Missing the Forest for the Trees: End-to-End AI Application Performance in Edge Data Centers

International Symposium on High Performance Computer Architecture
25 February 2020

Daniel Richins¹,², Dharmisha Doshi², Matthew Blackmore², Aswathy Thulaseedharan Nair², Neha Pathapati², Ankit Patel², Brainard Daguman², Daniel Dobrijalowski², Ramesh Illikkal², Kevin Long², David Zimmerman², Vijay Janapa Reddi¹,³

¹The University of Texas at Austin
²Intel
³Harvard University
Missing the Forest for the Trees
The AI Tax
Missing the Forest for the Trees
The AI Tax
Missing the Forest for the Trees
The AI Tax

The forest is the AI tax
Missing the Forest for the Trees
The AI Tax

The AI tax includes all the compute and infrastructure in an AI application that is necessary to *enable* the AI to execute but that isn’t AI itself.
The AI tax includes all the compute and infrastructure in an AI application that is necessary to *enable* the AI to execute but that isn’t AI itself.
Missing the Forest for the Trees
The AI Tax

The AI tax includes all the compute and infrastructure in an AI application that is necessary to *enable* the AI to execute but that isn’t AI itself.
1. **AI Tax**
   a. Definition
   b. Video Analytics
   c. Analysis

2. **AI Acceleration**
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. **Optimization**
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. Two Designs

4. **Conclusion**
1. AI Tax - A Case Study
   a. Definition
   b. Video Analytics
   c. Analysis

2. AI Acceleration
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. Optimization
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. Two Designs

4. Conclusion

Outline
1. **AI Tax - A Case Study**
   a. Definition
   b. Video Analytics
   c. Analysis

2. **AI Acceleration - Anticipating Future Bottlenecks**
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. **Optimization**
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. Two Designs

4. **Conclusion**

---

Outline
1. AI Tax - A Case Study
   a. Definition
   b. Video Analytics
   c. Analysis

2. AI Acceleration - Anticipating Future Bottlenecks
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. Optimization - Better Performance at Lower TCO
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. Two Designs

4. Conclusion
1. **AI Tax**
   a. Definition
   b. Video Analytics
   c. Analysis

2. **AI Acceleration**
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. **Optimization**
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. Two Designs

4. **Conclusion**

---

**AI Tax**
1. **AI Tax**
   a. **Definition**
   b. **Video Analytics**
   c. **Analysis**

2. **AI Acceleration**
   a. **Emulation Technique**
   b. **Results**
   c. **What's Breaking?**

3. **Optimization**
   a. **Fixing the Bottleneck**
   b. **Edge Data Centers**
   c. **Two Designs**

4. **Conclusion**

---

**AI Tax - Definition**
Missing the Forest for the Trees
Missing the Forest for the Trees
Missing the Forest for the Trees

AI Tax
Missing the Forest for the Trees

Supporting compute, storage, network, software infrastructure, etc. together constitute the AI Tax.
1. **AI Tax**
   a. Definition
   b. Video Analytics
   c. Analysis

2. **AI Acceleration**
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. **Optimization**
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. Two Designs

4. **Conclusion**

---

**AI Tax - Video Analytics**
Face Recognition
Algorithm
Face Recognition Algorithm

*Face Recognition* is Google’s FaceNet as a data center application.
Face Recognition Algorithm

*Face Recognition* is Google’s FaceNet as a data center application.
Face Recognition Algorithm

*Face Recognition* is Google’s FaceNet as a data center application.
Face Recognition Algorithm

*Face Recognition* is Google’s FaceNet as a data center application.
Face Recognition Algorithm

*Face Recognition* is Google’s FaceNet as a data center application.
Face Recognition Algorithm

*Face Recognition* is Google’s FaceNet as a data center application.
Face Recognition Algorithm

*Face Recognition* is Google’s FaceNet as a data center application.
Face Recognition Algorithm

*Face Recognition* is Google’s FaceNet as a data center application.
Face Recognition Algorithm

*Face Recognition* is Google’s FaceNet as a data center application.
Face Recognition Algorithm

*Face Recognition* is Google’s FaceNet as a data center application.
Face Recognition Algorithm

*Face Recognition* is Google’s FaceNet as a data center application.
Face Recognition Algorithm

*Face Recognition* is Google’s FaceNet as a data center application.

Diagram:
- Video Stream
- Ingestion
- Face Detection
- Feature Extraction
- Vector
- Classification
- Identity
- Face Thumbnail
- Frame

**Face Recognition Algorithm**

*Face Recognition* is Google’s FaceNet as a data center application.

![Diagram of Face Recognition Algorithm]

- **User**
  - Video Stream
  - Identity

- **Application**
  - Ingestion
  - Classification
  - Feature Extraction
  - Face Detection
  - Face Thumbnail

- Flow: Video Stream → Ingestion → Classification → Feature Extraction → Face Detection → Face Thumbnail → Video Stream
Face Recognition is Google’s FaceNet as a data center application.

**Face Recognition Algorithm**

**User**
- Video Stream
- Identity

**Application**
- Ingestion
- Frame
- Face Detection
- Face Thumbnail
- Vector
- Feature Extraction
- Classification

**AI Compute**
Face Recognition
Data Center Deployment

User

- Video Stream

Application

- Ingestion
- Face Detection
- Classification
- Feature Extraction

AI Compute
Face Recognition
Data Center Deployment

User

Application

Ingestion → Face Detection

Classification

Feature Extraction

AI Compute
Face Recognition
Data Center Deployment

User

Application

Ingest/Detect

Identification

AI Compute
Face Recognition
Data Center Deployment

User

Application

Producers

Ingest/Detect

Identification

Consumers

AI Compute
Face Recognition
Data Center Deployment

User

Application

Producers

Ingest/Detect

Identification

Consumers

AI Compute
Face Recognition
Data Center Deployment

User

Application

Producers

Ingest/Detect

Consumers

Identification

Brokers

AI Compute
Experimental Setup
Hardware
Experimental Setup
Hardware

2x Intel Xeon Platinum 8176
2x 28 cores, 2.10 GHz, 2x 38.5 MB LLC

384 GB DDR4 SDRAM

1x Intel SSD P4510
2.85 GB/s read
1.10 GB/s write

100 Gbps Ethernet
Experimental Setup

Hardware

-
Experimental Setup

Face Recognition

We allocate one core per container.
Hence, a server runs 56 containers.
Experimental Setup
Face Recognition

We allocate one core per container. Hence, a server runs 56 containers.
Experimental Setup
Face Recognition

We allocate one core per container. Hence, a server runs 56 containers.

840 total producers
1680 total consumers
Experimental Setup
Face Recognition

We allocate one core per container. Hence, a server runs 56 containers.

Brokers get their own server. This grants them full network and storage bandwidth.

Ingest/Detect

Identification

840 total producers
1680 total consumers
Experimental Setup
Face Recognition

We allocate one core per container. Hence, a server runs 56 containers.

Brokers get their own server. This grants them full network and storage bandwidth.

840 total producers
1680 total consumers
Experimental Setup
Face Recognition

We allocate one core per container. Hence, a server runs 56 containers.

Brokers get their own server. This grants them full network and storage bandwidth.

Ingest/Detect
840 total producers

Identification
1680 total consumers

Broker
3 brokers

3 brokers
1. **AI Tax**
   a. Definition
   b. Video Analytics
   c. Analysis

2. **AI Acceleration**
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. **Optimization**
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. Two Designs

4. **Conclusion**

---

**AI Tax - Analysis**
while True:
    frame = queue.get()
    faces = detect_faces(frame)
    producer.send(faces)
while True:
    frame = queue.get()
    start = time.time()
    faces = detect_faces(frame)
    end = time.time()
    producer.send(faces)
while True:
    frame = queue.get()
    start = time.time()
    faces = detect_faces(frame)
    end = time.time()
    producer.send(faces)
    size = sys.getsizeof(faces)
    log = {
        'start': start,
        'end': end,
        'size': size
    }
    logger.info(log)
while True:
    frame = queue.get()
    start = time.time()
    faces = detect_faces(frame)
    end = time.time()
    size = sys.getsizeof(faces)
    log = {
        'start': start,
        'end': end,
        'size': size
    }
    logger.info(log)
    producer.send(faces)

Logging is designed to raise the level of abstraction. We view application progress from the data center perspective.
Face Detection Latency
Face Detection Latency

Latency Breakdown

- Ingestion
- Detection
- Brokers
- Identification
Face Detection Latency

Latency Breakdown

- Ingestion
- Detection
- Brokers
- Identification

AI Tax: 5.4%
Face Detection Latency

Latency Breakdown

- Ingestion: 5.4%
- Brokers: 21.3%
- Detection: 21.3%
- Identification: 5.4%
- AI Tax: 5.4%
- AI Compute: 21.3%

Ingestion
Brokers
Detection
Identification
Face Detection Latency

Latency Breakdown

- Ingestion: 35.9%
- Detection: 21.3%
- Brokers: 5.4%
- AI Compute: 15%
- AI Tax: 25.4%

Ingestion: 35.9%
Detection: 21.3%
Brokers: 5.4%
AI Compute: 15%
AI Tax: 25.4%
Face Detection Latency

Latency Breakdown

- Ingestion: 5.4%
- Detection: 21.3%
- Brokers: 35.9%
- Identification: 37.4%

AI Tax

- AI Compute: 37.4%
- AI Tax: 21.3%
- AI Compute: 35.9%
Process Breakdowns
Process Breakdowns

Ingestion

100%

AI

AI Tax
Process Breakdowns

**Ingestion**

- 100% 
  - **AI**
  - **AI Tax**

**Face Detection**

- 58% 
  - **AI**
  - **AI Tax**

- 42% 
  - **AI**
  - **AI Tax**
Pre- and post-processing are heavily utilized within stages.
Al Tax
AI Tax
AI Tax

Pre AI Post

Time

Excitement
AI Tax

Time

Excitement

Pre Al Al Al Post
1. **AI Tax**
   a. Definition
   b. Video Analytics
   c. Analysis

2. **AI Acceleration**
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. **Optimization**
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. Two Designs

4. **Conclusion**

---

AI Acceleration
1. **AI Tax**
   a. Definition
   b. Video Analytics
   c. Analysis

2. **AI Acceleration**
   a. Emulation Technique
   b. Results
   c. What’s Breaking?

3. **Optimization**
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. Two Designs

4. **Conclusion**

---

AI Acceleration - Emulation Technique
Acceleration Emulation

Ingest/Detect

Identification

Brokers
Acceleration Emulation

Dial an Accelerator Speed

Brokers
while True:
    frame = queue.get()
    start = time.time()
    faces = detect_faces(frame)
    end = time.time()
    producer.send(faces)
    size = sys.getsizeof(faces)
    log = {
        'start': start,
        'end': end,
        'size': size
    }
    logger.info(log)
while True:
    frame = queue.get()
    start = time.time()
    faces = detect_faces(frame)
    end = time.time()
    producer.send(faces)
    size = sys.getsizeof(faces)
    log = {
        'start': start,
        'end': end,
        'size': size
    }
    logger.info(log)
while True:
    frame = queue.get()
    start = time.time()
    time.sleep(avg_time)
    end = time.time()
    producer.send(faces)
    size = sys.getsizeof(faces)
    log = {
        'start': start,
        'end': end,
        'size': size
    }
    logger.info(log)
while True:
    start = time.time()
    time.sleep(avg_time)
    end = time.time()
    frame = queue.get()
    producer.send(frame)
    size = sys.getsizeof(frame)
    log = {
        'start': start,
        'end': end,
        'size': size
    }
    logger.info(log)
while True:
    frame = queue.get()
    start = time.time()
    time.sleep(avg_time)
    end = time.time()
    producer.send(os.urandom(avg_size))
    size = avg_size
    log = {
        'start': start,
        'end': end,
        'size': size
    }
    logger.info(log)
Acceleration Emulation

```python
while True:
    frame = queue.get()
    start = time.time()
    time.sleep(avg_time)
    end = time.time()
    producer.send(os.urandom(avg_size))
    size = avg_size
    log = {
        'start': start,
        'end': end,
        'size': size
    }
    logger.info(log)
```
while True:
    frame = queue.get()
    start = time.time()
    time.sleep(avg_time)
    end = time.time()
    producer.send(os.urandom(avg_size))
    size = avg_size
    log = {
        'start': start,
        'end': end,
        'size': size
    }
    logger.info(log)
while True:
    frame = queue.get()
    start = time.time()
    time.sleep(avg_time)
    end = time.time()
    producer.send(os.urandom(avg_size))
    size = avg_size
    log = {
        'start': start,
        'end': end,
        'size': size
    }
    logger.info(log)
while True:
    frame = queue.get()
    start = time.time()
    time.sleep(avg_time/speedup)
    end = time.time()
    producer.send(os.urandom(avg_size))
    size = avg_size
    log = {
        'start': start,
        'end': end,
        'size': size
    }
    logger.info(log)
Acceleration Emulation

```python
while True:
    frame = queue.get()
    start = time.time()
    time.sleep(avg_time/speedup)
    end = time.time()
    producer.send(os.urandom(avg_size))
    size = avg_size
    log = {
        'start': start,
        'end': end,
        'size': size
    }
    logger.info(log)
```

With faster processing, we feed frames into the system faster to maximize throughput
1. **AI Tax**
   a. Definition
   b. Video Analytics
   c. Analysis

2. **AI Acceleration**
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. **Optimization**
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. Two Designs

4. **Conclusion**

---

**AI Acceleration - Results**
Accelerated AI: Reduced Latency and Increased Throughput
Accelerated AI: Reduced Latency and Increased Throughput

<table>
<thead>
<tr>
<th>Latency (ms)</th>
<th>Ingest/Detect</th>
<th>Broker</th>
<th>Identify</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>700</td>
<td>1x</td>
<td>4x</td>
<td>8x</td>
<td>70</td>
</tr>
<tr>
<td>600</td>
<td>2x</td>
<td>6x</td>
<td></td>
<td>60</td>
</tr>
<tr>
<td>500</td>
<td>3x</td>
<td>8x</td>
<td></td>
<td>50</td>
</tr>
<tr>
<td>400</td>
<td>4x</td>
<td></td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>300</td>
<td>5x</td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>200</td>
<td>6x</td>
<td></td>
<td></td>
<td>20</td>
</tr>
<tr>
<td>100</td>
<td>7x</td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>0</td>
<td>8x</td>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

Frames per Second (x1000)
Accelerated AI: Reduced Latency and Increased Throughput

Latency (ms)

<table>
<thead>
<tr>
<th>Ingest/Detect</th>
<th>Broker</th>
<th>Identify</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>100</td>
<td>200</td>
<td>300</td>
</tr>
<tr>
<td>200</td>
<td>400</td>
<td>500</td>
<td>600</td>
</tr>
<tr>
<td>300</td>
<td>600</td>
<td>700</td>
<td>800</td>
</tr>
</tbody>
</table>

Frames per Second (x1000)

- 1x: 0
- 2x: 10
- 4x: 20
- 6x: 30
- 8x: 40
Accelerated AI: Reduced Latency and Increased Throughput

![Graph showing latency and throughput comparisons]
Accelerated AI: Reduced Latency and Increased Throughput

[Bar chart showing latency and throughput for different acceleration levels (1x, 2x, 4x, 6x, 8x).]
Accelerated AI: Reduced Latency and Increased Throughput

At 8x speedup, the average latency goes to infinity. The longer the experiment runs, the greater the latency.
1. **AI Tax**
   a. Definition
   b. Video Analytics
   c. Analysis

2. **AI Acceleration**
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. **Optimization**
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. Two Designs

4. **Conclusion**

**AI Acceleration - What’s Breaking?**
Three Big Systems
Three Big Systems

Compute
Three Big Systems

Compute

Network
Three Big Systems

Compute

Network

Storage
Three Big Systems

- **Compute**
- **Network**
- **Storage**

[Diagram showing a crossed-out compute icon, a network icon, and a storage icon]
Three Big Systems

- **Compute**
- **Network**
- **Storage**
Three Big Systems

Compute

Network

Storage
Explaining the Bottleneck
Explaining the Bottleneck

Network Utilization

7% 6% 5% 4% 3% 2% 1% 0%

1x 2x 4x 6x 8x

Broker Read  Broker Write
Explaining the Bottleneck

Network Utilization

0% 1% 2% 3% 4% 5% 6% 7%
1x 2x 4x 6x 8x

Broker Read
Broker Write
Explaining the Bottleneck

Network Utilization

- 0%
- 1%
- 2%
- 3%
- 4%
- 5%
- 6%
- 7%

1x 2x 4x 6x 8x

- Broker Read
- Broker Write
Explaining the Bottleneck

Network Utilization

- 7% 6% 5% 4% 3% 2% 1% 0%
- 1x 2x 4x 6x 8x

Storage Utilization

- 70% 60% 50% 40% 30% 20% 10% 0%
- 1x 2x 4x 6x 8x

Broker Read

Broker Write
Explaining the Bottleneck

Network Utilization

Storage Utilization

Broker Read

Broker Write
Explaining the Bottleneck

Network Utilization

- Broker Read

Storage Utilization

- Broker Write
Explaining the Bottleneck

As storage utilization approaches the limits of the devices, it becomes the limiting factor to performance.
1. **AI Tax**
   a. Definition
   b. Video Analytics
   c. Analysis

2. **AI Acceleration**
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. **Optimization**
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. Two Designs

4. **Conclusion**

---

**Optimization**
1. AI Tax
   a. Definition
   b. Video Analytics
   c. Analysis

2. AI Acceleration
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. Optimization
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. Two Designs

4. Conclusion

Optimization - Fixing the Bottleneck
Fixing the Bottleneck
Fixing the Bottleneck

Additional Drives

Latency (ms)

1 Drive 2 Drives 3 Drives 4 Drives

8x 12x 16x 24x 32x
Fixing the Bottleneck

### Additional Drives

- 1 Drive
- 2 Drives
- 3 Drives
- 4 Drives

Latency (ms):
- 0
- 50
- 100
- 150
- 200

### Additional Brokers

- 3 Brokers
- 4 Brokers
- 6 Brokers
- 8 Brokers

Latency (ms):
- 0
- 50
- 100
- 150
- 200

Legend:
- 8x
- 12x
- 16x
- 24x
- 32x
Fixing the Bottleneck

Additional Drives

Latency (ms)

1 Drive  2 Drives  3 Drives  4 Drives

Additional Brokers

Latency (ms)

3 Brokers  4 Brokers  6 Brokers  8 Brokers

8x  12x  16x  24x  32x
Fixing the Bottleneck

### Additional Drives

<table>
<thead>
<tr>
<th>Drives</th>
<th>Latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8x</td>
</tr>
<tr>
<td>2</td>
<td>12x</td>
</tr>
<tr>
<td>3</td>
<td>16x</td>
</tr>
<tr>
<td>4</td>
<td>24x</td>
</tr>
</tbody>
</table>

### Additional Brokers

<table>
<thead>
<tr>
<th>Brokers</th>
<th>Latency (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>8x</td>
</tr>
<tr>
<td>4</td>
<td>12x</td>
</tr>
<tr>
<td>6</td>
<td>16x</td>
</tr>
<tr>
<td>8</td>
<td>24x</td>
</tr>
</tbody>
</table>

Colors:
- Blue: 8x
- Orange: 12x
- Green: 16x
- Red: 24x
- Purple: 32x
1. **AI Tax**
   a. Definition
   b. Video Analytics
   c. Analysis

2. **AI Acceleration**
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. **Optimization**
   a. Fixing the Bottleneck
   b. **Edge Data Centers**
   c. Two Designs

4. **Conclusion**

---

**Optimization - Edge Data Centers**
Advantages of an Edge Data Center
Advantages of an Edge Data Center

Smaller corporations are finding edge data centers more economical than the cloud.
Advantages of an Edge Data Center

Smaller corporations are finding edge data centers more economical than the cloud.

Edge data centers offer lower latency by serving local users.
Advantages of an Edge Data Center

Smaller corporations are finding edge data centers more economical than the cloud.

Edge data centers offer lower latency by serving local users.

Edge data centers can be built to target a specific application domain.

Sources
https://www.networkworld.com/article/2926448/7-key-criteria-for-defining-edge-data-centers.html
https://www.vxchnge.com/blog/what-is-an-edge-data-center
1. **AI Tax**
   a. Definition
   b. Video Analytics
   c. Analysis

2. **AI Acceleration**
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. **Optimization**
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. **Two Designs**

4. **Conclusion**

---

Optimization - Two Designs
Edge Data Center
Node Allocation
Edge Data Center
Node Allocation

We need to allocate enough brokers to handle 32x speedup.
Edge Data Center
Node Allocation

We need to allocate enough brokers to handle 32x speedup.
We need to allocate enough brokers to handle 32x speedup.
Targeted Data Center Design
Optimizing for Total Cost of Ownership
Targeted Data Center Design
Optimizing for Total Cost of Ownership

| Homogeneous | Heterogeneous |
### Targeted Data Center Design
Optimizing for Total Cost of Ownership

<table>
<thead>
<tr>
<th>Homogeneous</th>
<th>Heterogeneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Homogeneous Icon]</td>
<td>![Heterogeneous Icon]</td>
</tr>
<tr>
<td>56 core</td>
<td></td>
</tr>
<tr>
<td>1 Drive</td>
<td></td>
</tr>
<tr>
<td>100 GbE</td>
<td></td>
</tr>
</tbody>
</table>
Targeted Data Center Design
Optimizing for Total Cost of Ownership

Homogeneous

- 56 core
- 1 Drive
- 100 GbE

Heterogeneous

Compute

Broker
Targeted Data Center Design
Optimizing for Total Cost of Ownership

Homogeneous

56 core
1 Drive
100 GbE

160 switches

Heterogeneous

Compute
Broker

160 switches
Targeted Data Center Design
Optimizing for Total Cost of Ownership

Homogeneous

- 56 core
- 1 Drive
- 100 GbE

160 switches

Heterogeneous

Compute

Broker

28+14 switches

160 switches
Targeted Data Center Design
Optimizing for Total Cost of Ownership

Homogeneous

- 56 core
- 1 Drive
- 100 GbE

160 switches

Heterogeneous

- Compute
- Broker

28+14 switches

TCO
Heterogeneous Edge Data Center
Heterogeneous Edge Data Center

Compute Node
# Heterogeneous Edge Data Center

<table>
<thead>
<tr>
<th>Compute Node</th>
<th>Broker Node</th>
</tr>
</thead>
</table>


Heterogeneous Edge Data Center

Compute Node

Broker Node
Heterogeneous Edge Data Center

Compute Node

Broker Node
Heterogeneous Edge Data Center

Compute Node

Broker Node
Heterogeneous Edge Data Center

Compute Node

Broker Node

10 GbE
Heterogeneous Edge Data Center

**Compute Node**

- CPU
- Hard Drive
- 10 GbE

**Broker Node**
Heterogeneous Edge Data Center

Compute Node

Broker Node

10 GbE
Heterogeneous Edge Data Center

Compute Node

Broker Node

10 GbE
Heterogeneous Edge Data Center

Compute Node

Broker Node

10 GbE

50 GbE
Heterogeneous Edge Data Center Networking
Heterogeneous Edge Data Center Networking
Heterogeneous Edge Data Center Networking
Heterogeneous Edge Data Center Networking

- 100 Gbps Switch
- 100 Gbps Switch
- 100 Gbps
- 100 Gbps
- 50 Gbps

Broker Node
Heterogeneous Edge Data Center Networking

Broker Node

100 Gbps Switch

100 Gbps

100 Gbps

100 Gbps

100 Gbps

50 Gbps

40 Gbps

40 Gbps

100 Gbps
Heterogeneous Edge Data Center Networking

- **Broker Node**
- **Compute Node**

Diagram with bandwidths:
- 100 Gbps Switch
- 100 Gbps
- 100 Gbps
- 100 Gbps
- 100 Gbps
- 50 Gbps
- 40 Gbps
- 40 Gbps
- 10 Gbps
Heterogeneous Edge Data Center Networking

- **Broker Node**
- **Compute Node**

100 Gbps Switch

100 Gbps

100 Gbps

100 Gbps

50 Gbps

40 Gbps

40 Gbps

40 Gbps

40 Gbps

10 Gbps
Comparing Total Cost of Ownership

We assume a three-year amortization of costs.
Comparing Total Cost of Ownership

We assume a three-year amortization of costs.

<table>
<thead>
<tr>
<th></th>
<th>Compute</th>
<th>Networking</th>
<th>Power</th>
<th>Overall</th>
</tr>
</thead>
<tbody>
<tr>
<td>100%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Homogeneous**
- **Heterogeneous**
Comparing Total Cost of Ownership

We assume a three-year amortization of costs.

<table>
<thead>
<tr>
<th>Component</th>
<th>Homogeneous</th>
<th>Heterogeneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute</td>
<td>100%</td>
<td>89%</td>
</tr>
<tr>
<td>Networking</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td></td>
<td>100%</td>
</tr>
</tbody>
</table>

- **Homogeneous**
- **Heterogeneous**
Comparing Total Cost of Ownership

We assume a three-year amortization of costs.

<table>
<thead>
<tr>
<th></th>
<th>Homogeneous</th>
<th>Heterogeneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute</td>
<td>100%</td>
<td>89%</td>
</tr>
<tr>
<td>Networking</td>
<td>100%</td>
<td>23%</td>
</tr>
<tr>
<td>Power</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Overall</td>
<td>100%</td>
<td>100%</td>
</tr>
</tbody>
</table>
Comparing Total Cost of Ownership

We assume a three-year amortization of costs.
Comparing Total Cost of Ownership

We assume a three-year amortization of costs.
Comparing Total Cost of Ownership

We assume a three-year amortization of costs.

<table>
<thead>
<tr>
<th>Component</th>
<th>Homogeneous</th>
<th>Heterogeneous</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compute</td>
<td>100%</td>
<td>89%</td>
</tr>
<tr>
<td>Networking</td>
<td>100%</td>
<td>23%</td>
</tr>
<tr>
<td>Power</td>
<td>100%</td>
<td>100%</td>
</tr>
<tr>
<td>Overall</td>
<td>100%</td>
<td>84%</td>
</tr>
</tbody>
</table>

- Homogeneous: $12.9 million
- Heterogeneous: $10.8 million
Comparing Total Cost of Ownership

We assume a three-year amortization of costs.

The targeted, heterogeneous data center incurs 16% lower total cost of ownership.
Comparing Total Cost of Ownership

We assume a three-year amortization of costs.

The targeted, heterogeneous data center incurs 16% lower total cost of ownership.

Designing the data center to match the needs of the application, we created a better data center at lower cost.

- Compute: Homogeneous $12.9 million, Heterogeneous $10.8 million
- Networking: Homogeneous 84%, Heterogeneous 100%
- Power: Homogeneous 23%, Heterogeneous 100%
- Overall: Homogeneous 100%, Heterogeneous 100%
1. **AI Tax**
   a. Definition
   b. Video Analytics
   c. Analysis

2. **AI Acceleration**
   a. Emulation Technique
   b. Results
   c. What's Breaking?

3. **Optimization**
   a. Fixing the Bottleneck
   b. Edge Data Centers
   c. Two Designs

4. **Conclusion**
Calls to Action
Calls to Action

• To fully understand AI applications, **we must consider the overhead of the AI tax** in end-to-end performance.
Calls to Action

• To fully understand AI applications, **we must consider the overhead of the AI tax** in end-to-end performance.

• As we accelerate AI, **we must consider new bottlenecks** that manifest as AI tax.
Calls to Action

• To fully understand AI applications, **we must consider the overhead of the AI tax** in end-to-end performance.

• As we accelerate AI, **we must consider new bottlenecks** that manifest as AI tax.

• We cannot limit our view of AI to microarchitectural considerations. **We need data center-level optimizations to address data center-level bottlenecks.**
Thank You