iSwitch: Accelerating Distributed Reinforcement Learning with In-Switch Computing

Youjie Li  
Deming Chen

Iou-Jen Liu  
Alexander Schwing

Yifan Yuan  
Jian Huang

University of Illinois at Urbana-Champaign  
Electrical & Computer Engineering
AI Applications are Increasingly Operating in Dynamic Environments

- Autonomous Driving
- Robotics
- Games
AI Applications are Increasingly Operating in Dynamic Environments

Autonomous Driving  
Robotics  
Games

Reinforcement Learning Empowers AI Applications to Take Real-Time Intelligent Actions
What is Reinforcement Learning?

Agent

Environment
What is Reinforcement Learning?
What is Reinforcement Learning?

Agent

Action

Environment

State
What is Reinforcement Learning?
What is Reinforcement Learning?

Agent \rightarrow \text{Action} \rightarrow \text{Environment} \rightarrow \text{Next State} \rightarrow \text{Reward} \rightarrow \text{Agent}
What is Reinforcement Learning?

Model

Agent

Action

Environment

Next State

Reward
What is Reinforcement Learning?
What is Reinforcement Learning?
What is Reinforcement Learning?
What is Reinforcement Learning?

Train a Typical RL Agent on a Single GPU = 8 Days*

*Mnih, ICML’16
What is Reinforcement Learning?

RL Requires Distributed Training for Improved Performance

Train a Typical RL Agent on a Single GPU = 8 Days*

*Mnih, ICML’16
Centralized Distributed RL Training: Parameter-Server Based
Centralized Distributed RL Training: Parameter-Server Based
Centralized Distributed RL Training: Parameter-Server Based
Centralized Distributed RL Training: Parameter-Server Based
Centralized Distributed RL Training: Parameter-Server Based
Centralized Distributed RL Training: Parameter-Server Based

- Parameter Server
  - Sum
  - Update
  - Weight

- Switch

- Workers

- Model

- Gradient

- Multiple Network Hops
Centralized Distributed RL Training: Parameter-Server Based

Parameter Server
- Sum
- Update
- Weight

Switch

Central Bottleneck

Multiple Network Hops

Workers

Model

Gradient
Decentralized Distributed RL Training: AllReduce Based
Decentralized Distributed RL Training: AllReduce Based
Decentralized Distributed RL Training: AllReduce Based
Decentralized Distributed RL Training: AllReduce Based
Decentralized Distributed RL Training: AllReduce Based

Diagram:
- Workers
- Model
- Sum
- Switch
- Full
- Gradient
- Aggregated Gradient
- Ring-AllReduce
- Aggregation Complete!
Decentralized Distributed RL Training: AllReduce Based

Excessive Network Hops
Network Communication is the Bottleneck in Distributed RL Training

Centralized Design

Parameter Server

Switch

Gradient

Workers

Decentralized Design

Ring-AllReduce

Switch

Gradient

Workers
Network Communication is the Bottleneck in Distributed RL Training

Centralized Design

- Parameter Server
- Switch
- Workers

Centralized Design Network Hops = 4

Decentralized Design

- Ring-AllReduce
- Switch
- Workers

Decentralized Design
Network Communication is the Bottleneck in Distributed RL Training

Centralized Design

Parameter Server

Switch

Workers

Gradient

Network Hops = 4

Decentralized Design

Ring-AllReduce

Switch

Workers

Gradient

Network Hops = 4N - 4
The Unique Characteristic of Distributed RL Training: Latency Critical

<table>
<thead>
<tr>
<th>RL Benchmark</th>
<th>DQN-Atari</th>
<th>A2C-Atari</th>
<th>PPO-MuJoCo</th>
<th>DDPG-MuJoCo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient Size</td>
<td>6 MB</td>
<td>3 MB</td>
<td>40 KB</td>
<td>158 KB</td>
</tr>
<tr>
<td>Training Iterations</td>
<td>200 M</td>
<td>2 M</td>
<td>0.2 M</td>
<td>3 M</td>
</tr>
</tbody>
</table>
### RL Benchmark

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>DQN-Atari</th>
<th>A2C-Atari</th>
<th>PPO-MuJoCo</th>
<th>DDPG-MuJoCo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient Size</td>
<td>6 MB</td>
<td>3 MB</td>
<td>40 KB</td>
<td>158 KB</td>
</tr>
<tr>
<td>Training Iterations</td>
<td>200 M</td>
<td>2 M</td>
<td>0.2 M</td>
<td>3 M</td>
</tr>
</tbody>
</table>

### DNN Benchmark

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>AlexNet-ImageNet</th>
<th>ResNet50-ImageNet</th>
<th>VGG16-ImageNet</th>
<th>MLP-MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient Size</td>
<td>250 MB</td>
<td>100 MB</td>
<td>525 MB</td>
<td>4 MB</td>
</tr>
<tr>
<td>Training Iterations</td>
<td>320 K</td>
<td>600 K</td>
<td>370 K</td>
<td>10 K</td>
</tr>
</tbody>
</table>
# The Unique Characteristic of Distributed RL Training: Latency Critical

<table>
<thead>
<tr>
<th>RL Benchmark</th>
<th>DQN-Atari</th>
<th>A2C-Atari</th>
<th>PPO-MuJoCo</th>
<th>DDPG-MuJoCo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient Size</td>
<td>6 MB</td>
<td>3 MB</td>
<td>40 KB</td>
<td>158 KB</td>
</tr>
<tr>
<td>Training Iterations</td>
<td>200 M</td>
<td>2 M</td>
<td>0.2 M</td>
<td>3 M</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DNN Benchmark</th>
<th>AlexNet-ImageNet</th>
<th>ResNet50-ImageNet</th>
<th>VGG16-ImageNet</th>
<th>MLP-MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gradient Size</td>
<td>250 MB</td>
<td>100 MB</td>
<td>525 MB</td>
<td>4 MB</td>
</tr>
<tr>
<td>Training Iterations</td>
<td>320 K</td>
<td>600 K</td>
<td>370 K</td>
<td>10 K</td>
</tr>
</tbody>
</table>

88x Smaller Gradient Size
158x More Iterations
The Unique Characteristic of Distributed RL Training: Latency Critical

<table>
<thead>
<tr>
<th></th>
<th>DQN-Atari</th>
<th>A2C-Atari</th>
<th>PPO-MuJoCo</th>
<th>DDPG-MuJoCo</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gradient Size</strong></td>
<td>6 MB</td>
<td>3 MB</td>
<td>40 KB</td>
<td>158 KB</td>
</tr>
<tr>
<td><strong>Training Iterations</strong></td>
<td>200 M</td>
<td>2 M</td>
<td>0.2 M</td>
<td>3 M</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>AlexNet-ImageNet</th>
<th>ResNet50-ImageNet</th>
<th>VGG16-ImageNet</th>
<th>MLP-MNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Gradient Size</strong></td>
<td>250 MB</td>
<td>100 MB</td>
<td>525 MB</td>
<td>4 MB</td>
</tr>
<tr>
<td><strong>Training Iterations</strong></td>
<td>320 K</td>
<td>600 K</td>
<td>370 K</td>
<td>10 K</td>
</tr>
</tbody>
</table>

Distributed RL Training is Latency Critical

88x Smaller Gradient Size

158x More Iterations
Quantifying the Network Overhead in Distributed RL Training

Local Computation | Grad Aggregation
---|---
DQN | A2C | PPO | DDPG
Parameter Server

Local Computation | Grad Aggregation
---|---
DQN | A2C | PPO | DDPG
AllReduce
Quantifying the Network Overhead in Distributed RL Training

Gradient Aggregation over the Network Dominates the Training Time (50~83%)
Gradient Aggregation over the Network Dominates the Training Time (50~83%)
In-Switch Acceleration: A New Distributed Computing Paradigm

Programmable Switch + Aggregation Accelerator =
In-Switch Acceleration: A New Distributed Computing Paradigm

Programmable Switch

Aggregation Accelerator

Performance

Reduce End-to-End Network Latency
In-Switch Acceleration: A New Distributed Computing Paradigm

Programmable Switch + Aggregation Accelerator = Performance

Reduce End-to-End Network Latency

Programmability

Hardware-Algorithm Co-Design
In-Switch Acceleration: A New Distributed Computing Paradigm

Programmable Switch + Aggregation Accelerator = Performance

Programmability

- Hardware-Algorithm Co-Design

Scalability

- Scale Training at Rack Scale

Reduce End-to-End Network Latency
Challenges of In-Switch Acceleration

No Impact on Regular Switch Functions
Challenges of In-Switch Acceleration

No Impact on Regular Switch Functions

Limited On-Chip Resources
Challenges of In-Switch Acceleration

- No Impact on Regular Switch Functions
- Limited On-Chip Resources
- Scale with More Switches and Nodes
Basics of Programmable Switch

Control Plane

Data Plane
Basics of Programmable Switch

Control Plane

Data Plane
Basics of Programmable Switch

Control Plane

Data Plane
Basics of Programmable Switch

Control Plane

Data Plane
Basics of Programmable Switch

Control Plane

Data Plane
Basics of Programmable Switch

Control Plane

Data Plane
Basics of Programmable Switch

Control Plane

Data Plane

![Diagram showing input ports, output ports, and data flow]

Input Port  Output Ports

Head  Data
Basics of Programmable Switch

Control Plane

Data Plane

Packet Forwarding

Input Port

Output Ports

Head

Data
Basics of Programmable Switch

Control Plane

Forwarding Control

Data Plane

Packet Forwarding

Input Port

Output Ports

Head

Data
Basics of Programmable Switch

Control Plane

Forwarding Control
System Configuration

Data Plane

Packet Forwarding

Input Port
Output Ports

Head
Data
Basics of Programmable Switch

Control Plane
- Forwarding Control
- System Configuration
- ...

Data Plane

Packet Forwarding

Input Port

Output Ports

Head

Data

Packet Forwarding
Basics of Programmable Switch

Control Plane

Forwarding Control

System Configuration

Data Plane

Input Arbiter

Packet Process

Output Port Lookup

Receiver

RxQ

TxQ

Transmitter

Receiver

RxQ

TxQ

Transmitter

Receiver

RxQ

TxQ

Transmitter

Receiver

RxQ

TxQ

Transmitter

Receiver

RxQ

TxQ

Transmitter

Data Plane

Receiver

RxQ

TxQ

Transmitter

Receiver

RxQ

TxQ

Transmitter

Receiver

RxQ

TxQ

Transmitter

Receiver

RxQ

TxQ

Transmitter

Receiver

RxQ

TxQ

Transmitter
Basics of Programmable Switch

Control Plane

Forwarding Control

System Configuration

Data Plane

Receiver → RxQ → Input Arbiter → Packet Process → Output Port Lookup → Transmitter

RxQ

Receiver

RxQ

Receiver

RxQ

Receiver

RxQ

Header

Data

TxQ

Transmitter

TxQ

Transmitter

TxQ

Transmitter

TxQ

Transmitter
Basics of Programmable Switch

Control Plane

Forwarding Control

System Configuration

Data Plane

Receiver → RxQ

Header → Data

Input Arbiter

Packet Process

Output Port Lookup

TxQ → Transmitter

Receiver → RxQ

Receiver → RxQ

Receiver → RxQ

RxQ

RxQ

RxQ

Transmitter

Transmitter

Transmitter

Transmitter

Transmitter
Basics of Programmable Switch

Control Plane
- Forwarding Control
- System Configuration
- ...

Data Plane
- RxQ
- Input Arbiter
- Packet Process
  - Header
  - Data
- Output Port Lookup
- TxQ
- Transmitter
Basics of Programmable Switch

Control Plane
- Forwarding Control
- System Configuration

Data Plane
- Receiver → RxQ → Input Arbiter → Packet Process → Output Port Lookup → TxQ → Transmitter
- Receiver → RxQ → Input Arbiter → Packet Process → Output Port Lookup → TxQ → Transmitter
- Receiver → RxQ → Input Arbiter → Packet Process → Output Port Lookup → TxQ → Transmitter
- Receiver → RxQ → Input Arbiter → Packet Process → Output Port Lookup → TxQ → Transmitter

Packet Process
- Input Arbiter
- Output Port Lookup
- Header
- Data

Transmitter
Basics of Programmable Switch

Control Plane
- Forwarding Control
- System Configuration
- ...

Data Plane
- Receiver - RxQ
- Input Arbiter
- Packet Process
  - Output Port Lookup
- TxQ - Transmitter
- Header
- Data
Integrating Aggregation Accelerator into the Programmable Switch

Data Plane

Input Arbiter

Output Port Lookup

Packet Process

Receiver → RxQ → Input Arbiter → Output Port Lookup → TxQ → Transmitter

RxQ

TxQ

Transmitter

Receiver

Receiver

Receiver

Receiver

RxQ

RxQ

RxQ

RxQ

TxQ

TxQ

TxQ

TxQ
Integrating Aggregation Accelerator into the Programmable Switch

Data Plane

Core of Regular Functions

Receiver -> RxQ

Receiver -> RxQ

Receiver -> RxQ

Receiver -> RxQ

Input Arbiter

TxQ -> Transmitter

TxQ -> Transmitter

TxQ -> Transmitter

TxQ -> Transmitter
Integrating Aggregation Accelerator into the Programmable Switch
Integrating Aggregation Accelerator into the Programmable Switch

Data Plane

Receiver → RxQ

Input Arbiter

Packet Process

Output Port Lookup

Accelerator

RxQ

Transmitter

TxQ

Transmitter

Transmitter

Transmitter

Transmitter
Integrating Aggregation Accelerator into the Programmable Switch

Data Plane

Input Arbiter

Packet Process

Output Port Lookup

Accelerator

Receiver → RxQ

Receiver → RxQ

Receiver → RxQ

Receiver → RxQ

TxQ → Transmitter

TxQ → Transmitter

TxQ → Transmitter

TxQ → Transmitter

RxQ

RxQ

RxQ

RxQ

ECE ILLINOIS
Integrating Aggregation Accelerator into the Programmable Switch

Data Plane

Input Arbiter

Packet Process
Output Port Lookup

Accelerator

Receiver → RxQ

Receiver → Header

Receiver → RxQ

Receiver → RxQ

TxQ → Transmitter

TxQ → Transmitter

TxQ → Transmitter

TxQ → Transmitter
Integrating Aggregation Accelerator into the Programmable Switch

Data Plane

Input Arbiter

Regular

Packet Process

Output Port Lookup

Accelerator

Receiver

RxQ

Transmitter

TxQ

Transmitter

Receiver

Header

TxQ

Transmitter

RxQ

Transmitter

RxQ

Transmitter

RxQ
Integrating Aggregation Accelerator into the Programmable Switch

Data Plane

Input Arbiter

Regular

Packet Process

Output Port

Header

Accelerator

Receiver → RxQ

Receiver → RxQ

Receiver → RxQ

Receiver → RxQ

TxQ → Transmitter

TxQ → Transmitter

TxQ → Transmitter

TxQ → Transmitter
Integrating Aggregation Accelerator into the Programmable Switch
Integrating Aggregation Accelerator into the Programmable Switch

Data Plane

Receiver → Header
Receiver → RxQ
Receiver → RxQ
Receiver → RxQ

Input Arbiter

Packet Process
Output Port Lookup

Accelerator

TxQ → Transmitter
TxQ → Transmitter
TxQ → Transmitter
TxQ → Transmitter
Integrating Aggregation Accelerator into the Programmable Switch

Data Plane

Input Arbiter

Packet Process

Output Port Lookup

Receiver

RxQ

Header

Gradient

Accelerator

TxQ

Transmitter

RxQ

RxQ

RxQ

TxQ

Transmitter

TxQ

Transmitter

TxQ

Transmitter

Data Plane

Accelerator
Integrating Aggregation Accelerator into the Programmable Switch

Data Plane

Input Arbiter

Packet Process

Output Port Lookup

Gradient

Receiver ➔ RxQ ➔ Input Arbiter ➔ Gradient ➔ Header ➔ Packet Process ➔ Output Port Lookup ➔ TxQ ➔ Transmitter

Receiver ➔ RxQ ➔ Input Arbiter ➔ Gradient ➔ Header ➔ Packet Process ➔ Output Port Lookup ➔ TxQ ➔ Transmitter

Receiver ➔ RxQ ➔ Input Arbiter ➔ Gradient ➔ Header ➔ Packet Process ➔ Output Port Lookup ➔ TxQ ➔ Transmitter

Receiver ➔ RxQ ➔ Input Arbiter ➔ Gradient ➔ Header ➔ Packet Process ➔ Output Port Lookup ➔ TxQ ➔ Transmitter
Integrating Aggregation Accelerator into the Programmable Switch

Data Plane

Input Arbiter

Output Port Lookup

Packet Process

Receiver

RxQ

RxQ

RxQ

Receiver

RxQ

RxQ

RxQ

Receiver

RxQ

RxQ

RxQ

Accelerator

Receiver

RxQ

RxQ

RxQ

Transmitter

TxQ

TxQ

TxQ

Transmitter

TxQ

TxQ
Integrating Aggregation Accelerator into the Programmable Switch

Data Plane

Input Arbiter

Output Port Lookup

RxQ

TxQ

Receiver

Transmitter

Data Plane

Regular Traffic

Gradient Traffic

Receiver

RxQ

Input Arbiter

Receiver

RxQ

Accelerator

RxQ

Receiver

RxQ
Integrating Aggregation Accelerator into the Programmable Switch

Hardware Acceleration Isolated From Regular Switch Function
Integrating Aggregation Accelerator into the Programmable Switch

Hardware Acceleration Isolated From Regular Switch Function
Integrating Aggregation Accelerator into the Programmable Switch

Hardware Acceleration Isolated From Regular Switch Function
Hardware Acceleration Isolated From Regular Switch Function
Integrating Aggregation Accelerator into the Programmable Switch

Hardware Acceleration Isolated From Regular Switch Function
Integrating Aggregation Accelerator into the Programmable Switch

Hardware Acceleration Isolated From Regular Switch Function
Integrating Aggregation Accelerator into the Programmable Switch

Hardware Acceleration Isolated From Regular Switch Function
Developing Light-Weight Accelerator for Aggregation
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

| Seg 0 | Seg 1 | … | Seg i | … | Seg N |

In-Switch Accelerator
Developing Light-Weight Accelerator for Aggregation

Gradient Vector: Seg 0, Seg 1, ..., Seg i, ..., Seg N

Pkt i

In-Switch Accelerator
Developing Light-Weight Accelerator for Aggregation
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

Seg 0  Seg 1  ...  Seg i  ...  Seg N

Pkt i
Separator

In-Switch Accelerator
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

Segment 0 | Segment 1 | ... | Segment i | ... | Segment N

Pkt i

Separator

Header

Payload

In-Switch Accelerator
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

| Seg 0 | Seg 1 | ... | Seg i | ... | Seg N |

Parser

Header

Separator

Payload

In-Switch Accelerator
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

Parser

Header

Separator

Payload

In-Switch Accelerator

[Diagram showing the process flow with labeled segments: Seg 0, Seg 1, ..., Seg i, ..., Seg N]
Developing Light-Weight Accelerator for Aggregation

Gradient Vector: Seg 0, Seg 1, ..., Seg i, ..., Seg N

Parser

Header

Separator

Payload

Buffer Module

In-Switch Accelerator
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

Seg 0  Seg 1  …  Seg i  …  Seg N

Parser

Header

Buffer Module

Pkt i

Separator

Payload

In-Switch Accelerator
Developing Light-Weight Accelerator for Aggregation

Gradient Vector: Seg 0 Seg 1 ... Seg i ... Seg N

Parser

Header

Separator

Payload

Buffer Module

In-Switch Accelerator
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

Seg 0 Seg 1 … Seg i … Seg N

Parser

Header

Separator

Payload

Buffer Module

Slicer

Elements

In-Switch Accelerator
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

| Seg 0 | Seg 1 | … | Seg i | … | Seg N |

- **Parser**
  - Header
  - Payload

- **Separator**
  - Seg Idx

- **Buffer Module**
  - Elements

- **In-Switch Accelerator**

Pkt i
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

Parser

Header

Separator

Payload

Buffer Module

Slicer

Elements

In-Switch Accelerator
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

Seg 0  Seg 1  ...  Seg i  ...  Seg N

Parser

Header

Separator

Payload

Buffer Module

Slicer

Elements

In-Switch Accelerator

Pkt i

Switch

Accelerator

Slicer

Elements
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

Seg 0  Seg 1  ...  Seg i  ...  Seg N

Parser

Header

Separator

Counter Module

Buffer Module

In-Switch Accelerator

Increment

Elements

Pkt i

Seg Idx
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

| Seg 0 | Seg 1 | ... | Seg i | ... | Seg N |

Parser

Header

Separator

Payload

Counter Module

Buffer Module

Slicer

Elements

In-Switch Accelerator

Pkt i

Seg Idx

In-Switch Switch
Developing Light-Weight Accelerator for Aggregation
Developing Light-Weight Accelerator for Aggregation

Gradient Vector: Seg 0, Seg 1, ..., Seg i, ..., Seg N

Parser → Counter Module
Header → Seg Idx
Separator → Payload
Slicer → Elements

In-Switch Accelerator
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

Parser

Header

Separator

Counter Module

Segment 0 Segment 1 … Segment i … Segment N

Buffer Module

In-Switch Accelerator

Parser

Header

Separator

Counter Module

Buffer Module

Threshold

Payload

Elements

In-Switch Accelerator

ECE ILLINOIS
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

- Seg 0
- Seg 1
- …
- Seg i
- …
- Seg N

Parser

Header

Separator

Payload

Counter Module

Threshold

Buffer Module

In-Switch Accelerator

Output Module

In

- Switch

Accelerator

Elements

Slicer

Pkt i
Developing Light-Weight Accelerator for Aggregation

Gradient Vector

Counter Module

Threshold

Parser

Header

Separator

Payload

Buffer Module

Slicer

Elements

In-Switch Accelerator

Output Module

Pkt i
Developing Light-Weight Accelerator for Aggregation

Accelerator Resource Consumption: extra 18.6% of LUT, 17.3% of FF, and 17 DSP
Aggregating Gradient at Packet-Level for Improved Parallelism

Conventional Vector-Level Aggregation
Aggregating Gradient at Packet-Level for Improved Parallelism

Conventional Vector-Level Aggregation

Packet-Level Aggregation in Our iSwitch
Aggregating Gradient at Packet-Level for Improved Parallelism

Conventional Vector-Level Aggregation

Packet-Level Aggregation in Our iSwitch

Further Reduce Aggregation Time
Extending Network Protocol for In-Switch Computing

Regular Packet:

<table>
<thead>
<tr>
<th>ETH</th>
<th>IP</th>
<th>UDP</th>
<th>Application Data</th>
</tr>
</thead>
</table>
Extending Network Protocol for In-Switch Computing

Data Packet of iSwitch:

<table>
<thead>
<tr>
<th>ETH</th>
<th>IP</th>
<th>UDP</th>
<th>Application Data</th>
</tr>
</thead>
</table>
Extending Network Protocol for In-Switch Computing

Data Packet of iSwitch:

<table>
<thead>
<tr>
<th>ETH</th>
<th>IP</th>
<th>UDP</th>
<th>Application Data</th>
</tr>
</thead>
</table>

Type-of-Service Field
Extending Network Protocol for In-Switch Computing

Data Packet of iSwitch:

```
| ETH | IP | UDP | Seg | Gradient |
```

Type-of-Service Field
Extending Network Protocol for In-Switch Computing

Data Packet of iSwitch:

Control Packet of iSwitch:
Extending Network Protocol for In-Switch Computing

Data Packet of iSwitch:

- ETH
- IP
- UDP
- Seg
- Gradient

Control Packet of iSwitch:

- ETH
- IP
- UDP
- Action
- Value (optional)

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Join</td>
<td>Join the training job</td>
</tr>
<tr>
<td>Leave</td>
<td>Leave the training job</td>
</tr>
<tr>
<td>Reset</td>
<td>Clear the accelerator on the switch</td>
</tr>
<tr>
<td>SetH</td>
<td>Set aggregation threshold H on switch</td>
</tr>
<tr>
<td>FBcast</td>
<td>Force broadcast a segment on switch</td>
</tr>
<tr>
<td>Help</td>
<td>Request a lost data packet for a worker</td>
</tr>
<tr>
<td>Ack</td>
<td>Confirm the success of some actions</td>
</tr>
</tbody>
</table>
Extending Network Protocol for In-Switch Computing

Data Packet of iSwitch:

![Data Packet Diagram]

Control Packet of iSwitch:

![Control Packet Diagram]

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Join</td>
<td>Join the training job</td>
</tr>
<tr>
<td>Leave</td>
<td>Leave the training job</td>
</tr>
<tr>
<td>Reset</td>
<td>Clear the accelerator on the switch</td>
</tr>
<tr>
<td>SetH</td>
<td>Set aggregation threshold H on switch</td>
</tr>
<tr>
<td>FBcast</td>
<td>Force broadcast a segment on switch</td>
</tr>
<tr>
<td>Help</td>
<td>Request a lost data packet for a worker</td>
</tr>
<tr>
<td>Ack</td>
<td>Confirm the success of some actions</td>
</tr>
</tbody>
</table>

iSwitch extension will NOT affect regular network functions
Supporting Different (Sync & Async) Training Execution Modes

Synchronous Distributed Training

Diagram:
- Programmable Switch
- Aggregation Accelerator
- Gradient
- Result
Supporting Different (Sync & Async) Training Execution Modes

Synchronous Distributed Training

In-Switch Acceleration Directly Applies
Supporting Different (Sync & Async) Training Execution Modes

Synchronous Distributed Training

Asynchronous Distributed Training

In-Switch Acceleration Directly Applies
Supporting Different (Sync & Async) Training Execution Modes

Synchronous Distributed Training

Asynchronous Distributed Training

In-Switch Acceleration Directly Applies

Keep Computing
Supporting Different (Sync & Async) Training Execution Modes

Synchronous Distributed Training

Asynchronous Distributed Training

In-Switch Acceleration Directly Applies

Keep Aggregating

Keep Computing
Supporting Different (Sync & Async) Training Execution Modes

Synchronous Distributed Training

Asynchronous Distributed Training

In-Switch Acceleration Directly Applies

HW/Algo Co-Design For Improved Parallelism
Scaling In-Switch Computing in Rack-Scale Data Centers

The Typical Network Architecture at Data Center
Scaling In-Switch Computing in Rack-Scale Data Centers

The Hierarchical Aggregation of iSwitch
Scaling In-Switch Computing in Rack-Scale Data Centers

The Hierarchical Aggregation of iSwitch
Scaling In-Switch Computing in Rack-Scale Data Centers

The Hierarchical Aggregation of iSwitch

Core Switches

"Aggregate" Switches

Top-of-Rack Switches

Racks of Servers

Core

Core

AGG

AGG
Scaling In-Switch Computing in Rack-Scale Data Centers

The Hierarchical Aggregation of iSwitch
Scaling In-Switch Computing in Rack-Scale Data Centers

The Hierarchical Aggregation of iSwitch
Scaling In-Switch Computing in Rack-Scale Data Centers

The Hierarchical Aggregation of iSwitch

Core Switches

“Aggregate” Switches

Grad Pkt

Top-of-Rack Switches

ToR

Racks of Servers
Scaling In-Switch Computing in Rack-Scale Data Centers

The Hierarchical Aggregation of iSwitch
Scaling In-Switch Computing in Rack-Scale Data Centers

The Hierarchical Aggregation of iSwitch
Scaling In-Switch Computing in Rack-Scale Data Centers

The Hierarchical Aggregation of iSwitch
Scaling In-Switch Computing in Rack-Scale Data Centers

No Additional Cost or Topology Change for Scaling In-Switch Computing
In-Switch Computing Implementation

RL Training Benchmarks

NetFPGA-SUME Board

GPU Cluster

DQN  A2C  PPO  DDPG
Reducing the End-to-End Training Time with iSwitch

![Graph showing the relationship between Average Episode Reward and Training Time (min) of DQN.](image-url)
Reducing the End-to-End Training Time with iSwitch

![Graph showing the reduction in training time with iSwitch.](image)

- **Parameter Server (PS)**

<table>
<thead>
<tr>
<th>Training Time (min) of DQN</th>
<th>Average Episode Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-25</td>
</tr>
<tr>
<td>250</td>
<td>-20</td>
</tr>
<tr>
<td>500</td>
<td>-15</td>
</tr>
<tr>
<td>750</td>
<td>-10</td>
</tr>
<tr>
<td>1000</td>
<td>0</td>
</tr>
<tr>
<td>1250</td>
<td>5</td>
</tr>
<tr>
<td>1500</td>
<td>10</td>
</tr>
<tr>
<td>1750</td>
<td>15</td>
</tr>
<tr>
<td>2000</td>
<td>20</td>
</tr>
</tbody>
</table>

ECE ILLINOIS
Reducing the End-to-End Training Time with iSwitch

![Graph showing the comparison between AllReduce (AR) and Parameter Server (PS) in terms of average episode reward versus training time (minutes) of DQN. The graph illustrates the reduction in training time achieved with iSwitch.]
Reducing the End-to-End Training Time with iSwitch
Reducing the End-to-End Training Time with iSwitch

![Graph showing comparison of iSwitch (iSW), AllReduce (AR), and Parameter Server (PS) in terms of Average Episode Reward vs. Training Time (min) of DQN.]
Reducing the End-to-End Training Time with iSwitch

![Graph showing the comparison of training time for DQN with iSwitch (iSW), AllReduce (AR), and Parameter Server (PS). The x-axis represents the training time (min) of DQN, and the y-axis represents the average episode reward. The graph shows a significant reduction in training time with iSwitch compared to the other methods.]
Reducing the End-to-End Training Time with iSwitch

![Graph showing the comparison between iSwitch (iSW), AllReduce (AR), and Parameter Server (PS) in terms of Average Episode Reward and Training Time (min) of DQN. The graph highlights a 1.9x speedup for iSwitch and a 3.7x speedup for AllReduce compared to Parameter Server.](image)
Performance Breakdown for Each Training Iteration

- **Agent Action**
- **Environment**
- **Buffer Sampling**
- **Forward Pass**
- **Backward Pass**
- **Memory Alloc**
- **GPU Copy**
- **Weight Update**
- **Others**

Training Time (Norm)

Legend:
- **PS**
- **AR**
- **iSW**
- **DQN**
- **A2C**
- **PPO**
- **DDPG**
Performance Breakdown for Each Training Iteration

- **Agent Action**
- **Environment**
- **Buffer Sampling**
- **Memory Alloc**
- **Forward Pass**
- **Backward Pass**
- **GPU Copy**
- **Gradient Aggregation**
- **Weight Update**
- **Others**

<table>
<thead>
<tr>
<th>Method</th>
<th>PS</th>
<th>AR</th>
<th>iSW</th>
<th>PS</th>
<th>AR</th>
<th>A2C</th>
<th>PS</th>
<th>AR</th>
<th>PPO</th>
<th>PS</th>
<th>AR</th>
<th>DDPG</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A2C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PPO</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDPG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Performance Breakdown for Each Training Iteration

- Agent Action
- Environment
- Buffer Sampling
- Memory Alloc
- Forward Pass
- Backward Pass
- GPU Copy
- Others
- Weight Update
- Others

<table>
<thead>
<tr>
<th>Method</th>
<th>PS</th>
<th>AR</th>
<th>iSW</th>
</tr>
</thead>
<tbody>
<tr>
<td>DQN</td>
<td>1</td>
<td>0.3</td>
<td>0.8</td>
</tr>
<tr>
<td>A2C</td>
<td>0.7</td>
<td>0.9</td>
<td>0.1</td>
</tr>
<tr>
<td>PPO</td>
<td>0.5</td>
<td>0.6</td>
<td>0.9</td>
</tr>
<tr>
<td>DDPG</td>
<td>0.4</td>
<td>0.5</td>
<td>0.8</td>
</tr>
</tbody>
</table>
Performance Breakdown for Each Training Iteration

<table>
<thead>
<tr>
<th>Agent Action</th>
<th>Environment</th>
<th>Buffer Sampling</th>
<th>Memory Alloc</th>
<th>Forward Pass</th>
<th>Backward Pass</th>
<th>GPU Copy</th>
<th>Grad Aggregation</th>
<th>Others</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS</td>
<td>AR</td>
<td>iSW</td>
<td>PS</td>
<td>AR</td>
<td>iSW</td>
<td>PS</td>
<td>AR</td>
<td>iSW</td>
</tr>
</tbody>
</table>

- **DQN**
- **A2C**
- **PPO**
- **DDPG**
Improved Training Scalability with In-Switch Computing

Synchronous Training of PPO

<table>
<thead>
<tr>
<th>Number of Worker Nodes</th>
<th>PS</th>
<th>AR</th>
<th>iSW</th>
<th>Ideal</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>1</td>
<td>1.1</td>
<td>1.2</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1.15</td>
<td>1.25</td>
<td>1.3</td>
<td>1.25</td>
</tr>
<tr>
<td>9</td>
<td>1.5</td>
<td>1.55</td>
<td>1.6</td>
<td>1.55</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>2.1</td>
<td>2.2</td>
<td>2</td>
</tr>
</tbody>
</table>

Graph shows speedup as a function of the number of worker nodes.
Improved Training Scalability with In-Switch Computing

Synchronous Training of PPO

Asynchronous Training of PPO
Improved Training Scalability with In-Switch Computing

**Synchronous Training of PPO**

- PS
- AR
- iSW
- Ideal

**Asynchronous Training of PPO**

- PS
- iSW
- Ideal

Close-to Linear Speedup for Both Training Modes
In-Switch Computing Summary

Programmable Switch

Aggregation Accelerator

3.7x Speedup for Both Sync/Async Training

Scales at Rack-Scale Clusters
Thanks!

Youjie Li  
li238@Illinois.edu

Iou-Jen Liu  Yifan Yuan

Deming Chen  Alexander Schwing  Jian Huang

University of Illinois at Urbana-Champaign  
Electrical & Computer Engineering