E3: Energy-Efficient Microservices on SmartNIC-Accelerated Servers

Ming Liu, Simon Peter, Arvind Krishnamurthy, Phitchaya Mangpo Phothilimthana
Trend #1: Energy-efficiency has become a major factor for today’s DC

- US data centers consume 70 billion kilowatt-hours of energy per year
- Server CPUs consume the most energy

Trend #2: recent adoption of SoC SmartNICs in servers

- SoC SmartNICs are a new kind of heterogenous computing platform in the data center
  ✓ Present on the packet data path
  ✓ Process networking requests in short latency
  ✓ **Consume low power**
Trend #2: recent adoption of SoC SmartNICs in servers

- SoC SmartNICs are a new kind of heterogeneous computing platform in the data center
  - Present on the packet data path
  - Process networking requests in short latency
  - Consume low power

- LiquidIOII SmartNICs
  - OCTEON 12-core cnMIPS64 processor @1.2GHz
  - Domain-specific accelerators
    - Crypto/Pattern matching/Fetch-add engines
  - Wimpy memory hierarchy
    - 32KB/4MB/4GB L1/L2/DRAM
  - 2x 10Gbps ports
Trend #3: the rise of cloud microservices
Trend #3: the rise of cloud microservices

❖ Microservices
✓ Fine-grained -> small memory footprint
✓ Communication intensive -> invoked via RPCs
✓ Dataflow programming model -> explicit communication patterns

❖ Run by a cluster scheduler
✓ Examples: Azure Service Fabric, Google Application Engine, Nirmata
✓ Easy to explore architectural heterogeneity
Trend #3: the rise of cloud microservices

- We evaluate 8 microservice-based applications of 3 common types
  - Network function virtualization (NFV)
  - Real-time data analytics (RTA)
  - IoT hub (IoT)
- Each application comprises 60 ~ 108 microservices
Example: IoT thermostat analytics application

Stage 1: authentication
Stage 2: sensor logging
Stage 3: data analytics
Example: IoT thermostat analytics application

Stage 1: authentication  Stage 2: sensor logging  Stage 3: data analytics

Requests

Microservice

RPC request flow
Example: IoT thermostat analytics application

![Diagram](image)

Stage 1: authentication
Stage 2: sensor logging
Stage 3: data analytics

Microservice
RPC request flow

Requests
Example: IoT thermostat analytics application

Stage 1: authentication
Stage 2: sensor logging
Stage 3: data analytics

Microservice
RPC request flow

Requests

API Gateway → SQL store → Spike → EMA
API Gateway → SQL store → Recommend
API Gateway → SQL store → Spike → EMA

Stage 2: sensor logging

RPC request flow
Example: IoT thermostat analytics application

Stage 1: authentication
Stage 2: sensor logging
Stage 3: data analytics
E3 idea: run Microservices on SmartNIC-servers
E3 idea: run Microservices on SmartNIC-servers

- E3 goals:
  - Better energy-efficiency
  - Minimal latency cost

ToR switch

- Ethernet
  - Spike
  - EMA
  - Recommend
  - API Gateway
  - SQL store

SmartNIC-server

Intel XEON inside

PCIe
Two types of SmartNIC-servers

**Single-SmartNIC server cluster**

- 10GbE Ethernet
- PCIe

**Multi-SmartNIC server cluster**

- 4x10GbE Ethernet (breakout cable)
- PCIe
- QPI
Two types of SmartNIC-servers

**Single-SmartNIC server cluster**

- 10GbE Ethernet
- PCIe
- 1x 12-core E5-2680 v3 @2.5GHz
- 64GB DRAM
- 1x LiquidIOII

**Multi-SmartNIC server cluster**

- 4x 10GbE Ethernet (breakout cable)
- PCIe
- QPI
- 2x 8-core E5-2620 v4 @2.1GHz
- 128GB DRAM
- 4x LiquidIOII
Key question: Do SmartNIC-servers provide better energy efficiency?

VS.

Homogeneous/Heterogenous cluster

SmartNIC-server cluster
Key question: Do SmartNIC-servers provide better energy efficiency?

SmartNIC-server cluster vs. Homogeneous beefy cluster
Key question: Do SmartNIC-servers provide better energy efficiency?

- SmartNIC-server cluster
  - Ethernet
  - PCIe
  - QPI

- Homogeneous beefy cluster
  - Supermicro 1U server
    ✔ Intel 12-core E5-2680 v3 processor @2.5GHz
    ✔ 64GB DRAM
    ✔ 10Gbps Intel X710

Intel beefy server

Intel beefy server
Key question: Do SmartNIC-servers provide better energy efficiency?
Key question: Do SmartNIC-servers provide better energy efficiency?

**SmartNIC-server cluster**

- Ethernet
- PCIe
- QPI

**Homogeneous wimpy cluster**

- 1U Cavium CN6880 SoC
  - OCTEON 32-core cnMIPS64 processor @1.2GHz
  - 4GB DRAM
  - 2x 10Gbps XAUI ports
Key question: Do SmartNIC-servers provide better energy efficiency?
Key question: Do SmartNIC-servers provide better energy efficiency?

- Power measurement at each server
  - Onboard IPMI utility + WattsUp Pro meter
  - Report cluster power = aggregate server power

SmartNIC-server cluster

Heterogeneous cluster
Outline

✓ Three challenges of integrating SmartNICs
✓ E3 design
✓ Energy efficiency, cost & latency evaluation
✓ Conclusion
Three challenges of integrating SmartNICs with microservices
Three challenges of integrating SmartNICs with microservices

#1: Addressing and load balancing
Three challenges of integrating SmartNICs with microservices

1. Addressing and load balancing
2. SmartNIC overload
Three challenges of integrating SmartNICs with microservices

#1: Addressing and load balancing
#2: SmartNIC overload
#3: non-uniform communication costs
Outline

✓ Three Challenges of integrating SmartNICs
✓ E3 design
✓ Energy efficiency, cost & latency evaluation
✓ Conclusion
E3: a microservice execution platform

- Follows design philosophies of Azure Service Fabric [Eurosys’18]
- Adds three techniques to support SmartNICs
  - ECMP-based load balancing
  - Load-aware cluster manager
  - Communication-aware microservice placement
E3 technique #1: ECMP-based load balancing

- An intra-server addressing and load-balancing mechanism
E3 technique #1: ECMP-based load balancing

- An intra-server addressing and load-balancing mechanism
E3 technique #1: ECMP-based load balancing

- An intra-server addressing and load-balancing mechanism
E3 technique #1: ECMP-based load balancing

- An intra-server addressing and load-balancing mechanism
E3 technique #1: ECMP-based load balancing

- An intra-server addressing and load-balancing mechanism

Ingress traffic

ECMP @ ToR

Ethernet

PCIe

QPI

10.0.5
E3 technique #1: ECMP-based load balancing

- An intra-server addressing and load-balancing mechanism
E3 technique #1: ECMP-based load balancing

❖ An intra-server addressing and load-balancing mechanism
E3 technique #1: ECMP-based load balancing

- An intra-server addressing and load-balancing mechanism

- 2.5x higher throughput and 2.2x better energy-efficiency
E3 technique #2: load-aware cluster manager

❖ Purpose: avoid host starvation
  - Microservice interference with NIC firmware on SmartNIC memory/cache
❖ Solution:
  - Monitor ingress packet queue depth of SmartNIC, microservice CPU intensity
  - If above threshold, migrate CPU-intensive microservice
E3 technique #2: load-aware cluster manager

❖ Purpose: avoid host starvation
  - Microservice interference with NIC firmware on SmartNIC memory/cache
❖ Solution:
  - Monitor ingress packet queue depth of SmartNIC, microservice CPU intensity
  - If above threshold, migrate CPU-intensive microservice

2 fields added to SF periodic heartbeats:
✓ NIC queue depth
✓ CPU intensity
E3 technique #2: load-aware cluster manager

- Purpose: avoid host starvation
  - Microservice interference with NIC firmware on SmartNIC memory/cache
- Solution:
  - Monitor ingress packet queue depth of SmartNIC, microservice CPU intensity
  - If above threshold, migrate CPU-intensive microservice

❖ Our mechanism achieves 5.9x better energy-efficiency and 27.7% latency reduction
E3 technique #3: Communication-aware microservice placement

- Service Fabric cluster scheduler
  ✓ Simulated annealing
  ✓ Constraints
    - Static node information
      - # of CPUs, memory capacity, …
    - Runtime statistics of each computing node/microservice
      - CPU, network, memory utilization, …
  ❌ Ignores communication latency
E3 technique #3: Communication-aware microservice placement

- Service Fabric cluster scheduler
  - ✔ Simulated annealing
  - ✔ Constraints
    - Static node information
      - # of CPUs, memory capacity, …
    - Runtime statistics of each computing node/microservice
      - CPU, network, memory utilization, …
  - ❌ Ignores communication latency

- E3: hierarchical, communication-aware microservice placement (HCM)
  - ✔ Organize computing nodes into levels of communication distance
  - ✔ Place communicating microservices close to each other
  - ✔ Hierarchical -> prunes search space
E3 technique #3: Communication-aware microservice placement (cont’d)

- HCM algorithm input
  - ✓ $G$: microservice DAG
  - ✓ $V_{src}$: source microservice node of the DAG
  - ✓ $T$: server cluster topology graph
- HCM performs a breadth-first traversal of $G$
  - ✓ Map microservices to a cluster computing node in $T$

Subset of Service Fabric
E3 technique #3: Communication-aware microservice placement (cont’d)

- HCM algorithm input
  - $G$: microservice DAG
  - $V_{src}$: source microservice node of the DAG
  - $T$: server cluster topology graph
- HCM performs a breadth-first traversal of $G$
  - Map microservices to a cluster computing node in $T$

- 4 layers in a single rack
  - L1: the same computing node as $V$
E3 technique #3: Communication-aware microservice placement (cont’d)

- HCM algorithm input
  ✓ G: microservice DAG
  ✓ V_src: source microservice node of the DAG
  ✓ T: server cluster topology graph
- HCM performs a breadth-first traversal of G
  ✓ Map microservices to a cluster computing node in T

- 4 layers in a single rack
  - L1: the same computing node as V
  - L2: another computing node on the same server
E3 technique #3: Communication-aware microservice placement (cont’d)

- 4 layers in a single rack
  - L1: the same computing node as V
  - L2: another computing node on the same server
  - L3: a SmartNIC computing node on another server

- HCM algorithm input:
  ✓ $G$: microservice DAG
  ✓ $V_{src}$: source microservice node of the DAG
  ✓ $T$: server cluster topology graph

- HCM performs a breadth-first traversal of $G$
  ✓ Map microservices to a cluster computing node in $T$
HCM algorithm input
✓ $G$: microservice DAG
✓ $V_{src}$: source microservice node of the DAG
✓ $T$: server cluster topology graph
HCM performs a breadth-first traversal of $G$
✓ Map microservices to a cluster computing node in $T$

4 layers in a single rack
- L1: the same computing node as $V$
- L2: another computing node on the same server
- L3: a SmartNIC computing node on another server
- L4: a host computing node on other servers
E3 technique #3: Communication-aware microservice placement (cont’d)

- HCM algorithm input
  - $G$: microservice DAG
  - $V_{src}$: source microservice node of the DAG
  - $T$: server cluster topology graph
- HCM performs a breadth-first traversal of $G$
- Map microservices to a cluster computing node in $T$

- Compared with Service Fabric, HCM improves energy efficiency by 16.2% and reduces the latency by 13.0%

- L2: another computing node on the same server
- L3: a SmartNIC computing node on another servers
- L4: a host computing node on other servers

Subset of Service Fabric
Outline

✓ Three Challenges of integrating SmartNICs
✓ E3 design
✓ Energy efficiency, cost & latency evaluation
✓ Conclusion
Energy efficiency under peak utilization

- 3 Single-SmartNIC servers vs. 3 beefy servers
  - ✓ Deploy each application via E3, maximize client load without overload
  - ✓ Measure cluster throughput & power

<table>
<thead>
<tr>
<th>Application</th>
<th>Energy Efficiency (KRPJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NFV-FIN</td>
<td>15.2X</td>
</tr>
<tr>
<td>NFV-DIN</td>
<td>16.5X</td>
</tr>
<tr>
<td>NFV-IFID</td>
<td>12.3X</td>
</tr>
<tr>
<td>RTA-PTC</td>
<td>11.0X</td>
</tr>
<tr>
<td>RTA-SF</td>
<td>9.7X</td>
</tr>
<tr>
<td>RTA-SHM</td>
<td>8.4X</td>
</tr>
<tr>
<td>IOT-DH</td>
<td>6.1X</td>
</tr>
<tr>
<td>IOT-TS</td>
<td>5.8X</td>
</tr>
</tbody>
</table>

- NFV-FIN: 2.5X improvement
- NFV-DIN: 1.3X improvement
- NFV-IFID: 1.3X improvement
Average/tail latency under peak utilization

- 3 Single-SmartNIC servers vs. 3 beefy servers
- Up to 4% latency cost
Cluster cost efficiency over time of ownership

\[
\frac{Throughput \times T}{CAPEX + Power \times T \times Electricity}
\]
Cluster cost efficiency over time of ownership

\[
\frac{\text{Throughput} \times T}{\text{CAPEX} + \text{Power} \times T \times \text{Electricity}}
\]

Peak microservice throughput in time
Cluster cost efficiency over time of ownership

\[
\frac{Throughput \times T}{CAPEX + Power \times T \times Electricity}
\]

Total cost of ownership in time
Cluster cost efficiency over time of ownership

\[ \frac{\text{Throughput} \times T}{\text{CAPEX} + \text{Power} \times T \times \text{Electricity}} \]

Cluster capital cost
Cluster cost efficiency over time of ownership

\[
\frac{Throughput \times T}{CAPEX + Power \times T \times Electricity}
\]

Peak cluster energy cost in time
Cluster cost efficiency over time of ownership - **best case**

- Multi-SmartNIC cluster: up to 1.9x more cost efficient after 5 years
- RTA-SHM contains both compute and IO-intensive microservices

![Graph showing cost efficiency over time of ownership]
Wimpy cluster is most cost efficient when all microservices are IO-intensive
Multi-SmartNIC cluster ranks second (14.1% less after 5 years)
Other evaluations

- E3 power proportionality
- E3 control-plane/data-plane mechanisms perform @ scale
  ✔ Mechanism scalability
  ✔ Tail latencies
  ✔ Energy efficiency under power budgets
Conclusion

❖ SmartNICs are heterogenous computing units on the data path
❖ E3 enables energy-efficient microservices on SmartNIC-servers
  ✓ ECMP-based load balancing
  ✓ Load-aware cluster manager
  ✓ Communication-aware microservice placement
❖ Real system based energy efficiency evaluation
  ✓ Compare with homogenous and heterogeneous clusters
  ✓ SmartNIC-servers win:
    - Up to 3x better energy efficiency
    - Up to 4% latency cost
    - Up to 1.9x better cost efficiency after 5 years of ownership