Portable Accelerated Networking in Multicore Embedded Systems

SICS Multicore Day 2014
2014-10-08

Ola Dahl
Product Management
Enea
• Enea

1968

Services

Middleware

OSE

Linux

~400 employees
~400 MSEK revenue
Products and Services

• Myself

Copyright © 2014 Enea AB
Why?

Exabytes per Month

<table>
<thead>
<tr>
<th>Year</th>
<th>Exabytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>1.5 EB</td>
</tr>
<tr>
<td>2014</td>
<td>2.6 EB</td>
</tr>
<tr>
<td>2015</td>
<td>4.4 EB</td>
</tr>
<tr>
<td>2016</td>
<td>7.0 EB</td>
</tr>
<tr>
<td>2017</td>
<td>10.8 EB</td>
</tr>
<tr>
<td>2018</td>
<td>15.9 EB</td>
</tr>
</tbody>
</table>

61% CAGR 2013-2018

Source: Cisco VNI Mobile, 2014

Increasing amount of data
Increasing requirements on QoS

Hardware needs to change

Software needs to change

*Networking, data plane
What if

Software was portable

and

Software could utilize (and adapt to changes in) hardware acceleration
Even better

What if software was open source

AND

It was not controlled by a single company
Enter ODP

OpenDataPlane – http://opendataplane.org

Linaro Networking Group

Open source

An API for portable, accelerated, multicore networking

Reference implementation on Linux

Open development of reference implementation(s)

ODP implementations for different SoCs, from different vendors
Linaro Organization

- Founded in 2010 with 6 members*, now 28
- >200 OSS engineers distributed globally, $50M annual revenue

*ARM, IBM, Freescale, Samsung, ST-Ericsson, TI
Linaro Networking Group (LNG)

- Formed in February 2013
- Working on core open-source software for networking
  - Real Time Support
  - Virtualization for networking
  - Core isolation
  - Data Plane interfaces & frameworks
  - Big-endian legacy code support
- Developing cross-platform OpenDataPlane framework

http://wiki.linaro.org/LNG
ODP – conceptual model*

 Logical View of Networking System


Copyright © 2014 Enea AB
Hello ODP

sudo apt-get install autoconf libtool libssl-dev

git clone https://git.linaro.org/lnq/odp.git

cd odp

./bootstrap

./configure --prefix=$HOME/sw_install/odp

make

make install

cd example/odp_example/

./odp_example
Hello ODP – code snippets

* SPDX-License-Identifier: BSD-3-Clause

/* ODP main header */
#include <odp.h>

#define MAX_WORKERS 32 /**< Max worker threads */
#define MSG_POOL_SIZE (4*1024*1024) /**< Message pool size */

/** @private Barrier for test synchronisation */
static odp_barrier_t test_barrier;

buf = odp_queue_deq(queue);

buf = odp_schedule(&queue, ODP_SCHED_WAIT);

*source code at https://git.linaro.org/lng/odp.git/blob/HEAD:/example/odp_example/odp_example.c

Copyright © 2014 Enea AB
Hello ODP – code snippets*

* @internal Worker thread
static void *run_thread(void *arg)

thr = odp_thread_id();
printf("Thread %i starts on core %i\n", thr, odp_thread_core());

odp_barrier_sync(&test_barrier);

int main(int argc, char *argv[])

/*------------------------------------------------*/
Create and launch worker threads */
odph_linux_pthread_create(thread_tbl, num_workers, first_core,
run_thread, NULL);

Here we use threads – ODP can be used also in "bare-metal" mode (on data-plane cores)

*source code at https://git.linaro.org/lng/odp.git/blob/HEAD:/example/odp_example/odp_example.c

Copyright © 2014 Enea AB
Packet Buffer

Each element is a descriptor, referring to a buffer, like

Copyright © 2014 Enea AB
ODP L2 forwarding example – code snippets*

```c
#define MAX_WORKERS 32

#define SHM_PKT_POOL_SIZE (512*2048)

#define SHM_PKT_POOL_BUF_SIZE 1856

/* Use schedule to get buf from any input queue */
buf = odp_schedule(NULL, ODP_SCHED_WAIT);

pkt = odp_packet_from_buffer(buf);

/* Open a packet IO instance for this thread */
pktio = odp_pktio_open(args->srcif, pool);

pkts = odp_pktio_recv(thr_args->srcpktio, pkt_tbl, MAX_PKT_BURST);
```

*source code at https://git.linaro.org/lnf/odp.git/blob/HEAD:/example/l2fwd/odp_l2fwd.c

Copyright © 2014 Enea AB
ODP – Key concepts (cont.)

**Packet I/O** – representation of (physical and logical) network interfaces

**Timer** – send timer event buffer (descriptor) to a queue at a specified time

**Scheduler** – distribute work (e.g. packets, timer events) to cores – for the case of packet scheduling, the actual scheduling algorithm determines, for a given packet class, which core to send to

**Classification** – packet analysis and pattern matching, for the purpose of classifying packets into different categories

**Run-to-completion** – threads* are assumed to consume packets and process them - after processing is done the packets are delivered to a queue (or to a network interface) – all this is done without any context switches

*The ODP threads are assumed to execute in isolation, on the data plane assigned cores (e.g. by utilizing features available in Linux such as core isolation and NoHz Full, but also bare-metal operation – without OS on the data plane cores – is possible)
ODP – Run-time visualization

Queue groups

Queues

Queues can be dynamically created and added to and removed from queue groups

Cores/threads associated with queue groups

Scheduler

Control plane core

Cores can be dynamically added to and removed from queue groups

Queues can be dynamically created and added to and removed from queue groups

handlers associated with queues

IP-fwd
NAT
GTP-U
DPI
RoHC

Copyright © 2014 Enea AB
Can it be portable?

Portability vs Efficiency – is there a sweet spot?

* may be implemented with use of a vendor SDK
Does it work?
Several vendors are active
Is everything crystal clear?*

This patch provides API support for querying mac address of an interface

I think we want to stick with checkpatch as it is

yep, no problem, might be worth checking some of the other configure items

An ODP Release is a tag on the repository that denotes a specific set of (documented) functionality

Lets touch on this on next Tuesdays call

Here is a proposal for enhancing ODP crypto APIs

It's not only about the implementation. It's about the contract with the programmer

*quotes from http://lists.linaro.org/pipermail/lng-odp/
What about multicore?

Scheduling of work among cores – provides load balancing, and reduces the need of locking primitives

Thread affinity – one thread per data plane core

ODP atomic – atomic operations on data

ODP synchronization – locks, barriers

Utilizing Linux functionality, such as
- Core isolation – no other Linux activity (no other user processes, no kernel threads, no softirq execution, no tasklets, no worker threads), and no unwanted interrupts
- No Hz full – Linux functionality and additional patches
What about acceleration?

**Cisco DP on ODP - Block Diagram**

IPSec with HW acceleration
Demonstrated at LCU 2014
TI Keystone II hardware (also on Cavium, Freescale, Avago)
http://www.slideshare.net/linaroorg/lcu14-310-cisco-odp-v2
What's in the pipeline

ODP 1.0 – end 2014
Good enough for evaluation

ODP 2.0 – end 2015
Good enough for production
ODP and DPDK

ODP application layer

<table>
<thead>
<tr>
<th>ODP App</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODP App</td>
</tr>
<tr>
<td>ODP App</td>
</tr>
<tr>
<td>ODP App</td>
</tr>
</tbody>
</table>

ODP API

<table>
<thead>
<tr>
<th>ODP Implementation for Intel**</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intel Architecture</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ODP Implementation for SoC A*</th>
</tr>
</thead>
<tbody>
<tr>
<td>SoC A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ODP Implementation for SoC B*</th>
</tr>
</thead>
<tbody>
<tr>
<td>SoC B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ODP Implementation for SoC C*</th>
</tr>
</thead>
<tbody>
<tr>
<td>SoC C</td>
</tr>
</tbody>
</table>

* may be implemented with use of a vendor SDK

** is currently implemented with use of DPDK
Thank you

And here are some additional links

ODP software - http://www.opendataplane.org/downloads/

ODP docs - http://www.opendataplane.org/documentation/

ODP docs (alt.) - http://docs.opendataplane.org/

ODP arch doc git - https://git.linaro.org/lng/odp-architecture.git

Linaro Connect USA 2014 schedule (look for presentations about ODP – and other things as well) – http://lcu14.zerista.com/event?event_order=start&event_page=1&owner=other&owner_id=739721

ODP mailing list - http://www.opendataplane.org/mailing-list/