BigDL プライバシー保護マシンラーニングを使用したトラスト TorchServe

インテル® トラスト・ドメイン・エクステンションズとコンフィデンシャル・コンテナで AI をセキュアに

Qiyuan Gong、Dongjie Shi、Peter Zhu、Yabai Hu、Wesley Du、Hanyu Jin、Arron Wang、および Junrui Zhang インテル コーポレーション
Yao Zhang および Dian Chen ByteDance

デプロイメントは、モデルを現実世界のアプリケーションに導入する AI ライフサイクルの重要なステージです。TorchServe は、PyTorch* モデルを効率的かつ柔軟にデプロイするオープンソースのツールです。運用環境では、モデルとユーザーデータの機密性と整合性を重要事項として考慮する必要があります。この記事では、インテル® トラスト・ドメイン・エクステンションズ（インテル® TDX）、コンフィデンシャル・コンテナ（CoCo）、BigDL プライバシー保護マシンラーニング（PPML）上に構築された、トラスト PyTorch* モデル・サービング・ソリューションを紹介します。これらのテクノロジーは、パフォーマンスのオーバーヘッドを最小限に抑えながら、TorchServe の機密性と整合性を保護します。このソリューションのベスト・プラクティスとして、Jeddak サンドボックスも紹介します。

トラスト TorchServe ソリューション

コンフィデンシャル・コンピューティングは、処理中のデータの保護に重点を置いてセキュリティーとプライバシーを強化する手法です。信頼できる実行環境（TEE）は、ハードウェア・ベースのセキュリティー機能によりコンピューティング・システム内に分離された環境を作成するコンフィデンシャル・コンピューティングの基本的な構成要素です。最新の第 4 世代インテル® Xeon® スケーラブル・プロセッサーでは、インテル® TDX と呼ばれる特殊な TEE が提供され、トラストドメインと呼ばれる分離された仮想マシン（VM）内のデータを不正なソフトウェアから保護します。

CoCo（英語）は、ハードウェア TEE、リモート認証、関連ソフトウェア・テクノロジーを利用して、コンテナ化されたワークロードに機密性、整合性、信頼性を提供する Cloud Native Computing Foundation（CNCF）のサンドボックス・プロジェクトです。インテル® TDX コンフィデンシャル・コンテナ・ソリューションは CoCo ベースで、簡単に使用できます。アプリケーション開発者は、変更されていないコンテナイメージをインテル® TDX コンフィデンシャル・コンテナで実行できます。クラスターのオペレーター / 管理者は、共通のツールを使用して完全なソリューションをインストールし、Kubernetes* クラスターを管理できます。CoCo は、ランタイムだけでなくストレージやネットワーク上でも機密性の高い高価値のアプリケーション、データ、モデルに対するエンドツーエンドの保護を提供することにより、強力なセキュリティーを保証します（**図 1**）。

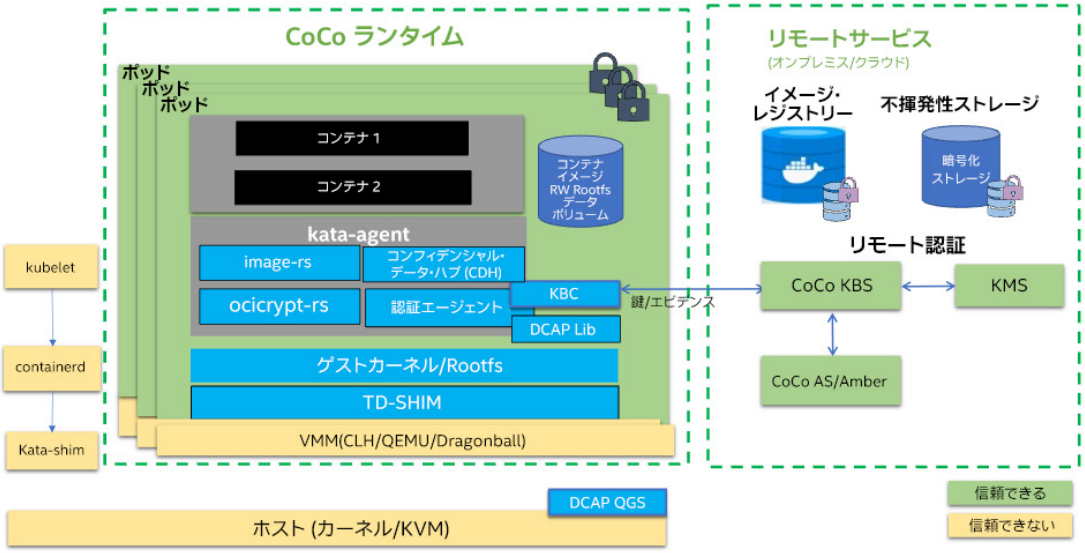

図 1. インテル® TDX コンフィデンシャル・コンテナ・アーキテクチャー

PPML（英語）は、エンドツーエンドの AI および分析アプリケーションのプライバシーとセキュリティーに対する新しい BigDL（英語）の機能です。インテル® ソフトウェア・ガード・エクステンションズ（インテル® SGX）、インテル® TDX、CoCo などの複数のセキュリティー・テクノロジーを組み合わせることにより、BigDL PPML を使用すると、ユーザーは既存のアプリケーションを変更することなく、ハードウェア保護とエンドツーエンドのセキュリティーを備えた状態で大規模に実行できます（**図 2**）。

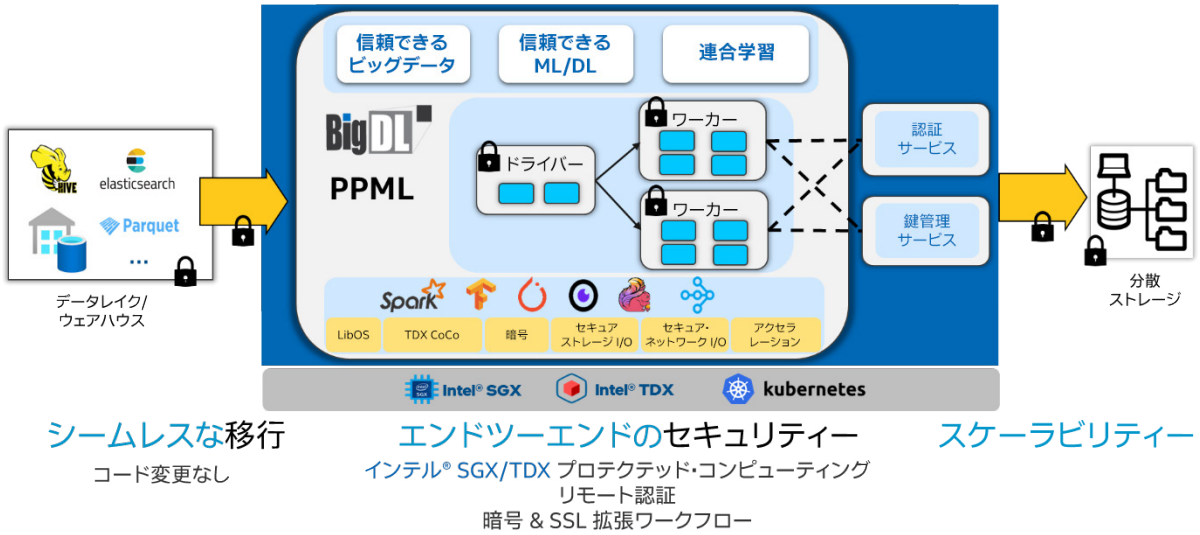

図 2. BigDL PPML アーキテクチャー

インテル® TDX CoCo をベースとし、BigDL PPML を使用して、Kubernetes* クラスターに簡単にデプロイできるトラスト TorchServe ソリューションを構築しました。**図 3** のように、次の 3 つの主要コンポーネントから構成されます。

- ロード可能なモデルが存在するモデルストア
- モデルを管理し、リクエストと応答を処理するフロントエンド
- 入力データを処理し、モデル推論を実行し、応答を生成するバックエンド

入出力データとモデルを保護するため、フロントエンドとバックエンドの両方がインテル® TDX CoCo 上で BigDL PPML を使用して実行されます。

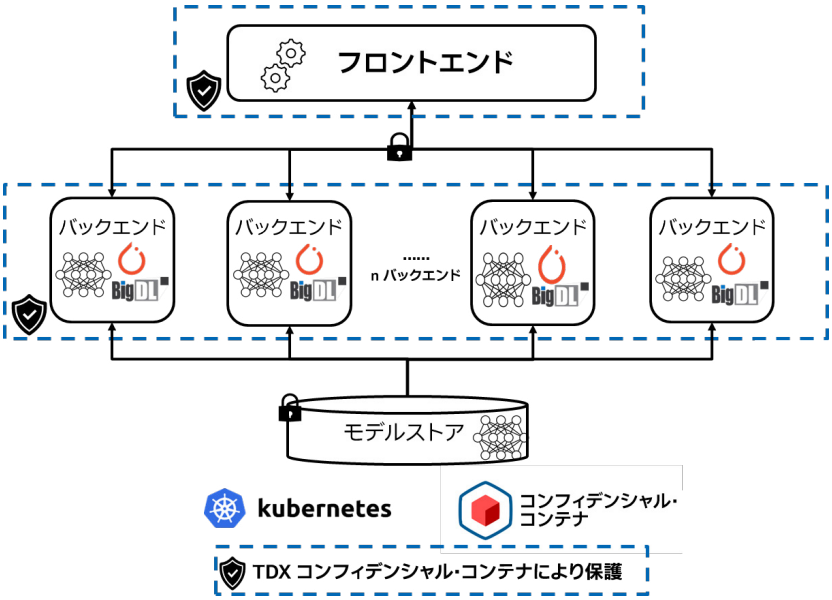

図 3. インテル® TDX コンフィデンシャル・コンテナと BigDL PPML 上のトラスト TorchServe

トラスト TorchServe ソリューションのワークフローは次のようになります。

1. モデルのオーナーが、モデルを準備してモデルストアに配置します。
2. モデルのオーナーが、CoCo ランタイムを使用してポッドを起動します。インテル® TDX により保護されたゲスト VM が作成され、VM 内でコンフィデンシャル・コンテナが開始します。
3. モデルのオーナーが、CoCo で TorchServe フロントエンド / バックエンドを開始します。
4. エンドユーザーが、トラスト TorchServe にサービングリクエストを送信し、推論の応答を受け取ります。

このワークフローでは、フロントエンドとバックエンドの両方がインテル® TDX CoCo により保護され、ユーザーモデルと入力データの機密性と整合性が保証されます。

セットアップ・ガイドおよびパフォーマンスのベスト・プラクティス

モデルを準備する

TorchServe でモデルをデプロイするには、特定の `mar` 形式のモデルファイル（すべてのモデルのアーティファクトを含む zip アーカイブ）が必要です。一般に、ユーザーはトレーニング済みモデル・チェックポイントとカスタム・モデル・ハンドラー・コードを準備し、TorchServe の torch-model-archiver ツールにフィードして、モデルファイルを生成する必要があります。インテル® プラットフォームのパフォーマンスを最大限に引き出せるように、BigDL Nano を使用してモデルを最適化することを推奨します。このツールは、ユーザーが PyTorch* 向けインテル® エクステンション（IPEX）、JIT、低精度のデータ型（BF16 や INT8 など）を含む、さまざまなソフトウェア最適化を活用するのに役立ちます。

環境をセットアップして設定する

トラスト TorchServe 環境をセットアップする前に、CoCo オペレーターが Kubernetes* クラスターにインストールされている（英語）ことを確認します。次に、トラスト TorchServe を実行するため、k8s ポッド内で CoCo を構成して開始します。フロントエンドとバックエンドのリソースを分離してパフォーマンスを向上するため、2 つのコンポーネントを異なるポッドに分割し、異なるセットの CPU コアを各ポッドに固定します。

Kubernetes* スタティック CPU マネージャー・ポリシーをセットアップして、kubelet 構成によるコンテナの CPU コアの固定を有効にします。

```
kubeReserved:
        cpu: "200m"
        memory: "1Gi"
systemReserved:
        cpu: "200m"
        memory: "1Gi"
cpuManagerPolicy: static
topologyManagerPolicy: best-effort
```

インテル® TDX CoCo でポッドを実行し、フロントエンドのポッドとバックエンドのポッドにリソースを割り当てるようにコンテナ・ランタイム・クラスを設定します。コンテナの CPU 固定を有効にするには、CPU リソースの `requests` と `limits` を整数の同じ値にする必要があることに注意してください。次に例を示します。

```
...
spec:
  runtimeClassName: kata-qemu-tdx
  containers:
  - name: torchserve-backend
    image: intelanalytics/bigdl-ppml-trusted-dl-serving-gramine-ref:multipods-8G
    imagePullPolicy: Always
...
    resources:
      limits:
        cpu: "48"
        memory: "64Gi"
      requests:
        cpu: "48"
        memory: "64Gi"
```

フロントエンドとバックエンドのゲスト VM のリソースを設定します。VCPU とメモリーリソースは、前のステップのポッドの設定と同じにする必要があることに注意してください。次に例を示します。

```
default_vcpus = 48
default_memory = 65536
```

TorchServe フロントエンドとバックエンドのポッドを起動します。

TorchServe を開始する

フロントエンドとバックエンドのポッドが起動され、コンフィデンシャル・コンテナがポッド内に自動的に作成されたら、対応するコンテナ内のフロントエンドのコンポーネントとバックエンドのコンポーネントを開始できます。マルチコアまたはマルチソケット・サーバーで最適なパフォーマンスを実現するには、複数のバックエンド・ワーカーを起動し、同じ NUMA ノードの CPU コアとメモリーを各バックエンド・ワーカーにバインドすることを推奨します。TorchServe 設定ファイルでバックエンドのワーカー番号を指定し、`numactl` ユーティリティーを使用してバックエンドのワーカープロセスを開始できます。

パフォーマンス・ベンチマーク

2 ソケットの第 4 世代インテル® Xeon® スケーラブル・プロセッサーを搭載したサーバー上で BERT-Large モデルの BigDL PPML を使用して、インテル® TDX CoCo 上でトラスト TorchServe のパフォーマンスを測定しました。ベンチマークでは、フロントエンドのポッドに 2 つのコアと 5GB メモリー、バックエンドのポッドに 48 個のコアと 64GB メモリーを割り当て、12 個のバックエンド・ワーカーを開始して、各ワーカーに 4 つのコアを固定しました。BigDL Nano を使用して、インテル® アドバンスト・マトリクス・エクステンション（インテル® AMX）アクセラレーションを利用した低精度推論によりモデル・サービング・パフォーマンスを最適化しました。FP32、BF16、INT8 精度のモデルサービングのエンドツーエンドのスループットを評価しました。インテル® TDX CoCo により保護されたトラスト TorchServe ソリューションのパフォーマンスは、保護されていない場合とほぼ同じです（**図 4**）。オーバーヘッドは通常 6% 未満です。

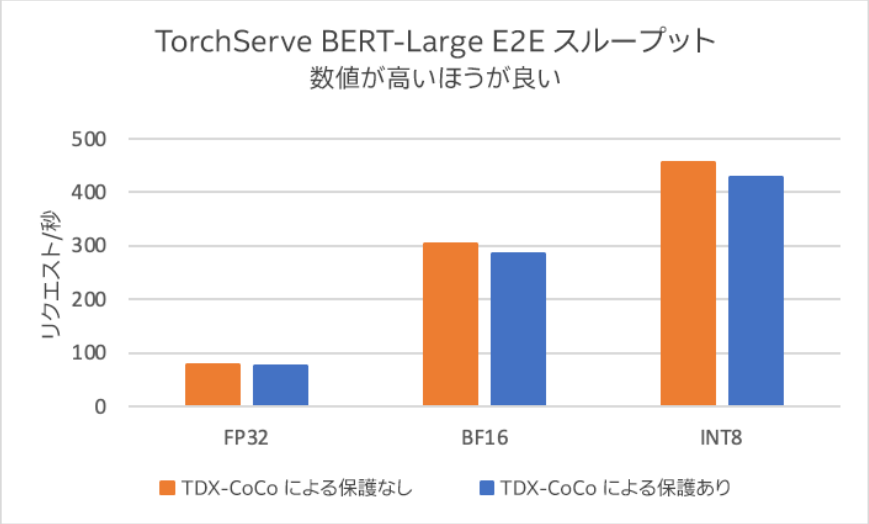

図 4. TorchServe BERT-Large のパフォーマンス

ByteDance のベスト・プラクティス

ByteDance は、社内および社外の企業ユーザーに安全かつ効率的なマシンラーニング・サービスを提供する Jeddak サンドボックスを発表しました。このソリューションは、ユーザーが AI シナリオにおけるさまざまなプライバシー・コンプライアンスの課題に対処するのを支援し、ユーザーがデータの価値を最大限に活用できるようにします。Jeddak サンドボックスは製品設計において BigDL と密接に連携し、そのセキュリティー機能とパフォーマンスの最適化を統合してユーザー・エクスペリエンスを向上させます（**図 5**）。

- BigDL Nano により提供される最適化手法を活用して、デプロイされたモデルを簡単に最適化し、モデル推論プロセスを加速して、ユーザー予測リクエストの応答レイテンシーを削減します。
- オンライン予測サービスのデプロイ方法としてインテル® TDX CoCo を採用しています。このアプローチは、サービングプロセス全体にハードウェア・ベースの保護を提供し、K8S の堅牢なコンテナ・オーケストレーション機能を組み合わせて、複雑で動的なリクエスト・ワークロードに対処するシームレスなスケーリングを実現します。

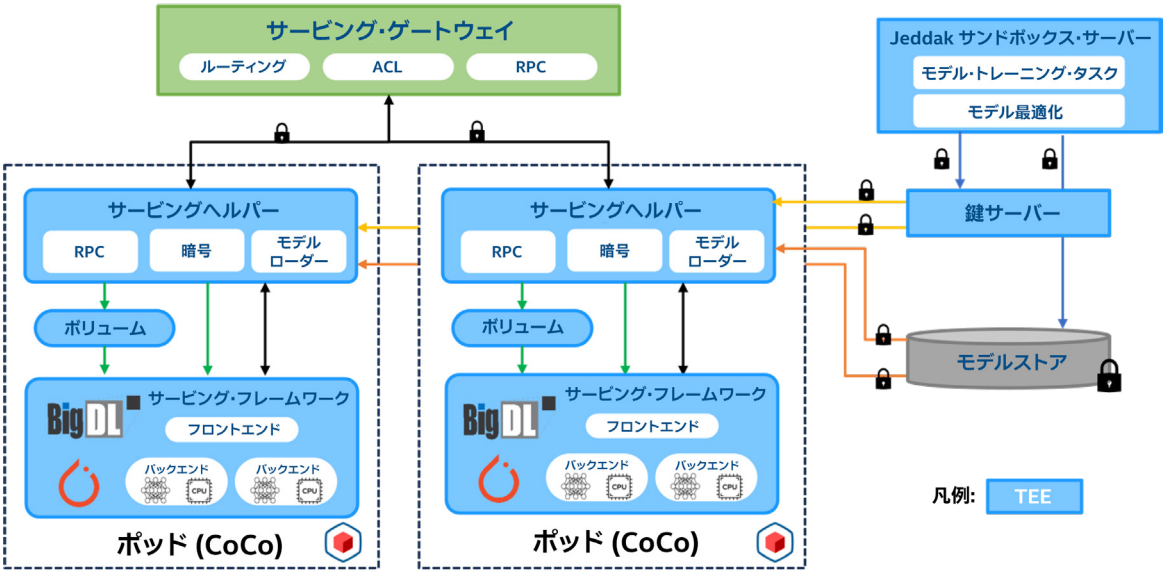

図 5. ByteDance の Jeddak サンドボックス・アーキテクチャー

Jeddak サンドボックスは、モデル予測プロセス中のさらなるセキュリティー強化を提供します。ユーザーのプライベート・データがハードウェアで保護された環境内でのみ復号化されることを保証する、ユーザーの推論リクエストのエンドツーエンド暗号化も含まれます。このソリューションにはモデルリクエストの認証および承認メカニズムも含まれており、ユーザーモデルへのアクセスを包括的に保護し、知的財産の漏洩を防ぎます。

まとめ

この記事では、サービング・コンポーネントが未承認のソフトウェアから保護されたトラストドメインに分離される、インテル® TDX 上の CoCo および BigDL PPML を利用したトラスト TorchServe ソリューションを紹介しました。インテル® TDX、CoCo、インテル® AMX アクセラレーションの長所を組み合わせることにより、ユーザーは、安全かつ柔軟でパフォーマンスの高い方法で PyTorch* モデルをデプロイできるようになります。また、トラスト TorchServe ソリューションを統合する ByteDance の Jeddak サンドボックスについても説明しました。今後も、モデルストアの暗号化 / 復号化や分散バックエンド・ワーカーのデプロイメントなどの新機能を継続的に追加し、モデル・サービング・パイプラインのエンドツーエンドのセキュリティーと柔軟性を強化します。

ハードウェアとソフトウェアの構成

ベンチマークは次のプラットフォームでインテルにより実施されました。

プラットフォーム	第 4 世代インテル® Xeon® スケーラブル・プロセッサー
サーバー	インテル® Xeon® Platinum 8475B プロセッサー インテル® ハイパースレッディング・テクノロジー無効 512GB DRAM
ホストシステムのソフトウェア	Ubuntu* 22.04.2 LTS カーネル 5.19.0-mvp15v2+4-generic
ゲストシステムのソフトウェア	Ubuntu* 20.04 カーネル vmlinuz-5.15-plus-TDX-102cc-tdx Qemu v7.1.0 Libvirt 8.6.0-2022.11.17.mvp2.el8 (tdxlibvirt-2022.11.17) TDVF mvp6-stable202211 TDX SEAM 1.0.01.01-mvp29.el8
クラウドネイティブのソフトウェア	Kubernetes* v1.24.0 CoCo v0.5.0
DL フレームワーク	TorchServe 0.7.1 PyTorch* 1.13.1 IPEX 1.13.100 Transformers 4.29.1 BigDL 2.4.0 snapshot
テスト実施日	2023 年 06 月 19 日