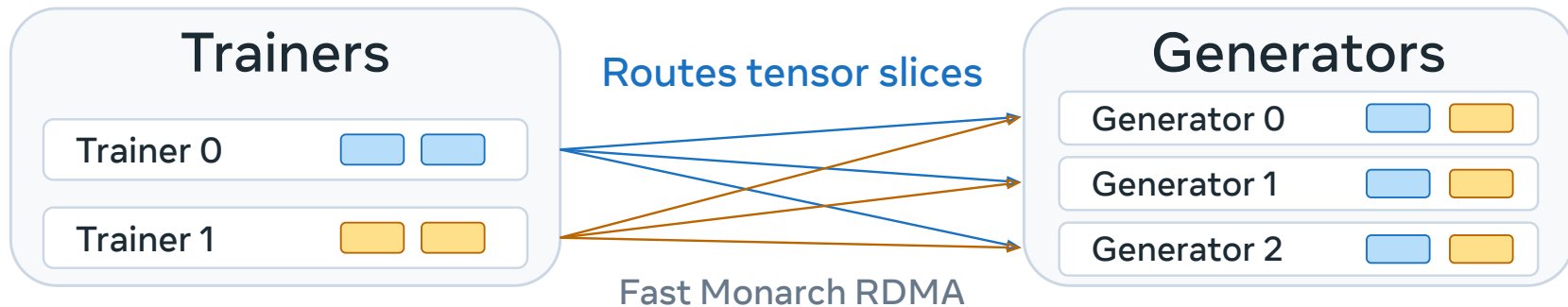


Fast weight sync from trainers to generators

Discovers each worker's tensor layout

Runs your existing weight loader with test tensors

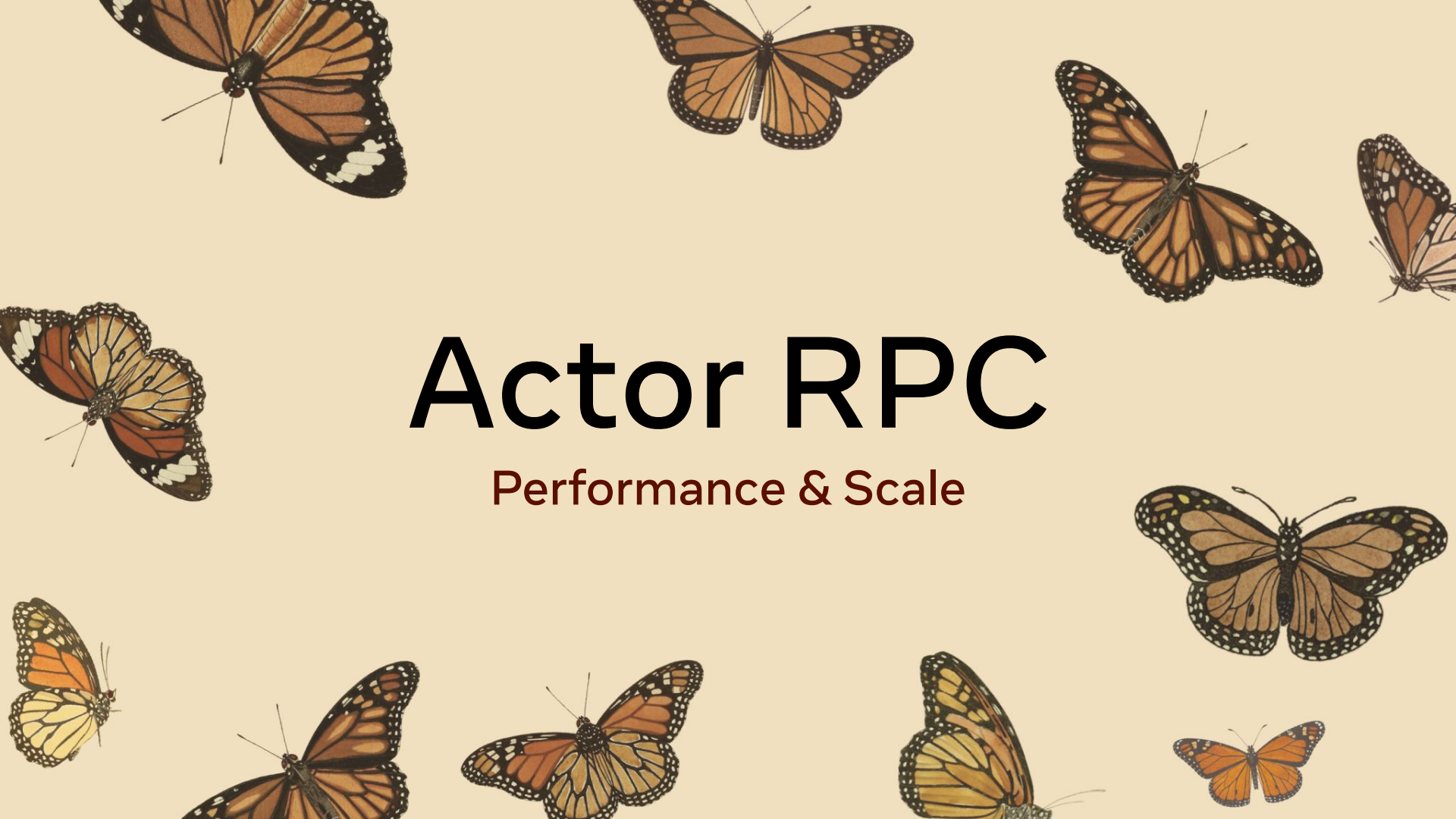
Reads where they land to find splits, reshapes and transposes



Each generator gets the weight slices it needs.

The image features a light beige background with the word "Demo" centered in a dark blue, sans-serif font. Surrounding the text are approximately 12 monarch butterflies of various sizes and orientations. The butterflies are depicted with detailed orange, black, and white patterns on their wings. Some are shown in profile, while others are facing forward or backward. The butterflies are scattered across the frame, with some appearing larger and more prominent than others, creating a decorative border around the central text.

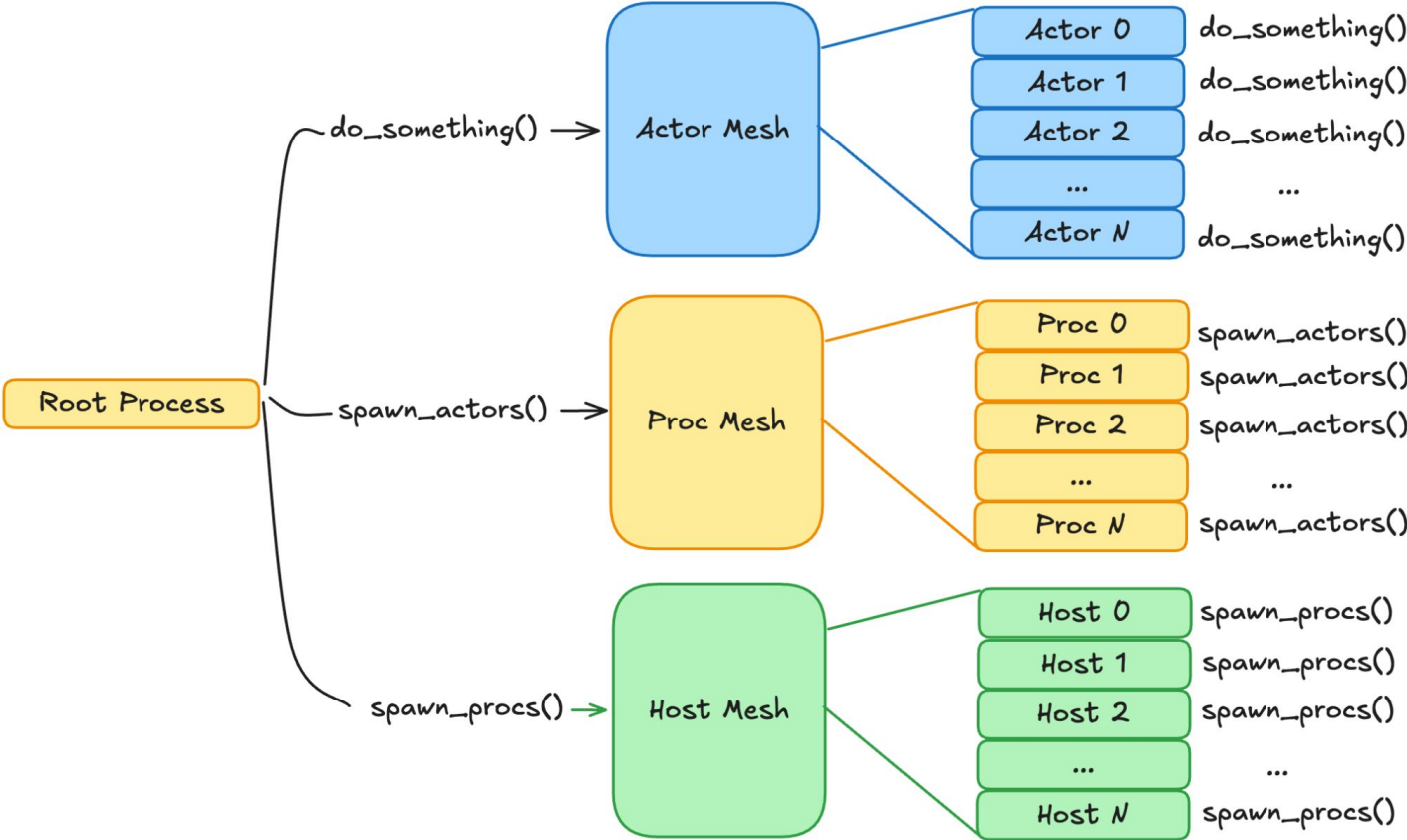
Demo

The background of the slide is a light beige color, populated with several detailed illustrations of monarch butterflies. These butterflies are scattered across the frame, with some in the upper corners, some in the lower corners, and others in the middle. They are shown in various orientations, some facing left, some right, and some slightly angled. The butterflies have orange wings with black veins and a black border with white spots. The central text is positioned in the middle of the slide, with the title 'Actor RPC' in a large, bold, black font, and the subtitle 'Performance & Scale' in a smaller, dark red font directly below it.

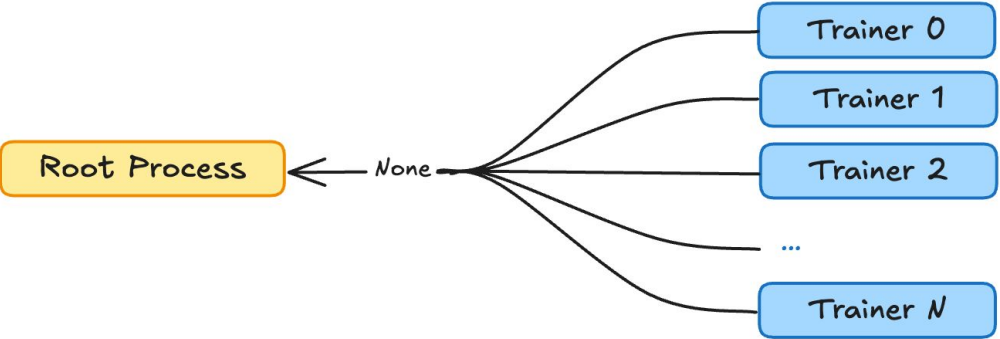
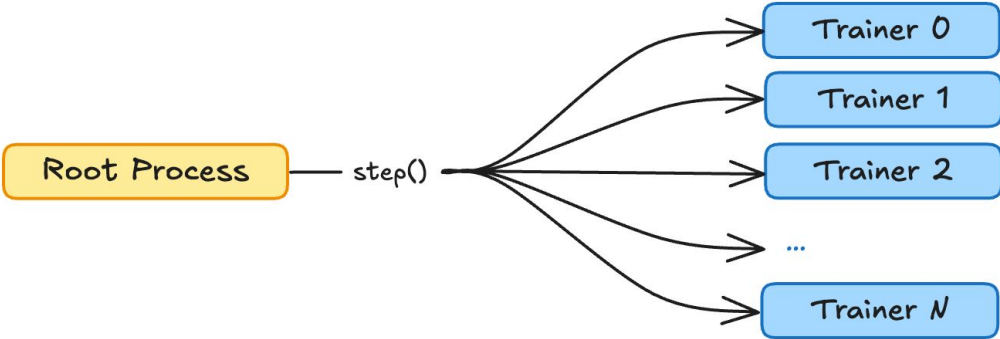
Actor RPC

Performance & Scale

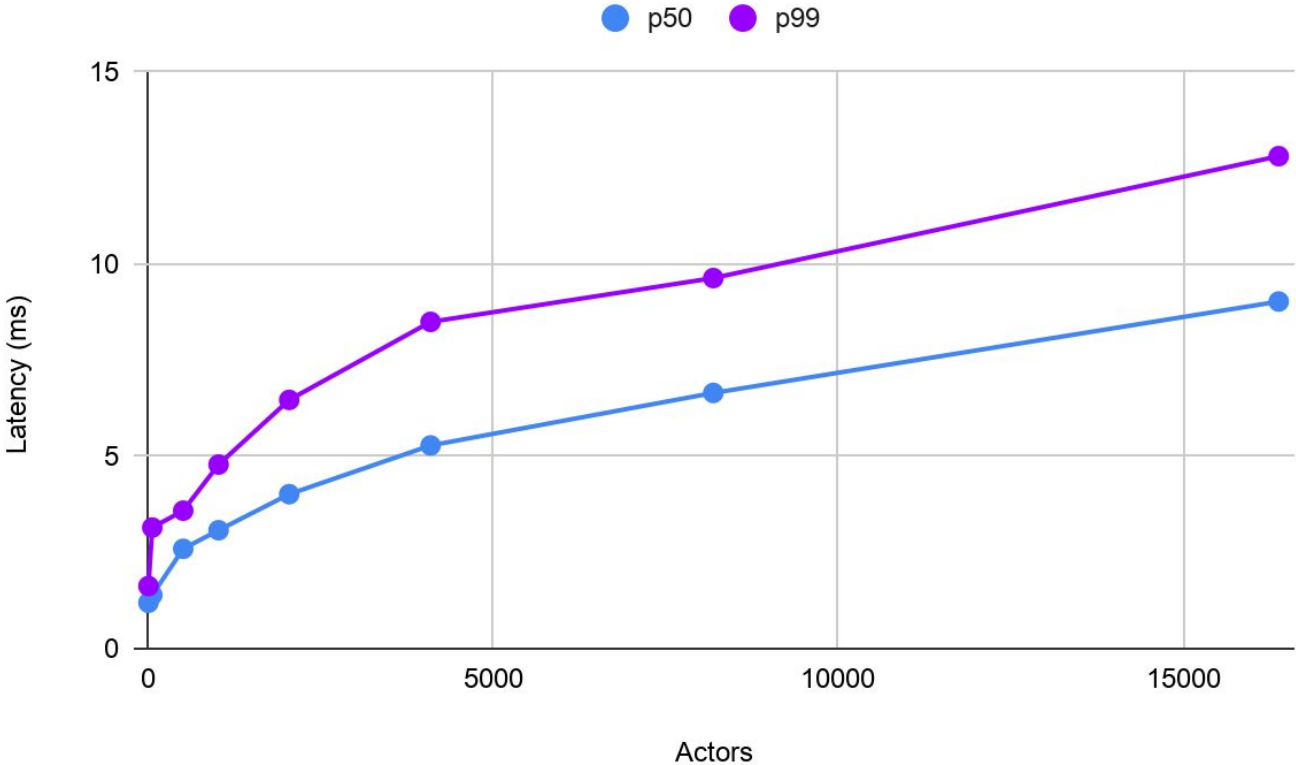
Monarch Orchestrates Meshes with RPCs



Orchestration Wants Low Latency at Scale



Actor RPC Scales Sublinearly to Tens of Thousands of Remote Actors

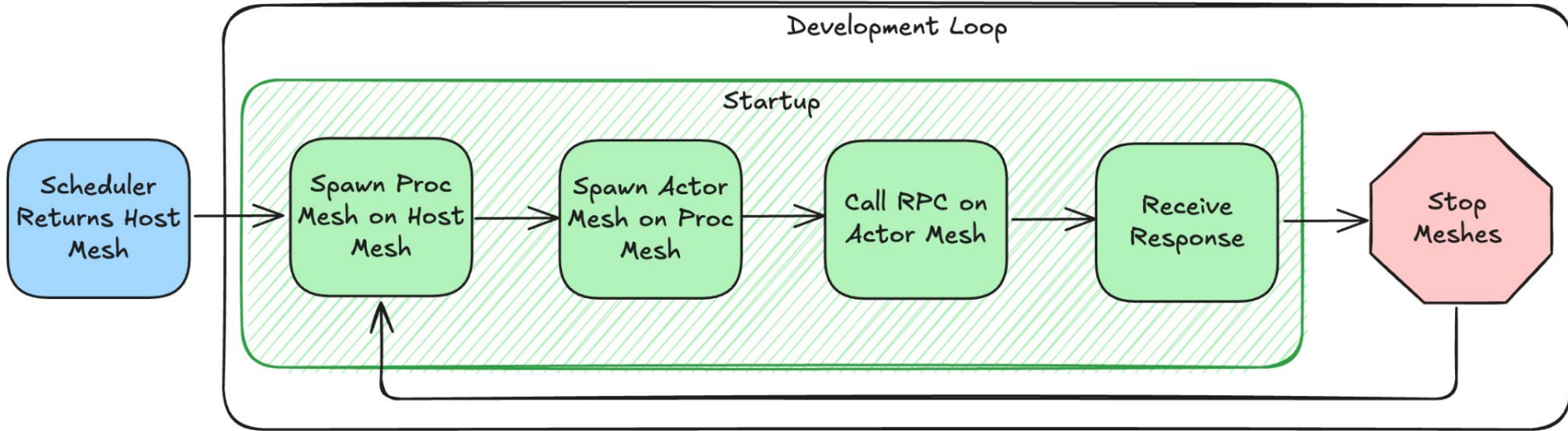


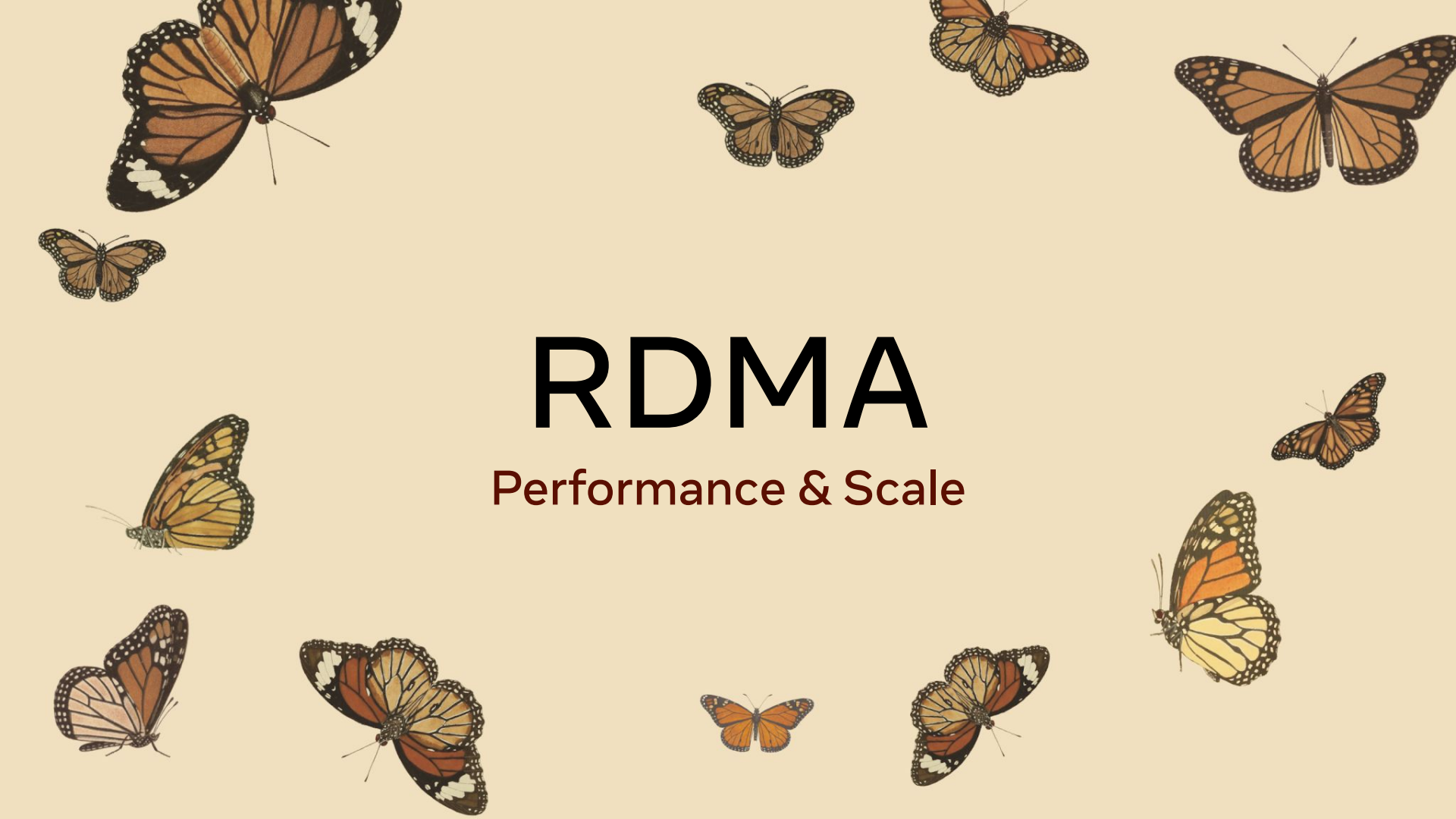
The background of the slide is a light beige color, decorated with several Monarch butterflies in various sizes and orientations. The butterflies have orange wings with black veins and a black border with white spots. They are scattered around the central text, with some in the top left, top right, bottom left, and bottom right corners, and others in between.

Startup

Performance & Scale

Monarch Starts Up 16k Remote Actors in 4.5 seconds

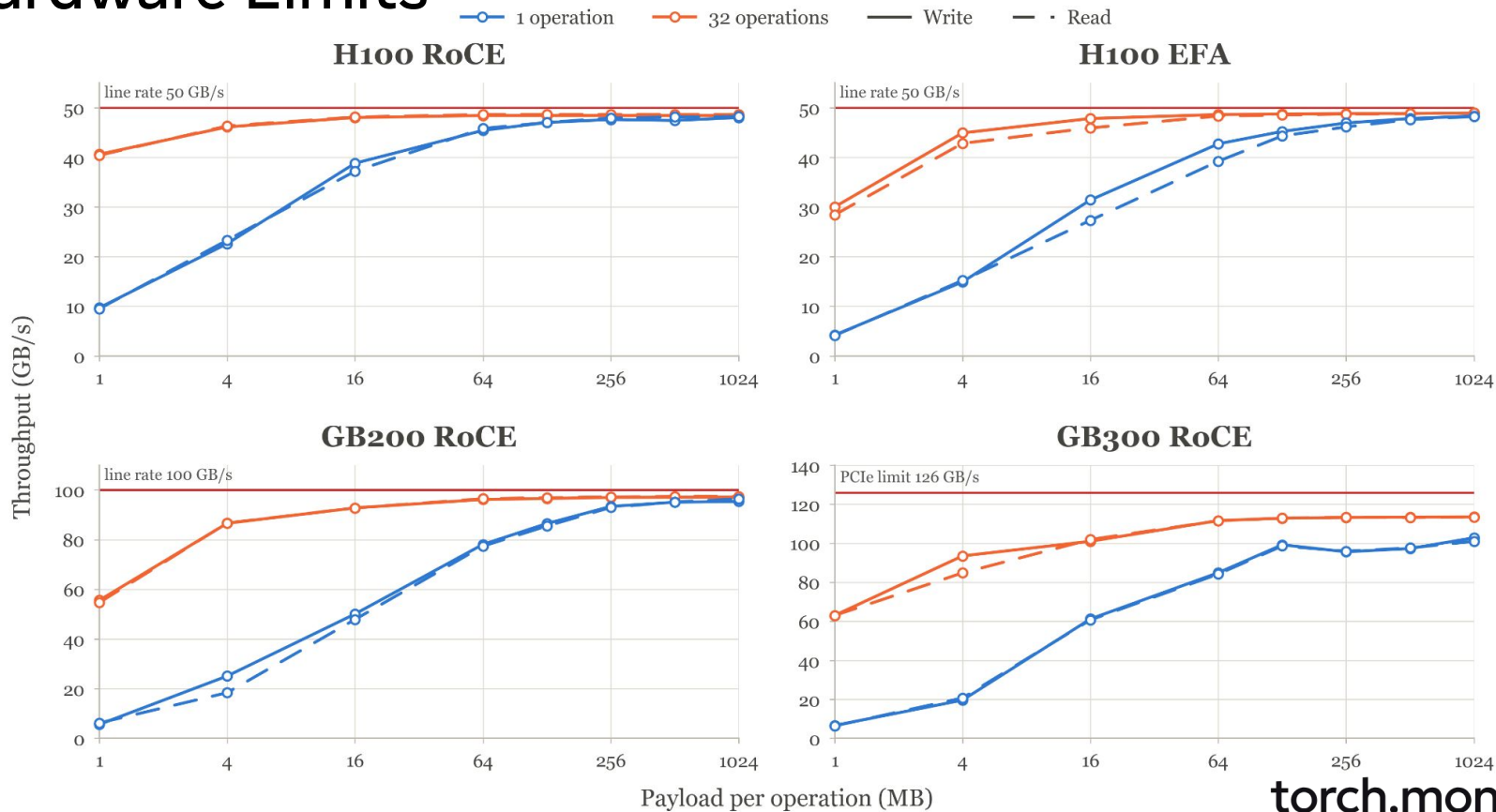


The background of the slide is a light beige color, decorated with several monarch butterflies in various sizes and orientations. The butterflies are detailed with orange, black, and white patterns on their wings.

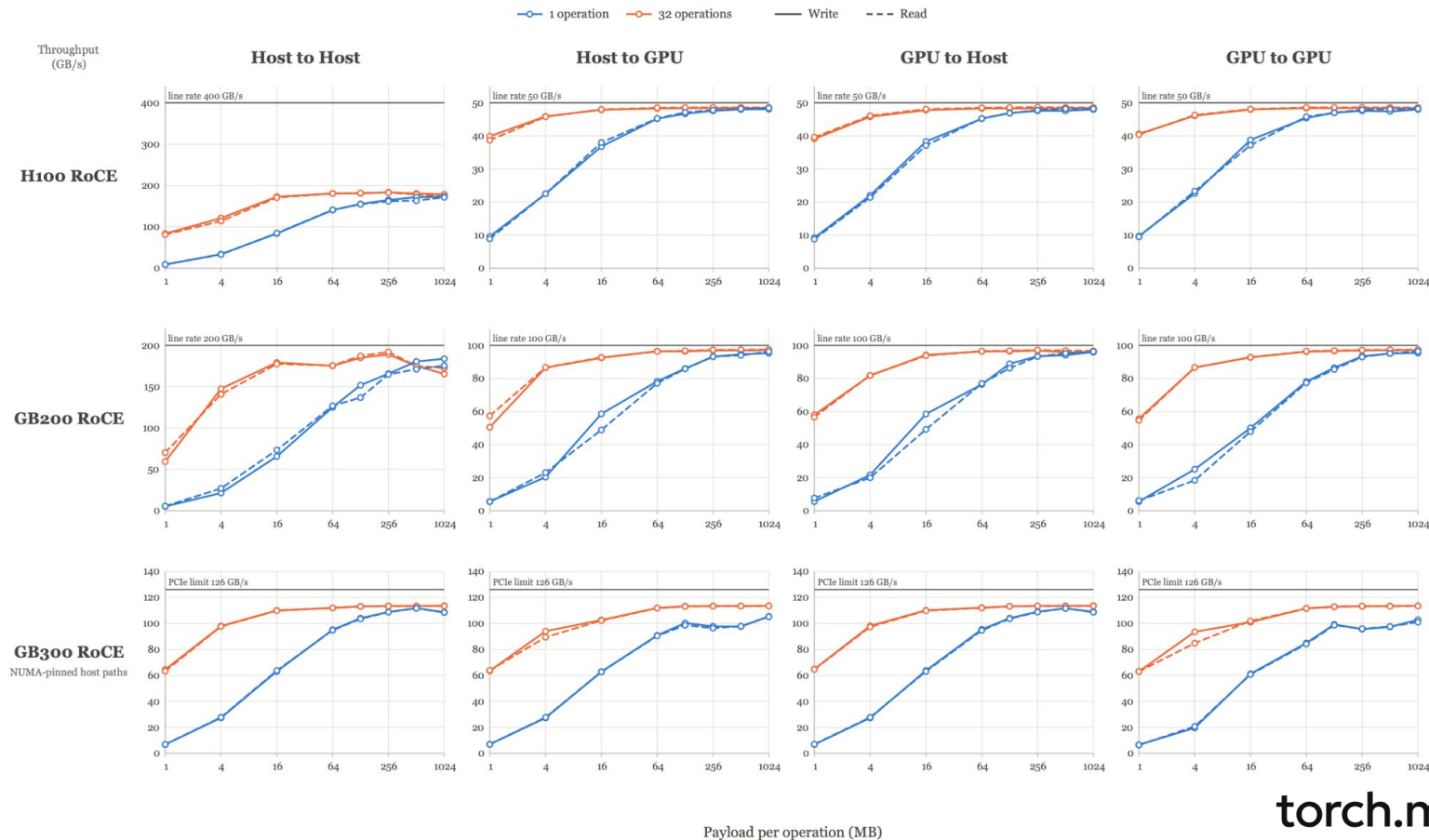
RDMA

Performance & Scale

GPU-GPU Read/Write Throughput Approaches Hardware Limits



RDMABuffer performs well across many configurations



Some Configurations Require Special handling: EFA

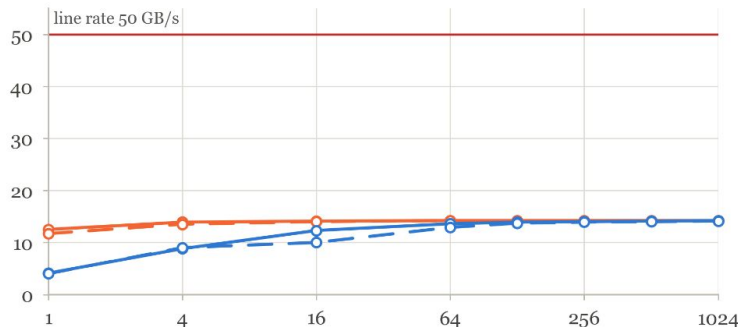
H100 EFA

—○— 1 operation —○— 32 operations — Write - - Read

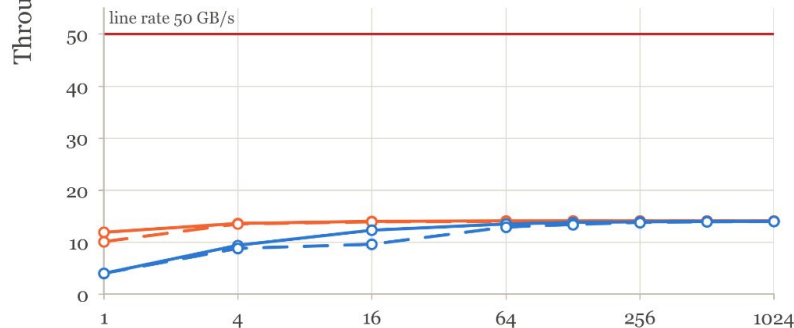
Host to Host



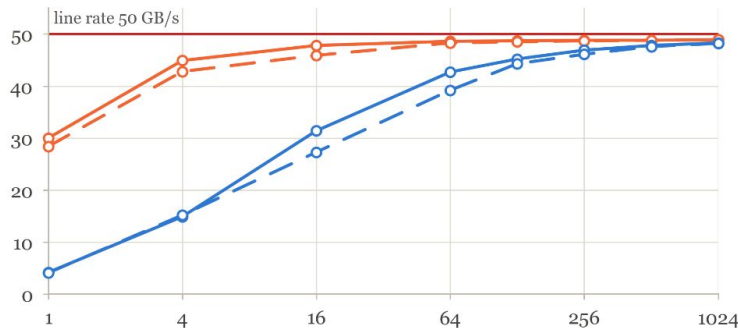
Host to GPU



GPU to Host



GPU to GPU



Payload per operation (MB)

Monarch an API for your supercomputer

- Agent and Human friendly API
- Sublinear scaling to tens of thousands of hosts
- Start a 16k job in under 5 seconds
- Line Rate RDMA
- Develop locally, scale remotely
 - Code transfer: **remote/gather mount**
 - Weight syncing: **WeightBridge**
 - (Soon) Containerization for agents

