Xenic: SmartNIC-Accelerated Distributed Transactions

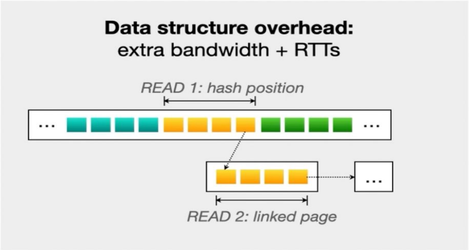
- Rishideep Reddy Rallabandi

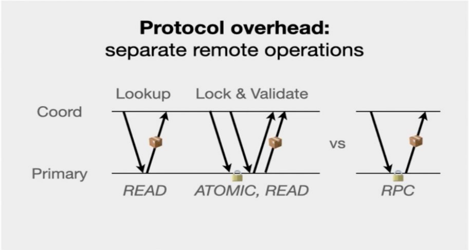
Distributed Transactions

- Distributed transactions refer to operations that involve multiple interconnected systems or resources, where a single transaction spans across these distributed entities.
- Advantages:
  - Scalability
  - Fault Tolerance
  - Geographic Distribution

Challenges in optimizing Distributed transactions

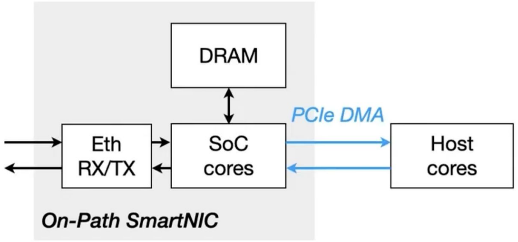
- Distributed transactions pose inherent challenges due to the need for coordination across multiple nodes, leading to increased latency and reduced throughput.
- Traditional approaches rely on two-sided Remote Direct Memory Access (RDMA) and Remote Procedure Calls (RPCs) for transaction execution, often leading to bottlenecks in network utilization and increased communication overhead.





SmartNIC Technology

- The evolution of Smart Network Interface Cards (SmartNICs) represents a breakthrough in distributed systems by integrating processing capabilities within network interface cards.
- SmartNICs leverage hardware acceleration, including RDMA, offloading computation-intensive tasks from the CPU to the NIC, promising significant improvements in transactional processing and network efficiency.



Pros and cons of Smart NIC.

Pros :

- Flexible CPU-bypass remote operations
- Latency savings via stateful NIC operations, efficient PCIe DMA
- Efficient NIC-to-NIC communication

Cons:

- Software packet pipeline latency overhead
- Limited NIC resources

Motivation Behind Xenic’s Development

- Xenic—a system designed to address existing inefficiencies of SmartNICs.
- The motivation is to leverage SmartNICs to optimize transactional processing, reduce communication overhead, and enhance both throughput and latency in distributed transactional scenarios.
- By reimagining the utilization of SmartNICs and their integration into the transactional workflow, Xenic seeks to offer substantial improvements in performance and efficiency over existing systems.

Xenic

Distributed transactions accelerated with on-path SmartNICs

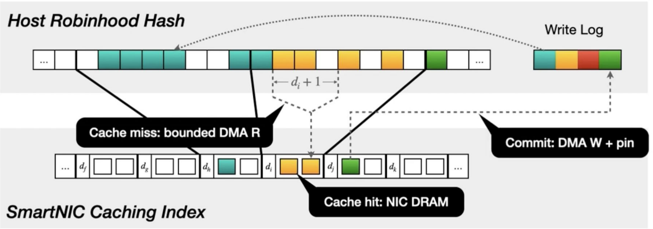
1. Co-designed data store, spread across NIC + host DRAM :
    - Minimize lookup overhead, utilizing NIC's on-board memory
2. SmartNIC function shipping :
    - Offload transaction logic to avoid PCIe crossings
3. Multi-hop OCC protocols:
    - Reduce communication with optimized message patterns
4. Stateful, asynchronous SmartNIC operation framework :
    - Exploit the SmartNIC's hardware interfaces

Xenic - Robinhood data store.

Host DRAM contains all objects; SmartNIC caches objects and lookup hints

Critical path accesses: NIC memory hit or DMA read, DMA log write

- Lookup hints limit DMA cost for cache misses
- OCC + pinning ensure NIC/host consistency



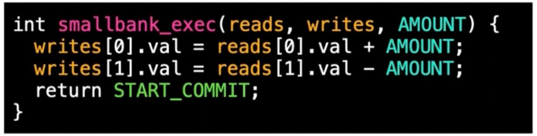
Xenic - smartNIC function shipping

Xenic provides SmartNIC cores as a function shipping target

Shipping execution can reduce overhead, depending on application-level computation and state requirements

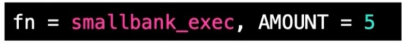
SmartNIC function shipping saves coordinator PCIe crossings

Define execution transformation on NIC



```
int smallbank_exec(reads, writes, AMOUNT) {
  writes[0].val = reads[0].val + AMOUNT;
  writes[1].val = reads[1].val - AMOUNT;
  return START_COMMIT;
}
```

Append annotation to transactions on host

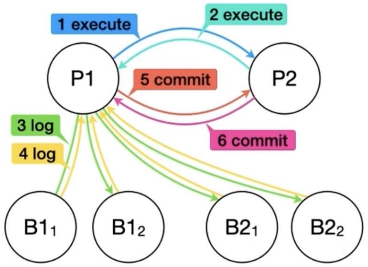


```
fn = smallbank_exec, AMOUNT = 5
```

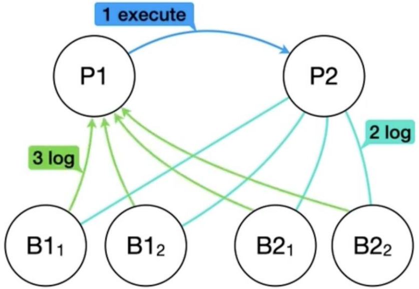
Xenic - Multi-hop OCC protocols

Xenic also ships execution to remote smartNICs

Multi-hop NIC to NIC communication increases network efficiency.



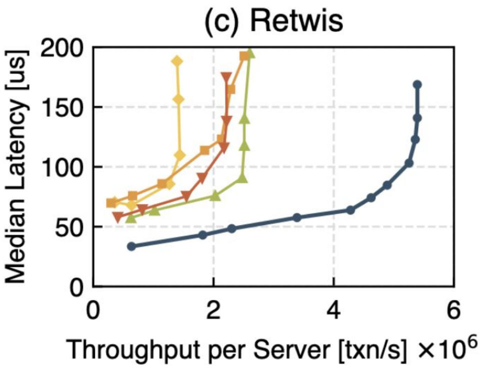
Local write (P1) + remote write (P2)
Execution at coordinator P1



Local write (P1) + remote write (P2)
Shipped to remote primary P2

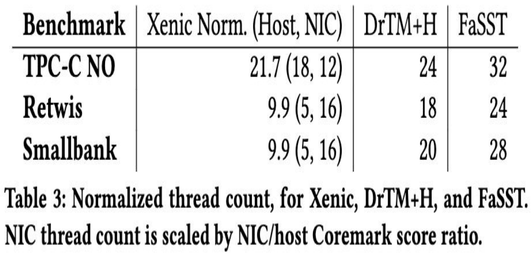
Evaluations:

- Retwis simulates Twitter-like transactions (50% read-only, 1-10 keys per transaction).
- Replication factor of 3, 1M keys per server.
- 2.07× higher peak throughput vs. DrTM+H.
- 42% lower median latency at low load.
- Xenic outperforms DrTM+H and FaSST in Retwis.
- Xenic maximizes network bandwidth, ensuring higher efficiency.
- FaSST approaches peak throughput but faces latency challenges due to its RPC design.



Evaluations:

- Xenic demonstrates reduced host thread requirements for Retwis and Smallbank (2 application threads and 3 worker threads). TPC-C necessitates 18 host threads due to its computationally intensive operations.
- Xenic saves 2.3 threads for TPC-C, 8.1 threads for Retwis, and 10.1 threads for Smallbank compared to DrTM+H.
- Xenic's superior throughput and core efficiency compared to FaSST and DrTM+H stem from lower utilization



Benchmark	Xenic Norm. (Host, NIC)	DrTM+H	FaSST
TPC-C NO	21.7 (18, 12)	24	32
Retwis	9.9 (5, 16)	18	24
Smallbank	9.9 (5, 16)	20	28

Table 3: Normalized thread count, for Xenic, DrTM+H, and FaSST. NIC thread count is scaled by NIC/host Coremark score ratio.

Summary

High-performance, CPU-efficient distributed transactions

By leveraging on-path SmartNICs:

- Xenic avoids RDMA compromises
- Xenic provides a new, remote access-optimized data store
- Xenic selectively offloads transaction logic
- Xenic applies multi-hop communication patterns

Thank you