4th Gen Intel® Xeon® Processors On-Die Accelerators: Intel® Dynamic Load Balancer (Intel® DLB)
Contents

1. Intel Dynamic Load Balancer in Sapphire Rapids
2. Intel DLB Technology & Value Proposition
3. Applications & Performance
Accelerators on 4th Gen Intel Xeon
4\textsuperscript{th} Gen Intel® Xeon® Processor Accelerator Architecture

- Intel DLB, Intel® Data Streaming Accelerator (Intel® DSA), Intel® QuickAssist Technology (Intel®QAT), and Intel® In-Memory Analytics Accelerator (Intel® IAA) sit on a “Data Accelerator Complex” (DAC) outside the CPUs CHA, LLC, and core mesh.

- Intel® Advanced Matrix Extensions (Intel® AMX) physically sits on each core.
4th Gen Intel® Xeon® Processor Data Accelerator Complex (XCC)

Data Accelerator Complex: Houses Intel DSA, Intel IAA, Intel QAT, and Intel DLB
- Accelerators are PCIe root-complex integrated end point devices.
- XCC clusters 1x Intel DSA, 1x Intel IAA, 1x Intel QAT and 1x Intel DLB device into a single DAC, and instantiates that DAC 4 times.
- Each accelerator is designed to operate independent of one another.
- CMI is the common path used by accelerators to access upstream CPU cache and memory, and generate PCIe messages.
- Accesses from downstream SW to the accelerators will also use CMI.
- Total CMI BW in each DAC is approximately equivalent to PCIe5 BW of 50GBs.
- Each DAC has a separate and dedicated CMI.
4th Gen Intel® Xeon® Processor Data Accelerator Complex (MCC)

Data Accelerator Complex: Houses Intel DSA, Intel IAA, Intel QAT, and Intel DLB

- Accelerators are PCIe root-complex integrated end point devices.
- MCC clusters 1x Intel DSA, 1x Intel IAA, 2x Intel QAT and 2x Intel DLB device into a single DAC, and instantiates that DAC once.
- Each accelerator is designed to operate independent of one another.
- CMI is the common path used by accelerators to access upstream CPU cache and memory, and generate PCIe messages.
- Accesses from downstream SW to the accelerators will also use CMI.
- Total CMI BW in each DAC is approximately equivalent to PCIe5 BW of 50GBs.
- Each DAC has a separate and dedicated CMI.
Intel Dynamic Load Balancer
Intel Dynamic Load Balancer

- HW-managed system of queues & arbiters linking producers and consumers
- Enables pipelined packet processing models for load balancing & packet queueing
- Enqueue & Dequeue capability from software
Intel Dynamic Load Balancer Overview

➢ What is it?
Dedicated HW (on-die PCIe root-complex endpoint devices) that provides intelligent dynamic, balanced distribution of network traffic across CPU cores.

➢ Why it matters to customers

TARGET WORKLOADS/USAGES
▪ Streaming Data Processing
▪ IPSec Gateway Handling
▪ VPP Router
▪ Elephant Flow Handling
▪ Restore order of network data packets processed simultaneously by CPU cores

Without Intel DLB: CPU Utilization Per Core

With Intel DLB: CPU Utilization Per Core

Load Balanced Processing
Queue Management : Before & After Intel DLB

SW Queue Management (Without Intel DLB)
- Queues and pointers stored in system memory
- Queues managed by software
- Requires lock to enqueue/dequeue
- Impacted by lock latency, memory latency, cache behaviors, polling multiple queues

With Intel DLB:
- Queues and pointers stored in DLB local memory
- Queues managed by hardware
- Improvement in number of operations per second
- More complex load balancing algorithms
- Better determinism, cycles freed on cores
Intel DLB Traffic Types

**Type: Direct**
- m->1 communication

**Type: Unordered**
- m->n load balanced across Cs

**Type: Ordered**
- m->n load balanced across workers with order recovery

**Type: Atomic**
- m->n load balancing with dynamic affinity. Allows Cons to operate on per-flow variables without atomics

---

**Non-Atomic Types**

**Order tracking**
- Reorder

**Queue per FlowID**
- Dynamic affinity

FlowID -> Consumer
## Intel Dynamic Load Balancer Performance Snap Shot

<table>
<thead>
<tr>
<th>Function</th>
<th>Business Value</th>
<th>Software Support</th>
<th>Use Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Dynamic redistribution of data load across cores when static NIC distribution causes a load-imbalance</td>
<td>• Improves system performance related to handling network data on multi-core Intel® Xeon® Scalable processors</td>
<td>• Intel® Data Mover Library</td>
<td>• IPSec security gateway, VPP router, UPF, vSwitch, Streaming data processing, Elephant flow handling</td>
</tr>
<tr>
<td></td>
<td>• Improved performance for distributed processing, dynamic load balancing and dynamic network processing reordering</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Performance gains

<table>
<thead>
<tr>
<th>Microservices</th>
<th>Performance gains vs not using these accelerators</th>
<th>Performance gains vs prior generation products</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Up to 96% lower latency at the same throughput with built-in Intel® DLB vs. software for Istio ingress gateway</td>
<td>Up to 89% lower latency and 57% lower CPU utilization at same core count with built-in Intel® DLB vs. prior generation</td>
</tr>
</tbody>
</table>

## Intel DLB vs SQM (Software Queue Manager)

<table>
<thead>
<tr>
<th>P-W Pairs</th>
<th>SQM (mpps)</th>
<th>DLB (mpps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1P1W</td>
<td>14.68</td>
<td>20.38</td>
</tr>
<tr>
<td>2P2W</td>
<td>15.07</td>
<td>36.22</td>
</tr>
<tr>
<td>3P3W</td>
<td>15.97</td>
<td>48.43</td>
</tr>
<tr>
<td>4P4W</td>
<td>16.01</td>
<td>63.10</td>
</tr>
<tr>
<td>5P5W</td>
<td>16.02</td>
<td>77.17</td>
</tr>
<tr>
<td>6P6W</td>
<td>15.91</td>
<td>89.49</td>
</tr>
<tr>
<td>7P7W</td>
<td>15.89</td>
<td>99.99</td>
</tr>
</tbody>
</table>

Results may vary. See Config 1
Intel DLB – Packet Application Value Drivers & Markets

Multi-function Pipeline (SD-WAN, Edge)

Atomic Load Balancing (Routers, NGFW, LB, Data Movement & Storage)

Elephant Flow Management (VPN/Security GW)

Packet Latency Bounding (Web Servers, Storage, Cloud)

Connect Stages w/ SW Rings

Connect stages with DLB

Without DLB: CPU Utilization Per Core

With DLB: CPU Utilization Per Core

All workers serve to minimize wait time

Load Balanced Network Processing
Intel DLB Processing Flow Summary

1. Intel DLB receives and stores a list of 8x 8byte pointers (QIDs) to packets in memory, via Rx core.

2. Intel DLB uses QIDs to direct associated packets to cores for execution, only targeting cores that, with these packets, will still operate at max performance and efficiency.

3. Tx cores transmit data back out once all packet instructions are complete.

LDB = Load balanced
DIR = Direct linked
DC = Distributor core

WC = Worker Core
CQ = Consumer Queue
WC2_1 = worker core #1 of stage #2

High priority traffic
Normal traffic
Mixed traffic
Intel DLB: How Does It Interface With Your WL?

**Linux kernel driver:**

- Provides Intel DLB device configuration, resource management, and Physical function (PF) – to – Virtual Function (VF) communication via ioctl, mmap, and sysfs interfaces.

**User-space libraries:**

- DPDK Poll Mode Driver (PMD)
  - Bifurcated PMD (shown): configuration requests via the kernel driver
  - PF PMD (not shown): configuration done by PMD
- **Libdlb** – user space library that runs on the top of dlb kernel driver.
Intel DLB: Under the Hood

- Thread writes HCW to PP with any transmitted packets and noting completion if required.
- QE, unless reordered, gets inserted into specified QID. Reordered QEs are held until order recovered.
- Eventually bubbles to top of QID, becomes available for scheduling.
- Scheduling done across CQs which have available space.
- For each such CQ, select only from up to 8 mapped QIDs. Both QID and QE priorities apply.
- QE entry written to CQ in memory.
Applications
Performance IPsec (Intel DLB Vs SW) - Gbps

Why it matters to customers

Higher Performance - More efficient load balancing translates into higher performance for packet throughput.

Packet flows across more than a core (An Elephant Flow) are effectively balanced across cores with the Intel DLB.

Note: The configuration consists of 1 HT core for producer, 1 HT core for consumer (both on a single physical core) and # worker cores (separate physical cores). SQM configuration consists of 1 EXTRA service core for scheduling.

Results using pre-production 4th Gen Intel® Xeon® Scalable Processor, systems, and software. Performance varies by part, use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex. See backup for workloads and configurations. Results may vary. All product plans, roadmaps, and performance are subject to change without notice.

Results may vary. See Config 2
Elephant Flow Management in Security Load Balancer

- **Why it matters to customers**: Higher Performance – More Efficient distribution of flows to workers results in higher performance vs SW solutions.

Intel DLB PoC with leading cloud vendors shows greater throughput with the addition of Intel DLB distributing packets across cores.

Results may vary. See Config 3.
Why it matters to customers

- Reduced Latency - Faster response times for TLS applications – Faster Content Delivery,
- More Connections per Sec = More Client traffic.
- Up to 1.4X improvement in average latency for DLB vs NO DLB on nginx for higher number of connections with operations >1 million across all object types.

Results may vary. See Config 4
Enabling
Where to get the SW & What is enabled?

- Releases are posted externally and internally.

Intel® Dynamic Load Balancer

**ID**  | **Date**  | **Version**  
--- | --- | ---  
686372  | 12/15/2022  | 8.0.0 (Latest)  

**Introduction**
This package contains the Intel® Dynamic Load Balancer Driver.

**Available Downloads**
- **Linux®**
  - Size: 696.8 KB
  - SHA1: 48F1B934F9B016E5D0F679443520B8B6B3FAE8E

Documentation
- User Guides
  - DLB_Driver_User_Guide.pdf
- README Text Files
  - DLB_Readme.txt

Intel DLB Value Proposition

Core Recovery
Elimination of Software Scheduler yields CPU cores back for application usage.

Performance
Finer work distribution allows for improved packet processing/data movement performance.

Determinism
Work can be fed to cores evenly and with an efficiency that provides fine granularity and controlled latencies for applications like data plane policing or packet processing pipelines.

Lower Latency
Finer tuned performance & work distribution provide for lower latencies in IPSec, PDCP, TLS protocols.
Notices & Disclaimers

Performance varies by use, configuration and other factors. Learn more on the Performance Index site.

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.

Your costs and results may vary.

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


© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.
Thank You
Performance Configurations
### System Config 1 - Eventdev

#### Platform Hardware
- **manufacturer**: Intel Corporation
- **product**: ArcherCity
- **sockets**: 1
- **cpu_model**: Genuine Intel(R) CPU 50000M@P
- **base_frequency**: 1000 MHz
- **maximum_frequency**: 1.001GHz
- **all-core_maximum_frequency**: 1.9GHz
- **cores_per_cpu**: 56
- **total_cpu_threads**: 512
- **tdp**: 300 watts
- **family**: 5
- **model**: 143
- **stepping**: 2
- **architecture**: Unknown Intel
- **dimm_config**: 1 x 32 GB <OUT OF SPEC> 4000 MT/s
- **pmem_firmware_version**: MemTotal 32636812 kB

#### Platform Config
- **microcode**: 0xe004200
- **bios_version**: EGSOCR81.86B.0058.D16.2104291438
- **bios_settings**: NUMA Nodes 1
- **NUMA disabled**: SNC
- **thread(s) per core**: 2
- **turbo**: Enabled
- **power & perf policy**: Performance
- **prefetchers**: DCU HW, DCU IP, L2 HW, L2 Ad

#### System Software
- **name**: System Config 1 - Eventdev
- **time**: Thu Jul 8 10:17:15 PM UTC 2021
- **os**: Ubuntu 21.04
- **kernel**: 5.11.0-18-generic
- **gcc_version**: gcc (Ubuntu 10.3.0-1ubuntu1) 10.3.0
- **glibc_version**: ld64 (Ubuntu GRIBC 2.33-0ubuntu5) 2.33
- **binutils_version**: GNU ld (GNU Binutils for Ubuntu) 2.36.1
- **frequency_driver**: acpi-cpufreq
- **frequency_governer_setting**: performance
- **pmem_node**: 1
- **huge_pages_total**: 8
- **huge_pages_size**: 1048576 kB
- **transparenthuge_pages**: madvisor
- **automatic_NUMA_balancing**: Disabled
- **irq_balance**: Enabled

#### CVEs
- **CVE-2017-5753**: OK
- **CVE-2017-5715**: OK
- **CVE-2017-5734**: OK
- **CVE-2018-3640**: OK
- **CVE-2018-3639**: OK
- **CVE-2018-3615**: OK
- **CVE-2018-3620**: OK
- **CVE-2018-3646**: OK

#### Driver(s)
- **NAME**: sda
- **model**: WDC_WD5500G280A-00SM50
- **size**: 465.8G
- **fstype**: ext4
- **rg-size**: 64
- **MIN-ID**: 512
- **NUMA node**: 0

- **NAME**: sda2
- **model**: 465.3G
- **fstype**: ext4
- **rg-size**: 64
- **MIN-ID**: 512
- **NUMA node**: 0

- **NAME**: sda1
- **model**: 512M
- **fstype**: ext4
- **rg-size**: 64
- **MIN-ID**: 512
- **NUMA node**: 0

#### NIC(s)
- **NAME**: enp1s0
- **model**: Ethernet Controller I225-LM
- **speed**: 1000M/s
- **link**: yes
- **driver_version**: 5.11.0-18-generic
- **firmware_version**: N/A
- **numa_node**: 0

- **NAME**: enp2pt1
- **model**: Ethernet Controller XXV710 for 25Gbe SPF28
- **speed**: Unknown
- **link**: no
- **driver_version**: 5.11.0-18-generic
- **firmware_version**: 6.01 0x80003554 1.1747.0
- **numa_node**: 0

#### GPU(s)
- **model**: N/A
- **version**: N/A
- **driver**: N/A

#### svr_info
- **version**: 1.2.4 2021-02-10 016F182 internal

---

2021.27.07 test date
# System Config 2 - IPSec

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>spr-dlb-ipsec-sys</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>Thu 11 Jan 2022</td>
</tr>
<tr>
<td><strong>Manufacturer</strong></td>
<td>Intel Corporation</td>
</tr>
<tr>
<td><strong>Product Name</strong></td>
<td>ArcherCity</td>
</tr>
<tr>
<td><strong>BIOS Version</strong></td>
<td>EGSDCRB1.86B.0062.D04.2107281526, ITP Tuning applied</td>
</tr>
<tr>
<td><strong>OS</strong></td>
<td>Ubuntu 20.04.3 LTS</td>
</tr>
<tr>
<td><strong>Kernel</strong></td>
<td>5.4.0-40-generic</td>
</tr>
<tr>
<td><strong>Microcode</strong></td>
<td>0x8d000260</td>
</tr>
<tr>
<td><strong>IRQ Balance</strong></td>
<td>Disabled</td>
</tr>
<tr>
<td><strong>CPU Model</strong></td>
<td>SPR D0, QDF: QYE3</td>
</tr>
<tr>
<td><strong>Base Frequency</strong></td>
<td>1.8GHz</td>
</tr>
<tr>
<td><strong>Maximum Frequency</strong></td>
<td>4.0GHz</td>
</tr>
<tr>
<td><strong>All-core Maximum Frequency</strong></td>
<td>2.7GHz</td>
</tr>
<tr>
<td><strong>CPU(s)</strong></td>
<td>120</td>
</tr>
<tr>
<td><strong>Thread(s) per Core</strong></td>
<td>2</td>
</tr>
<tr>
<td><strong>Core(s) per Socket</strong></td>
<td>60</td>
</tr>
<tr>
<td><strong>Socket(s)</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>NUMA Node(s)</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Prefetchers</strong></td>
<td>DCU HW, DCU IP, L2 HW, L2 Adj.</td>
</tr>
<tr>
<td><strong>Turbo</strong></td>
<td>Disabled</td>
</tr>
<tr>
<td><strong>PPIN(s)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Power &amp; Perf Policy</strong></td>
<td>Performance</td>
</tr>
<tr>
<td><strong>TDP</strong></td>
<td>350 watts</td>
</tr>
<tr>
<td><strong>Frequency Driver</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Frequency Governor</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Frequency (MHz)</strong></td>
<td>1800</td>
</tr>
<tr>
<td><strong>Max C-State</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong>Installed</strong></td>
<td>256GB (8x32GB &lt;OUT OF SPEC&gt; 4800MT/s [4800MT/s]) - Dual Rank</td>
</tr>
<tr>
<td><strong>Huge Pages Size</strong></td>
<td>1048576 kB</td>
</tr>
<tr>
<td><strong>Transparent Huge Pages</strong></td>
<td>Never</td>
</tr>
<tr>
<td><strong>Automatic NUMA Balancing</strong></td>
<td>Disabled</td>
</tr>
<tr>
<td><strong>NIC Summary</strong></td>
<td>Intel Corporation Ethernet Controller E810-C for QSFP (rev 02)</td>
</tr>
<tr>
<td><strong>Drive Summary</strong></td>
<td>2x100 GbE link, 1x 240G INTEL_SSD</td>
</tr>
<tr>
<td><strong>Workload</strong></td>
<td>DPDK* IPsec with Intel® Dynamic Load Balancer (DLB) and Software queue management (SQM)</td>
</tr>
<tr>
<td><strong>OS</strong></td>
<td>Ubuntu* 20.04.2 LTS</td>
</tr>
<tr>
<td><strong>Kernel</strong></td>
<td>5.4.0-40-generic</td>
</tr>
<tr>
<td><strong>Workload Version</strong></td>
<td>DPDK* IPsec-20.08</td>
</tr>
<tr>
<td><strong>Compiler</strong></td>
<td>gcc 9.3.0</td>
</tr>
<tr>
<td><strong>CPU Utilization (active cores)</strong></td>
<td>100%</td>
</tr>
</tbody>
</table>
System Config 3 - HDSLB

Name  | gtk-12-3-2
Time  | Thu Nov 3 01:38:06 PM UTC 2022
Manufacturer  | Intel Corporation
Product Name  | ArcherCity
BIOS Version  | EGSDORB1.SYS.901.P01.2209200243
OS  | Ubuntu 22.04 LTS
Kernel  | 5.15.0-27-generic
Microcode  | 0x2b0000a1
IRQ Balance  | Disabled
QDF/Stepping  | E5 Stepping
Base Frequency  | 1.8GHz
Maximum Frequency  | 3.6GHz
All-core Maximum Frequency  | 2.8GHz
CPU(s)  | 104
Thread(s) per Core  | 2
Core(s) per Socket  | 52
Socket(s)  | 1
NUMA Node(s)  | 1
Prefetchers  | L2 HW, L2 Adj, DCU HW, DCU IP
Turbo  | Disabled
PPIN(s)  | 28df8c9350f4a0f
Power & Perf Policy  | Performance
TDP  | 300 watts
Frequency Driver
Frequency Governor
Frequency (MHz)  | 3600
Max C-State  | 1
Installed Memory  | 256GB (8x32GB DDR5 4800 MT/s [4800 MT/s])
Huge Pages Size  | 1048576 kB
Transparent Huge Pages  | madvise
Automatic NUMA Balancing  | Disabled
NIC Summary  | 2x Ethernet Controller E810-C for QSFP
Drive Summary  | INTEL SSDSC2KB240G8

<table>
<thead>
<tr>
<th>Config1 (baseline)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workload &amp; version</td>
</tr>
<tr>
<td>Compiler</td>
</tr>
<tr>
<td>DPDK</td>
</tr>
<tr>
<td>CPU Utilization</td>
</tr>
<tr>
<td>Name</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>Time</td>
</tr>
<tr>
<td>Manufacturer</td>
</tr>
<tr>
<td>Product Name</td>
</tr>
<tr>
<td>BIOS Version</td>
</tr>
<tr>
<td>OS</td>
</tr>
<tr>
<td>Kernel</td>
</tr>
<tr>
<td>Microcode</td>
</tr>
<tr>
<td>IRQ Balance</td>
</tr>
<tr>
<td>QDF/Stepping</td>
</tr>
<tr>
<td>Base Frequency</td>
</tr>
<tr>
<td>Maximum Frequency</td>
</tr>
<tr>
<td>All-core Maximum Frequency</td>
</tr>
<tr>
<td>CPU(s)</td>
</tr>
<tr>
<td>Thread(s) per Core</td>
</tr>
<tr>
<td>Core(s) per Socket</td>
</tr>
<tr>
<td>Socket(s)</td>
</tr>
<tr>
<td>NUMA Node(s)</td>
</tr>
<tr>
<td>Turbo</td>
</tr>
<tr>
<td>PPIN(s)</td>
</tr>
<tr>
<td>Power &amp; Perf Policy</td>
</tr>
<tr>
<td>TDP</td>
</tr>
<tr>
<td>Frequency Driver</td>
</tr>
<tr>
<td>Frequency Governor</td>
</tr>
<tr>
<td>Frequency (MHz)</td>
</tr>
<tr>
<td>Max C-State</td>
</tr>
<tr>
<td>Installed Memory</td>
</tr>
<tr>
<td>Huge Pages Size</td>
</tr>
<tr>
<td>Transparent Huge Pages</td>
</tr>
<tr>
<td>Automatic NUMA Balancing</td>
</tr>
<tr>
<td>Drive Summary</td>
</tr>
</tbody>
</table>

**Config1 (baseline): CPU only**

<table>
<thead>
<tr>
<th>Workload &amp; version</th>
<th>nginx/1.16.1</th>
<th>nginx/1.16.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compiler</td>
<td>gcc (GCC) 8.5.0 20210514 (Red Hat 8.5.0-4)</td>
<td>gcc (GCC) 8.5.0 20210514 (Red Hat 8.5.0-4)</td>
</tr>
<tr>
<td>DPDK</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>wrk (client side)</td>
<td>wrk 4.2.0 [epoll] Copyright (C) 2012 Will Glozer</td>
<td>wrk 4.2.0 [epoll] Copyright (C) 2012 Will Glozer</td>
</tr>
<tr>
<td>BKC#</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>DLB Driver</td>
<td>NA</td>
<td>RELEASE_VER_7.4.0.V1</td>
</tr>
<tr>
<td>Iterations and result choice (median, average, min, max)</td>
<td>180sec, average of 3 runs</td>
<td>180sec</td>
</tr>
<tr>
<td>Raw Results (latency in ms)</td>
<td>3990</td>
<td>2720</td>
</tr>
<tr>
<td>Operating Frequency</td>
<td>2.72Ghz</td>
<td>2.72Ghz</td>
</tr>
<tr>
<td>CPU Utilization</td>
<td>100%</td>
<td>98%</td>
</tr>
</tbody>
</table>

**Config2 (new): with DLB**

<table>
<thead>
<tr>
<th>Workload &amp; version</th>
<th>nginx/1.16.1</th>
<th>nginx/1.16.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compiler</td>
<td>gcc (GCC) 8.5.0 20210514 (Red Hat 8.5.0-4)</td>
<td>gcc (GCC) 8.5.0 20210514 (Red Hat 8.5.0-4)</td>
</tr>
<tr>
<td>DPDK</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>wrk (client side)</td>
<td>wrk 4.2.0 [epoll] Copyright (C) 2012 Will Glozer</td>
<td>wrk 4.2.0 [epoll] Copyright (C) 2012 Will Glozer</td>
</tr>
<tr>
<td>BKC#</td>
<td>47</td>
<td>47</td>
</tr>
<tr>
<td>DLB Driver</td>
<td>NA</td>
<td>RELEASE_VER_7.4.0.V1</td>
</tr>
<tr>
<td>Iterations and result choice (median, average, min, max)</td>
<td>180sec, average of 3 runs</td>
<td>180sec</td>
</tr>
<tr>
<td>Raw Results (latency in ms)</td>
<td>3990</td>
<td>2720</td>
</tr>
<tr>
<td>Operating Frequency</td>
<td>2.72Ghz</td>
<td>2.72Ghz</td>
</tr>
<tr>
<td>CPU Utilization</td>
<td>100%</td>
<td>98%</td>
</tr>
</tbody>
</table>