SynCron
Efficient Synchronization Support
for Near-Data-Processing Architectures

Christina Giannoula
Nandita Vijaykumar, Nikela Papadopoulou, Vasileios Karakostas
Ivan Fernandez, Juan Gómez Luna, Lois Orosa
Nectarios Koziris, Georgios Goumas, Onur Mutlu

SAFARI
ETHzürich
CSLab
UNIVERSITY OF TORONTO
UNIVERSITAS MALACITANA
UNIVERSIDAD DE MÁLAGA
Executive Summary

Problem:

Synchronization support is challenging for NDP systems

Prior schemes are not suitable or efficient for NDP systems

Contribution:

SynCron: the first end-to-end synchronization solution for NDP architectures

Key Results:

SynCron comes within 9.5% and 6.2% of performance and energy of an Ideal zero-overhead synchronization scheme
Near-Data-Processing (NDP) Systems

Graph Analytics

Recommendation Systems

Neural Networks

Bioinformatics
Synchronization is Necessary

**Single Source Shortest Path (SSSP)**

for v in Graph:
    for u in neighbors[v]:
        if distance[v] + edge_weight[v, u] < distance[u]
            lock_acquire(u)
        if distance[v] + edge_weight[v, u] < distance[u]
            distance[u] = distance[v] + edge_weight[v, u]
            lock_release(u)
Synchronization challenges in NDP systems:

1. Lack of hardware cache coherence support
2. Expensive communication across NDP units
3. Lack of a shared level of cache memory
NDP Synchronization Solution Space

(1) Shared Memory
- Hardware Cache Coherence
- Remote Atomics
- Specialized Hardware Support

(2) Message-passing
- Software-based Schemes
- Specialized Hardware Support
NDP Synchronization Solution Space

(1) Shared Memory
- Hardware Cache Coherence
- Remote Atomics
- Specialized Hardware Support

(2) Message-passing
- Software-based Schemes
- Specialized Hardware Support

Lack of hardware cache coherence support

CPUs:
- Hierarchical CLH Locks [EuroPar'06]
- Cohort Locks [TOPC'15]
- Ticket Locks [TOCS'91] ...

MPPs:
- QOLB [ASPLOS'89]
NDP Synchronization Solution Space

(1) Shared Memory
- Hardware Cache Coherence
- Remote Atomics
- Specialized Hardware Support
  - GPUs: Fermi GF100 [IEEE Micro’10] ...
  - MPPs: SGI Origin [ISCA’97] Cray T3E [ASPLOS’96] ...
  - CPUs: SSB [ISCA’07] Lock Cache [CASES’01] ...
  - MPPs: Full/Empty Bits [ISCA’83] ...

(2) Message-passing
- Software-based Schemes
  - NDPs: Tesseract [ISCA’15]
  - GPUs: HQL [IPDPS’13] ...

Expensive communication across NDP units
**NDP Synchronization Solution Space**

1. **Shared Memory**
   - **Hardware Cache Coherence**
   - **Remote Atomics**
   - **Specialized Hardware Support**

   **CPUs:**
   - SSB [ISCA’07]
   - Lock Cache [CASES’01]
   - BarrierFilter [MICRO’06]

2. **Message-passing**
   - **Software-based Schemes**
   - **Specialized Hardware Support**

   **NDPs:**
   - Tesseract [ISCA’15], Near-Data Processing for In-memory Analytics [PACT’15]

   **CPUs:**
   - MiSAR [ISCA’15]
   - GPUs: HQL [IPDPS’13]

---

**Lack of a shared level of cache memory**
Prior schemes are not suitable or efficient for NDP systems
NDP Synchronization Solution Space

(1) Shared Memory
- Hardware support
- Remote Atomics
- Specialized Hardware Support

(2) Message-passing
- Software-based Schemes
- Specialized Hardware Support

SynCron’s Key Techniques:
1. **Hardware support** for synchronization acceleration
2. **Direct buffering** of synchronization variables
3. **Hierarchical** message-passing **communication**
4. Integrated hardware-only **overflow management**
1. Hardware Synchronization Support

- No Complex Cache Coherence Protocols
- No Expensive Atomic Operations
- Low Hardware Cost
2. Direct Buffering of Variables

NDP Unit 0

- NDP Core 0
- NDP Core 1
- Main Memory
- Synchronization Engine 0

NDP Unit 1

- NDP Core 0
- NDP Core 1
- Main Memory
- Synchronization Engine 1

Synchronization Processing Unit

Indexing Counters

Synchronization Table

Local lock acquire

| Address | ...
| --- | ---
| -- | ...
| -- | ...
| -- | ...
| -- | ...

2. Direct Buffering of Variables

- No Costly Memory Accesses
- Low Latency
3. Hierarchical Communication
3. Hierarchical Communication

NDP Unit 0
- NDP Core 0
- NDP Core 1
- Main Memory
- Synchronization Engine 0

NDP Unit 1
- NDP Core 0
- NDP Core 1
- Main Memory
- Synchronization Engine 1
- syncronVar

NDP Unit 2
- NDP Core 0
- NDP Core 1
- Main Memory
- Synchronization Engine 2

NDP Unit 3
- NDP Core 0
- NDP Core 1
- Main Memory
- Synchronization Engine 3

Local lock acquire
3. Hierarchical Communication

Minimize Expensive Traffic
4. Integrated Overflow Management

- Low Performance Degradation
- High Programming Ease
SynCron

The first end-to-end synchronization solution for NDP architectures

SynCron’s Benefits:

1. High System Performance
2. Low Hardware Cost

SynCron comes within 9.5% and 6.2% of performance and energy of Ideal zero-overhead synchronization
SynCron
Efficient Synchronization Support for Near-Data-Processing Architectures

Christina Giannoula
Nandita Vijaykumar, Nikela Papadopoulou, Vasileios Karakostas
Ivan Fernandez, Juan Gómez Luna, Lois Orosa
Nectarios Koziris, Georgios Goumas, Onur Mutlu