Berkeley Extensible Software Switch (BESS)

Sangjin Han – UC Berkeley
Christian Maciocco – Intel

DPDK US Summit - San Jose - 2016
BESS: A Virtual Switch Tailored for NFV

Sangjin Han, Aurojit Panda, Brian Kim, Keon Jang, Joshua Reich, Saikrishna Edupuganti, Christian Maciocco, Sylvia Ratnasamy, Scott Shenker

SPAN: Software Principle for Advanced Networking

• Does Open vSwitch meet all the requirements for NFV?

1. Performance
   - OVS (~1Mpps) → OVS-DPDK (~15Mpps)
   - Vanilla DPDK (~59 Mpps/core)
   - Packet I/O is only half of the problem

2. Flexibility
   - Custom actions?
   - Stateful packet processing?

3. Extensibility
   - Must enable NFV controller evolution
   - Easily add support for new/niche protocols
Alternative Approach with BESS

- Modular pipeline as a dataflow graph
- Each module can run arbitrary code
  - Not limited by Match/Action semantics
  - Independently extensible & optimizable
- Everything is programmable, not just flow tables
- You pay only for what you use.
  - No performance cost for unused features
Berkeley Extensible Software Switch (BESS)

- BESS is a programmable platform for vSwitch dataplane
- Clean-slate internal architecture with NFV in mind
  - Highly extensible & customizable
  - Readily deployable with backward compatibility
  - ... all with extreme performance:
    - Sub-microsecond latency
    - Line-rate 40Gbps with min-sized packets on two cores
    - (> 2x faster than existing virtual switches)
What can you Build on BESS?

- NFV virtual switches

- Other possible usages:
  - Network virtualization for multi-tenant datacenters
  - Network functions (firewall, VPN, ADC, etc.)
  - Traditional L2/L3 switches
  - “Smart” NICs
  - ...

Our use Case: Elastic Edge (E2)

- E2 is a research prototype of our NFV platform
  - “E2: A Framework for NFV Applications”, Palkar et al., In ACM SOSP, 2015
Packet buffer allocation/deallocation
- ~10 CPU cycles per packet

CPU scheduling
- ~50 CPU cycles per round
- Scales well with thousands of traffic classes

Dynamic per-packet metadata attributes
- Zero instruction overhead for access
- Optimal CPU cache-line usage
Performance Evaluation

- OPNFV VSPERF usage models

VSPERF LTD Supported Deployment Scenarios
## System Configuration

<table>
<thead>
<tr>
<th>Hardware</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Platform</td>
<td>Wildcatpass S2600WT2</td>
</tr>
<tr>
<td>CPU</td>
<td>Intel(R) Xeon(R) CPU E5-2697 v3 @ 2.60GHz</td>
</tr>
<tr>
<td>Chipset</td>
<td>Intel® C610 series chipset (Wellsburg)</td>
</tr>
<tr>
<td>No of CPU</td>
<td>1</td>
</tr>
<tr>
<td>Cores per CPU</td>
<td>14 (HT Enabled. Total: 28)</td>
</tr>
<tr>
<td>LL CACHE</td>
<td>35840K</td>
</tr>
<tr>
<td>QPI/DMI</td>
<td>Auto</td>
</tr>
<tr>
<td>PCIe</td>
<td>Port3A and Port3C(x8)</td>
</tr>
<tr>
<td>MEMORY</td>
<td>Micron 16GB 1Rx4 PC4-2133MHz, 16GB per channel, 4 Channels, 64GB Total</td>
</tr>
<tr>
<td>NIC</td>
<td>2 x Intel® Ethernet X710-DAX2 Adapter (Total: 4 Ports)</td>
</tr>
<tr>
<td>NIC Mbps</td>
<td>10000</td>
</tr>
<tr>
<td>BIOS</td>
<td>Version: SE5C610.86B.01.01.0008.021120151325 &amp; Date: 02/11/2015</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td>Fedora 23</td>
</tr>
<tr>
<td>Kernel version</td>
<td>4.2.3-300.fc23.x86_64</td>
</tr>
<tr>
<td>Host</td>
<td>Hugepage size = 1G ; No. of Hugepages = 16</td>
</tr>
<tr>
<td>Machine</td>
<td>Hugpage size=2MB; No. of Hugepages = 2048</td>
</tr>
<tr>
<td>boot setting</td>
<td>isolcpus=1-9,21-29</td>
</tr>
<tr>
<td>Software version</td>
<td>DPDK 16.07</td>
</tr>
<tr>
<td>BIOS settings</td>
<td>firewall, iptables, Selinux, network Manager disabled</td>
</tr>
<tr>
<td></td>
<td>ip_forward = 0</td>
</tr>
<tr>
<td></td>
<td>set uncore frequency to the max ratio</td>
</tr>
<tr>
<td></td>
<td>kill -9 dhclient</td>
</tr>
<tr>
<td></td>
<td>rmmod ipmi_si ipmi_devintf ipmi_msghandler</td>
</tr>
<tr>
<td></td>
<td>lpc_ich bridge</td>
</tr>
<tr>
<td></td>
<td>sepci -s 00:03.0 184.l=1408</td>
</tr>
<tr>
<td></td>
<td>sepci -s 00:03.2 184.l=1408</td>
</tr>
<tr>
<td>IXIA TEST</td>
<td>RFC 2544 0% PACKET LOSS, 2 flows total/two ports</td>
</tr>
<tr>
<td>BIOS settings</td>
<td>P-state Disabled, C-State Disabled, HT ON and Turbo Boost Disabled</td>
</tr>
<tr>
<td>Fortville NIC</td>
<td>FW 4.33 API 1.2 NVM 04.04.02 eetrack 8000191c</td>
</tr>
<tr>
<td>FW Version</td>
<td>RFC 2544 0% PACKET LOSS, 4 total flows/4 ports</td>
</tr>
</tbody>
</table>

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit [www.intel.com/benchmarks](http://www.intel.com/benchmarks).

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804.
Phy-to-Phy Performance (1/2)

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Optimization Notice: Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804

Source: Irene Liew, Anbu Murugesan – Intel Network Packet Group
Phy-to-Phy Performance (2/2)

Data sources:
- BESS, Vanilla DPDK, VPP: measured on a 2.6GHz Xeon E5-2650 v2 machine
- OVS-DPDK: Intel ONP 2.1 Performance Test Report
- mSwitch: (link bottlenecked w/ large batch sizes @ 3.2GHz) Honda et al. "mSwitch: A Highly-Scalable, Modular Software Switch", SOSR 2015
Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Optimization Notice: Intel’s compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804
VM (Fedora 23)  VM (Fedora 23)

BESS x Phy-NF-Phy Performance

NIC (4 x 10G E - 40 Gigabit Ethernet)

Fedora 23 / KVM

Soft-switch (BESS, OVS-DPDK)

Source: Irene Liew, Anbarasan Murugesan – Intel NPG Architecture

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Optimization Notice: Intel’s compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice Revision #20110804
- BESS outperforms OVS-DPDK by a factor of 4-5x*

* Source: Intel ONP 2.1 Performance Test Report
Multi-Core / Thread Scalability

Source: Irene Liew, Anbarasan Murugesan – Intel NPG Architecture
Round-Trip Latency

RTT: 8.22us

- Increase of 0.60us
  (0.15us per BESS traverse)
BESS Architecture Overview (1/3)

BESS daemon (running in user space)
BESS Architecture Overview (2/3)

BESS daemon (running in user space)

DPDK PMD

Linux VMM

Host

Zerocopy

Container

DPDK VHost

Socket APP

Native APP

APP

APP

APP

APP

APP

APP

APP

OS

OS

VM

VM
BESS Architecture Overview (3/3)

BESS Controller

Socket APP  Native APP  APP  APP  APP
Linux VMM
Host  Zerocopy  Container  DPDK VHost

Legacy

DPDK PMD
External ports are interconnected with “modules” in a dataflow graph (like the Click modular router).

- You can compose modules to implement your own datapath.
- Developing a new module is easy.
Building Modules

• BPF filter
• Exact match table
• Wildcard match table
• Load balancer
• Encapsulation / decapsulation
• L2 forward
• IP Lookup
• 802.1q / 802.1ad
• Metadata
• ...

BPF('tcp port 80')

HashLB(L4)

Sink

Update
dst IP: 10.0.2.1

Simple L4 load balancer for HTTP
• Modules can tag metadata attributes to packets
  – Dynamic key-value table for each packet
  – 0-instruction access overhead, as compared to static “C struct”
  – Optimal usage of memory (cf. metadata bloat in Linux sk_buff)
  – Enables more decoupled and coherent design of modules
• BESS allows flexible scheduling policies for the data path
  – In terms of CPU utilization and bandwidth. Examples:
BESS allows flexible scheduling policies for the data path.

In terms of CPU utilization and bandwidth. Examples:
- BESS allows flexible scheduling policies for the data path.
  - In terms of CPU utilization and bandwidth. Examples:
BESS allows flexible scheduling policies for the data path.
  - In terms of CPU utilization and bandwidth. Examples:

  "Limit by 1 Mpps / 5 Gbps"
BESS allows flexible scheduling policies for the data path.
- In terms of CPU utilization and bandwidth. Examples:

“Should not consume more than 30% of CPU”
BESS Control Interface

Your global controller (optional)

Your server agent

libbess-python

JSON-like structured msg over socket

BESS daemon

REST, NETCONF, OpenFlow, ...

CLI

Configuration language

bessctl

Your server agent

Developer / operator

libbess

-python

CLI

Configuration language

bessctl

Your server agent

REST, NETCONF, OpenFlow, ...

Your global controller (optional)
BESS Control Interface

[2007] sangjin@c6:/tess/bessctl/conf/samples [develop] $ cat update.bess
import scapy.all as scapy

eth = scapy.Ether(src='02:1e:67:9f:4d:ae', dst='06:16:3e:1b:72:32')
ip = scapy.IP(src='192.168.1.1', dst='10.0.0.1')
udp = Ether() / IP() /Raw()

BESS Control Interface (2/5)
Controller / operator

Port initialization
BESS Control Interface (4/5)

Controller / operator

Module initialization

Dataflow setup
BESS Control Interface

Controller / operator

Module configuration (e.g., flow table update)

Port stats monitoring
BESS is an ideal vSwitch platform for NFV

- High performance
  - Sub-microsecond latency/jitter
  - Small packet 40Gbps throughput with only 1-2 cores
- Full flexibility and extensibility

Available on GitHub: https://github.com/netsys/bess

- Under BSD3 License
- ~34k lines in C and Python, supporting
  - Linux 3.x / 4.x (x86_64), DPDK 16.04
  - QEMU/KVM virtual machines, Docker/LXC containers
No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Intel technologies’ features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. **No computer system can be absolutely secure.** Check with your system manufacturer or retailer or learn more at intel.com.

© 2016 Intel Corporation. Intel, the Intel logo, Intel. Experience What’s Inside, and the Intel. Experience What’s Inside logo are trademarks of Intel. Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.
Questions?

Sangjin Han
sangjin@eecs.berkeley.edu

Christian Maciocco
christian.maciocco@intel.com