GPUnet: networking abstractions for GPU programs

Mark Silberstein
Technion – Israel Institute of Technology

Sangman Kim, Seonggu Huh, Xinya Zhang
Yige Hu, Emmett Witchel
University of Texas at Austin

Amir Wated
Technion

What

A socket API for programs running on GPU

Why

GPU-accelerated servers are hard to build

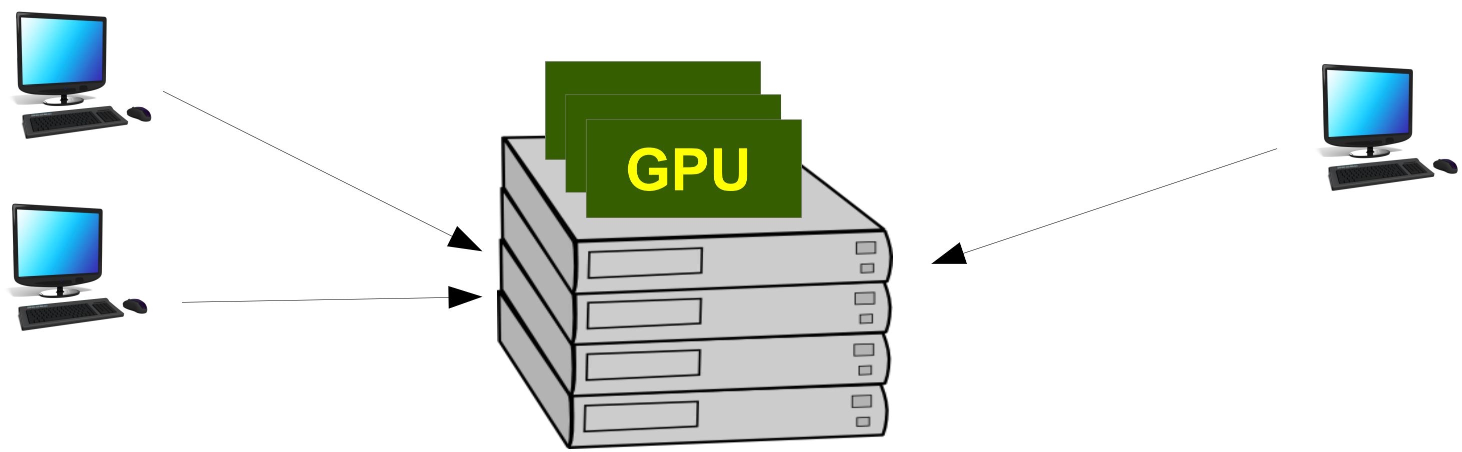
Results

GPU vs. CPU

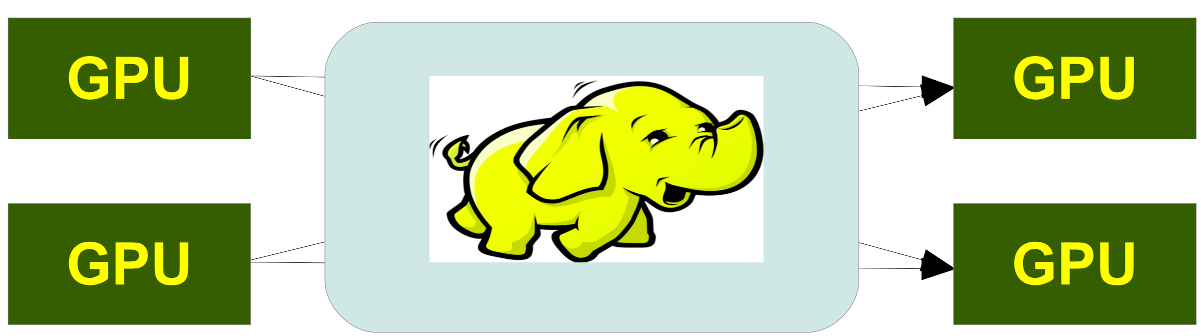
50%↑ throughput, 60%↓ **latency**, ½ LOC

Motivation: GPU-accelerated networking applications

Data processing server



MapReduce



Recent GPU-accelerated networking applications

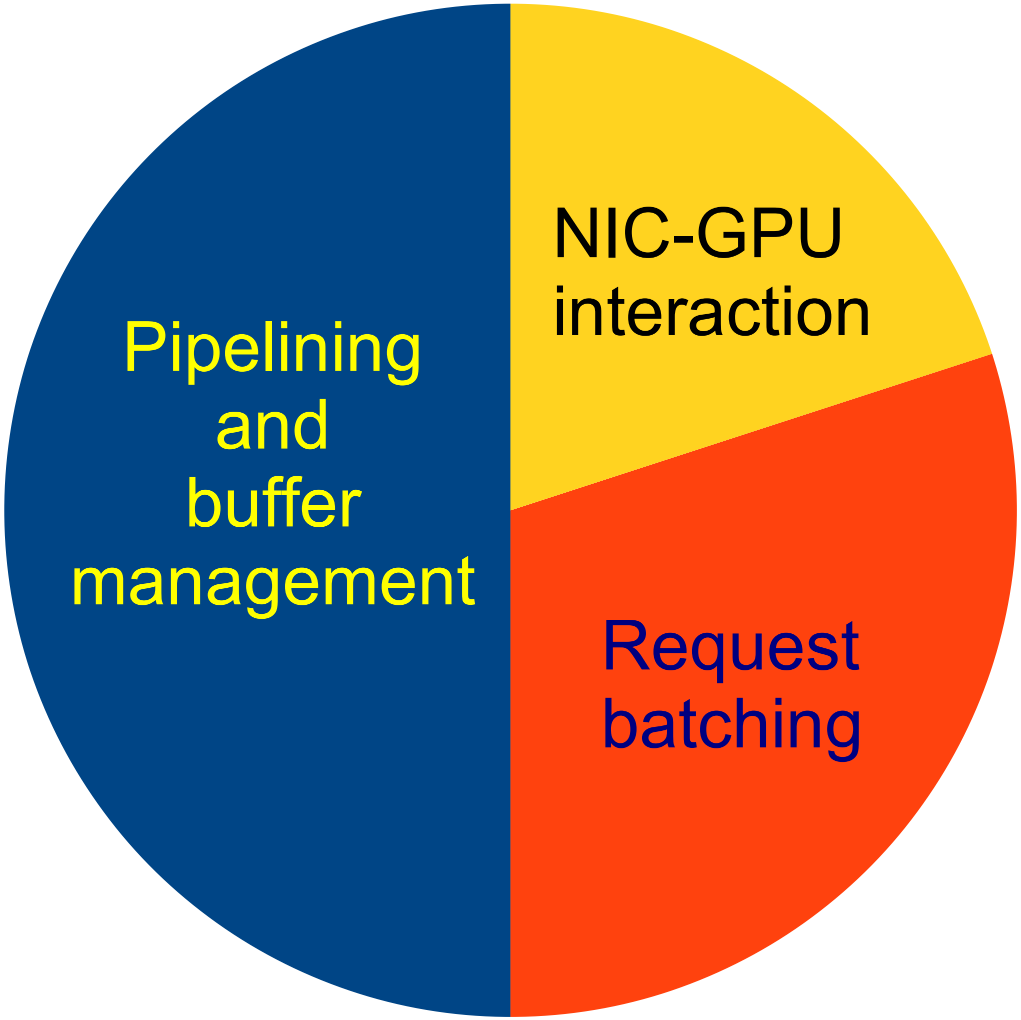
SSLShader (Jang 2011), GPU MapReduce (Stuart 2011), Deep Neural Networks (Coates 2013), Dandelion (Rossbach 2013), Rhythm (Agrawal 2014) ...

Recent GPU-accelerated networking applications

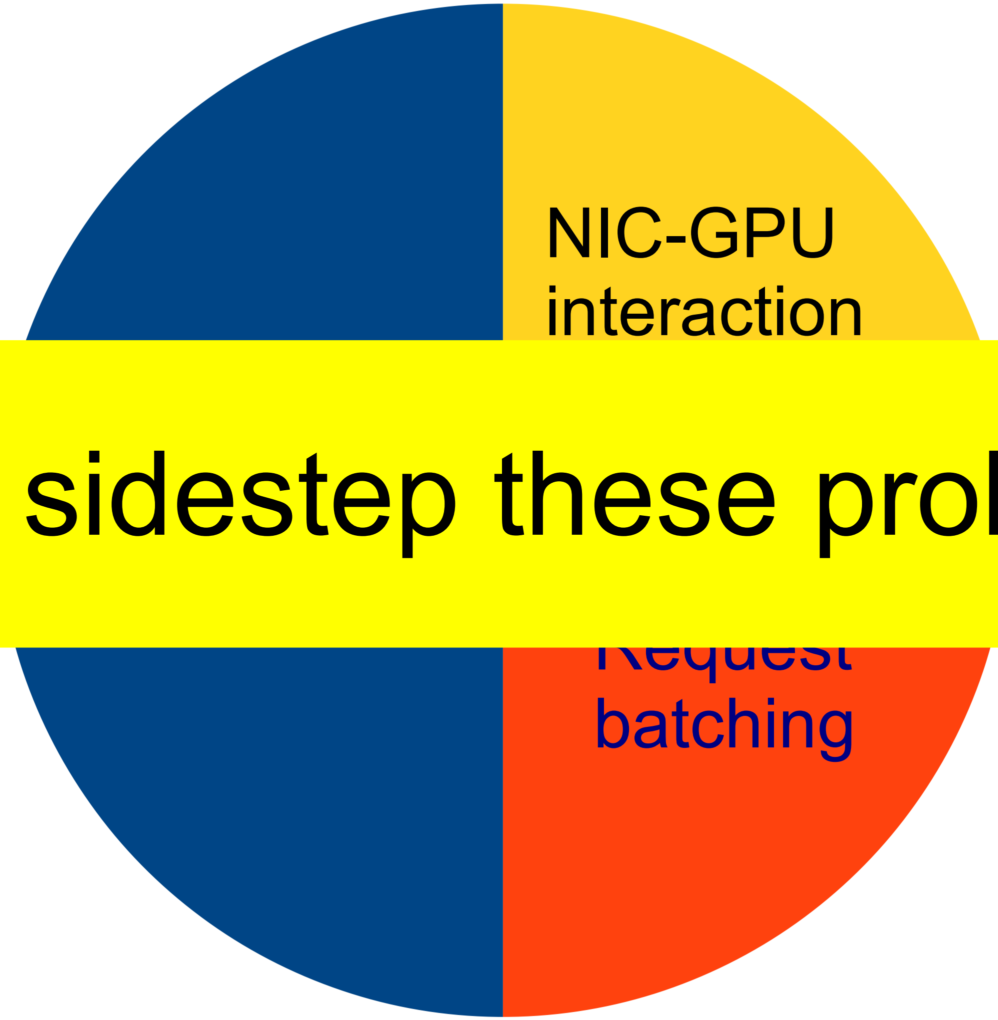
SSLShader (Jang 2011), GPU MapReduce (Stuart 2011), Deep Neural Networks (Coates 2013), Dandelion (Rossbach 2013), Rhythm (Agrawal 2014) ...

required **heroic** efforts

GPU-accelerated networking apps: Recurring themes

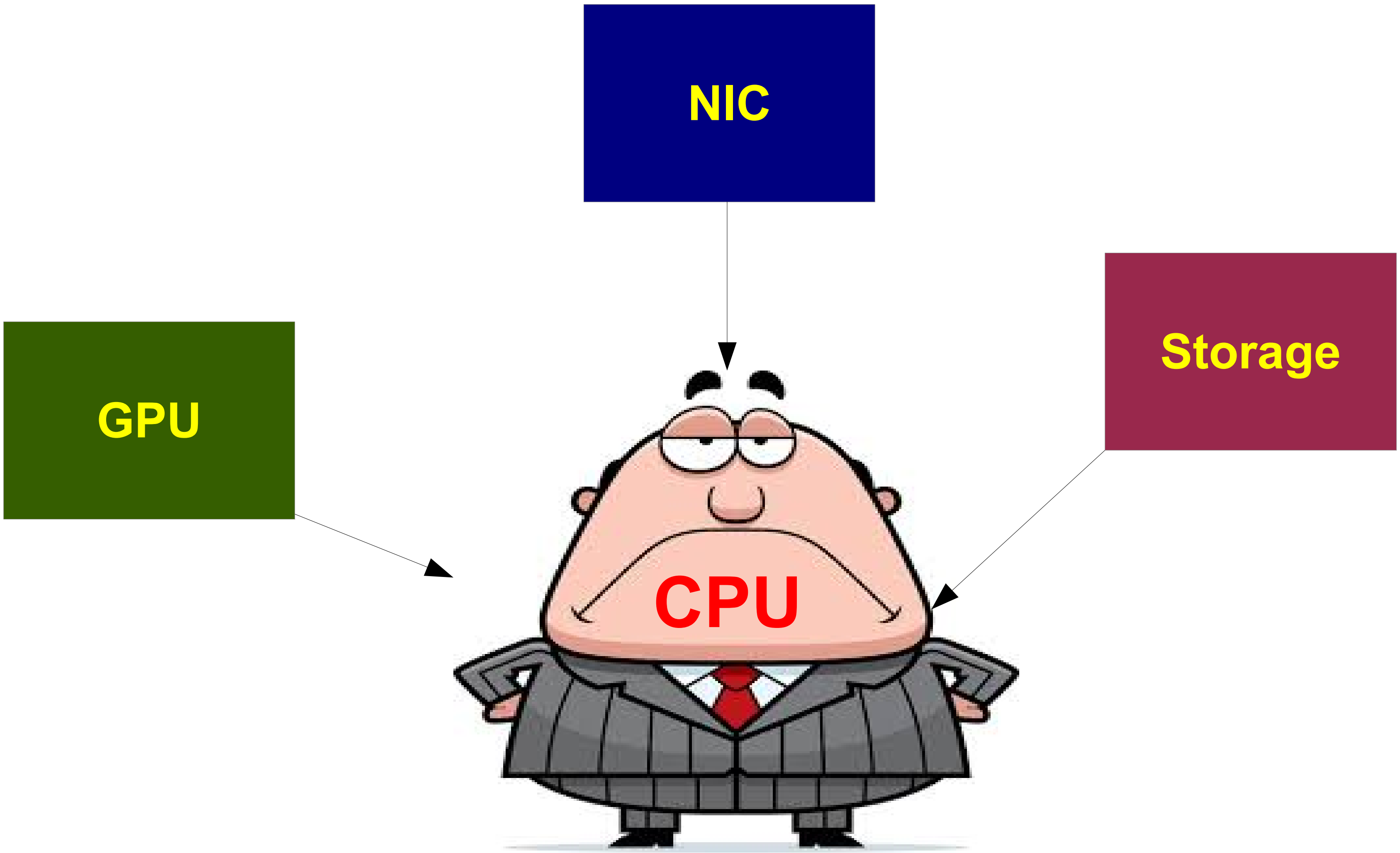


GPU-accelerated networking apps: Recurring themes

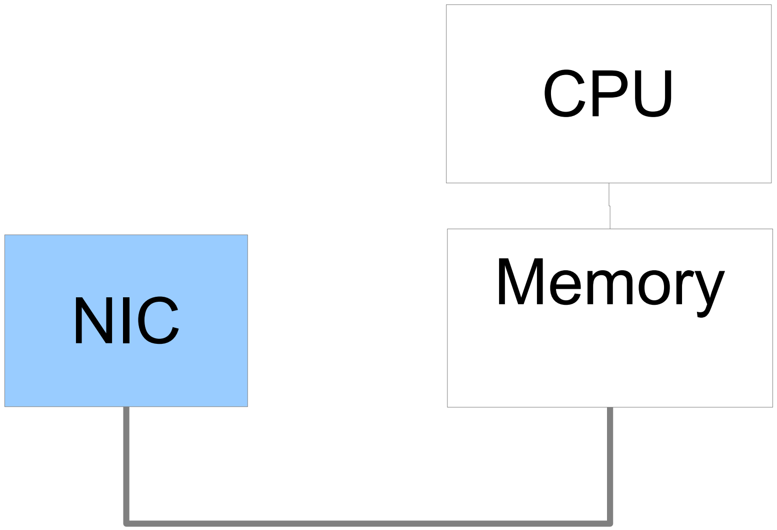


We will sidestep these problems

The **real** problem: CPU is the ***only*** boss

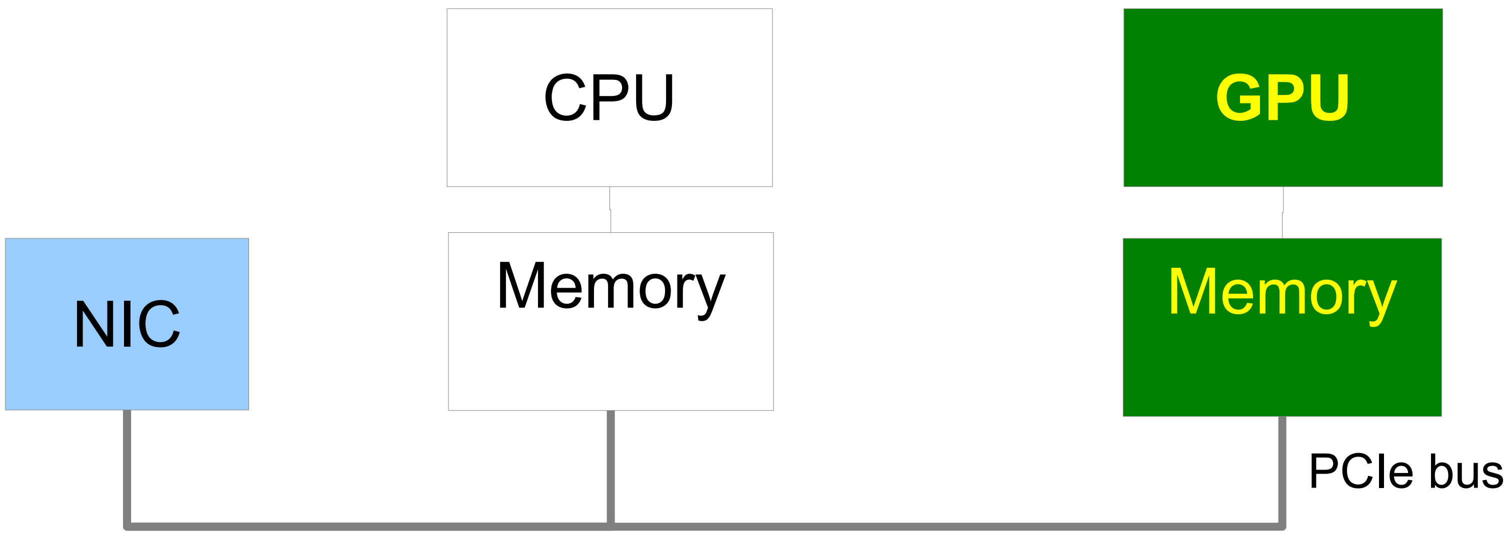


Example: CPU server




```
recv()
compute()
send()
```

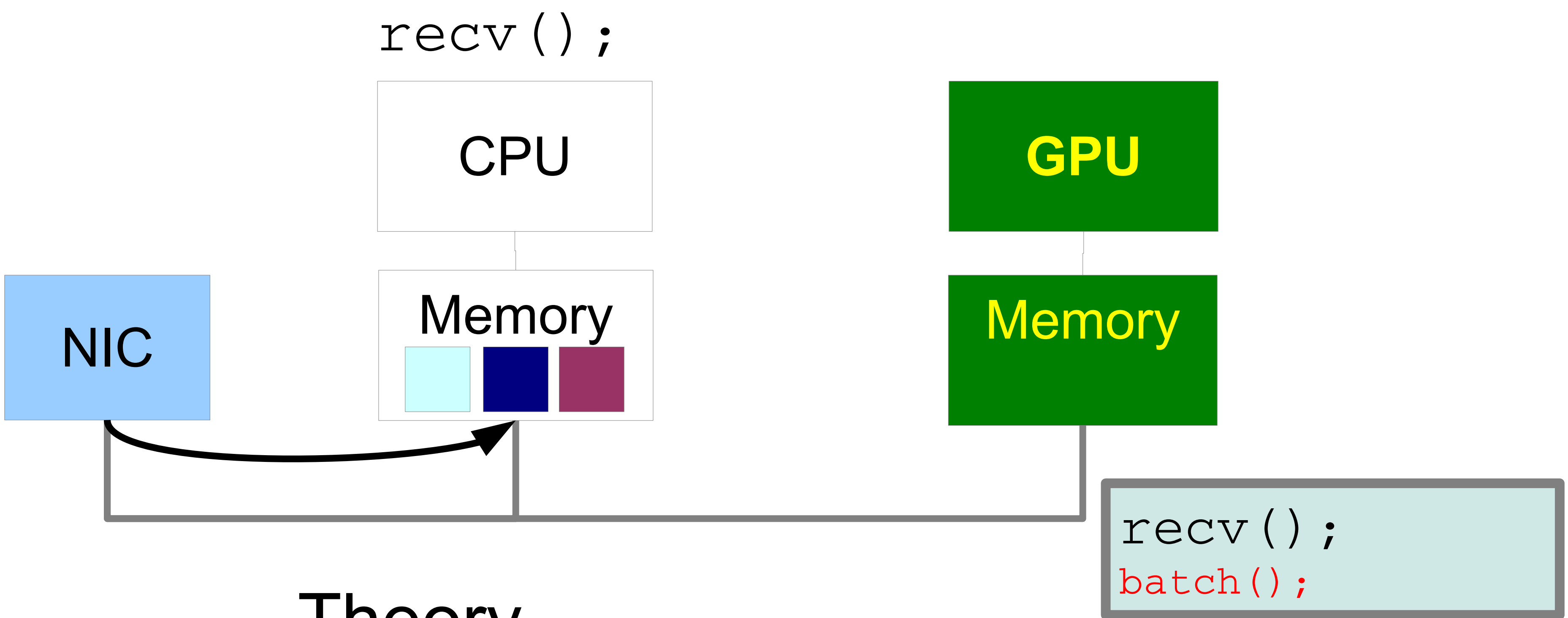
Inside a GPU-accelerated server



Theory

```
recv()
GPU_compute()
send()
```

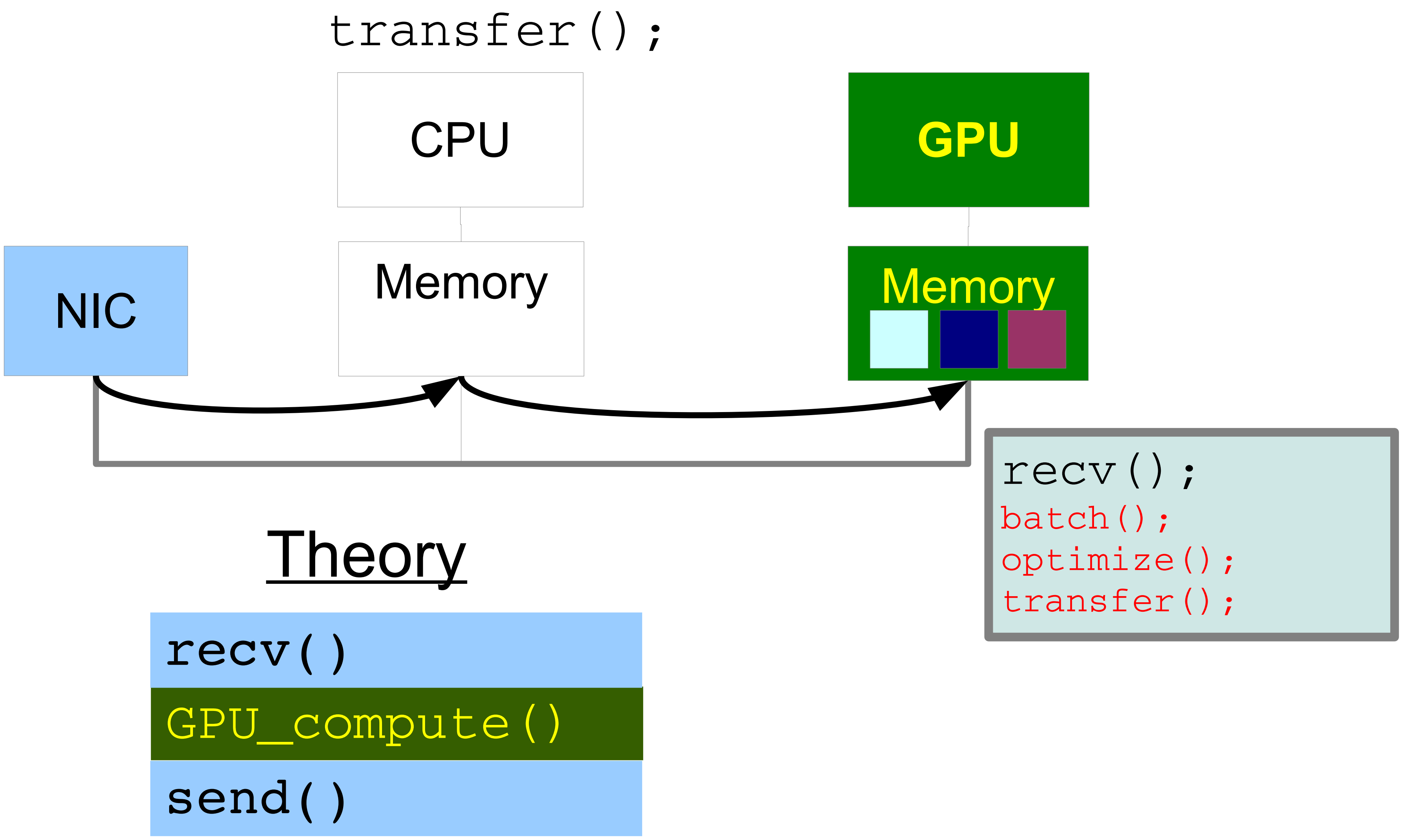
Inside a GPU-accelerated server



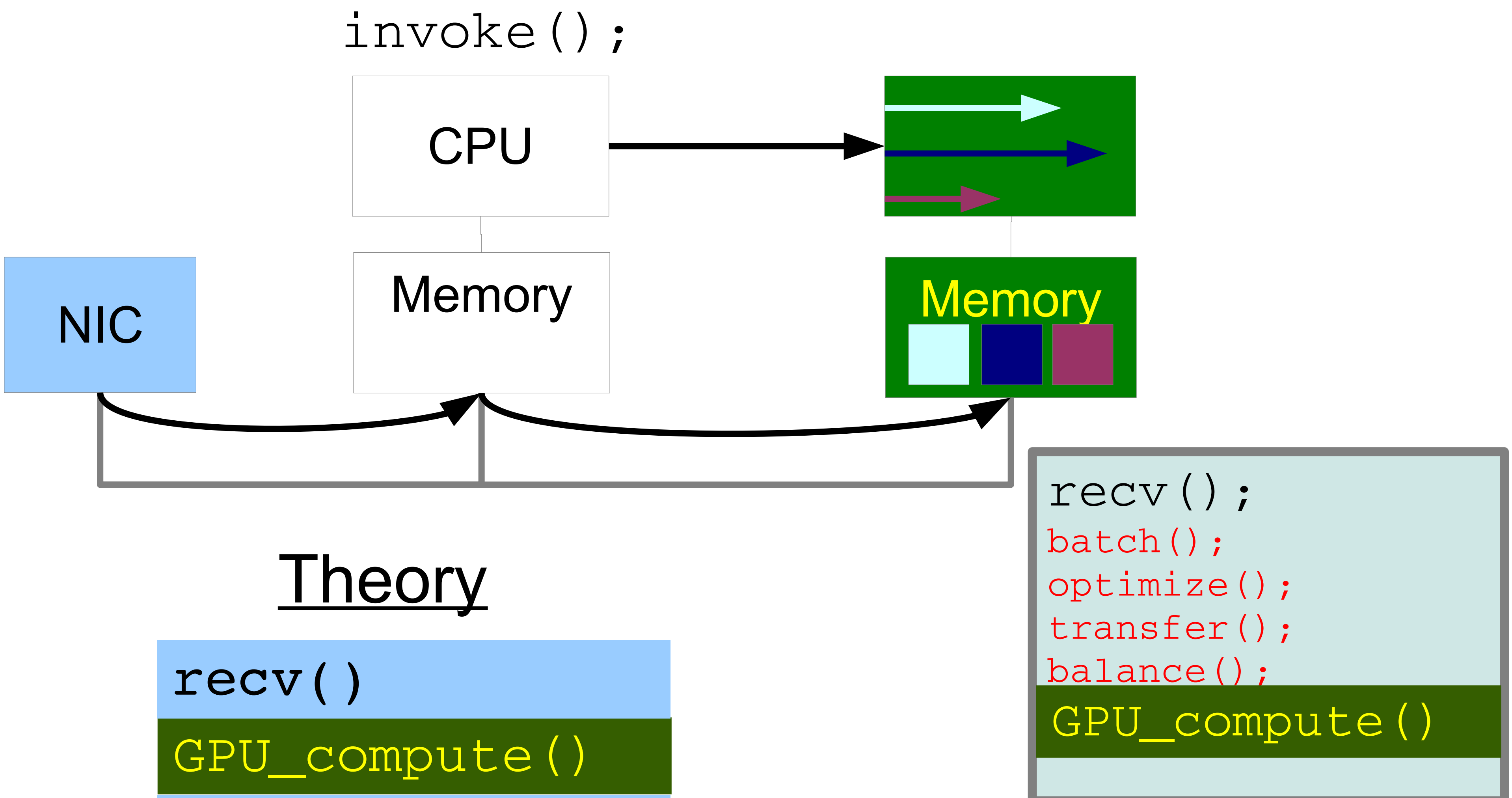
Theory

```
recv()
GPU_compute()
send()
```

Inside a GPU-accelerated server



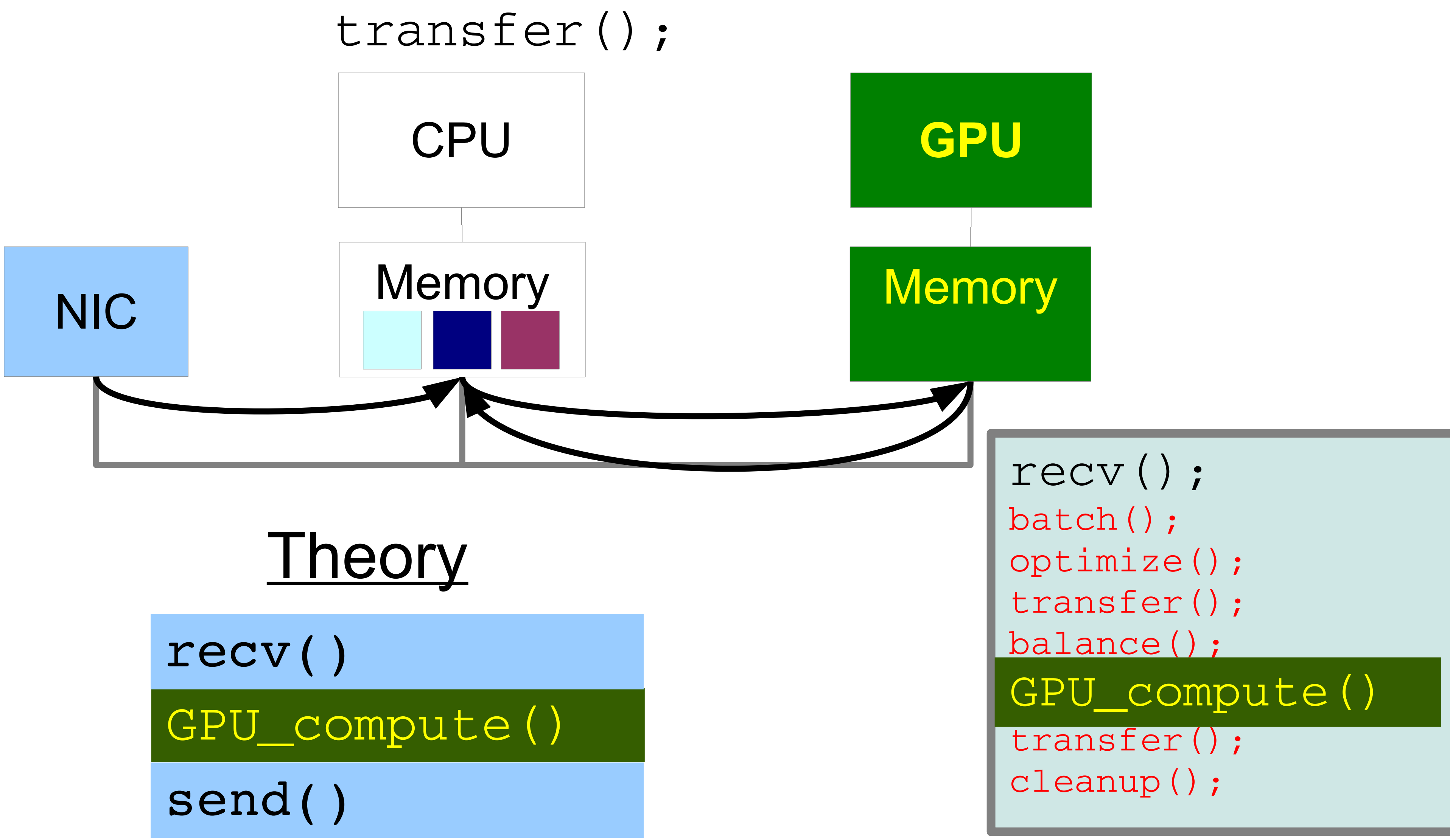
Inside a GPU-accelerated server



Theory

```
recv()
GPU_compute()
send()
```

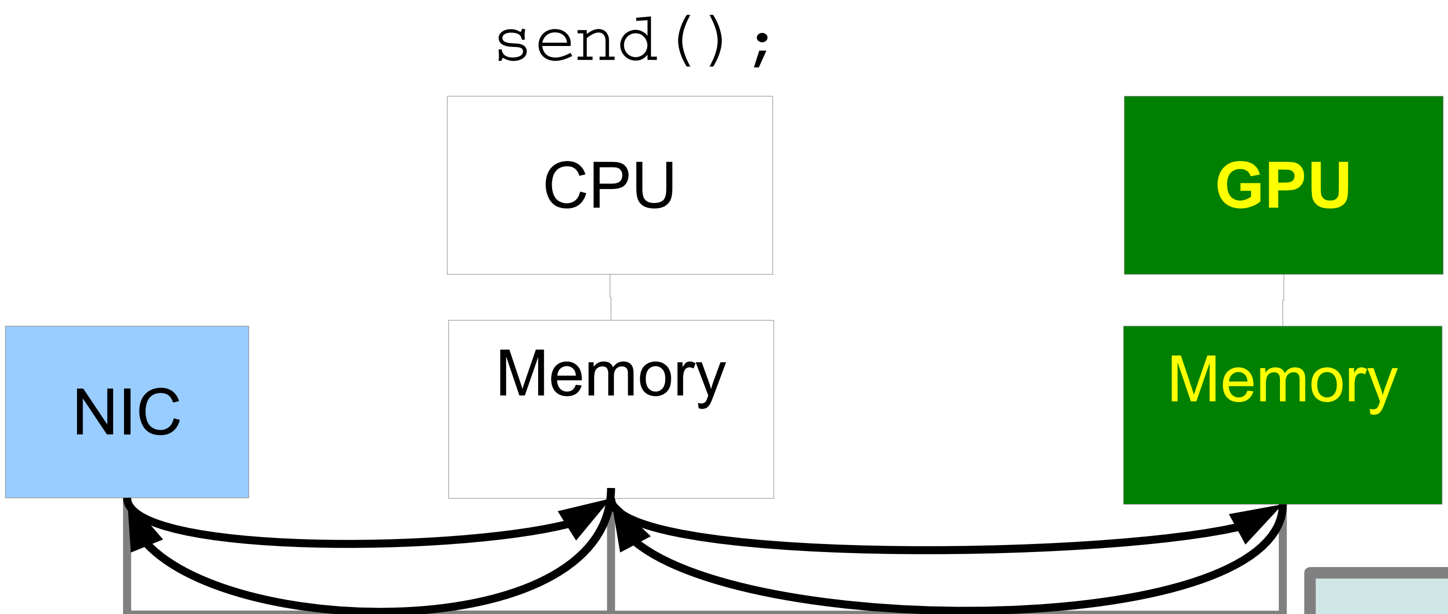
Inside a GPU-accelerated server



Theory

recv()
GPU_compute()
send()

Inside a GPU-accelerated server

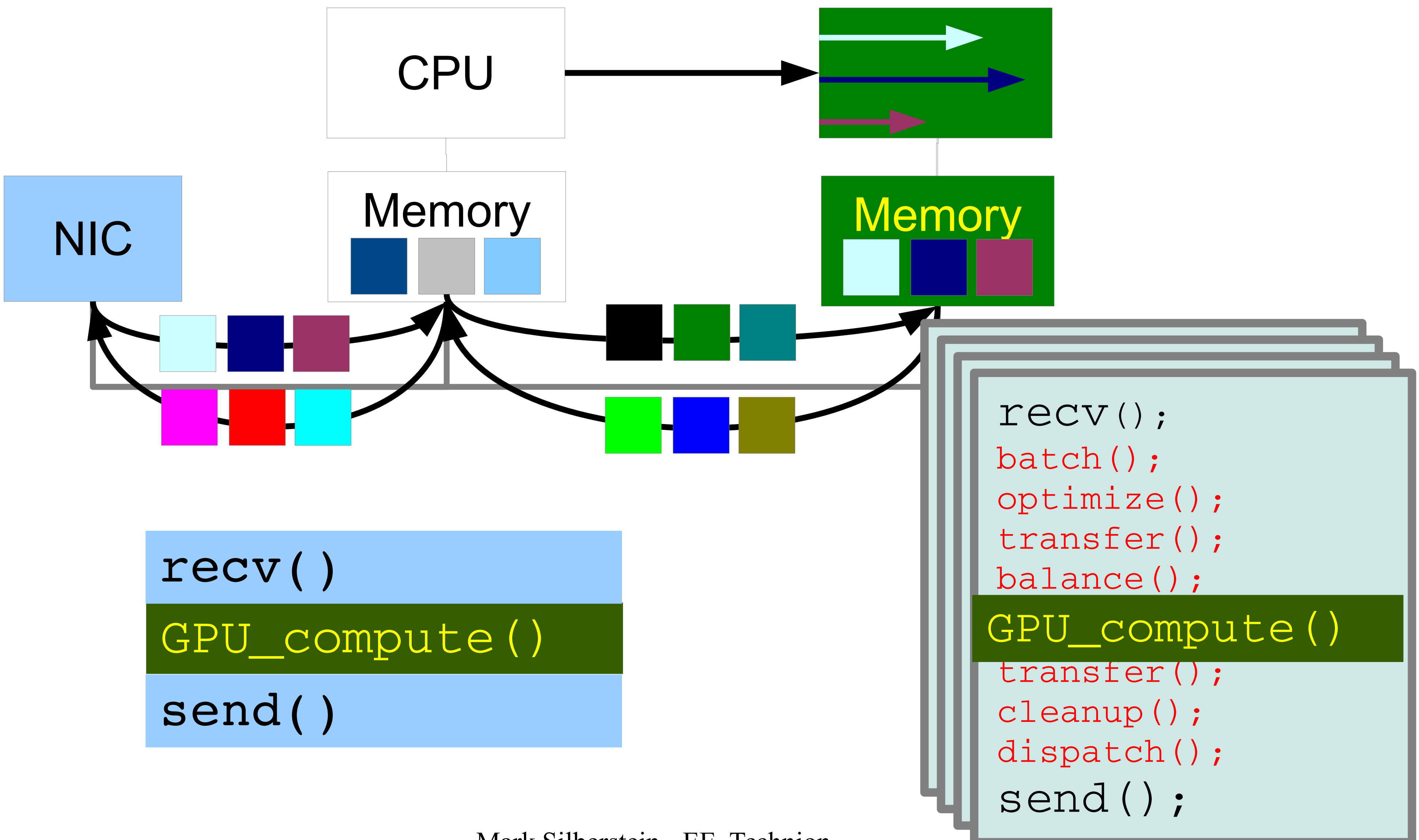


Theory

```
recv()
GPU_compute()
send()
```

```
recv();
batch();
optimize();
transfer();
balance();
GPU_compute()
transfer();
cleanup();
dispatch();
send();
```

Aggressive pipelining
Double buffering, asynchrony, multithreading



This code is for a CPU to manage a GPU

```
batch();
optimize();
transfer();
balance();
GPU_compute()
transfer();
cleanup();
dispatch();
```

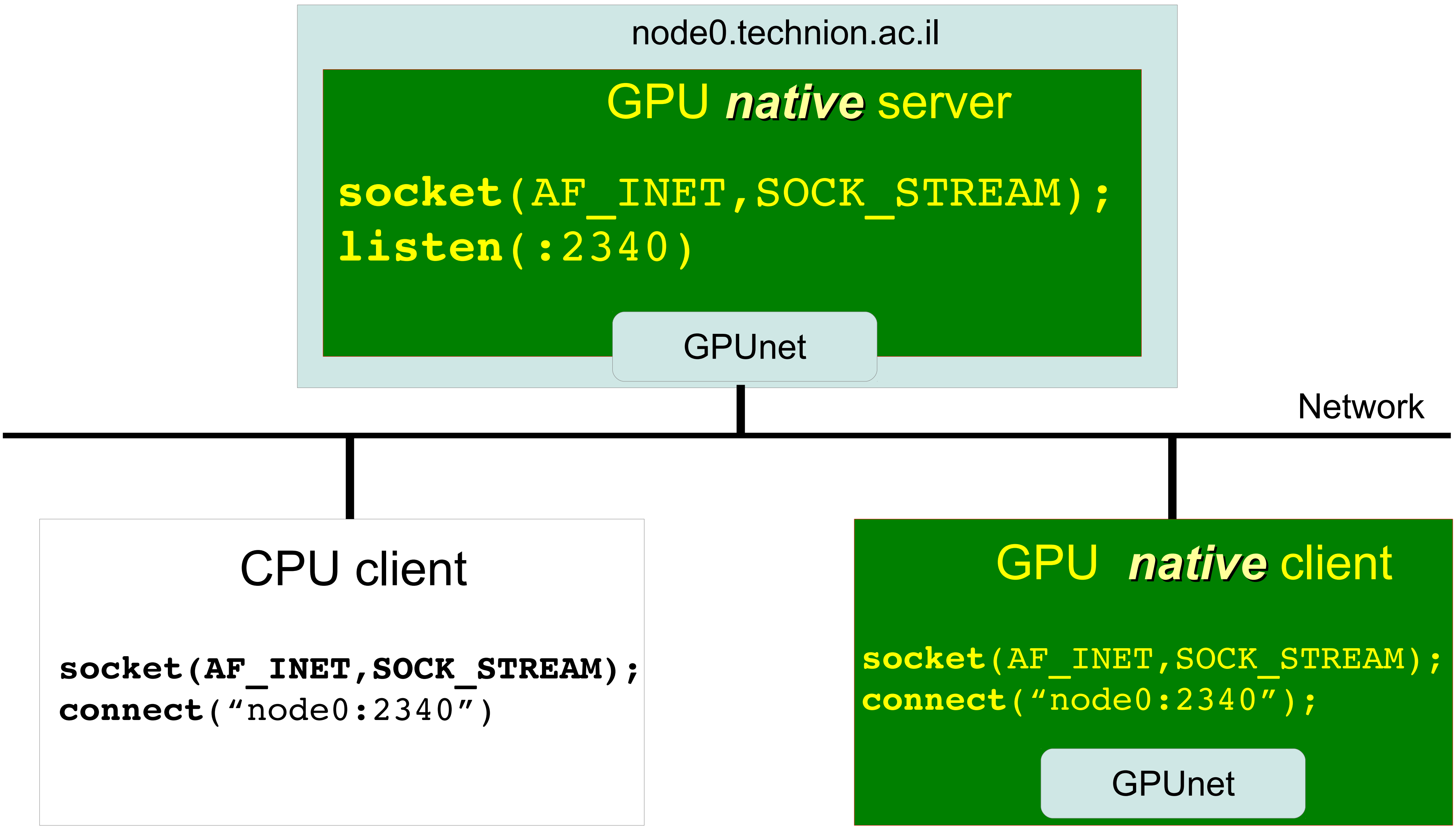
GPUs are not ***co-***processors

GPUs are **peer-**processors

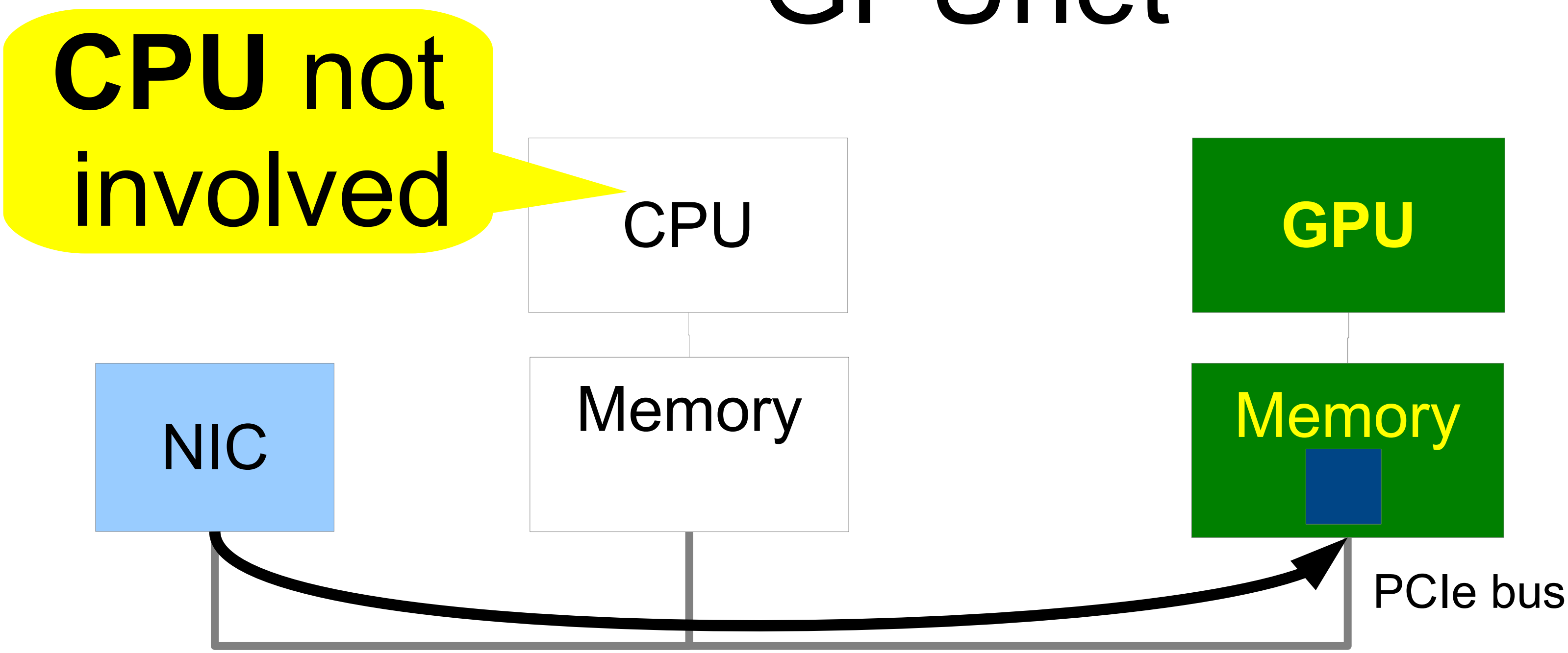
They need I/O **abstractions**

File system I/O – [GPUfs ASPLOS13]
Network I/O – this work

GPUnet: socket API for GPUs
Application view

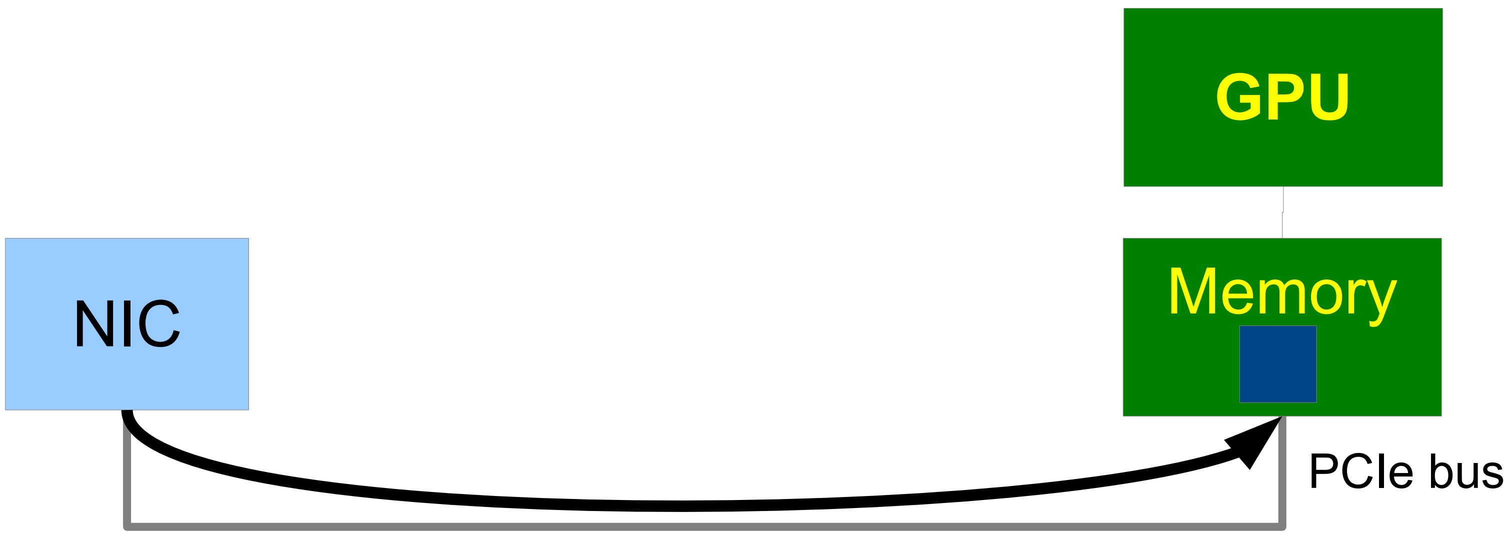


GPU-accelerated server with GPUnet



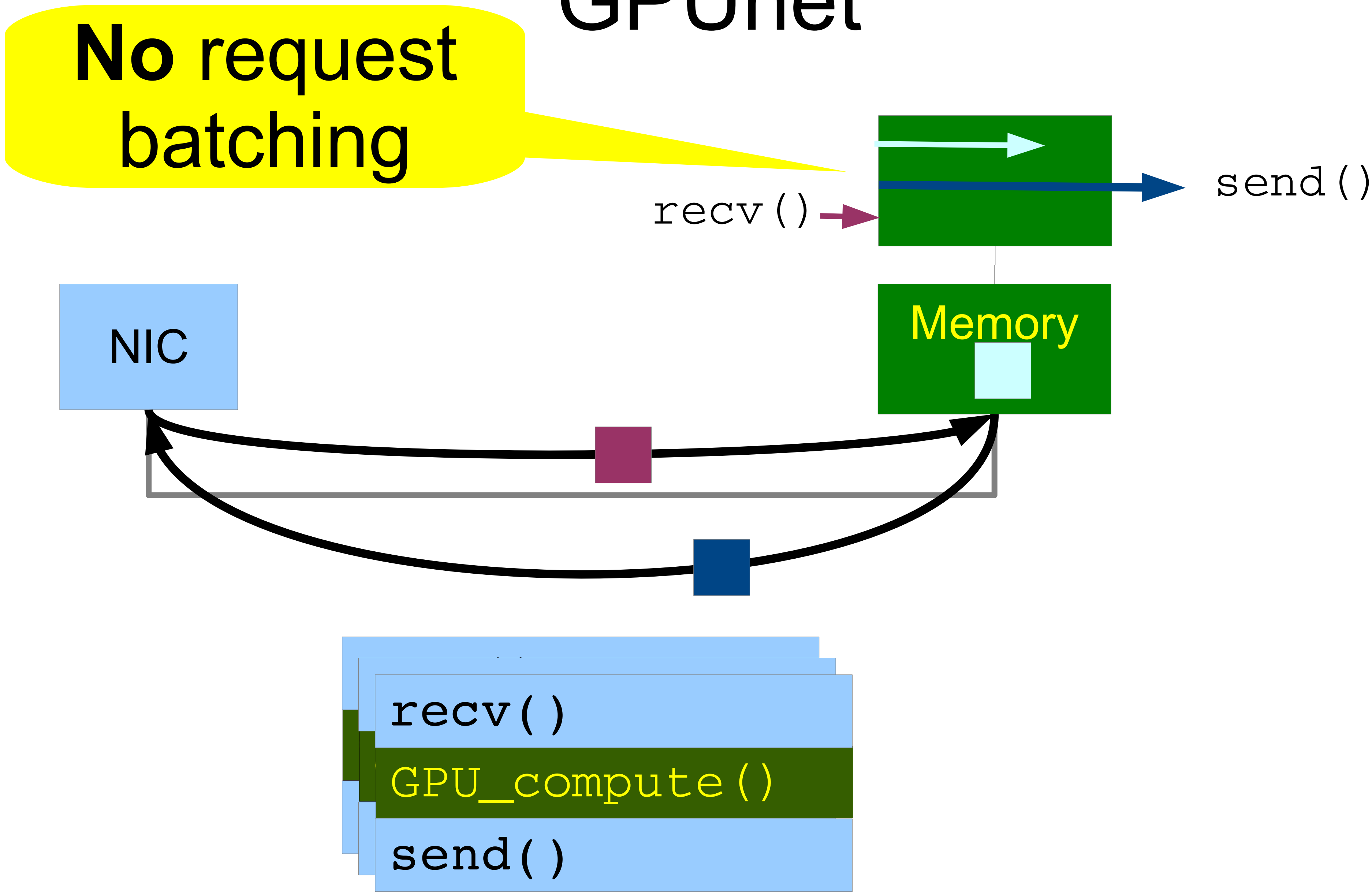

```
recv()
GPU_compute()
send()
```

GPU-accelerated server with GPUnet

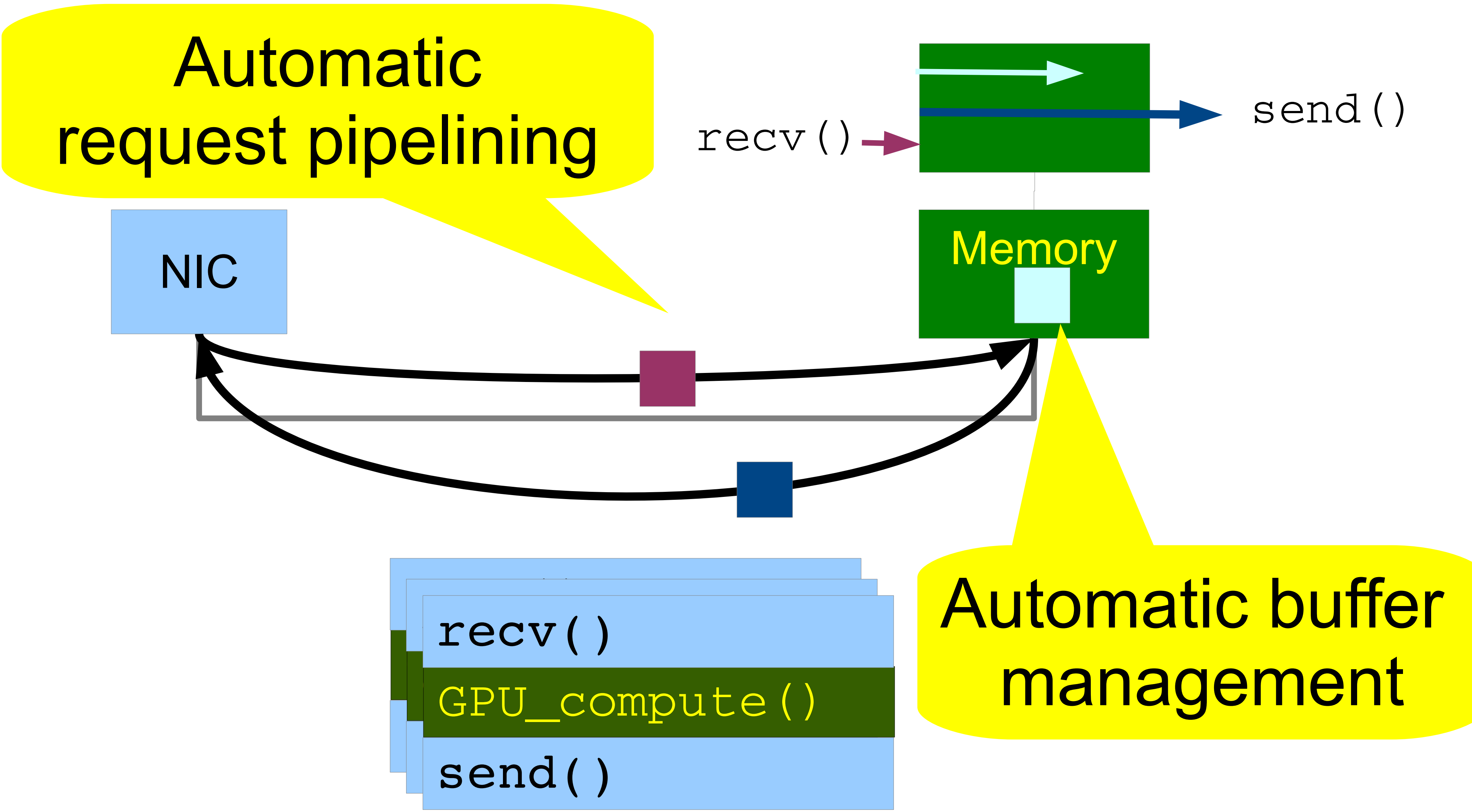



```
recv()
GPU_compute()
send()
```

GPU-accelerated server with GPUnet

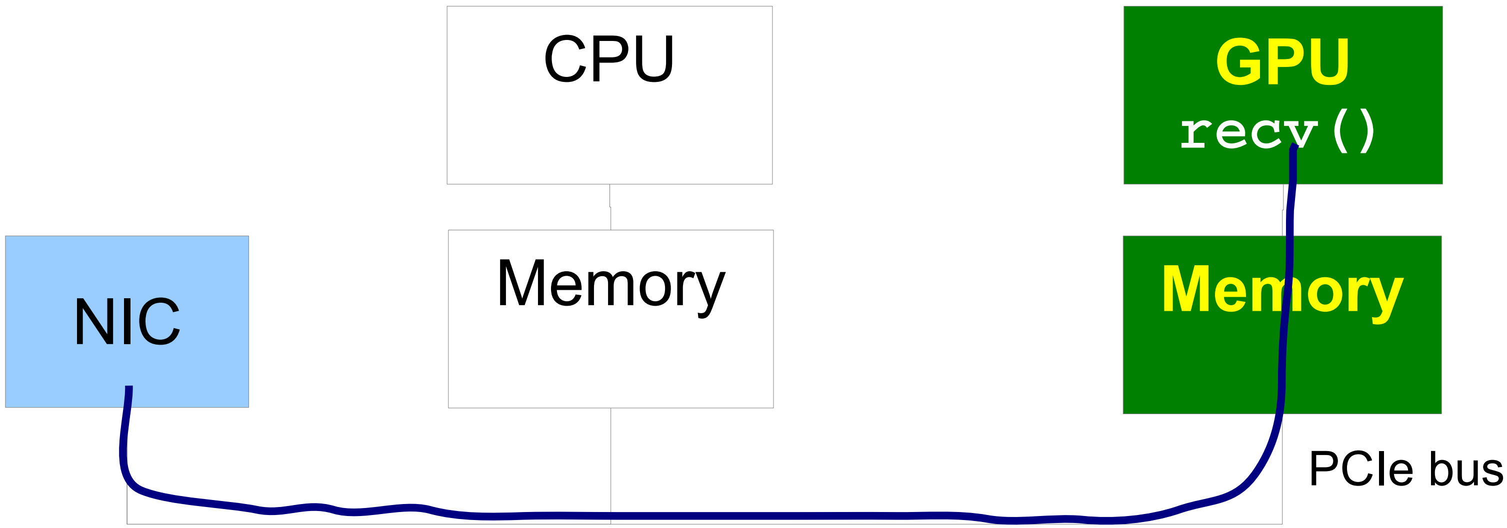


GPU-accelerated server with GPUnet



Building a socket abstraction for GPUs

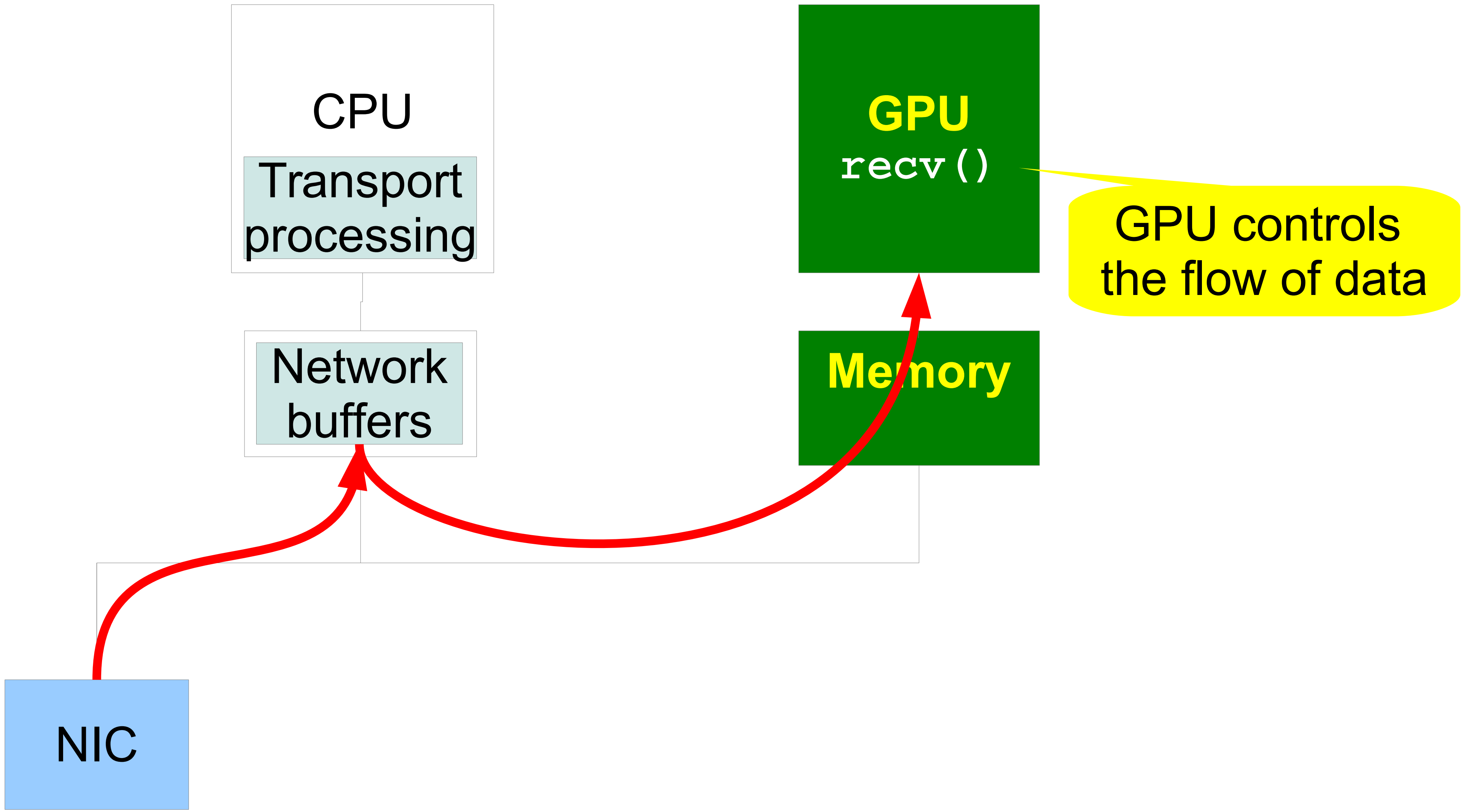
Goals



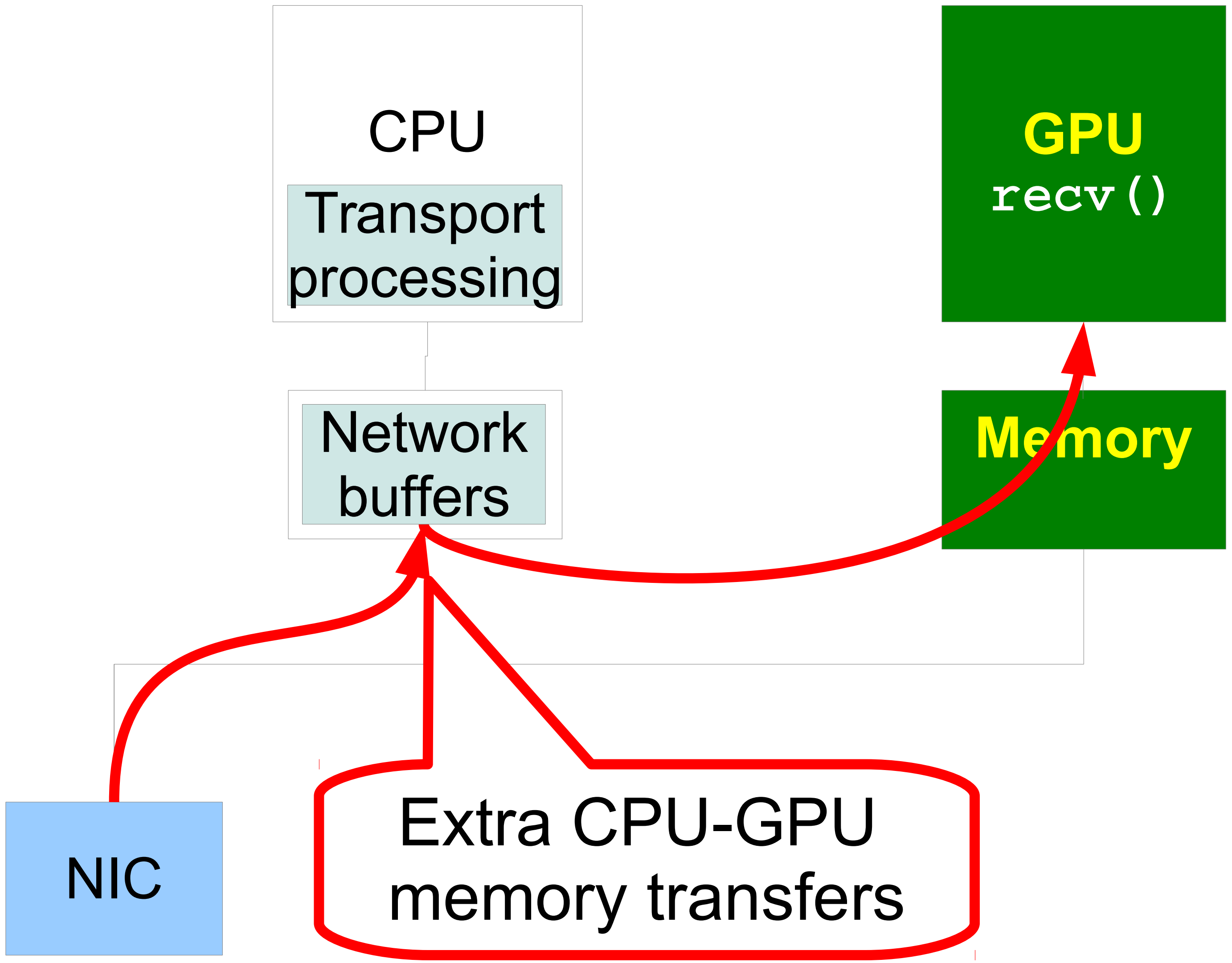
Simplicity
Reliable streaming abstraction for GPUs

Performance
NIC → GPU
data path optimizations

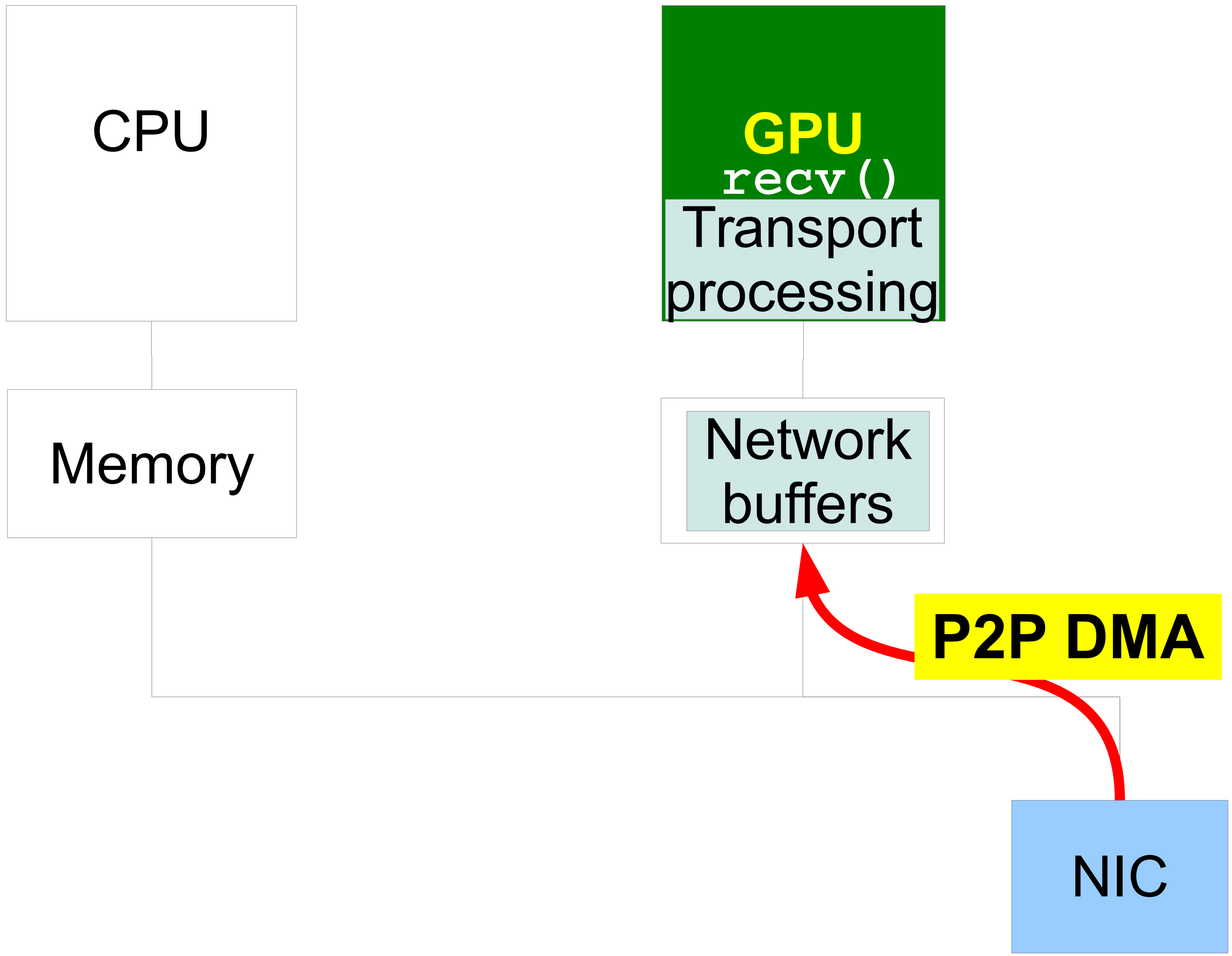
Design option 1: Transport layer processing on CPU



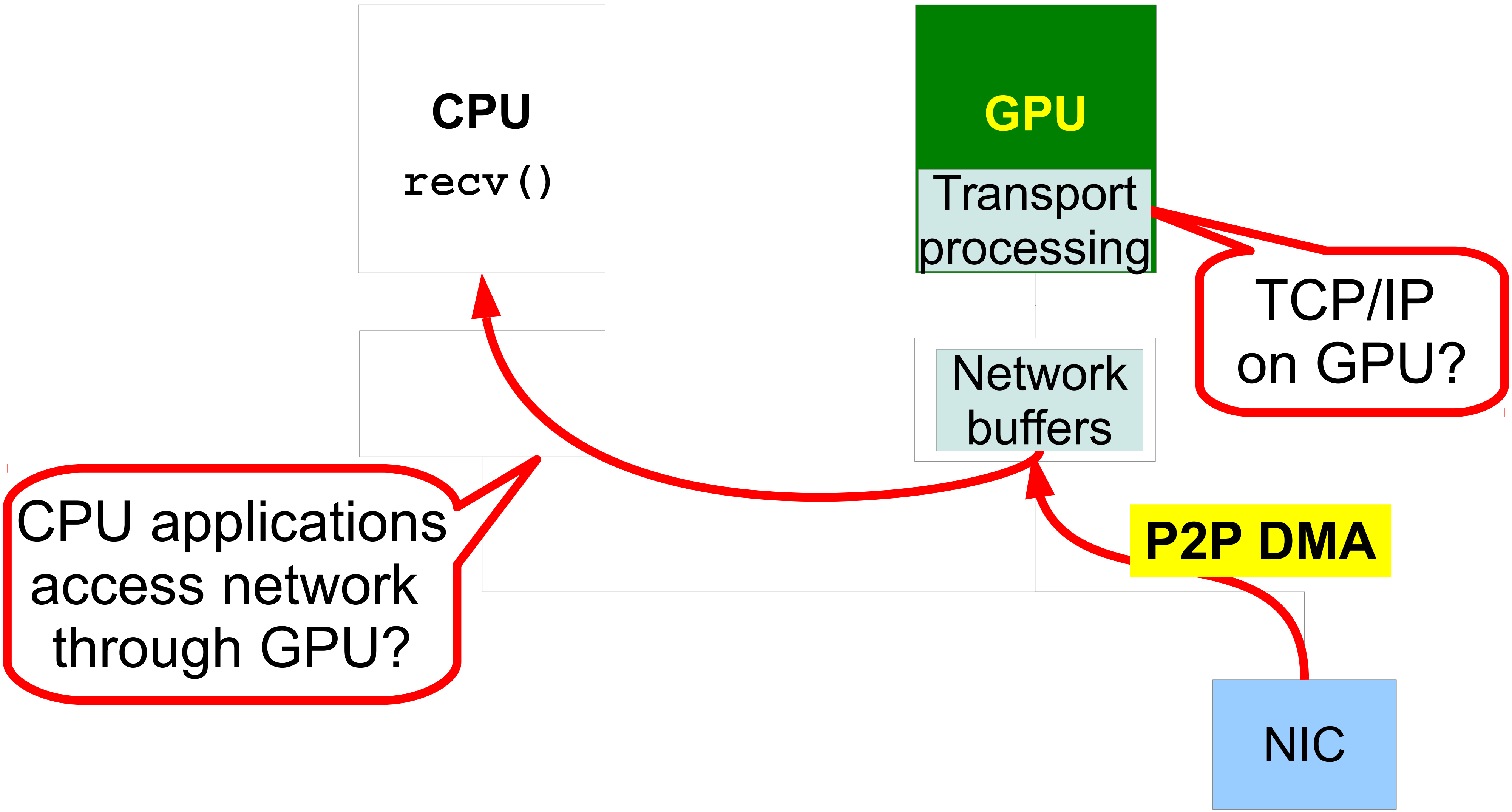
Design option 1: Transport layer processing on CPU



Design option 2:
Transport layer processing on GPU



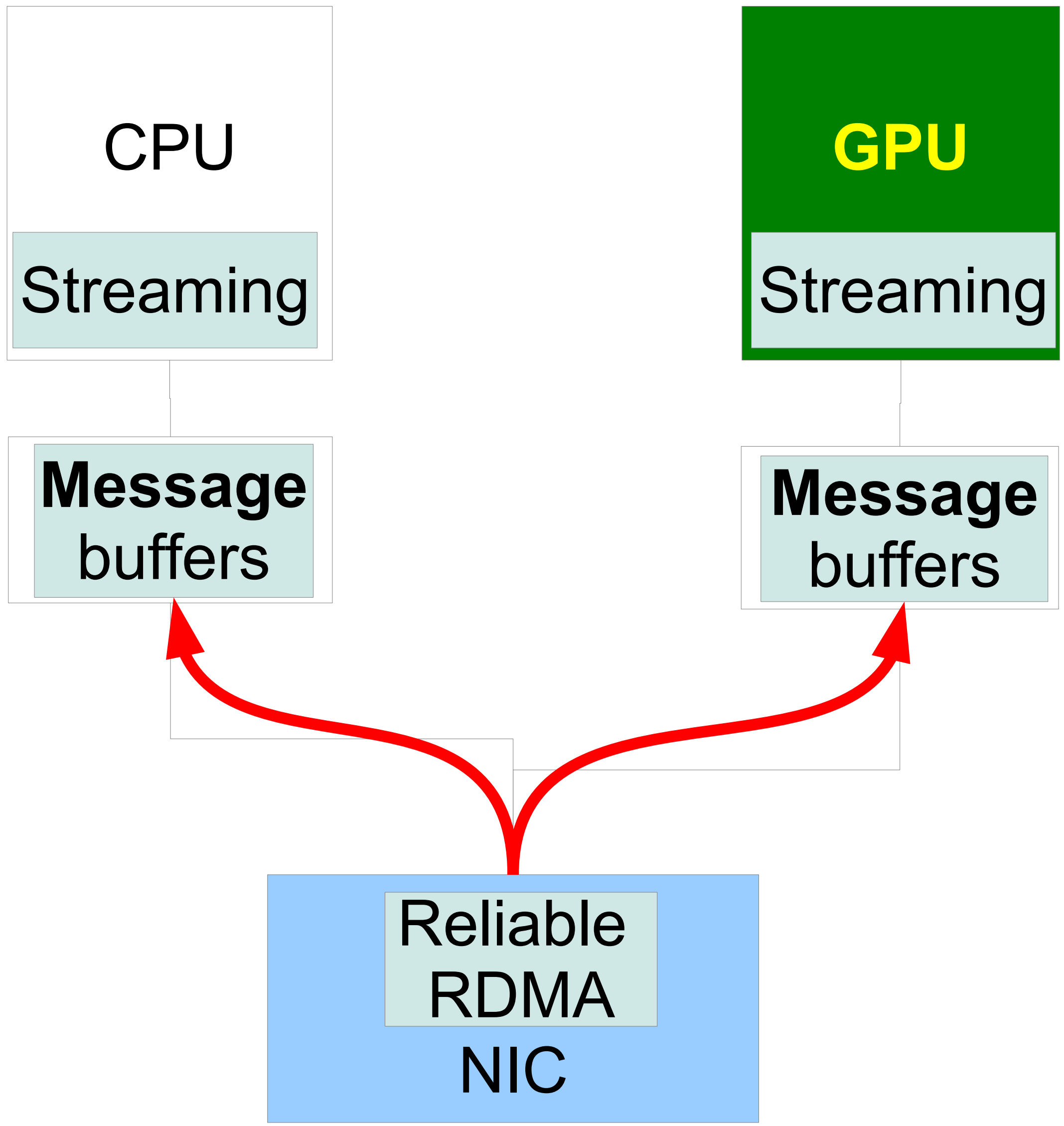
Design option 2: Transport layer processing on GPU



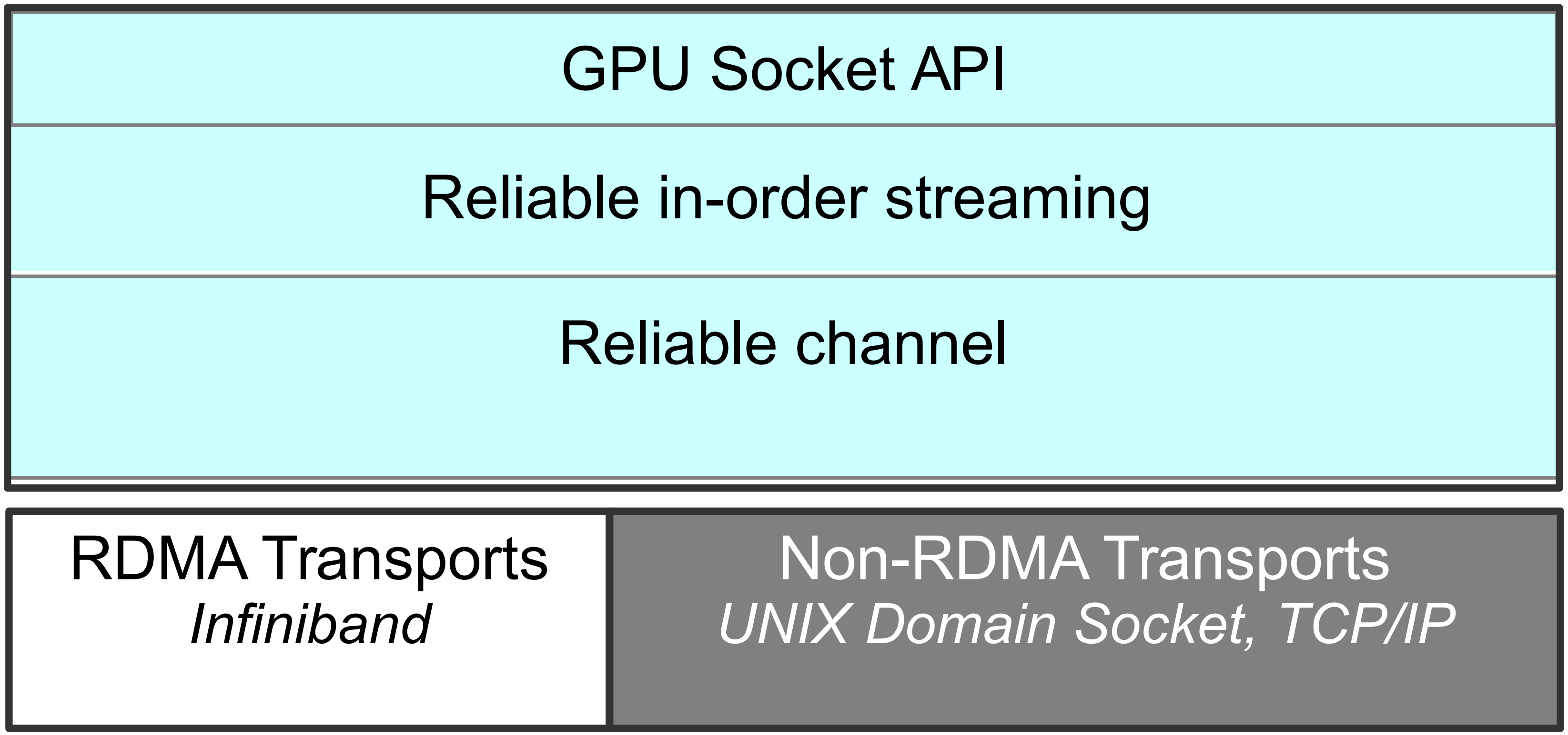
Not CPU, Not GPU

We need help from NIC hardware

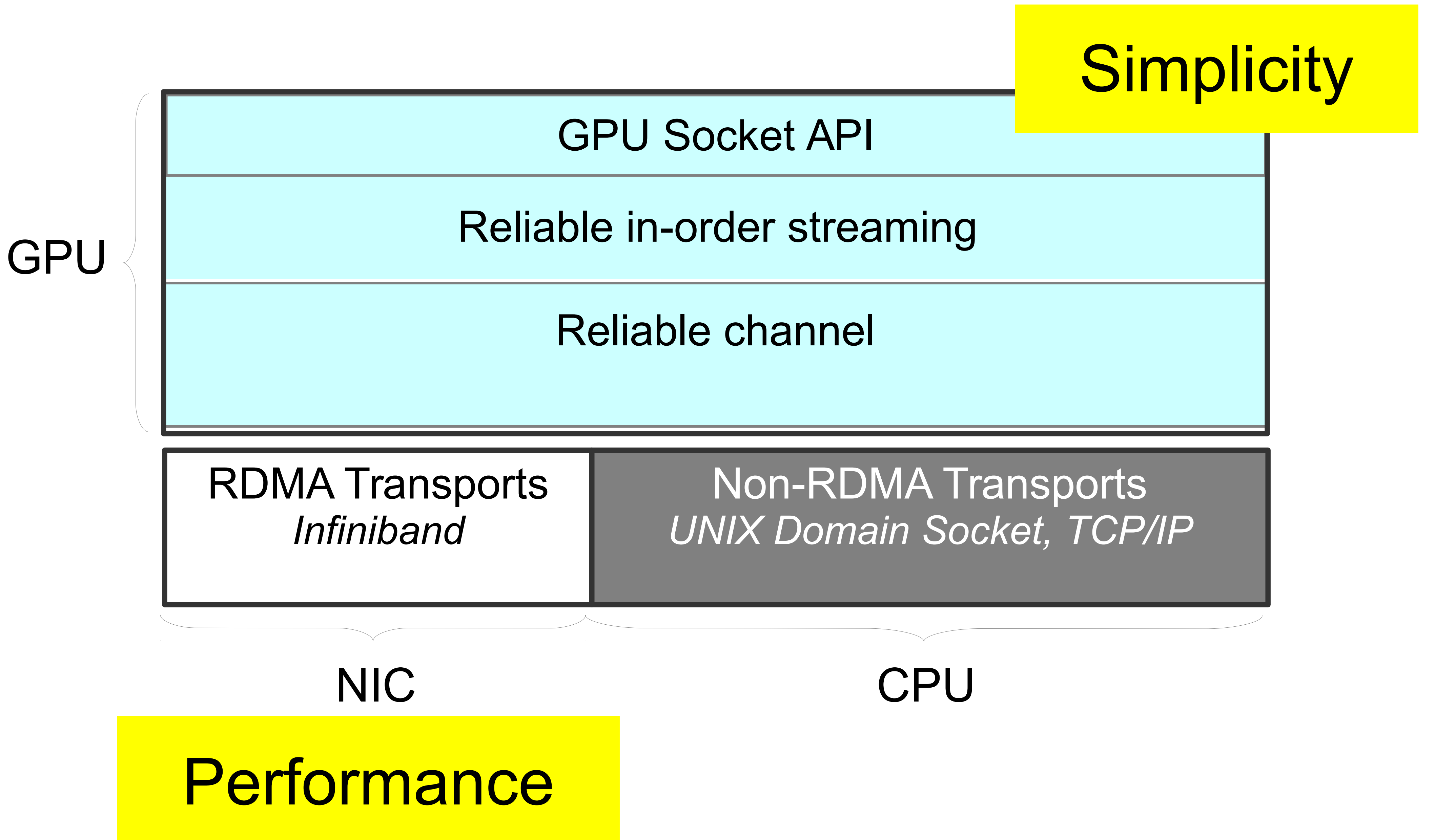
RDMA: offloading transport layer processing to NIC



GPUnet layers



GPUnet layers



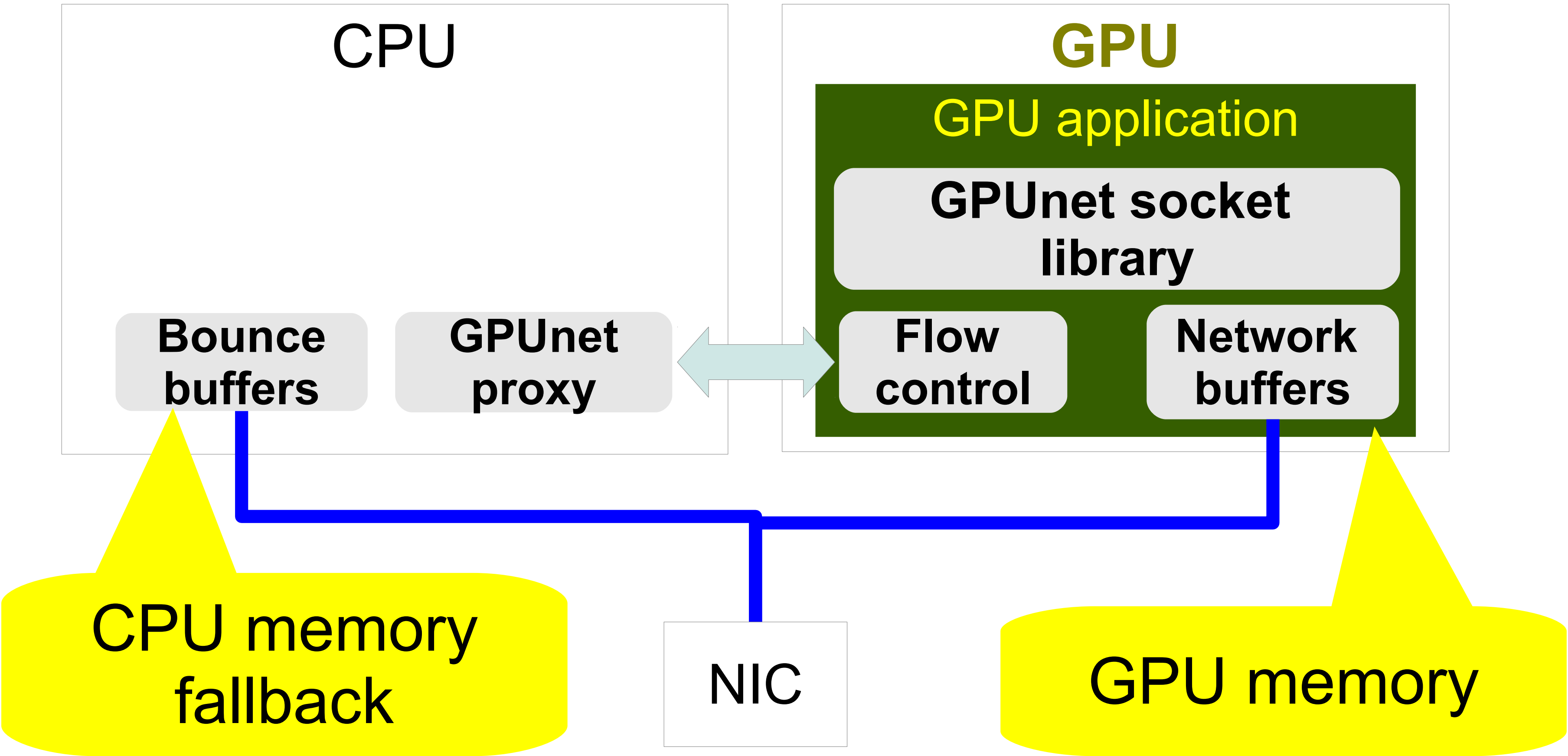
See the paper for

- Coalesced API calls
- Latency-optimized GPU-CPU flow control
- Memory management
- Bounce buffers
- Non-RDMA support
- GPU performance optimizations

Implementation

- Standard API calls, blocking/nonblocking
- **libGPUnet.a**: AF_INET, Streaming over Infiniband RDMA
  - Fully compatible with CPU **rsocket** library
- **libUNIXnet.a**: AF_LOCAL: Unix Domain Sockets support for inter GPU/CPU-GPU

Implementation

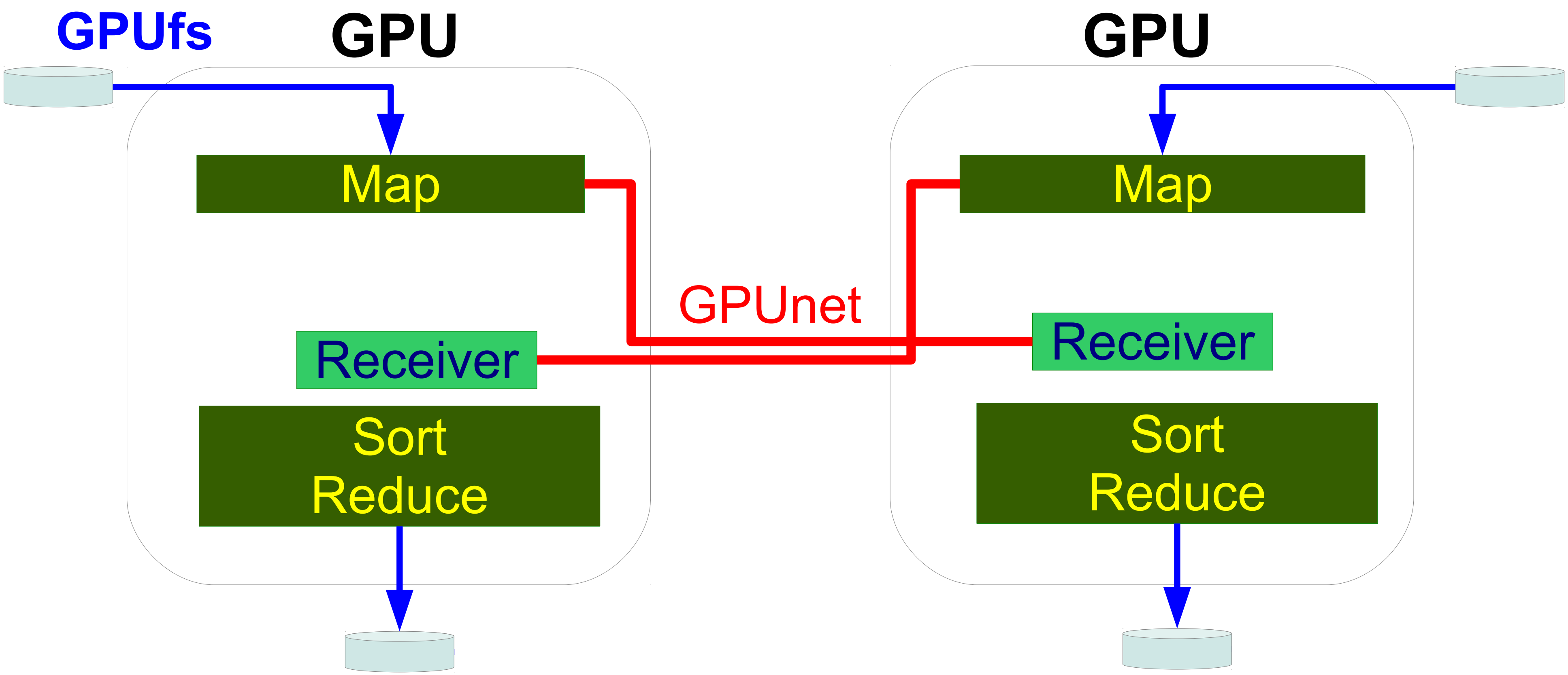


Evaluation

- Analysis of GPU-native server design
  - Matrix product server
- **In-GPU-memory MapReduce**
- **Face verification server**

2x6 Intel E5-2620, NVIDIA Tesla K20Xm GPU, Mellanox Connect-IB HCA, Switch-X bridge

In-GPU-memory MapReduce

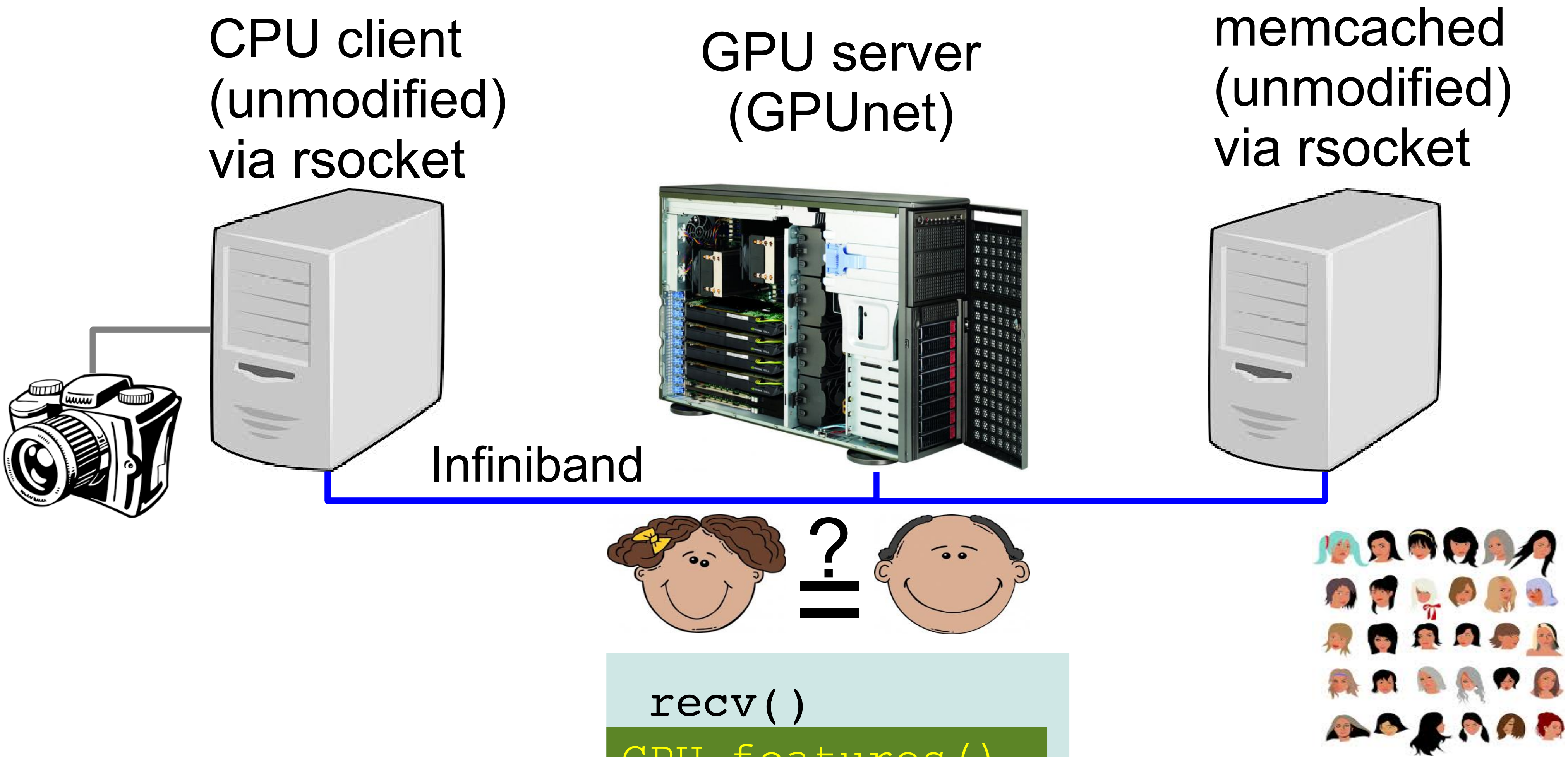


In-GPU-memory MapReduce: Scalability

	1 GPU (no network)	4 GPUs (GPUnet)
K-means	5.6 sec	1.6 sec (**3.5x**)
Word-count	29.6 sec	10 sec (**2.9x**)

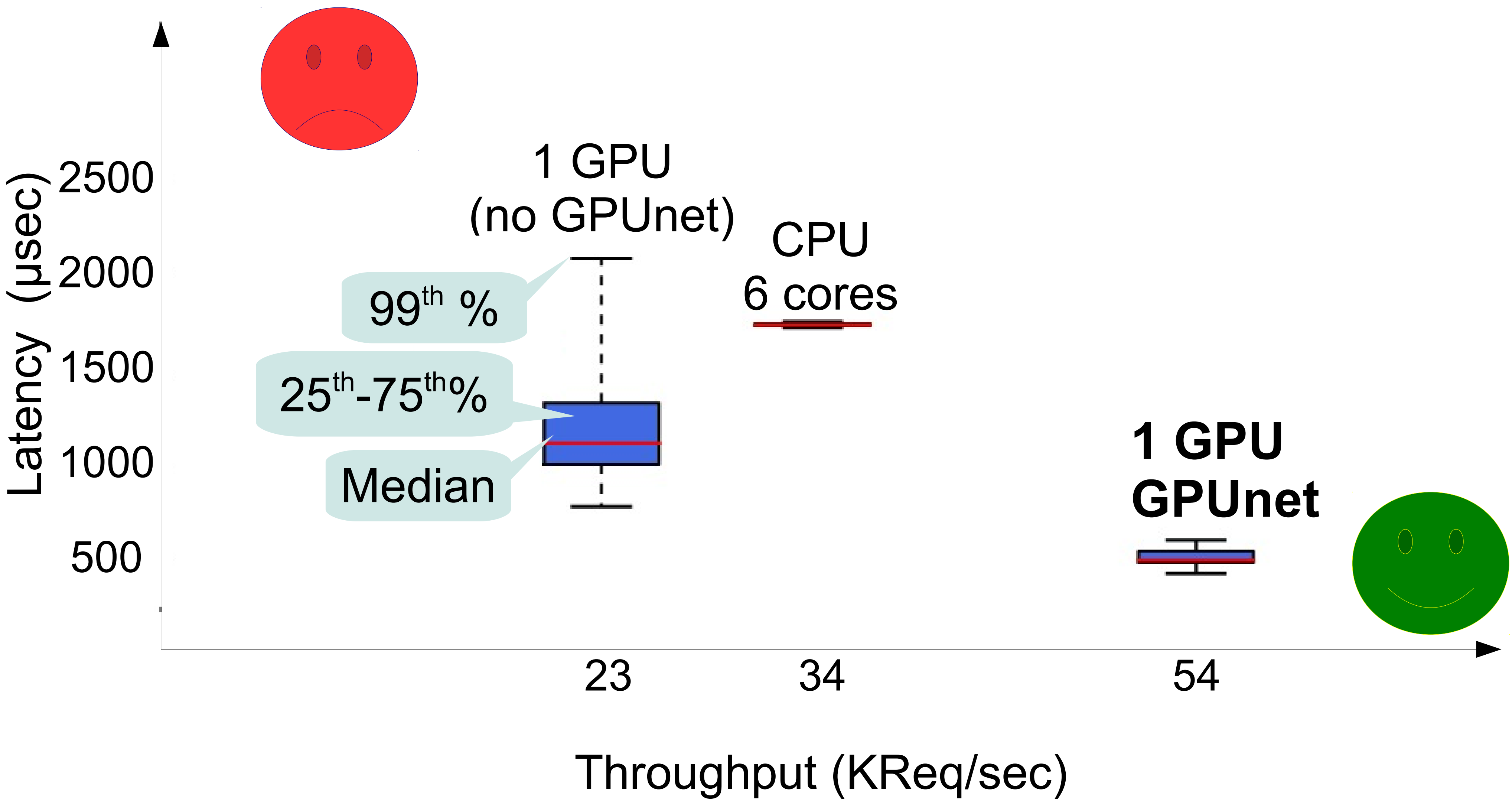
GPUnet enables scale-out
for GPU – accelerated systems

Face verification server

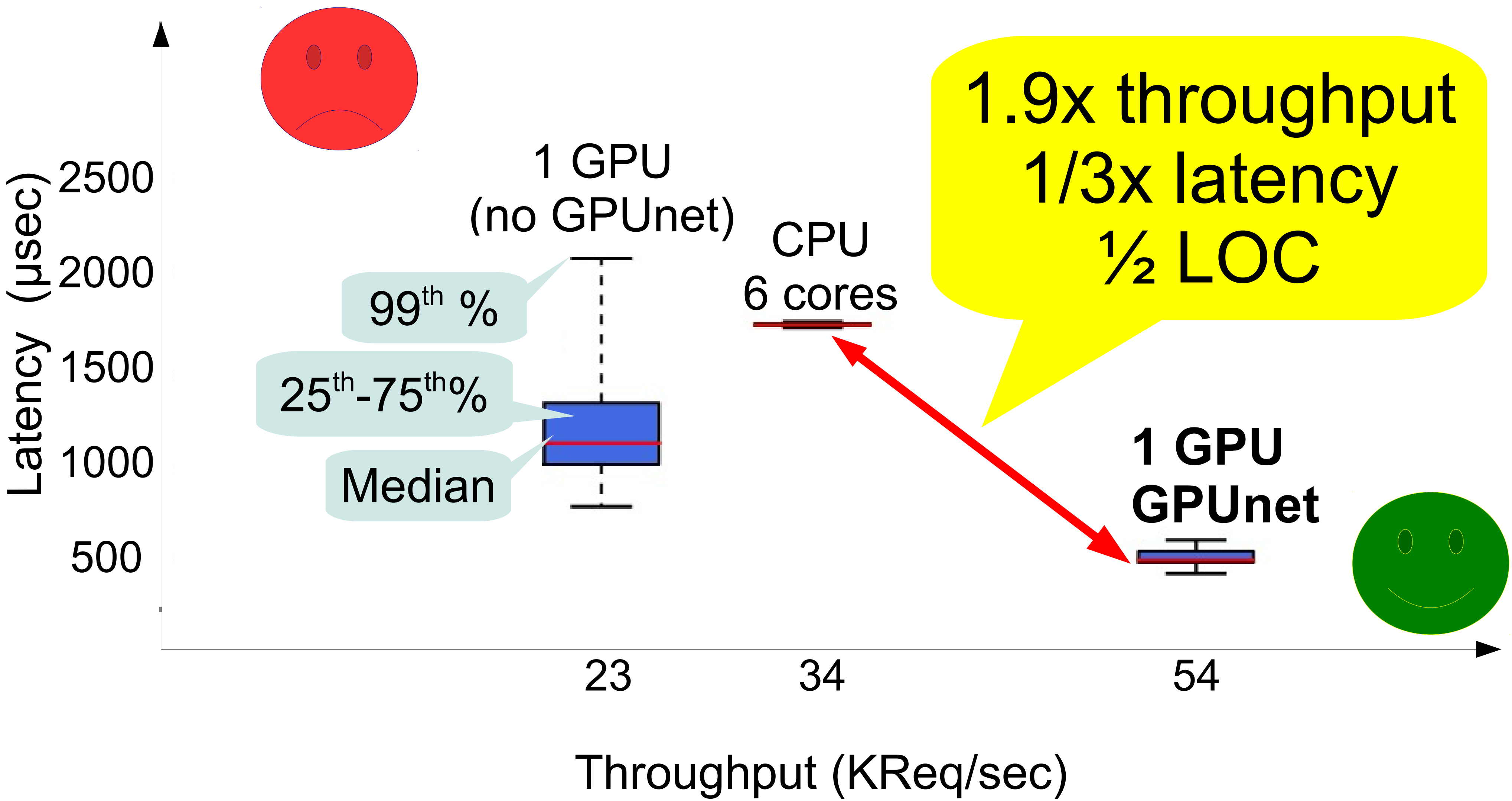



```
  recv()
GPU_features()
  query DB()
GPU_compare()
  send()
```

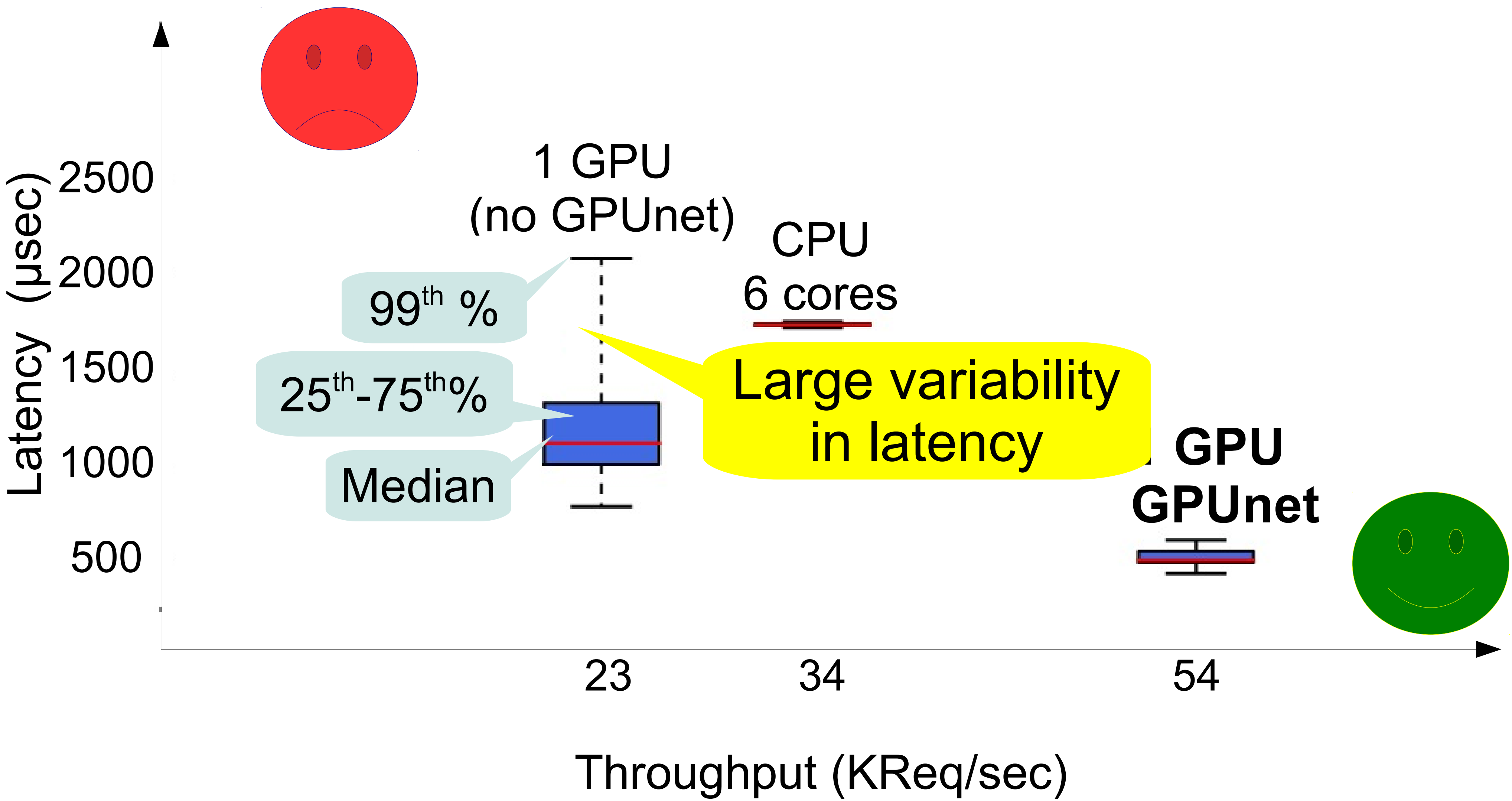
Face verification: Different implementations



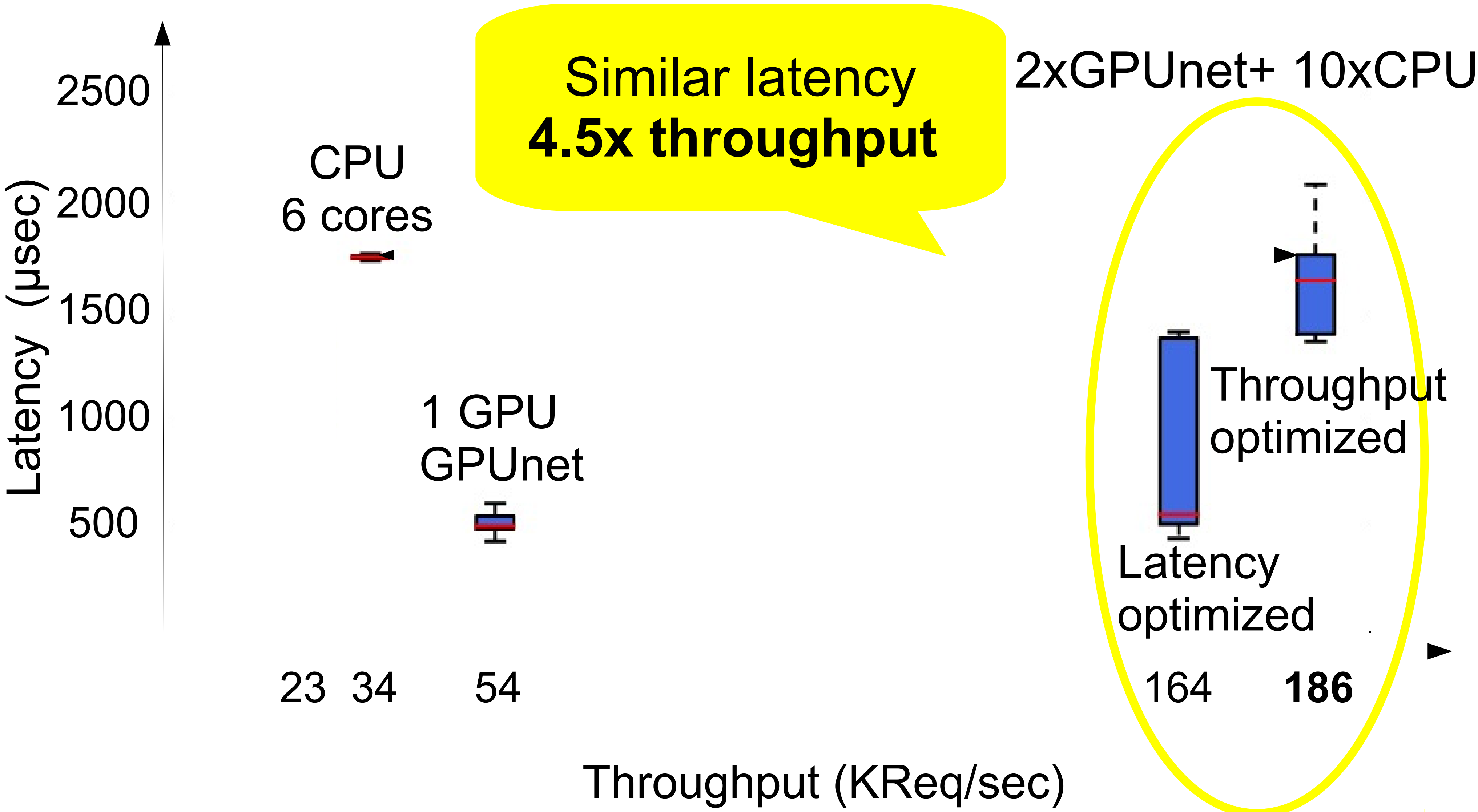
Face verification: Different implementations



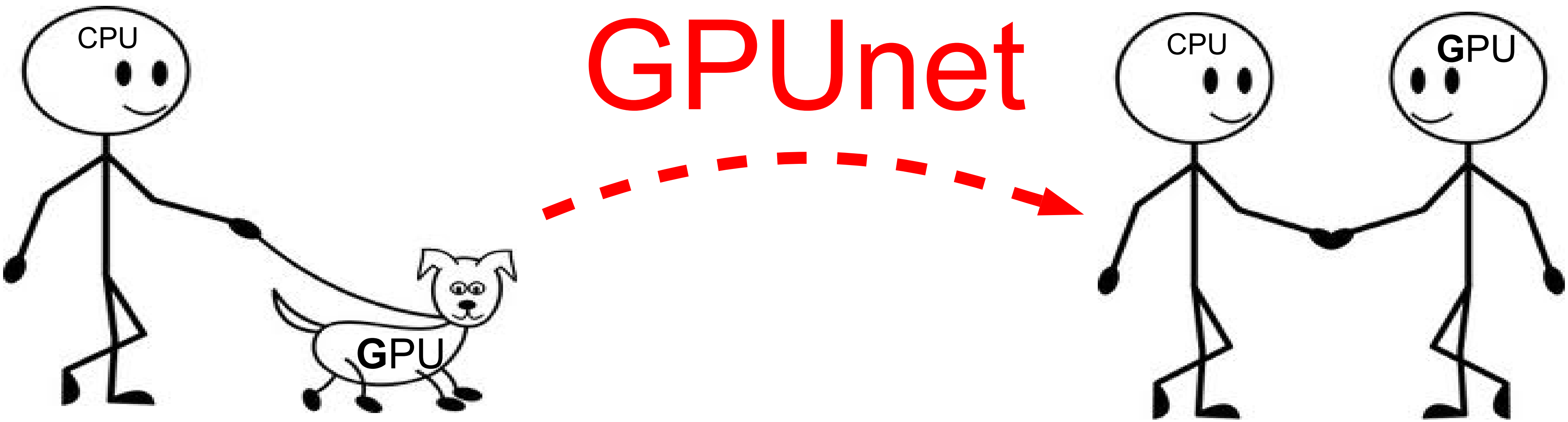
Face verification: Different implementations



Face verification on all processors 2xGPU + 10xCPU



Set GPUs free!



GPUnet is a library providing networking abstractions for GPUs

https://github.com/ut-osa/gpunet

 mark@ee.technion.ac.il