Technology Guide

Intel® Dynamic Load Balancer (Intel® DLB) - Accelerating Elephant Flow

Authors

Yunfeng Bi

Hongjun Ni

Zhijun Tang

Dong Wang

Pan Zhang

Tao Zhu

Mrittika Ganguli

1 Introduction

Software based network gateways are facing the challenge of elephant flow. In traditional Data Plane Development Kit (DPDK) based applications, single traffic flow can be processed by only one single CPU core. When the volume of the flow exceeds the processing capability of one single CPU core, packets drop will be inevitable.

The industry is working on resolving this challenge by various attempts. Leveraging software queues to distribute the packets to multiple CPU cores seems like a straightforward solution but due to the facts such as lock contention and core-to-core data transfer penalty, the solution has not been widely deployed.

This guide will explain how the Intel® Dynamic Load Balancer (Intel® DLB) integrated in the 4th Gen Intel® Xeon® Scalable processor is used to achieve hardware assisted core to core queue communication, which subsequentially enables coordination and collaboration of multiple CPU cores on elephant flow handling and linear scalability.

This document is intended for cloud service providers, or anyone looking to deal with elephant flow by using software and hardware co-design method in their network. The technologies enabled here can be used as a reference point for improving performance in any scenarios that need collaboration of multiple CPU cores or pipeline working models.

This document is part of the Network Transformation Experience Kits.

Table of Contents

Figures

Tables

Document Revision History

Revision	Date	Description
001	February 2023	Initial release.

1.1 Terminology

Table 1. Terminology

Abbreviation	Description
CPS	Connections-per-second
DPDK	Data Plane Development Kit
FDIR	Flow Director
HDSLB	High Density Scalable Load Balancer
Intel® DLB	Intel® Dynamic Load Balancer
NFV	Network Function Virtualization
PCI	Peripheral Component Interconnect
RSS	Receiver Side Scaling

1.2 Reference Documentation

Table 2. Reference Documents

Reference	Source
AVX-512 Overview	https://www.intel.com/content/www/us/en/architecture-and-technology/avx-512-overview.html
Intel® Ethernet 800 Series Network Adapter Overview	https://ark.intel.com/content/www/us/en/ark/products/series/184846/intel-ethernet-network-adapter-e810-series.html
Intel® Dynamic Load Balancer	http://doc.dpdk.org/guides/eventdevs/dlb2.html
VPP Wiki	https://wiki.fd.io/view/VPP

2 Overview

Load Balancer is widely used in the age of the Internet to ensure quality of service and security. However, there is a challenge to process the elephant flow. The current load balancer solutions that use one core for elephant flow processing cannot handle the elephant flow very efficiently. Some companies developed the software multi-core solution but were not able to obtain a good core scaling performance due to lock contention. Our customers expect the hardware accelerated solution for the elephant flow, which can get better scalability and performance.

We propose a brand-new solution based on Intel® DLB device to process the elephant flow and enhance the throughput of HDSLB.

2.1 Technology Description

2.1.1 High Density Scalable Load Balancer (HDSLB)

High Density Scalable Load Balancer (HDSLB) is a software L4 Load Balancer project. It aims at building an industry leading Load Balancer, reaching 150 Mpps Level throughput, 100 million Concurrent Connections, 10 million Level Connections-per-second (CPS) per node and linear scaling. It optimizes the key data path leveraging advanced Intel® architecture capabilities. Key features are deeply optimized with Intel's latest platforms (that is, 3rd Gen Intel® Xeon® Scalable processors, Intel® Xeon® D-1700 and D-2700 processors and 4th Gen Intel® Xeon® Scalable processors), covering Intel® AVX-512, Flow Director (FDIR), and Intel® DLB. It is delivered as a customer reference with best practice of deploying Load Balancer solutions on an Intel® architecture platform.

2.1.2 Intel® Dynamic Load Balancer

Intel® Dynamic Load Balancer (Intel® DLB) is a hardware managed system of queues and arbiters connecting producers and consumers. It accelerates software synchronization in multiplied Core-to-Core communication environment and provides several dynamic work distribution types that are critical for packet processing in communications and cloud business. It is a PCI device envisaged to live in the 4th Gen Intel Xeon Scalable processors and can interact with software running on cores.

There are four queue models in Intel® DLB.

Table 3. Intel DLB Queue Models

Queue model	Description
Direct Queue	Multiple producers and a single consumer
Unordered Queue	Multiple producers and multiple consumers, ignore the order of tasks
Ordered Queue	Multiple producers and multiple consumers, re-arrange the task in the original order after processing
Atomic Queue	Multiple producers and multiple consumers. Tasks are grouped according to certain rules and share the same resources in the group. The order of tasks in the group is concerned.

In our Intel® DLB accelerated HDSLB solution, ordered queue type is the choice since we need to keep the order of the packets.

2.1.3 Intel® Ethernet 800 Series Network Adapter

Intel® Ethernet Network Adapter E810-CQDA2 is a dual 100 Gbps port PCIe 4.0 network adapter designed for optimizing networking workloads including Network Function Virtualization (NFV). For more information about Intel® Ethernet Network Adapter E810, please refer to: https://ark.intel.com/content/www/us/en/ark/products/series/184846/intel-ethernet-network-adapter-e810-series.html

2.1.4 Elephant Flow Handling Mechanism

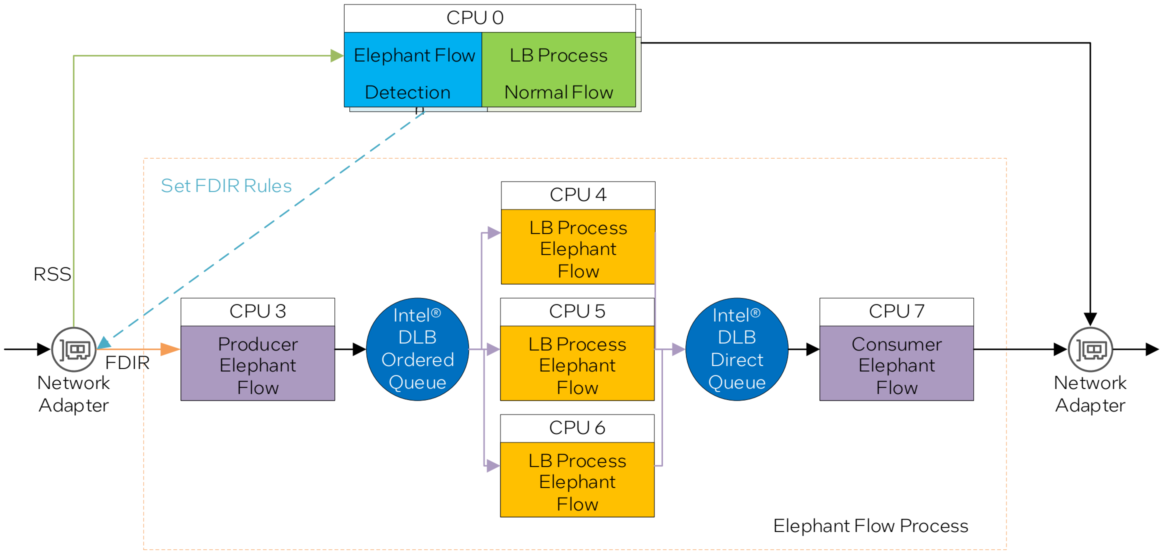

Figure 1. Intel® DLB accelerated HDSLB pipeline

From Figure 1, the incoming traffic from the network adapter is divided into two parts.

The general mice traffic and unmarked traffic are distributed to the top process through RSS (CPU 0 in Figure 1). In this process, there are two jobs to be performed. The first module is an elephant flow detection (as depicted in the blue box in Figure 1. From this module, we can detect if a flow is an elephant flow. If it is detected as an elephant flow, we will set an FDIR rule for this flow in the network adapter, and it will be redirected to a pre-specified queue when the next packet comes. If it not an elephant flow, then the flow will be processed in the second module in this pipeline, which is the load balance process (as depicted in the green box in Figure 1). The original workload is performed here, after which, the packet will be sent to the network adapter for the next stage.

When a marked flow (which is an elephant flow) comes, it will be redirected to the below pipeline through FDIR, and it will wake up the elephant flow process.

The CPU 3 on the left, as a producer, receives elephant flow from the last stage, and enqueue to the ordered queue. The ordered queue helps to record the order of packets, and then distribute to serval workers.

The CPU 4–6 in the middle of the picture are workers, three are drawn here, but can be added or removed according to the actual situation. We can use more cores to process an elephant flow, thereby increasing the processing capability of the system. The actual workloads are deployed on these cores.

The CPU 7 on the right, as a consumer, dequeues data from direct queue. With the help of Intel DLB, we can obtain the original sequence of packets, and then it sends the packet to the next stage.

3 Deployment

Table 4. System Setup

Item	Description
CPU	Intel® Xeon® Platinum 8471N @1.8GHz
Microcode	0x2b0000a1
Intel Turbo Boost	Disabled
Hyperthreading	Enabled
Memory	256GB (8x32GB 4800 MT/s [4800 MT/s])
Disk	1x 223.6G INTEL SSDSC2KB240G8
Operating System	Ubuntu 22.04 LTS
Network Adapter	Intel® Ethernet Network Adapter E810-CQDA2
BIOS	EGSDCRB1.SYS.8901.P01.2209200243
Linux Kernel Version	5.15.0-27-generic
ICE driver version	1.9.11
Software Version	HDSLB- 80aa6e659e04da6a704fd49f4ab7f738401581ab
DPDK Version	21.11.0
Compiler	Ninja 1.10.0 gcc (Ubuntu 11.2.0-19ubuntu1) 11.2.0
Test date	November 2022
Test by	yunfeng.bi@intel.com

3.1 Deployment Setup

To test the performance of the above-mentioned Intel DLB solution and the original HDSLB solution, we used a pre-production 4th Gen Intel Xeon Scalable processer server as our experimental platform, which connects to the IXIA* hardware traffic generator with two 100 GbE connection. This test setup is depicted in Figure 2. The HDSLB application established the connection with the two network adapter ports and the flows are processed in the application. We used two solutions with the same hardware setup.

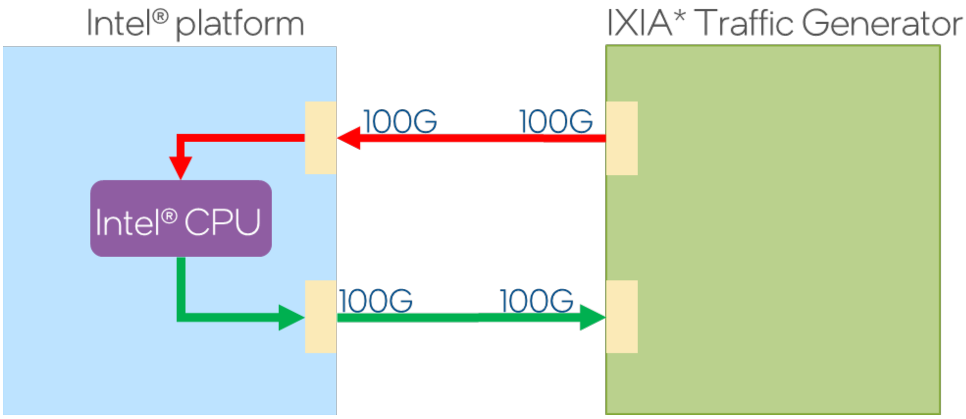

Figure 2. HDSLB Performance Test Setup Diagram

3.1.1 Network Adapter Ports and IXIA* Ports Connection

The system was equipped with two Intel® Ethernet Network Adapters E810-CQDA2, and uses one port per network adapter. All Ethernet ports used in this test were 100 Gigabit Ethernet (GbE) ports. As shown in Figure 2, the IXIA* traffic generator sends preconfigured flows to the first port and receives the return traffic from the other port.

3.1.2 IXIA* Flow Configuration

There is a single elephant flow, which has fixed IP, UDP, and MAC addresses. The packet length is fixed to 64 bytes in the small packets test, and 512 bytes in the big packets test. The throughput of the traffic is set to the line rate.

3.1.3 HDSLB Startup Configuration

The original HDSLB solution uses the config file to control the CPU resources, and the new Intel DLB part starts specified worker according to the parameters in header file. We use a single Intel DLB device, in which we enable one ordered queue and one direct queue.

4 Results

After the system setup is complete and HDSLB application starts up and runs with all commands injected, we started the IXIA traffic generator to transmit the flows and collected the data after the traffic is stable (for about 20 seconds). The packet loss is allowed during the switch in this test.

The test started from one worker and added one by one, until the maximum performance of the system is reached. The data is displayed in two parts, namely small packet performance and large packet performance. The horizontal coordinate of the chart is the number of worker cores. According to the section "Elephant Flow Handling Mechanism", the hardware solution requires two additional cores, so the total number of cores used for data processing is from 1+2 to 8+2.

We observed that the performance of the original HDSLB solution does not improve with more cores allocated, since it does not have the scalability function. The Intel DLB solution has more throughput and reaches a maximum number. CPU utilization is 100% during the running.

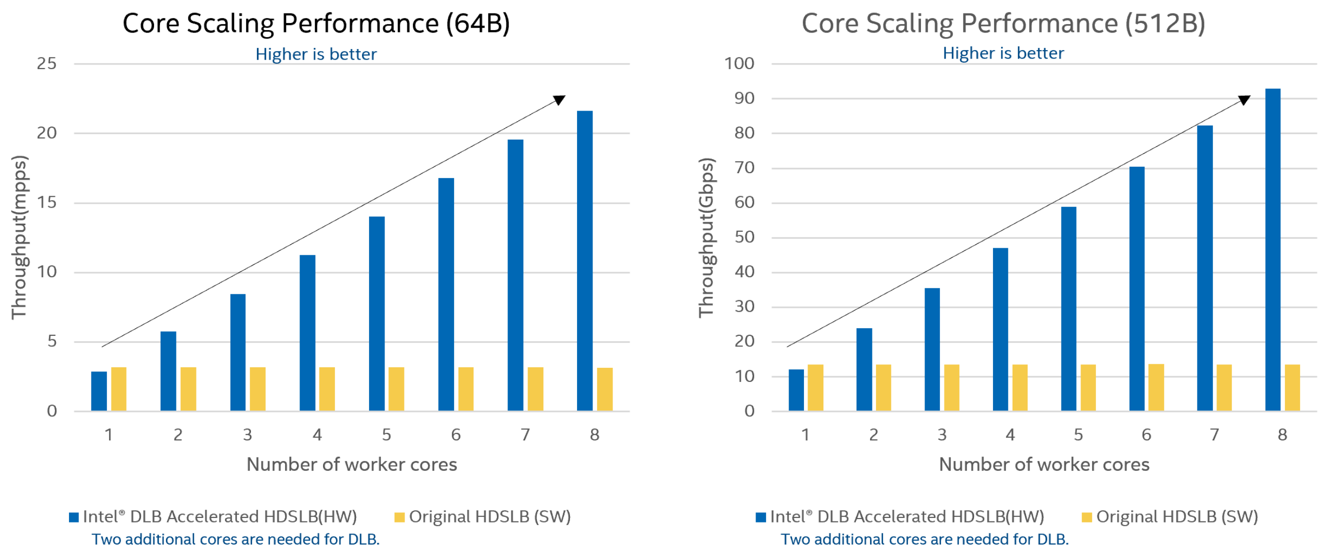

Figure 3. HDSLB performance

As shown in Figure 3, we can see that the Intel DLB accelerated HDSLB solution has less single core performance compared to the original HDSLB, for its extra workload to handle an Intel DLB device. On the other hand, the Intel DLB solution has great scalability, it gets higher performance than single core's limitation and reaches about 22 Mpps, which is up to 7x compared to the original HDSLB solution. As you can see from the two graphs, this data is not affected by the packet length, and it can reach about 90Gbps when packet length is 512 B. The linearity is good before the maximum number is reached, which is 8 cores in this test. When faced with heavier workloads, fewer packets can be handled by a single core, then the hardware solution, due to its scalability, using more worker cores, will make the total throughput more advantageous.

The performance is consistent with different packet sizes and stays with or without the background mice flows since Intel DLB solution has kept the original pipeline to process the mice flow.

5 Summary

This guide demonstrates how the Intel DLB technology provided in the latest 4th Gen Intel Xeon Scalable processors is leveraged on the elephant flow handling and linear scalability, to reach excellent CPU core scaling and achieve a maximum of up to 22 Mpps throughput for a single flow.

This guide has detailed the challenges faced, the underlying technologies, and the software design to achieve this scalability. While the software design and configuration are specific to elephant flow handling, the technologies enabled by Intel DLB accelerator and software framework are intended to be used as a reference point for improving performance and scalability in any scenario that needs the collaboration of multiple CPU cores.

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

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.

Intel technologies may require enabled hardware, software or service activation.

Intel does not control or audit third-party data. You should consult other sources to evaluate accuracy.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

0223/DN/WIT/PDF 769116-001US