第 4 世代インテル® Xeon® スケーラブル・プロセッサー上の DPDK およびインテル® データ・ストリーミング・アクセラレーターを使用した Open vSwitch* (OvS*) のチューニング ガイド

この記事は、インテル® デベロッパー・ゾーンに公開されている「Tuning Guide for Open vSwitch* (OvS) with DPDK* and Intel® Data Streaming Accelerator on 4th Gen Intel® Xeon® Scalable Processors」の日本語参考訳です。原文は更新される可能性があります。原文と翻訳文の内容が異なる場合は原文を優先してください。

はじめに

本ガイドは、Open vSwitch* (OvS*) と DPDK (データプレーン開発キット) を使用するユーザー向けです。ほとんどの状況で最高のパフォーマンスを達成するハードウェアとソフトウェアの設定に関する推奨事項を示します。ただし、OvS* と DPDK の展開方法はさまざまであり、特定のシナリオに合わせてこれらの設定を慎重に検討する必要があります。

Open vSwitch* は、ネットワーク・トラフィックのモニタリング、キューイング、シェーピングを行うマルチレイヤー・ソフトウェア・スイッチです。VLAN だけでなく、トンネル (VXLAN、Geneve) の分離やトラフィックのフィルタリングにも使用できます。Open vSwitch* は、VM 環境の仮想スイッチとして機能するのに適しています。DPDK は、パケット処理ワークロードを高速化するデータプレーン開発キットです。

第 4 世代インテル® Xeon® スケーラブル・プロセッサー・プラットフォームは、HPC、AI、ビッグデータ、ネットワークなど、さまざまなワークロードを高速化し、パフォーマンスと総所有コスト (TCO) 効率を高めるように最適化された、ユニークで拡張性のあるプラットフォームです。

このワークロードにおいて特に興味深い改良点は以下のとおりです。

- 1 ソケットあたり最大 56 コア、8 ソケット・プラットフォームでは最大 448 コアを搭載
- DDR5 によるメモリーの帯域幅と速度の向上 (対 DDR4)
- PCIe* 5.0 による最大 2 倍の I/O 帯域幅で、レイテンシーに敏感なワークロードに高いスループットを提供
- インテル® アドバンスト・ベクトル・エクステンション (インテル® AVX)
- インテル® データ・ストリーミング・アクセラレーター (インテル® DSA)

現在、多くの OvS* 展開では、vHost/virtio のような仮想インターフェイスを広範に使用して、仮想マシン (VM) との間でパケットを受け渡しています。これらの仮想インターフェイスは、VM とホスト上で動作する OvS* 間のメッセージ転送にパケットコピーを使用します。このようなシナリオでは、パケットコピーにかかる計算コストは、パケットサイズに比例して大きくなります。インテル® DSA は、ゲストまたは VM と OvS* 間の高コストの CPU パケットコピーを非同期にオフロードするのに使用されます。

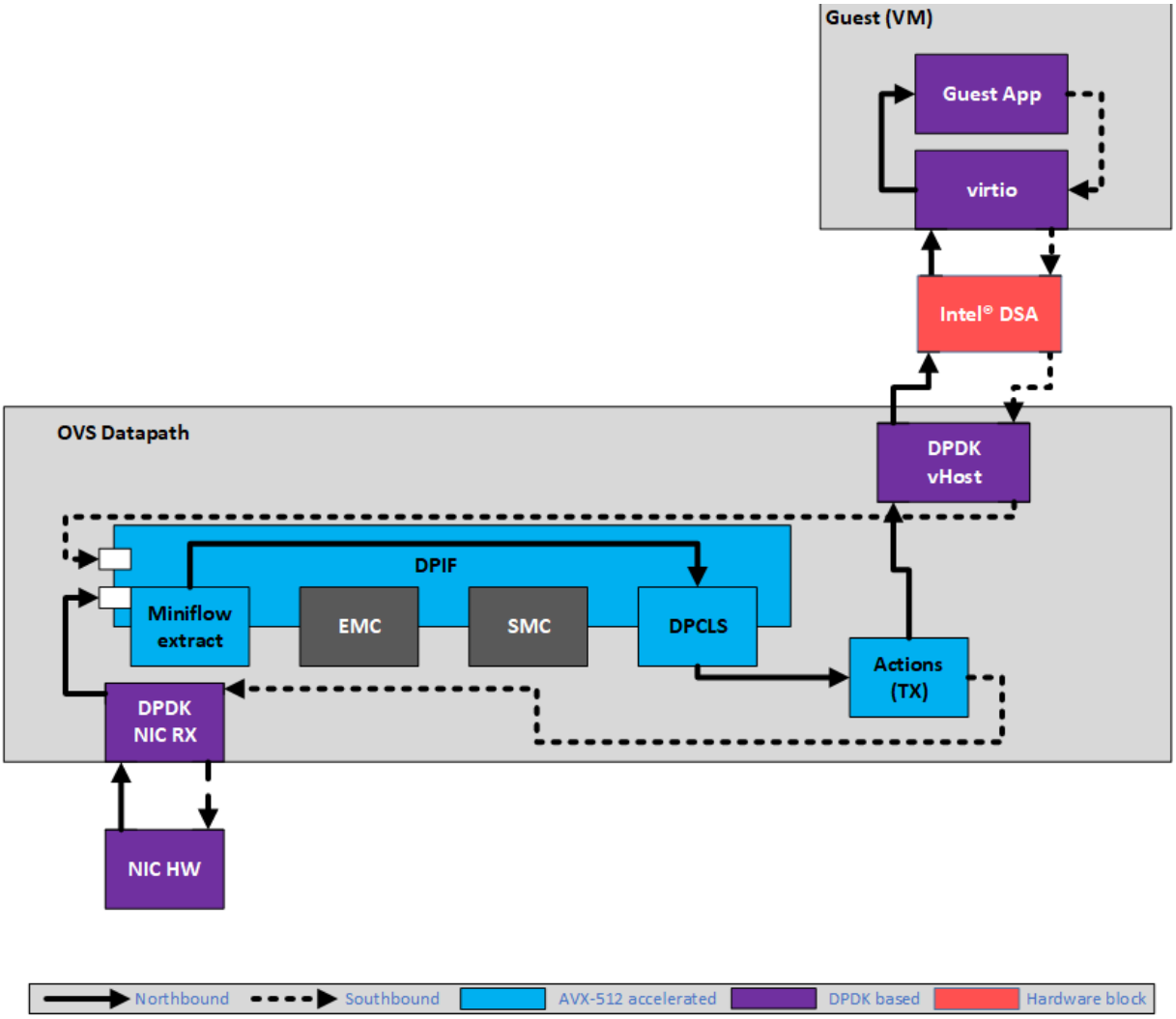

図 1: PVP (PHY-VM-PHY) セットアップでのインテル® DSA の使用

ソフトウェア構成

オペレーティング・システムのチューニングを含むソフトウェアの設定は不可欠です。汎用的なデフォルト設定では、特殊なワークロードで最適なパフォーマンスが得られる可能性はほとんどありません。

Linux* カーネルの最適化設定

インテル® DSA idxd ドライバーを使用するには、5.11 以降のカーネルが必要です。カーネル要件を満たせない場合、インテル® DSA デバイスを vfio-pci などのユーザー空間ドライバーにバインドできます。

Linux* ブート・パラメーターは次のとおりです。

ホストの設定

```
default_hugepagesz=1G hugepagesz=1G hugepages=<huge_pages> isolcpus=<core_list>
rcu_nocbs=<core_list> nohz_full=<core_list> intel_pstate=disable
processor.max_cstate=1 intel_idle.max_cstate=1 intel_iommu=on,sm_on iommu=on
nmi_watchdog=0 audit=0 nosoftlockup hpet=disable mce=off tsc=reliable
numa_balancing=disable workqueue.power_efficient=false
```

VM の設定

```
default_hugepagesz=1G hugepagesz=1G hugepages=<huge_pages> isolcpus=<core_list>
rcu_nocbs=<core_list> intel_pstate=disable processor.max_cstate=1
intel_idle.max_cstate=1 nmi_watchdog=0 audit=0 nosoftlockup
```

このワークロードに適用される一般的なパフォーマンスの推奨事項は、最新のパフォーマンス・レポート (英語) に記載されています。

必要なツールと情報

デバイスの有無

インテル® DSA は PCI デバイスとして列挙されているため、lspci を使用してその存在を確認できます。

```
lspci -vvv | grep "0b25"
```

出力は次のようになります。

```
6a:01.0 System peripheral: Intel Corporation Device 0b25
```

ここで、0b25 は、第 4 世代インテル® Xeon® プロセッサー上のインテル® DSA のデバイス ID です。この時点では pci-id データベースのエントリーがないため、デバイス ID を使用してインテル® DSA を照会します。

あるいは、DPDK の dpdk-devbind.py スクリプトを使用して、DPDK がサポートするプラットフォームで DMA が使用可能かどうかを確認することもできます。

```
$DPDK_DIR/usertools/dpdk-devbind.py --status-dev dma
```

出力は次のようになります。

```
DMA devices using kernel driver
===============================
0000:e7:01.0 'Device 0b25' drv=idxd unused=vfio-pci
```

accel-config

これは、カーネルの idxd ドライバーにバインドされたインテル® DSA インスタンスを設定するツールです。https://github.com/intel/idxd-config (英語)

例:

ドキュメントに従ってツールをビルドしてインストールした後、それぞれ 32 のキュー深度/バッチ記述子を持つ 4 つの専用ワークキュー (DWQ) で構成された 1 つのインテル® DSA インスタンスを設定するには、以下のコマンドを使用します。

```
accel-config disable-device dsa0
accel-config config-engine dsa0/engine0.0 --group-id=0
accel-config config-engine dsa0/engine0.1 --group-id=1
accel-config config-engine dsa0/engine0.2 --group-id=2
accel-config config-engine dsa0/engine0.3 --group-id=3
```

```
accel-config config-wq dsa0/wq0.0 --group-id=0 --mode=dedicated --priority=10 --
wq-size=32 --type=user --name=dpdk_ovs
accel-config config-wq dsa0/wq0.1 --group-id=1 --mode=dedicated --priority=10 --
wq-size=32 --type=user --name=dpdk_ovs
accel-config config-wq dsa0/wq0.2 --group-id=2 --mode=dedicated --priority=10 --
wq-size=32 --type=user --name=dpdk_ovs
accel-config config-wq dsa0/wq0.3 --group-id=3 --mode=dedicated --priority=10 --
wq-size=32 --type=user --name=dpdk_ovs
accel-config enable-device dsa0
accel-config enable-wq dsa0/wq0.0 dsa0/wq0.1 dsa0/wq0.2 dsa0/wq0.3
accel-config list
ls /dev/dsa # this command should output: wq0.0 wq0.1 wq0.2 wq0.3
```

注: accel-config を使用してインテル® DSA ワークキューを構成する場合、DPDK によって認識されるように、ワークキューにプリフィクス dpdk_ (上記の例のように) で始まる名前を付けることが重要です。

インテル® DSA Performance Micros (英語)

このツールを使用して、スループットや 1 秒あたりの操作数など、さまざまなインテル® DSA メトリックを検証します。インテル® DSA を OvS* やアプリケーションで使用する前に、このツールでテストすることを推奨します。

例:

ドキュメントに従ってツールをビルドし、上記のコマンドを使用してインテル® DSA を設定した後、以下のコマンドで 64B から 4KB までのパケットサイズのインテル® DSA のスループットを確認できます。

```
b=32;qd=32;n=`expr $b \* $qd`;for sz in 64 128 256 512 1k 1518 2k 4k; do
y=`sudo ./src/dsa_perf_micros -b$b -n$n -i1000 -jcf -s$sz -zD,D -
K[0]@dsa0,0,[1]@dsa0,1,[2]@dsa0,2,[3]@dsa0,3 |egrep "GB per sec"|cut -d ' ' -
f5,17`; echo $sz $y;done
```

同様に、vfio-pci/ユーザー空間ドライバーによってデバイスにバインドされている場合は、以下のコマンドを使用します。

```
b=32;qd=32;n=`expr $b \* $qd`;for sz in 64 128 256 512 1k 1518 2k 4k; do
y=`sudo ./src/dsa_perf_micros -b$b -n$n -i1000 -jcf -s$sz -zD,D -u  |egrep "GB
per sec"|cut -d ' ' -f5,17`; echo $sz $y;done
```

上記のオプションの詳細は、ツールのドキュメントまたはコマンドライン・ヘルプを参照してください。

インテル® DSA の使用方法の詳細は、『インテル® データ・ストリーミング・アクセラレーター (インテル® DSA) ユーザーガイド』(英語) を参照してください。

インテル® DSA 対応の DPDK

インテル® DSA ドライバーサポートは、DPDK 21.11 以降で dmadev デバイス (英語) として追加されました。

DPDK DSA ドライバーに関する詳細は、http://doc.dpdk.org/guides/dmadevs/idxd.html (英語) を参照してください。

注: インテル® DSA ドライバーに対応した現在の DPDK は、専用ワークキュー (DWQ) のみをサポートしており、共有ワークキューはサポートしていません。

accel-config または vfio-pci などのユーザー空間ドライバーへのバインドによりインテル® DSA デバイスを設定した後、DPDK の dmafwd サンプル・アプリケーションを使ってパフォーマンスを検証することもできます。DPDK ライブラリーがすでにビルドされていると仮定して、以下のコマンドを使用して dmadfwd サンプル・アプリケーションをビルドし、実行します。

```
cd $DPDK_BUILD_DIR;
meson configure -Dexamples=dma
ninja
./examples/dpdk-dma -l <core_list> <traffic source> -- -i 1 -s 2048 -c hw -p 0x1
-q 1
```

<traffic source> は、NIC ポート/pcap デバイス/null デバイスなどです。

アプリケーションに関する詳細は、『サンプル・アプリケーション・ユーザー・ガイド』(英語) を参照してください。

注: dmadev ライブラリーはテレメトリーをサポートしているため、DPDK アプリケーションで使用する場合は、テレメトリー・インターフェイスを介してインテル® DSA デバイス情報と統計情報を取得できます。詳細については、「デバイス統計のクエリー」(英語) を参照してください。

コードの可用性

DPDK

DPDK リポジトリーは、次の場所にあります: https://github.com/istokes/dpdk/tree/dma-tracking (英語)。クローンして dma-tracking ブランチにいることを確認してください。

```
cd dpdk
git checkout dma-tracking
```

インテル® DSA を有効にするのに必要なビルドオプションや変更はないため、通常どおり DPDK をビルドします。

Open vSwitch* (OvS*)

OvS* リポジトリー (英語) を探してクローンします。

クローン後、dpdk-dma-tracking ブランチにいることを確認してください。

```
cd ovs
git checkout dpdk-dma-tracking
```

インテル® DSA を有効にするのに必要なビルドオプションや変更はないため、通常どおり OvS* をビルドします。

注: この機能で OvS* で使用される DPDK vhost および dmadev API は、現在 experimental (実験的) とマークされています。

OvS* で DPDK を使用してインテル® DSA を有効にする

OvS* でインテル® DSA を有効にするには、以下のコマンドを使用します。

```
ovs-vsctl --no-wait set Open_vSwitch . other_config:vhost-async-support=true
```

- これを有効にするには、ovs-vswitchd デーモンを再起動する必要があります。
- デフォルト値は false です。
- OvS* を起動する前に、インテル® DMA デバイスを vfio-pci などのユーザー空間ドライバーにバインドするか、DPDK ドライバーのドキュメントに記載されている機能を使って、DPDK から見えるようにしておく必要があります。
- 現在の割り当てモデルは、OvS* のデータプレーン・スレッドごとに 1 つの DPDK dmadev デバイスを割り当てます。スレッドにインテル® DMA デバイスが見つからない場合、CPU コピーにフォールバックしますが、パフォーマンスに影響する可能性があります。したがって、OvS* を起動する前に、インテル® DMA デバイスが利用可能であることを確認することを推奨します。
- dmadev デバイスのデータプレーン・スレッドの NUMA 割り当ては、(dmadev-id に基づいて) 利用可能な dmadev デバイスを繰り返し処理することで自動的に行われ、デバイスごとにサポートされている vchannel は 1 つだけであると想定されます。
- dmadev 設計/アーキテクチャーの詳細は、DPDK ドキュメント (英語) を参照してください。

パフォーマンス・チューニング

小さなパケット向けに CPU コピーを最適化

インテル® DSA のコピーは、大きなパケットでは CPU に比べて高いパフォーマンスを提供することが確認されていますが、小さなパケットでは記述子を作成して完了を待つコストが、CPU でコピーを実行する時間に比べて大きくなります。そのため、小さなパケットのコピーは、CPU で実行したほうが高速な場合もありますが、パケットの順序を保持しながら、大きなパケットにはインテル® DSA を、小さなパケットには CPU を使用することで、両方のモードの利点が得られます。

以下のコマンドを使用して、パケットを vHost ライブラリーによって dmadev にオフロードするしきい値 (バイト単位) を実行時に選択することが可能です。

```
ovs-appctl netdev-dpdk/set-vhost-async-thresh <threshold_size>
```

デフォルトのしきい値は 128 バイトに設定されています。

同様に、以下のコマンドを使用して、現在のしきい値を取得することもできます。

```
ovs-appctl netdev-dpdk/get-vhost-async-thresh
```

VFIO-PCI またはカーネルのインテル® DSA ドライバー

実験結果から、第 4 世代インテル® Xeon® プロセッサーのインテル® DSA 1.0 の場合、このワークロードでは、IDXD カーネルドライバーをデバイスにバインドするよりも、vfio-pci ドライバーをインテル® DSA デバイスにバインドすることを推奨します。

インテル® アドバンスト・ベクトル・エクステンション 512 (インテル® AVX-512)

OvS* のパケット処理パイプラインのほとんど (Miniflow Extract、DPCLS、DPIF など) は、インテル® AVX-512 で最適化されています。これにより、vHost maddev オフロードの前処理サイクルが大幅に短縮され、dmadev エンキュー速度が向上し、OvS* 全体のスループットが向上します。したがって、CPU やプラットフォームが提供する機能を最大限に活用するには、dmadev オフロードとともにインテル® AVX-512 を有効にすることを強く推奨します。

以下のコマンドを使用して、OvS* と DPDK でインテル® AVX-512 サポートを有効にします。

ovs-vswitchd 起動前:

```
ovs-vsctl --no-wait set Open_vSwitch . other_config:dpdk-extra="--force-max-simd-bitwidth=512"
```

ovs-vswitchd 起動後:

```
ovs-appctl dpif-netdev/miniflow-parser-set study
ovs-appctl dpif-netdev/dpif-impl-set dpif_avx512
ovs-appctl dpif-netdev/subtable-lookup-prio-set avx512_gather 3
ovs-appctl odp-execute/action-impl-set avx512
```

上記のコマンドに関する詳細は、OvS* ドキュメント (英語) を参照してください。

テストでは、インテル® AVX-512 をインテル® DSA とともに有効にすると、パフォーマンスが大幅に向上することが確認されました。

出力のバッチ処理

バッチ記述子の使用率を向上することでスループットの向上につながる可能性があるため、特に分散トラフィックでは、tx-flush-interval オプションを使用して出力バッチのサイズを増やします。

```
ovs-vsctl set Open_vSwitch . other_config:tx-flush-interval=<time in microseconds>
```

テストでは、tx-flush-interval 値を 50 マイクロ秒に設定すると、良好なスループットの向上が得られました。

注: さらにスループットを向上すると、レイテンシーの増加を招く可能性があります。詳細は、OvS* ドキュメント (英語) を参照してください。

NUMA 対応

マルチ NUMA システムでは、すべての OvS* PMD コア、VM、NIC、インテル® DSA は同じ NUMA 上にあるべきです。OvS* は現在、PMD コアに基づく NUMA 対応のインテル® DSA 割り当てスキームを使用しています。同じ PMD コアの NUMA 上にインテル® DSA デバイスが見つからない場合、別の NUMA からインテル® DSA を割り当てようとします。この場合、OvS* は以下のような警告を出力します。

```
No available DMA device found on numa node <numa_id>, assigning DMA <dmadev_name> with dev id: <dmadev_id> on numa <numa_id> to pmd for vhost async copy offload.
```

インテル® DSA のワークキュー (WQ) 深度

インテル® DSA の DWQ で使用可能なバッチ記述子またはスロットが不足し、予備オプションとして CPU コピーが使用されると、スループットがわずかに低下することがあります。この動作は、インテル® DSA の WQ 深度が低いか、バッチあたりのパケット数が少ない場合に発生する可能性があり、インテル® DSA のバッチ記述子の使用が最適化されません。したがって、このようなシナリオを回避するため、WQ 深度を WQ ごとに 16/32 のような十分な値に設定することが推奨されます。vfio-pci によりバインドされたインテル® DSA デバイスの場合、DPDK はデフォルトで、デバイスごとに利用可能な WQ サイズの合計 (第 4 世代インテル® Xeon® スケーラブル・プロセッサーでは 128) に基づき、WQ 深度を均等に分割して割り当てます (第 4 世代インテル® Xeon® スケーラブル・プロセッサーでは 8 DWQ、したがって DWQ あたり 16 キュー深度)。

HugePage メモリー

DPDK ドキュメント (英語) に記載されているように、実験によると、インテル® DSA 上の DPDK を使用した OvS* では、2MB の HugePage よりも 1GB の HugePage を使用したほうがパフォーマンスは向上しました。

その他の考察

レイテンシーの影響

インテル® DMA オフロードを使用した固定パケットレートでの最大および平均レイテンシーは、大きなパケットサイズでは CPU よりも優れている可能性があります。また、ピーク NDR 帯域幅では、所定のパケットサイズに対して、DMA オフロードは CPU に比べてレイテンシーが高くなることがありますが、この場合、帯域幅の差は非常に大きい (DMA オフロード帯域幅のほうが高い) ことに注意してください。

パケット分割

セグメント化/チェーン化されたパケットでは、インテル® DSA へのオフロードコストが増加する可能性があります。例えば、mbuf サイズ (デフォルトは 2KB) を超えるパケットが複数の mbuf に分割される場合などです。この場合、パケットあたりのインテル® DSA のエンキュー数が増えるため、オフロードコストが増加し、パフォーマンスに影響する可能性があります。

用語

略語	説明
DPIF	DataPath InterFace (データパス・インターフェイス)。ソフトウェア・データパス全体を表現する OvS* データパス・コンポーネント。
DPCLS	Datapath Classifier (データパス分類器)。パケットのワイルドカード一致を行う OvS* ソフトウェア・データパス・コンポーネント。
D2K	Direct-to-UPI。
EMC	Exact Match Cache (完全一致キャッシュ)。パケットの完全一致を行う OvS* ソフトウェア・データパス・コンポーネント。
IP	Internet Protocol (インターネット・プロトコル)。
MFEX	Miniflow Extract (ミニフロー抽出)。
NIC	Network Interface Card (ネットワーク・インターフェイス・カード)。
NUMA	Non-Uniform Memory Access (不均等メモリーアクセス)。
NDR	No Drop Rate (ノー・ドロップ・レート)。
OvS*	Open vSwitch*
PMD	Poll Mode Driver (ポール・モード・ドライバー)。
SMC	Signature Match Cache (署名一致キャッシュ)。パケットのワイルドカード一致を行う OvS* ソフトウェア・データパス・コンポーネント。
SNC	Sub-NUMA Cluster (サブ NUMA クラスター)。
インテル® SST	インテル® スピード・セレクト・テクノロジー (インテル® SST)
インテル® SST-BF	インテル® スピード・セレクト・テクノロジー - ベース・フリークエンシー (インテル® SST-BF)
VXLAN	Virtual Extensible LAN (仮想拡張 LAN)。
VLAN	Virtual local area network (仮想 LAN)。
XPT	eXtended Prediction Table (拡張予測テーブル)。

関連情報

accel-config ユーティリティー・ライブラリー (英語)
DPDK dmadev ドキュメント (英語)
DPDK ドキュメント (英語)
インテル® Data Mover Library (インテル® DML) (英語)
インテル® DSA アーキテクチャー仕様 (英語)
インテル® DSA ドライバー GitHub* リポジトリー (英語)
インテル® DSA Performance Micros (英語)
IDXD カーネルドライバー (dmadev) ドキュメント (英語)
01.org オープンソース・ブログ (英語)
OvS* 最適化デプロイメント・ベンチマーク・テクノロジー・ガイド (英語)
OvS* ドキュメント (英語)