Combining Hardware and Simulation for Datacenter Scaling Studies

ONDM 2017, 21st International Conference on Optical Network Design and Modeling

Sarah Ruepp*, Artur Pilimon*, Jakob Thrane, Michael Galili, Michael Berger and Lars Dittmann

Department of Photonics Engineering, Technical University of Denmark, Kgs. Lyngby, Denmark
E-mail: artpil@fotonik.dtu.dk

DTU Fotonik
Department of Photonics Engineering
Agenda

Introduction

Goals

Work methodology:
- A Simulation and Hardware setup
- Packet conversion latency measurements

Results

Takeaway points and future work
Introduction. DC traffic growth (1)

- Tremendous increase of intra-DCN traffic globally
- Fast proliferation of Cloud and network virtualization technologies

Introduction. DC traffic growth (2)

- Most (~77%) of the generated traffic remains within a DC!!!
Introduction. Global trends

- Increase in DCN scale and power
  - Cloud DCs, MDCs

- Diverse traffic patterns
  - Mice/Elephant flows
  - Multi-tier applications

- New DCN architectures
  - Jupiter, RoR, Helios, c-Through, OPMDC, COSIGN, etc.

- Optical switching
  - 2-D/3-D MEMS, optical cross-bar, etc.
**Introduction. Research: COSIGN approach**

**Manual, Painful, Error-prone**
- Workload deployment requires time and intervention of different admin roles
- Connectivity of workload components is static
- Performance is achieved with over provisioning, questionable scaling characteristics

**Discrete, Heterogeneous, Complex**
- Multiple management roles and domains
- Unable to make global decisions efficiently and dynamically
- Requires expertise in different vendor technologies

**Traditional fat-tree designs**
- Over provisioned
- Inefficient for east-west communications
- Restricts (virtual) server placement decisions

**Cabling mess**
- Many different incompatible interconnects

**Consolidated, Converged, Programmable**
- Unified IT and SDN orchestrator
- Converged IT + Network virtualization for data centers
- Efficient and optimized virtual resource utilization and allocation
- Enhanced abstraction mechanisms for emerging optical technologies

**Automated, Streamlined, Optimized**
- Rapid and flexible workload management
- On demand resource allocation and release
- Scalable and efficient virtualized workload components connectivity

**Converged DCN interconnect**
- Flattened DC network architecture
- Fast TOR switch
- 3D stacked transceiver
- InP fast switch
- Low loss beam-streering switch
- Hollow-core and Multi-core fibers

**High-performance optical solutions**
- Rapid and flexible workload management
- On demand resource allocation and release
- Scalable and efficient virtualized workload components connectivity

**Data Plane**
- Converged DCN interconnect
- Cabling mess

**Control Plane**
- Unified Control Plane
- Plan, Deploy, Operate

**Orchestrated Management**
- Automated, Streamlined, Optimized
- Hour(s)-Days-Weeks
- App level request
- Infrastructure Reconfigured

**Manual, Painful, Error-prone**
- Workload deployment requires time and intervention of different admin roles
- Connectivity of workload components is static
- Performance is achieved with over provisioning, questionable scaling characteristics

**Discrete, Heterogeneous, Complex**
- Multiple management roles and domains
- Unable to make global decisions efficiently and dynamically
- Requires expertise in different vendor technologies

**Traditional fat-tree designs**
- Over provisioned
- Inefficient for east-west communications
- Restricts (virtual) server placement decisions

**Cabling mess**
- Many different incompatible interconnects

**Consolidated, Converged, Programmable**
- Unified IT and SDN orchestrator
- Converged IT + Network virtualization for data centers
- Efficient and optimized virtual resource utilization and allocation
- Enhanced abstraction mechanisms for emerging optical technologies

**Automated, Streamlined, Optimized**
- Rapid and flexible workload management
- On demand resource allocation and release
- Scalable and efficient virtualized workload components connectivity

**Converged DCN interconnect**
- Flattened DC network architecture
- Fast TOR switch
- 3D stacked transceiver
- InP fast switch
- Low loss beam-streering switch
- Hollow-core and Multi-core fibers

**High-performance optical solutions**
- Rapid and flexible workload management
- On demand resource allocation and release
- Scalable and efficient virtualized workload components connectivity

**Data Plane**
- Converged DCN interconnect
- Cabling mess

**Control Plane**
- Unified Control Plane
- Plan, Deploy, Operate

**Orchestrated Management**
- Automated, Streamlined, Optimized
- Hour(s)-Days-Weeks
- App level request
- Infrastructure Reconfigured
Challenges and Solutions

- Datacenters for “trying out things” are not really available for researchers
- New research approaches and technologies may not be implemented in commercial products
- Mass-scaling of datacenters: high CAPEX

**SOLUTION**: Use simulation, and combine it with the hardware that is already available in the datacenter
Goals

1. Analysis of the functional capabilities of a simulation tool
2. Feasibility of building a hybrid (simulation - real hardware) research testbed
3. Preliminary performance evaluation (Packet conversion latency measurements)
Work methodology
Test scenarios

Data Center 1

Data Center 2

Data Center 3

Data Center 1

Data Center 2

Data Center 3
## Work methodology
Choosing a M&S environment: Hardware-in-the-Loop

<table>
<thead>
<tr>
<th>Simulation tool</th>
<th>Real-time</th>
<th>HIL</th>
<th>SDN</th>
<th>Complexity</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMNET++</td>
<td>+/-</td>
<td>+/-</td>
<td>-</td>
<td>+/-</td>
<td>APL/GNU GPL*</td>
</tr>
<tr>
<td>Qualnet</td>
<td>+/-</td>
<td>-</td>
<td>-</td>
<td>+/-</td>
<td>Commercial</td>
</tr>
<tr>
<td>CORE</td>
<td>+/-</td>
<td>+/-</td>
<td>-</td>
<td>+/-</td>
<td>GNU GPL</td>
</tr>
<tr>
<td>Riverbed/OPNET</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+/-*</td>
<td>Commercial</td>
</tr>
<tr>
<td>NS2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+/-</td>
<td>GNU GPL</td>
</tr>
<tr>
<td>NS3</td>
<td>+/-</td>
<td>+/-</td>
<td>-</td>
<td>+/-</td>
<td>GNU GPL</td>
</tr>
<tr>
<td>DCSim</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>+/-</td>
<td>GNU GPL</td>
</tr>
</tbody>
</table>
Work methodology
Riverbed modeler: System-in-the-Loop principle

Real Network

SITL Module

Riverbed Simulation

Real packet arrives at network interface of workstation running simulation

Real packet is converted to a simulation packet

Simulation packet is forwarded through the simulation, where it experiences real-time delays and protocol effects

Simulation packet is mapped at external interface and is converted to real packet

Packet conversion latency

Real packet sent out on network interface of simulation workstation
Work methodology
A Hybrid simulation-real-hardware testbed (1)

Virtual link to PHY NIC’s driver

Virtual 10G Eth link

10G/40G NIC

“Data Center 1”

High Performance L2-3/4-7 Traffic Generator

Network node

“Data Center 3”

High Performance L2-3/4-7 Traffic Generator

Cluster node

“Data Center 1”

High Performance L2-3/4-7 Traffic Generator

DCN GW

“Data Center 3”

High Performance L2-3/4-7 Traffic Generator

Virtual link to PHY NIC’s driver

Virtual 10G Eth link

10G/40G NIC

ToR switch

Server rack

Virtual GW

High Performance Workstation/Server

DCN GW

Master Ring

Cluster node

DCN Simulation model

Network node

Server rack

Virtual GW

High Performance Workstation/Server

DCN GW

Master Ring

Cluster node

DCN Simulation model
Work methodology
Experimental setup (2)
Work methodology
Packet conversion efficiency

Packet translation depth (level) at the real/simulated interface
Results

Real traffic generation experiments (1)
Results
Packet conversion latency measurements

CDFs of packet conversion delay

Conversion delay at the SITL interface, µs
Results
Packet conversion latency measurements

CDFs of packet conversion delay

95% < 4.5 μs

Conversion Delay at the SITL interface, μs

Data Flow
Results
Packet conversion latency measurements

TCP ACK Flow

95% < 6 μs
Results
Packet queueing latency measurements

CDFs of the queueing delay

- SITL.CDF of Queuing Delay for Incoming Real Packets, TCP traffic flow
- SITL.CDF of Queuing Delay for Incoming Real Packets, TCP ACKs

95% < 4 ms
Results

Real traffic generation experiments (2)

Data Flow:
3.41 GB File transfer
Results
Packet conversion latency measurements

Data rate: 48.4 – 216 Mbit/s
Takeaway points

• Building a hybrid simulation-real-hardware experimental setup for DCN performance studies – not a trivial task!
• Two critical latency components must be taken into consideration: packet conversion delay and packet queueing delay (SITL gateway).
• Packet conversion delay:
  – Packet translation depth (traffic type)
  – Specifics of packet capture by the WinPcap [9] (libPcap for Linux) module
  – Conversion functions (code efficiency)
  – NIC characteristics/functionality
• SITL gateway adds a conversion delay in the order of microseconds ($\mu$s) as well as load-dependent buffering delays ($ms$)
Future steps

• Consider the impact of more realistic high bit-rate mixed traffic patterns, bursty workloads (DC workload traces or a real Map-Reduce cloud application)

• Simulation model on a powerful workstation with multiple electro-optical interfaces (10G/40G)

• Wider range of performance evaluation scenarios and metrics

• Integration with an SDN framework (hybrid SDN-controlled setup) and OCS physical nodes
Thank you for your attention!

Questions?
Suggestions?

Contact: Artur Pilimon, artpil@fotonik.dtu.dk
Thank You for Your attention
Results
Packet conversion latency measurements

![Graph showing packet conversion delay vs simulation time]

- **Conversion to Real Delay (μsec)**
- **Conversion to Simulated Delay (μsec)**

Simulation Time, s

Average conversion delay, μs
Results

Packet conversion latency measurements

![Graph showing instantaneous queue size in SITL GW, (pkts) over simulation time (s). The x-axis represents simulation time in seconds ranging from 0 to 600, and the y-axis represents instantaneous queue size in packets ranging from 0 to 12. The blue line indicates the queue size for incoming real packets.](Image)
Software models and Tester interface