Flexplane: An Experimentation Platform for Resource Management in Datacenters

Amy Ousterhout\textsuperscript{1}, Jonathan Perry\textsuperscript{1}, Hari Balakrishnan\textsuperscript{1}, Petr Lapukhov\textsuperscript{2}

\textsuperscript{1}MIT, \textsuperscript{2}Facebook
Datacenter Networks

• Applications have diverse requirements
• Dozens of new resource management schemes
  – Low latency: DCTCP
  – Min FCT: PDQ, RCP, pFabric, PERC
  – Deadlines: $D^3$, $D^2$TCP
• Difficult to experiment with schemes in real networks
  – Requires changes to hardware routers
Experimentation with Resource Management

• Experimentation in real networks
  – Programmable hardware - limited flexibility
  – Software routers - limited throughput
  – Emulation – limited throughput

By Altera Corporation - Altera Corporation, CC BY 3.0
Experimentation with Resource Management

• Experimentation in simulation (e.g., ns, OMNeT++)
  – Does not accurately model real network stacks, NICs, and distributed applications
  – Does not run in real time

No existing approach to experimentation provides accuracy, flexibility, and high throughput
Our Contributions

• Key idea: whole-network emulation
• Flexplane: a platform for faithful experimentation with resource management schemes
  – Accurate – predicts behavior of hardware
  – Flexible – express schemes in C++
  – High throughput – 761 Gbits/s
Approach: Whole-Network Emulation

Real Network

Emulated Network

class MyScheduler {...}

class MyAQM {...}
Abstract Packets

• Resource management schemes are data-independent
• Concise representation of one MTU
  – Source, destination, flow, ID
  – Custom per-scheme fields
• Transmitted over the real network
  – Consume 1-2% of network bandwidth
Emulator

• Real-time network simulator
• Faster than standard network simulators
  – Time divided into abstract-packet-sized timeslots
  – Omits endpoint software
Sending Real Packets

• Drop packet if dropped in emulation
• Otherwise, modify and send
  – ECN marks, rates (e.g., in RCP, PDQ, PERC)
Accuracy

• Goal: predict behavior of a hardware network
• Hardware latency:
• Added latency of Flexplane:
  – RTT to emulator
  – Unloaded delay
  – Queuing delay in real network
Flexplane API

- Decouples schemes from framework

<table>
<thead>
<tr>
<th>Emulator (pattern)</th>
<th>int route(AbstractPkt *pkt)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>int classify(AbstractPkt *pkt, int port)</td>
</tr>
<tr>
<td></td>
<td>enqueue(AbstractPkt *pkt, int port, int queue)</td>
</tr>
<tr>
<td></td>
<td>AbstractPkt *schedule(int output_port)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Emulator (generic)</th>
<th>input(AbstractPkt **pkts, int n)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>output(AbstractPkt **pkts, int n)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Endpoints</th>
<th>prepare_request(sk_buff *skb, char *request_data)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>prepare_to_send(sk_buff *skb, char *allocation_data)</td>
</tr>
</tbody>
</table>

incoming packets –> route classify enqueue schedule –> outgoing packets
Multi-core Emulator Architecture

• Common approaches are ill-suited for Flexplane
  – Parallel (RouteBricks, IX)
  – Pipeline (Fastpass)
Multicore Emulator Architecture

• Pin network components to cores, connect in a graph
  – Router state not shared across cores
• Loose synchronization, batching, prefetching

![Diagram of multicore emulator architecture showing ToR, CPU core, aggregation, and FIFO queues.](image-url)
Implementation

- Emulator uses Intel DPDK for low-latency NIC access
- Endpoints run a Linux qdisc
Evaluation

- Accuracy
- Utility
- Emulator throughput
Flexplane is Accurate

- Bulk TCP: 5 senders, 1 receiver
- Throughput 9.2-9.3 Gbits/s vs. 9.4 Gbits/s in hardware
- Similar queue occupancies

<table>
<thead>
<tr>
<th></th>
<th>Hardware</th>
<th>Flexplane</th>
</tr>
</thead>
<tbody>
<tr>
<td>DropTail</td>
<td>931</td>
<td>837</td>
</tr>
<tr>
<td>RED</td>
<td>138</td>
<td>104</td>
</tr>
<tr>
<td>DCTCP</td>
<td>61</td>
<td>51</td>
</tr>
</tbody>
</table>
Flexplane is Accurate

- RPC web search workload

- Accurate to within 2-14% for loads up to 60%
- Observe behavior not visible in simulations
Flexplane is Easy to Use

- Implemented several schemes in dozens of lines of code

<table>
<thead>
<tr>
<th>scheme</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>drop tail queue manager</td>
<td>39</td>
</tr>
<tr>
<td>RED queue manager</td>
<td>125</td>
</tr>
<tr>
<td>DCTCP queue manager</td>
<td>43</td>
</tr>
<tr>
<td>priority queueing scheduler</td>
<td>29</td>
</tr>
<tr>
<td>round robin scheduler</td>
<td>40</td>
</tr>
<tr>
<td>HULL scheduler</td>
<td>60</td>
</tr>
<tr>
<td>pFabric QM, queues, scheduler</td>
<td>251</td>
</tr>
</tbody>
</table>
Example: DCTCP in Flexplane

```c
void enqueue(AbstractPkt *pkt, uint32_t port, uint32_t queue) {
    uint32_t qlen = m_bank->occupancy(port, queue);
    if (qlen >= m_dctcp_params.q_capacity) {
        /* no space to enqueue, drop this packet */
        drop(pkt, port);
        return;
    }
    /* mark if queue occupancy is greater than K */
    if (qlen > m_dctcp_params.K_threshold) {
        /* Set ECN mark on packet */
        mark_ecn(pkt, port);
    }
    m_bank->enqueue(port, queue, pkt);
}
```
Flexplane Enables Experimentation

- Evaluating trade-offs between resource management schemes
Flexplane Enables Experimentation

- Experiment with real distributed applications such as Spark

\[
\begin{array}{|c|c|c|}
\hline
\text{% Change in Completion Time} & \text{Relative to DropTail} \\
\hline
\text{Coordinate descent} & \text{Sort} \\
\hline
\text{DCTCP} & +4.4\% & -4.8\% \\
\hline
\text{HULL} & +29.4\% & -2.6\% \\
\hline
\end{array}
\]

- Performance depends on network and CPU
Emulator Throughput

- Emulator provides 761 Gbits/s of aggregate throughput with 10 total cores
Emulator Throughput

• Throughput varies by 29% across schemes

• 81x as much throughput per clock cycle as RouteBricks
Flexplane: an Experimentation Platform

• Whole-network emulation
• Flexplane: a platform for faithful experimentation with resource management schemes
  – Accuracy, flexibility, and high throughput

https://github.com/aousterh/flexplane