Collie: Finding Performance Anomalies on RDMA subsystems

Xinhao Kong, Yibo Zhu, Huaping Zhou, Zhuo Jiang, Jianxi Ye,
Chuanxiong Guo, Danyang Zhuo

Duke University
ByteDance
RDMA are getting rapidly adopted

- Remote procedure call: FaSST (OSDI ‘16), eRPC (NSDI ‘19), Flock (SOSP ‘21)
- Graph processing: Wukong (OSDI ‘16), A1 (SIGMOD ‘20)
- Deep learning: GPUNet (OSDI ‘14), TensorFlow (OSDI ‘16), BytePS (OSDI ‘20)
There exist unexpected *performance anomalies*

Why the **NIC** sends a large amount of **PFC pause frames** when the RX traffic is only \(\sim 5\text{Gbps}\)?

PFC pause frame storms are catastrophic, e.g., can trigger deadlocks [1]

There exist unexpected *performance anomalies*

Throughput is 195 Gbps, Great!

Why it drops to less than 40 Gbps?

The same application migrates...

200 Gbps RNIC
Host 1 of type A

200 Gbps RNIC
Host 2 of type A
We need test!
Vendors have done extensive tests

This figure is based on public resources and do not contain proprietary information.
Vendors have done extensive tests
Unfortunately, we still need integration tests over the entire RDMA subsystems.
Unfortunately, we still need integration tests over the entire RDMA subsystems.
Unfortunately, we still need integration tests over the entire RDMA subsystems.
Unfortunately, we still need integration tests over the entire RDMA subsystems.
Unfortunately, we still need integration tests over the entire RDMA subsystems.
Existing integration tests

- Run simple benchmarks (e.g., Perftest) to conduct basic throughput and latency tests.

- Run a representative set of the applications (e.g., distributed machine learning application) before real deployment.
Why are they insufficient?

- Workload triggers anomaly
- Application possible workloads

![Diagram showing Workload Space with Perftest, DML application, and Storage application with crosses indicating possible workloads.](image)
Why are they insufficient?

- Workload triggers anomaly
- Application possible workloads

Further development
Why are they insufficient?

- Workload triggers anomaly
- Application possible workloads

![Diagram with Workload Space, Perftest, Storage application, and DML Application (modified)]
Strawman solutions are not enough

- Run simple benchmarks (e.g., Perftest) to conduct basic throughput and latency tests.
- Run a representative set of the applications (e.g., distributed machine learning application) before real deployment.

They only test with existing workloads and cannot uncover potential performance anomalies.
Collie

• **Systematic search** for application workloads that can trigger performance anomalies

This will benefit:

• Datacenter operators: ensure the network infrastructure runs with high performance and reliability

• Vendors: understand their bugs and hardware limitations

• Developers: develop better RDMA applications
Question: How to define performance anomaly?

This is hard in general.

• Hardware specifications contain little information of what is the hardware’s expected performance!

We take the first step to focus on two types of concrete anomalies:
1. No network congestion -> PFC pause frames!
2. Throughput is substantially lower than what’s specified in RDMA NIC specification in terms bits per second or messages per second
Challenge #1: Comprehensive Search Space
Challenge #2: Efficient Search Algorithm

No search signal
Solution #1: Finding the narrow waist

- The narrow waist RDMA programming abstraction is clearly defined and stable.

![Diagram showing RDMA Programming Abstractions and various RDMA subsystems]

- HERD
- eRPC
- FaSST
- FaRM
- Pangu
- DrTM

RDMA Programming Abstractions

Various RDMA subsystems
Solution #1: Finding the narrow waist

**DRAM or GPU memory**
- `ibv_reg_mr(…)`
  - Dimension (1)
  - Dimension (2)

**WQ depth**
- `ibv_create_qp(…)`
- `ibv_modify_qp(…)`
  - Dimension (3)

**CQ depth**
- `ibv_create_cq(…)`
  - Dimension (3)

**Memory Region (MR)**
- `ibv_post_send(…)`
- `ibv_post_recv(…)`
  - Dimension (4)

**Queue Pair (QP)**
- `ibv_poll_cq(…)`
  - Dimension (4)

**Completion Queue (CQ)**

**Work Queue Element (WQE)**

Request pattern (size, opcode, etc)

Generate CQ element
Solution #2: Hardware counters as search signal

• We propose to use two types of counters as the search signal.
  
  • Performance counters (e.g., bits per second)
  
  • Diagnostic counters (e.g., PCIe backpressure)

• The lower/higher the performance/diagnostic counter is, the test case is more likely to trigger an anomaly.

• Collie uses simulated annealing to maximize/minimize hardware counters to search for anomalies.
Solution #2: Hardware counters as search signal
Solution #2: Hardware counters as search signal

No anomaly

Diagnostic counter ↑

mutate
Solution #2: Hardware counters as search signal
Solution #3: Minimal Feature Set (MFS)

An anomaly is a region rather than a single point!

Redundant!
Solution #3: Minimal Feature Set (MFS)

We find a new anomaly
Solution #3: Minimal Feature Set (MFS)

We conduct a few tests on the new anomaly.
Solution #3: Minimal Feature Set (MFS)

The region is marked as found

Redundant!
Solution #3: Minimal Feature Set (MFS)

The region is marked as found

Redundant!
Implementation

Workload Generator
- Simulated Annealing

Workload Engine
- Workload Setup

Anomaly Monitor
- Anomaly Detection
- MFS Generation
- Updated MFS Set

RDMA Subsystem
- Set up workload
- Counters (Perf and diagnostic)
- Throughput, PFC pause frames
Evaluation

- Performance anomalies found by Collie
- The efficiency of Collie
## Evaluation Settings

<table>
<thead>
<tr>
<th>Type</th>
<th>RNIC</th>
<th>Speed</th>
<th>CPU</th>
<th>PCIe</th>
<th>NPS</th>
<th>Memory</th>
<th>GPU</th>
<th>BIOS</th>
<th>Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>CX-5 DX</td>
<td>25 Gbps</td>
<td>Intel(R) Xeon(R) CPU 1</td>
<td>3.0 x 16</td>
<td>1</td>
<td>128 GB</td>
<td>-</td>
<td>INSYDE</td>
<td>4.19</td>
</tr>
<tr>
<td>B</td>
<td>CX-5 DX</td>
<td>100 Gbps</td>
<td>Intel(R) Xeon(R) CPU 2</td>
<td>3.0 x 16</td>
<td>1</td>
<td>768 GB</td>
<td>-</td>
<td>AMI</td>
<td>4.14</td>
</tr>
<tr>
<td>C</td>
<td>CX-5 DX</td>
<td>100 Gbps</td>
<td>Intel(R) Xeon(R) CPU 2</td>
<td>3.0 x 16</td>
<td>1</td>
<td>384 GB</td>
<td>V100</td>
<td>AMI</td>
<td>5.4</td>
</tr>
<tr>
<td>D</td>
<td>CX-6 DX</td>
<td>100 Gbps</td>
<td>Intel(R) Xeon(R) CPU 2</td>
<td>3.0 x 16</td>
<td>1</td>
<td>768 GB</td>
<td>-</td>
<td>AMI</td>
<td>4.14</td>
</tr>
<tr>
<td>E</td>
<td>CX-6 DX</td>
<td>200 Gbps</td>
<td>AMD EPYC CPU 1</td>
<td>4.0 x 16</td>
<td>1</td>
<td>2 TB</td>
<td>A100</td>
<td>AMI</td>
<td>5.4</td>
</tr>
<tr>
<td>F</td>
<td>CX-6 DX</td>
<td>200 Gbps</td>
<td>Intel(R) Xeon(R) CPU 3</td>
<td>4.0 x 16</td>
<td>1</td>
<td>2 TB</td>
<td>A100</td>
<td>AMI</td>
<td>5.4</td>
</tr>
<tr>
<td>G</td>
<td>CX-6 VPI</td>
<td>200 Gbps</td>
<td>AMD EPYC CPU 1</td>
<td>4.0 x 16</td>
<td>2</td>
<td>2 TB</td>
<td>-</td>
<td>AMI</td>
<td>5.4</td>
</tr>
<tr>
<td>H</td>
<td>P2100G</td>
<td>100 Gbps</td>
<td>Intel(R) Xeon(R) CPU 2</td>
<td>3.0 x 16</td>
<td>1</td>
<td>384 GB</td>
<td>-</td>
<td>AMI</td>
<td>5.4</td>
</tr>
</tbody>
</table>

We test with 8 types of RDMA subsystems, including 6 types of RNICs.
- Performance anomalies found by Collie

<table>
<thead>
<tr>
<th>#</th>
<th>RNIC</th>
<th>Dir.</th>
<th>Transport</th>
<th>MTU</th>
<th>WQE</th>
<th>SGE</th>
<th>WQ depth</th>
<th>Message Pattern</th>
<th># of QPs</th>
<th>Symptom</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>CX-6</td>
<td>-</td>
<td>UD SEND</td>
<td>-</td>
<td>&gt;64</td>
<td>-</td>
<td>≥256</td>
<td>-</td>
<td>-</td>
<td>pause frame</td>
</tr>
<tr>
<td>#2</td>
<td>CX-6</td>
<td>-</td>
<td>UD SEND</td>
<td>-</td>
<td>≤8</td>
<td>-</td>
<td>≥1024</td>
<td>≤1KB</td>
<td>≥≈16</td>
<td>low throup.</td>
</tr>
<tr>
<td>#3</td>
<td>CX-6</td>
<td>-</td>
<td>RC READ</td>
<td>1K</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>≥16KB</td>
<td>-</td>
<td>pause frame</td>
</tr>
<tr>
<td>#4</td>
<td>CX-6</td>
<td>Bi-</td>
<td>RC READ</td>
<td>-</td>
<td>≥32</td>
<td>≥4</td>
<td>-</td>
<td>-</td>
<td>≥≈160</td>
<td>pause frame</td>
</tr>
<tr>
<td>#5</td>
<td>CX-6</td>
<td>-</td>
<td>RC SEND</td>
<td>1K</td>
<td>≥64</td>
<td>-</td>
<td>≥1024</td>
<td>≥2KB and ≤8KB</td>
<td>-</td>
<td>pause frame</td>
</tr>
<tr>
<td>#6</td>
<td>CX-6</td>
<td>-</td>
<td>RC SEND</td>
<td>1K</td>
<td>≤16</td>
<td>≥2</td>
<td>≥1024</td>
<td>≤1KB</td>
<td>≥≈32</td>
<td>low throup.</td>
</tr>
<tr>
<td>#7</td>
<td>CX-6</td>
<td>-</td>
<td>RC WRITE</td>
<td>-</td>
<td>No</td>
<td>-</td>
<td>-</td>
<td>≤1KB and ≥≈12K MRs</td>
<td>-</td>
<td>low throup.</td>
</tr>
<tr>
<td>#8</td>
<td>CX-6</td>
<td>-</td>
<td>RC WRITE</td>
<td>-</td>
<td>No</td>
<td>-</td>
<td>≤16</td>
<td>≤1KB</td>
<td>≥≈500</td>
<td>low throup.</td>
</tr>
<tr>
<td>#9</td>
<td>CX-6</td>
<td>Bi-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>≥3</td>
<td>-</td>
<td>mix of ≤1KB &amp; ≥64KB</td>
<td>-</td>
<td>pause frame</td>
</tr>
<tr>
<td>#10</td>
<td>CX-6</td>
<td>Bi-</td>
<td>RC WRITE</td>
<td>-</td>
<td>≥64</td>
<td>-</td>
<td>-</td>
<td>mix of ≤1KB &amp; ≥64KB</td>
<td>≥≈320</td>
<td>pause frame</td>
</tr>
<tr>
<td>#11</td>
<td>CX-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Bidirectional cross-socket traffic on particular AMD servers</td>
</tr>
<tr>
<td>#12</td>
<td>CX-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Particular GPU-Direct RDMA traffic on particular servers</td>
</tr>
<tr>
<td>#13</td>
<td>CX-6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Co-existence of loop traffic and receiving traffic</td>
</tr>
<tr>
<td>#14</td>
<td>P2100</td>
<td>Bi-</td>
<td>RC</td>
<td>4K</td>
<td>-</td>
<td>≥4</td>
<td>-</td>
<td>-</td>
<td>≥≈1300</td>
<td>low throup.</td>
</tr>
<tr>
<td>#15</td>
<td>P2100</td>
<td>-</td>
<td>UD SEND</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>≥64</td>
<td>-</td>
<td>≥≈32</td>
<td>pause frame</td>
</tr>
<tr>
<td>#16</td>
<td>P2100</td>
<td>-</td>
<td>RC READ</td>
<td>1K</td>
<td>≥8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>≥≈500</td>
<td>pause frame</td>
</tr>
<tr>
<td>#17</td>
<td>P2100</td>
<td>-</td>
<td>RC SEND</td>
<td>-</td>
<td>≤16</td>
<td>-</td>
<td>≥128</td>
<td>≤1KB</td>
<td>≥≈64</td>
<td>pause frame</td>
</tr>
<tr>
<td>#18</td>
<td>P2100</td>
<td>Bi-</td>
<td>RC</td>
<td>1K</td>
<td>≥32</td>
<td>-</td>
<td>-</td>
<td>≤64KB</td>
<td>≥≈30</td>
<td>pause frame</td>
</tr>
</tbody>
</table>
• The speed of Collie
  • Random: Random Search
  • BO: Bayesian Optimization

• We run each approach for 10 hours
The speed of Collie
- Random: Random Search
- BO: Bayesian Optimization

We run each approach for 10 hours
• The speed of Collie
  • Random: Random Search
  • BO: Bayesian Optimization

• We run each approach for 10 hours
Lessons and Future Work

- Holistic performance tuning over entire RDMA subsystems is crucial.
  - MTU, PCIe, NUMA, IOMMU, etc.

- Opaque resource limitation of the RDMA subsystems.
  - New challenges for virtualization & isolation.

- End-to-end flow control for RDMA is very important.
Conclusion

• Performance anomalies in RDMA are a real threat for datacenters
  • Pause frame storm
  • Unexpected poor performance for applications

• Integration tests are critical. However, existing approaches only test known workloads and are thus insufficient.

• Collie: systematic search for application workloads that trigger anomalies
  • Comprehensive search space design based on the verbs abstraction
  • Simulated annealing using hardware counters
  • 15 new performance anomalies
  • https://github.com/bytedance/Collie/
Thank you!