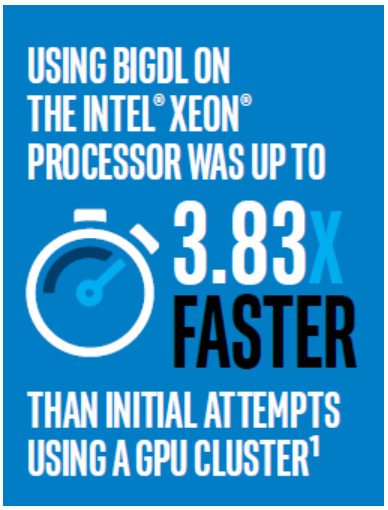
JD は GPU から CPU への変更により画像解析を高速化

この記事は、Tech.Decoded に公開されている「JD Speeds Up Image Analysis By Replacing GPUs with CPUs」の日本語参考訳です。

中国のクラウド・サービス・プロバイダー JD は、GPU を使用して膨大な製品画像データベースから情報を抽出しようとしていましたが、既存のインテル® Xeon® プロセッサー・ベースのサーバーを使用することでパフォーマンスを 3.83 倍に向上できました [1]。



概要

- ストレージと GPU 解析クラスター間のデータのコピーにかかる時間は合計プロセス時間の半分を占めていました。
- 当初の GPU を使用した高速化は開発および実行が困難でした。
- JD はインテル® Xeon® プロセッサー E5 ファミリーベースの Spark* クラスターで BigDL ライブラリーを使用して Caffe* モデルを実行しました。
- パフォーマンスは 3.83 倍になりました [1]。

JD は、中国有数の小売業者およびクラウド・サービス・プロバイダーです。製品画像のカタログには多くの情報が含まれており、それらの情報を解析できれば、ビジュアル検索や価格比較アプリケーションの基本情報として利用できます。JD は当初、GPU を使用して膨大な商品画像データベースを解析しようとしていましたが、管理が大変なことに加えて、ストレージとデータクラスター間のデータのコピー処理に時間がかかっていることが判明しました。そこで、インテルの協力の下、画像を格納するインテル® Xeon® プロセッサー・ベースのサーバーに BigDL ディープラーニング・ライブラリーを配備したところ、パフォーマンスを 3.83 倍に向上できました [1]。これは、JD が製品画像を新しいサービスの基本情報として使用する際の敏捷性を大きく高めることになりました。

課題

- JD のコンピューター、おもちゃ、衣類を含む、広範なカテゴリーにわたる、膨大な製品全体のカタログから画像の特徴抽出を行う。
- 膨大なデータベースにスケーリング可能な、パフォーマンス効率に優れた画像解析のインフラストラクチャーを作成する。
- 開発が容易で、新しい画像解析アプリケーションの作成に利用できるクラウド解析プラットフォームを構築する。

ソリューション

- JD は、データを格納しているインテル® Xeon® プロセッサー E5 ファミリーベースのサーバーに、BigDL を使用して既存の Caffe* モデルを配備しました。
- インフラストラクチャーは、新しいインテル® Xeon® プロセッサー・ベースのサーバーを追加することにより効率的にスケールアウトできます。
- Apache Hadoop* および Spark* フレームワークでリソース管理を行うことにより、パフォーマンス効率を低下させることなく、将来新しいアプリケーションを簡単に開発できます。

結果

- パフォーマンスは、当初の GPU ベースのソリューションと比較して 3.83 倍に向上しました [1]。
- JD は、内部利用およびパブリック・クラウド・サービス向けの新しいアプリケーションを簡単に作成できる、革新的なプラットフォームを確立しました。
- JD は、解析に既存のハードウェア資産を再利用することにより、別の GPU クラスターを利用した場合と比較して、ソリューションの総所有コストを削減できました。

効率的な画像解析を可能にする

JD にとって、クラウドはビジネスの基礎をなすものです。JD は、中国で有数の小売業者であり、オンライン販売のプラットフォームに加えて、パブリッククラウドも提供しています。小売側の提案により追加された機能がパブリッククラウドの顧客に提供されることもあります。

小売サイトで販売している膨大な製品のカタログには、数億の製品画像が含まれています。これらの画像は Apache HBase* に格納され、Hadoop* フレームワークのビッグデータに分散されます。JD は、異なる製品の画像の特徴をマッチしたいと考えていました。この機能が利用できれば、例えば、ビジュアル検索で、顧客は興味のある製品のスナップ写真を撮ることができ、JD は類似の商品を検索して表示できます。また、製品とほかの Web サイトの製品をマッチさせて、その製品に対抗する価格を設定することもできます。

JD のチームは、グラフィックス・プロセシング・ユニット (GPU) を使用してアプリケーションをマッチする機能を構築しようとしていましたが、GPU を適切にスケーリングしてデータベースを制御するのは困難なことが分かりました。JD は、マルチ GPU サーバーと GPU クラスターの両方を使用しようとしていましたが、クラスター設定でメモリーエラーが頻繁に発生し、GPU のメモリー不足が原因でプログラムがクラッシュしていました。クラスターの個々の GPU カードのリソース管理と割り当ては複雑であり、エラーが発生しやすいことが分かりました。マルチ GPU サーバーでは、JD の開発者がデータのパーティショニング、タスクのバランス、フォールトトレランスを手動で管理する必要がありました。また、CUDA* のような多くの依存性は、プロダクション環境への配備を困難にしていました。

GPU で画像処理を実行しながら、解析するデータを HBase* から GPU にコピーし、結果をコピーして戻すためにかかる時間が原因の遅延も発生していました。プロセスのこの部分は、特徴抽出の合計時間の約半分を占めていました。リソース管理、データ処理、フォールトトレランスをサポートするソフトウェア・フレームワークがなかったため、画像の前処理も課題でした。

JD は、スケーラブルで持続性のある方法で画像データベースの特徴抽出パイプラインをサポートするインフラストラクチャーを必要としていました。

BigDL を使用したスケーラブルなディープラーニング

JD は、インテル® Xeon® プロセッサー E5-2650 v4 ベースのサーバーを使用して CPU で特徴抽出ワークロードを処理するため、Apache Spark* で分散型ディープラーニング・ライブラリー、BigDL を使用しました。BigDL は、Scala または Python* を使用してスケーラブルな Spark* クラスターベースのディープラーニング・アプリケーションを作成し、数百あるいは数千のサーバーにスケールアウトできます。パフォーマンスを向上するため、BigDL はインテル® マス・カーネル・ライブラリー (インテル® MKL) と並列コンピューティングを利用してインテル® Xeon® プロセッサーの機能を活用しています。

BigDL を利用することで、GPU を使用して訓練していた Caffe* モデルを、画像を格納している既存の CPU アーキテクチャーに配備することができました。JD のアプリケーションでは、Single Shot MultiBox Detector (SSD) モデルを使用して画像のオブジェクトを検出した後、DeepBit モデルを使用してオブジェクトから特徴を抽出しています。

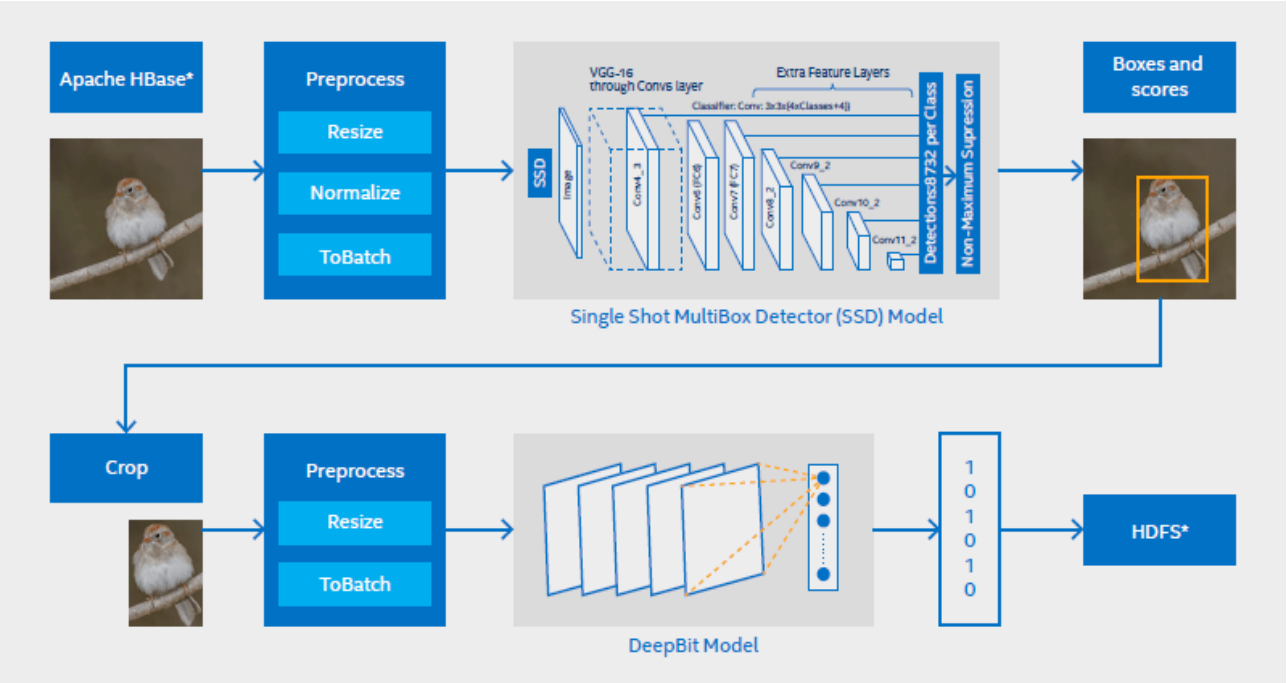


図 1 - JD の特徴抽出ワークフロー。BigDL を使用して SSD モデル (オブジェクト検出) と DeepBit モデル (特徴抽出) を管理しています。

ワークフローを次に示します (**図 1** も参照)。

1. HBase* から数億の画像を Resilient Distributed Datasets (RDD) として読み取ります。
2. BigDL を使用してこれらの画像を SSD モデル向けに前処理します (サイズ変更、正規化、バッチを含む)。BigDL は、一般的な変換と拡大をサポートする、OpenCV ベースの画像前処理ライブラリーを提供します。
3. Spark* 上で大規模な分散型オブジェクト検出を行う Single Shot MultiBox Detector (SSD) モデルをロードし、検出されたオブジェクトの座標と信頼度スコアを生成します。
4. 最もスコアが高いオブジェクトの座標で元の画像を切り取ります。
5. ターゲット画像を DeepBit モデル向けに前処理します (サイズ変更とバッチを含む)。
6. BigDL を使用して Spark* 上で分散型特徴抽出を行う DeepBit モデルをロードし、対応する特徴をベクトル float として生成します。
7. 抽出されたオブジェクトの特徴の RDD を Hadoop* Distributed File System (HDFS) に格納します。

1,200 の論理コアを搭載した高度な並列アーキテクチャーを使用して、データベースから画像データを読み取るプロセスは大幅に高速化され、全体的なパフォーマンスは 3.83 倍になりました [1]。

ソリューションには、JD の既存の CPU 資産である、インテル® Xeon® プロセッサー E5-2650 v4 (2.20GHz) を使用しました。24 の物理コアを搭載し、インテル® ハイパースレッディング・テクノロジー (インテル® HT テクノロジー) を有効にしたサーバーを、Apache Hadoop* Yet Another Resource Negotiator (YARN) クラスター管理テクノロジーを使用して 50 の論理コアをサポートするように設定しました。24 台のサーバーを使用したソリューション (合計 1,200 論理コア) は、高度な並列ワークフローを提供します。

BigDL を使用することにより、JD は、GPU を使用して訓練していたモデルを既存のインテル® Xeon® プロセッサー・ベースのサーバーで再利用することができました。追加の GPU カードが不要になり、GPU サーバーと同じ構成を CPU サーバーで実行できたため、特徴抽出を別の GPU クラスターで実行していた場合と比較して、コストは大幅に削減されました。また、BigDL ワークロードを夜間に処理し、昼間は別のタスクを実行することで、CPU クラスターの使用効率も高まりました。高度な並列データロードにより特徴抽出にかかる時間は大幅に短縮され、Spark* フレームワークを使用してリソース、フォールトトレランス、タスクのバランスを管理できるようになりました。

パフォーマンスが向上し、標準的なサーバーを追加してソリューションを簡単にスケールアウトできるようになったことで、JD は画像解析で大規模なデータセットを処理できるようになりました。

ソリューションのテクニカル・コンポーネント

- **BigDL**。BigDL は Apache Spark* 上で動作する分散型ディープラーニング・ライブラリーです。開発者は、Scala または Python* を使用して Spark* クラスター向けのディープラーニング・アプリケーションを作成できます。オープンソースで、データを格納した同じ Hadoop* または Spark* クラスター上でデータを解析できます。

- **インテル® Xeon® プロセッサー E5 ファミリー。**次世代のデータセンター設計のために開発されたインテル® Xeon® プロセッサー E5 ファミリーは、データセンターやクラウドの各種ワークロードを柔軟に処理します。
- **インテル® マス・カーネル・ライブラリー (インテル® MKL)。**インテル® MKL は、最小限の労力で将来のインテル® プロセッサー向けにコードを最適化できます。各プロセッサー・ファミリーでパフォーマンスを最大限に引き出すように高度に最適化、スレッド化、ベクトル化された数学関数が含まれています。
- **Apache Spark*。**Apache Spark* は、Java*、Scala*、Python* または R から使用できる、大規模なデータ処理向けの高速なエンジンです。
- **Apache Hadoop*。**Apache Hadoop* ソフトウェア・ライブラリーは、計算クラスター間で大規模なデータの分散処理を行うことができるフレームワークです。JD は、このモジュールに含まれる Hadoop* Distributed File System (HDFS) を使用して画像から抽出した特徴データを格納しています。Hadoop* YARN は、CPU 上でのジョブ・スケジューリングおよびクラスターリソース管理のフレームワークを提供します。
- **インテル® イーサネット・サーバー・アダプター I350 およびインテル® イーサネット・コンバージド・ネットワーク・アダプター X710。**JD は、アジャイルなデータセンターの要件に適したインテル製のアダプターを使用しています。

インテルの協力

新しい特徴抽出機能は、インテルのリサーチエンジニアと開発エンジニアの協力の下で開発されました。JD とインテルは長年の協力関係にあり、昨年はビッグデータと解析アプリケーションの開発を行いました。インテルの中国の研究開発チームは、BigDL のようなオープンソースのソリューションを利用してクラウド・サービス・プロバイダーを支援しており、これまで多くの配備に取り組んだ経験があります。

「弊社のビッグデータ・クラスターをベースとする大規模なディープラーニング・アプリケーションの作成は大きな課題でした。インテルは BigDL テクノロジーの豊富な知識を備えており、弊社の実装にも大いに役立ちました。インテルのチームは、開発期間を短縮し、継続的な革新が可能な専門知識と経験をもたらしました。」(JD、シニア・ソフトウェア・エンジニア (アルゴリズム)、Zhenhua Wang 氏)

革新のプラットフォーム

JD は、画像マッチと特徴抽出をベースにした新しいサービスの作成に利用できるプラットフォームを確立しました。このプラットフォームは、ほかのディープラーニング・アプリケーションや人工知能アプリケーションを開発するテンプレートとしても利用できます。BigDL フレームワークを利用することで、JD は、専用ハードウェアに投資することなく、汎用ハードウェアで Caffe*、Torch および TensorFlow* などのフレームワークから訓練済みのモデルを使用して、新しいサービスを迅速にテストして開始できるようになりました。JD は、内部アプリケーションとクラウドベースのサービスの両方で、分散型モデル訓練を含む、広範なディープラーニング・アプリケーションに BigDL の適用を続けています。パブリッククラウドでは、トピック別に記事を分類する BigDL

ベースのテキスト分類モデルをすでに提供しています。JD は、今後もインテルと協力して、さまざまな新しいテクノロジーに取り組んでいきます。

JD について

JD は Jingdong Group の EC サイトとして 2004 年に創設されました。2017 年 3 月時点の正社員数は 12 万人以上で、携帯電話、デジタル・テクノロジー、コンピューターなどを扱う、中国で最大のオンライン市場の 1 つです。同社のカタログには、家庭用品、コンピューター、おもちゃ、紳士服、婦人服、靴、書籍、贈答品、スポーツ用品、自動車のアクセサリーを含む、さまざまなカテゴリーの商品が掲載されています。2014 年 5 月に、米国の NASDAQ に上場しました。

www.jd.com (英語)

この記事から分かること

ほかのクラウド・サービス・プロバイダーは、JD の経験が参考になるでしょう。

- データを格納する同じクラスターでディープラーニング解析を実行すれば、別の解析クラスターにデータをコピーする時間を省くことができます。JD のケースでは、この時間が解析ワークロード全体の時間の約半分を占めていました。
- BigDL は、Caffe* などのフレームワークで GPU を使用して訓練していたモデルを CPU 上の Spark* で使用できるフレームワークを提供します。BigDL は、サードパーティーの訓練済みモデルも使用できるため、開発期間を短縮できます。
- 画像特徴抽出機能の確立により、JD は、パブリッククラウド (テキスト分類など) や E コマースビジネス (画像検索など) 向けに革新的なアプリケーションを開発して配備できるようになりました。

組織に合ったソリューションを見つけてください。詳細は、インテルの担当者まで、またはこちらからお問い合わせください。

関連情報

- インテル® Xeon® プロセッサー E5 ファミリー
- BigDL: Apache Spark* 上の分散型ディープラーニング (英語)
- インテル® マス・カーネル・ライブラリー (インテル® MKL)
- JD.com の BigDL を使用した大規模な画像特徴抽出の構築 (英語)

#CodeModernization (英語)

参考資料 (英語)

[1] Building Large-Scale Image Feature Extraction with BigDL at JD.com, https://software.intel.com/en-us/articles/building-large-scale-image-feature-extraction-with-bigdl-at-jdcom.

インテル® テクノロジーの機能と利点はシステム構成によって異なり、対応するハードウェアやソフトウェア、またはサービスの有効化が必要となる場合があります。実際の性能はシステム構成によって異なります。絶対的なセキュリティーを提供できるコンピューター・システムはありません。詳細については、各システムメーカーまたは販売店にお問い合わせいただくか、https://www.intel.co.jp/ を参照してください。

性能に関するテストに使用されるソフトウェアとワークロードは、性能がインテル® マイクロプロセッサー用に最適化されていることがあります。SYSmark* や MobileMark* などの性能テストは、特定のコンピューター・システム、コンポーネント、ソフトウェア、操作、機能に基づいて行ったものです。結果はこれらの要因によって異なります。製品の購入を検討される場合は、他の製品と組み合わせた場合の本製品の性能など、ほかの情報や性能テストも参考にして、パフォーマンスを総合的に評価することをお勧めします。詳細については、www.intel.com/benchmarks (英語) を参照してください。

ベンチマーク結果は、「Spectre」および「Meltdown」と呼ばれる脆弱性への対処を目的とした最新のソフトウェア・パッチおよびファームウェア・アップデートの適用前に取得されたものです。パッチやアップデートを適用したデバイスやシステムでは同様の結果が得られないことがあります。

インテルは、本資料で参照しているサードパーティーのベンチマーク・データまたはウェブサイトについて管理や監査を行っていません。本資料で参照しているウェブサイトにアクセスし、本資料で参照しているデータが正確かどうかを確認してください。

記載されているコスト削減シナリオは、指定の状況と構成で、特定のインテル® プロセッサー搭載製品が今後のコストに及ぼす影響と、その製品によって実現される可能性のあるコスト削減の例を示すことを目的としています。状況はさまざまであると考えられます。インテルは、いかなるコストもコスト削減も保証いたしません。

ここに記載されているすべての情報は、予告なく変更されることがあります。インテルの最新の製品仕様およびロードマップをご希望の方は、インテルの担当者までお問い合わせください。

この記事のパフォーマンス・テストは、20 枚の NVIDIA Tesla* K40 GPU を搭載したシステムと 24 台のインテル® Xeon® プロセッサー E5-2650 v4 ベースのサーバー (2.20GHz、1,200 論理コア) を比較したものです。24 の物理コアを搭載し、インテル® ハイパースレッディング・テクノロジー (インテル® HT テクノロジー) を有効にしたサーバーを、Apache Hadoop* Yet Another Resource Negotiator (YARN) で 50 の論理コアをサポートするように設定しました。

コンパイラーの最適化に関する詳細は、最適化に関する注意事項を参照してください。