https://software.intel.com/en-us/videos/accelerating-lossless-data-compression-code-for-cloud-and-edge-applications

クラウドおよびエッジ・アプリケーション向けデータ圧縮コードの高速化

intel Software

内容

- インテル® インテグレーテッド・パフォーマンス・プリミティブ (インテル® IPP) の概要
  - インテル® IPP 2018 の新機能
- データ圧縮の使用例
- インテル® IPP のデータ圧縮関数
  - zlib
  - LZ4
  - LZO
  - bzip2
- アプリケーションに最適な圧縮アルゴリズムを選択する方法

インテル® インテグレーテッド・パフォーマンス・プリミティブ (インテル® IPP)

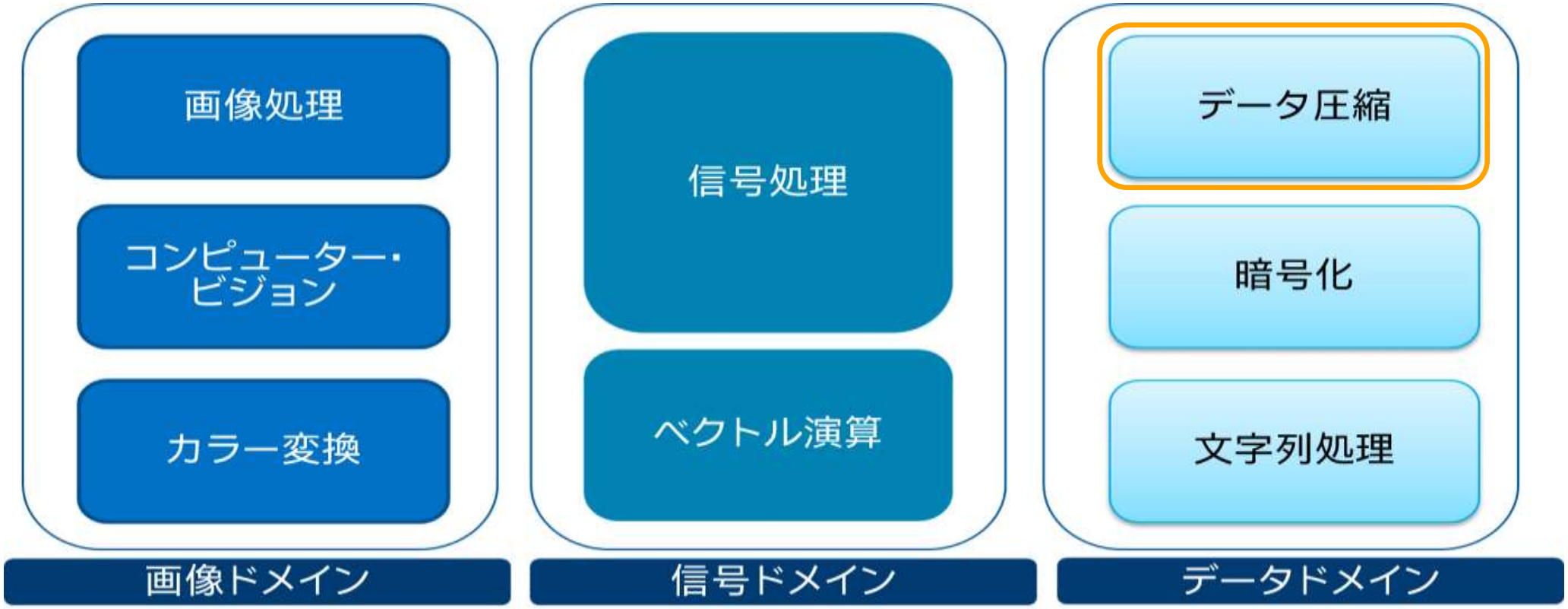
インテル® IPP は、画像、信号、データ、暗号化処理作業を高速化する、すぐに使える、プロセッサー向けに最適化された関数

- マルチコア、マルチ OS、マルチプラットフォームに対応した計算集約型の高度に最適化された関数
- プラグイン API によりアプリケーションのパフォーマンスを素早く向上
- ソフトウェアの開発と保守にかかる費用を軽減し、開発期間を短縮
- インテルのエンジニアに技術的な質問を直接問い合わせることができる プライオリティー・サポート (英語) (有償バージョンのみ)

インテル® IPP 2018 の新機能

- LZ4 データ圧縮/展開をサポートする新しい関数
- データ圧縮ドメイン関数の新しい最適化
  - インテル® アドバンスト・ベクトル・エクステンション 2 (インテル® AVX2) およびインテル® ストリーミング SIMD 拡張命令 4.2 (インテル® SSE4.2) における LZO データ圧縮関数のパフォーマンスが向上
  - LZO 展開にレベル 999 サポートを追加
  - ユーザー定義のハフマンテーブルをサポートする新しい zlib 関数を追加
- スタンドアロンの暗号化パッケージ
- 新しい楕円曲線暗号関数を追加
- GraphicsMagick* ソースコード向けの新しいドロップインの最適化
  - GraphicsMagick* 1.3.25 をサポートし、次の GraphicsMagick* API 向けの最適化を提供: ResizeImage、ScaleImage、GaussianBlurImage、FlipImage、および FlopImage
  - API のパフォーマンスを最大 4 倍向上 (機能、入力パラメーター、プロセッサーに依存)
- インテル® AVX-512 およびインテル® AVX2 命令セット向けの最適化を拡張

インテル® IPP
ハイパフォーマンスで使いやすいプロダクション環境対応の API



インテル® アーキテクチャー・ベースのプラットフォーム

オペレーティング・システム: Windows*、Linux*、macOS*[1]

[1]インテル® Parallel Studio XE Composer Edition でのみ利用可能

インテル® IPP のアクティブな分野 - データ圧縮

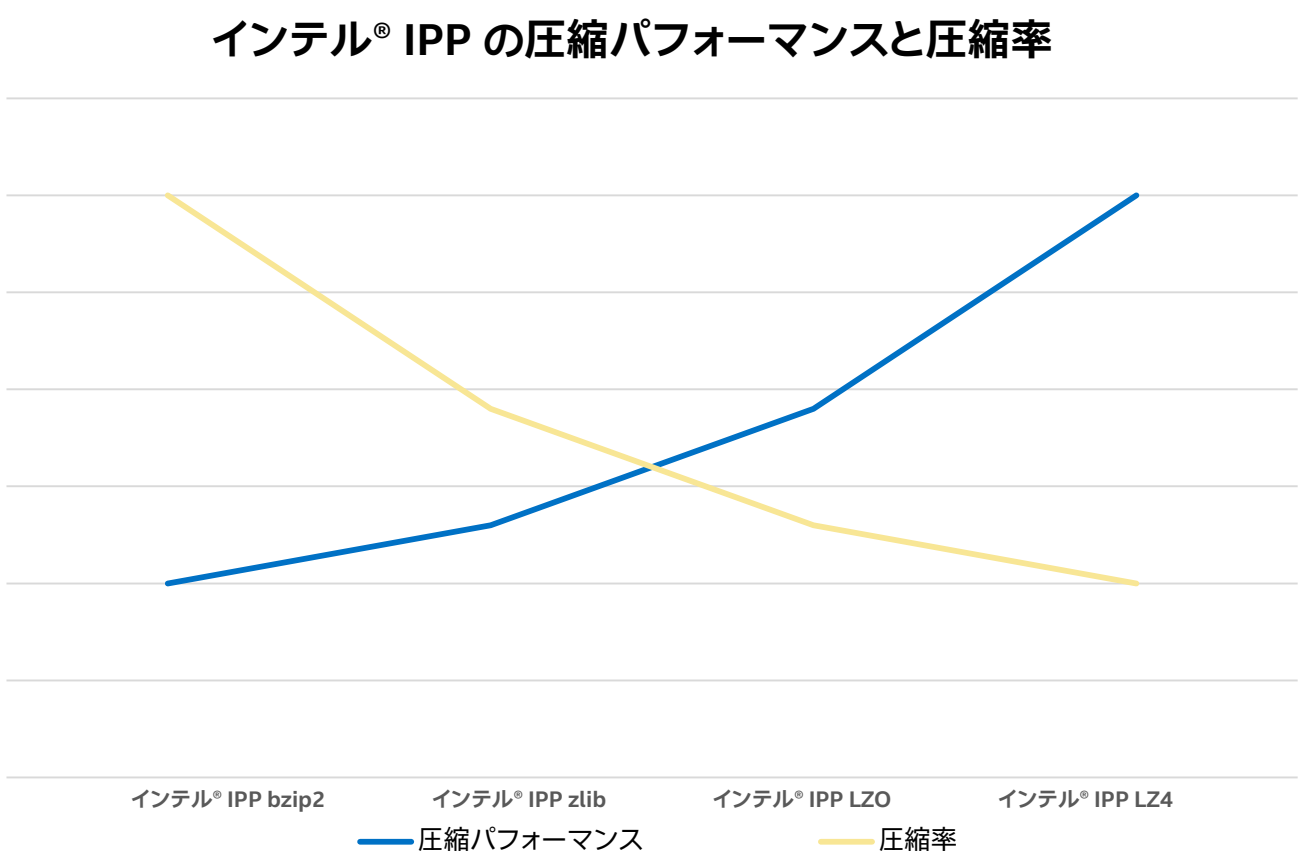
インテル® IPP 2018 のデータ圧縮のアップデート

- zlib の機能
  - 圧縮/展開パフォーマンスの向上
  - 圧縮率を下げてパフォーマンスを大幅に向上する最速の圧縮レベル
  - 圧縮率を向上する訓練済みハフマンテーブル
- 新しい LZ4 関数
- LZO の機能
  - 圧縮/展開パフォーマンスの向上

データ圧縮の使用例

可逆データの使用例

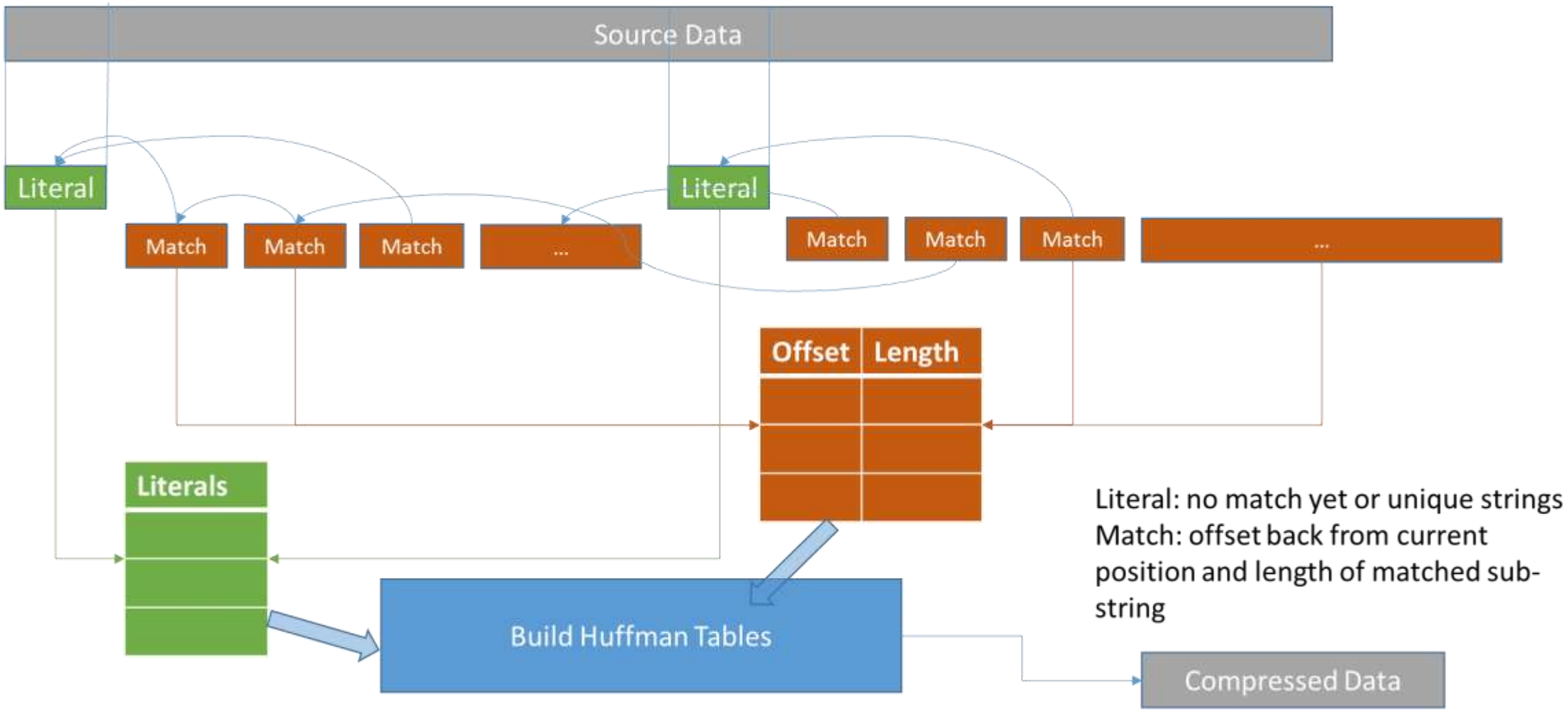
- ケース 1: 圧縮率の高いアルゴリズム
  - サーバー側で 1 回圧縮して、データを転送し、複数回展開する
  - データストレージ
- ケース 2: 高速圧縮アルゴリズム
  - リアルタイムのデータ圧縮とデータ転送
  - 低性能システム上でのデータ圧縮
- ケース 3: 圧縮率とパフォーマンスのバランスが良いアルゴリズム
  - zlib ベースのアーカイブ



zlib データ圧縮

- 辞書ベースのコーディング (LZ77 アルゴリズムとハフマン符号)
- 広範なデータ型をサポート
- オープンソース、クロスプラットフォームの可逆データ圧縮ライブラリー
- GNU* ライセンス不要
- 圧縮率とパフォーマンスのバランスが良い
- 使用モデル
  - アーカイブ: gzip、WinZip*、PKZIP* など
  - グラフィックス・ライブラリー: PNG (ポータブル・ネットワーク・グラフィックス)
  - ネットワーク PPP プロトコルのセット
  - リアルタイム圧縮の特定の高度に調整されたタスク (ソフトウェア・データ・キャッシュ、高速バックアップ・ユーティリティー、圧縮関連の Java* クラス)

zlib アルゴリズム



zlib の利点と制限

- 利点
  - 優れた圧縮率: 詳細なソースデータ統計 (エントロピー) を収集
  - ビット単位のコーディング: 各出力ビットが重要
  - 長年にわたって事実上の業界標準: 膨大な数の顧客/アプリケーション
- 制限
  - 非常に多くの計算リソースを必要とする
  - ソースデータ・バッファーを複数回パススルー
  - ビット単位のコーディング: 各出力ビットが CPU 時間を消費

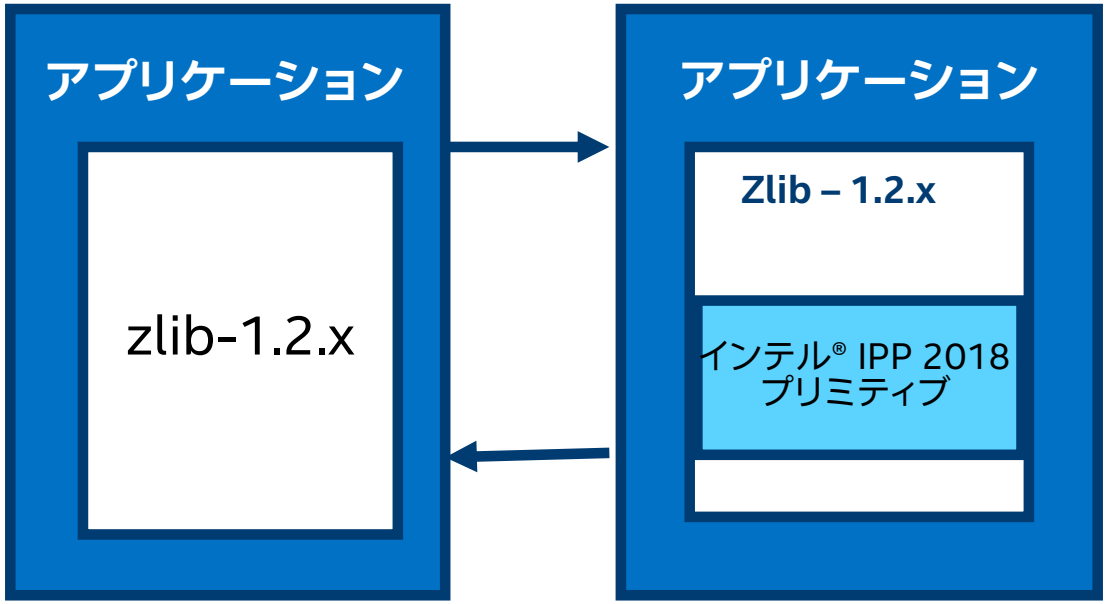
インテル® IPP の zlib 関数の使用

zlib API の使用

- include “zlib.h”
- 基本圧縮関数の呼び出し
  deflateInit();
  deflate();
  deflateEnd();
- 基本展開関数の呼び出し
  inflateInit();
  inflate();
  inflateEnd()
- 圧縮レベルは Z_DEFAULT_COMPRESSION (0-9) で定義
  レベル 1: スピードを優先
  レベル 9: 圧縮率を優先
  レベル 6 (デフォルト): スピードと圧縮率のバランスを考慮

詳細は http://www.zlib.net (英語) を参照

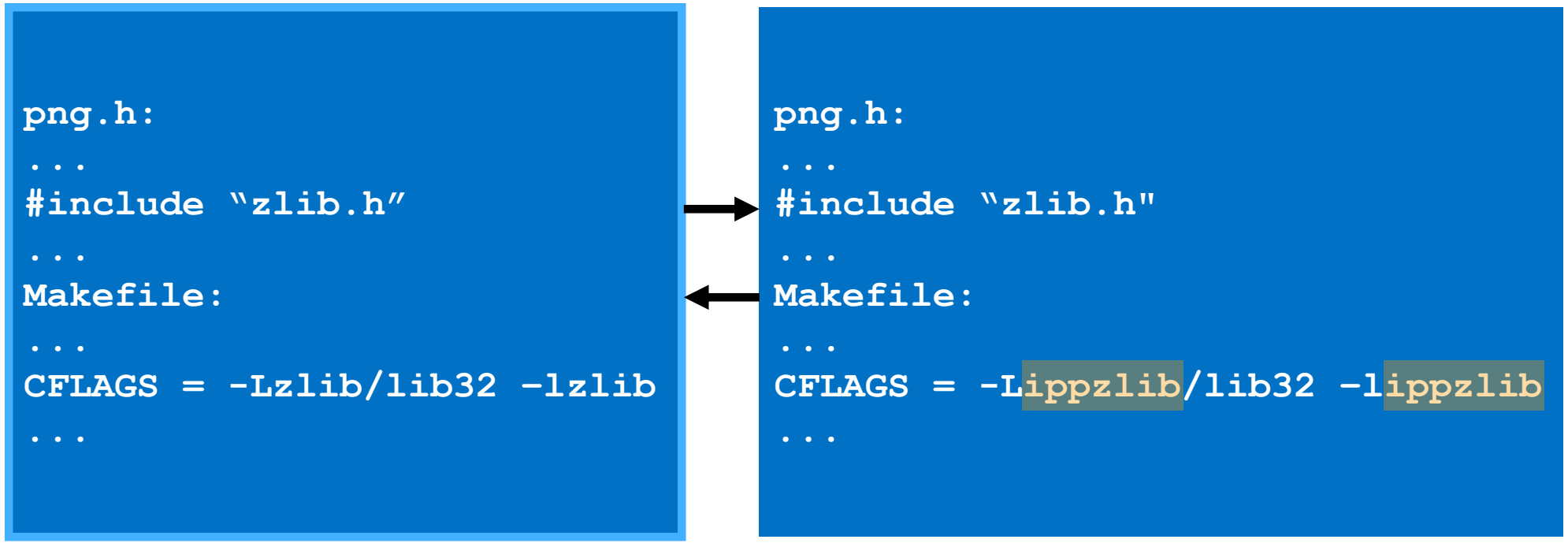
zlib ライブラリーから簡単に移行可能



intel Software

zlib1.2.x からインテル® IPP の zlib への移行

- ソースコードの変更は不要
- インテル® IPP の zlib ライブラリーに再リンクするだけ
  - 例: PNG ライブラリーの Makefile の変更



インテル® IPP zlib の機能

- すべての zlib メジャーバージョンとレベル 1-9 をサポート
- 訓練済みハフマンテーブル
- 最速の圧縮 (レベル -2)
  - 標準 zlib と完全に互換性のある圧縮データ
  - 標準 zlib レベル 1 と同等
  - 圧縮率を下げてパフォーマンスを大幅に向上

データ圧縮コーパス	圧縮率 (レベル "fast"、1)/パフォーマンス (MB/秒)	圧縮率 (レベル "fastest"、-2)/パフォーマンス (MB/秒)
Large Calgary	2.83/78	2.39 (-0.44)/331 (+253、4.2 倍)
Canterbury	3.15/85	2.58 (-0.57)/372 (+287、4.4 倍)
Large (3 ファイル)	2.81/70	2.18 (-0.63)/300 (+230、4.3 倍)
Silesia	3.01/78	2.56 (-0.45)/361 (+283、4.6 倍)

システム構成 – ソフトウェア: Red Hat* Enterprise Linux* Server 7.1 (Maipo)、インテル® IPP 2018。
ハードウェア: インテル® Xeon® プロセッサー E5-2680 v3、2.50GHz、2x12 コア + HT、30MB SmartCache、264GB RAM。

インテル® IPP zlib の最速圧縮

- 一致検索の緩和
- 入力データを 1 回パススルー (一般的な zlib は 2 パス・アルゴリズムを採用: 1 回目でエントロピーを収集し、2 回目で出力ストリームを生成)
- デフォルトでは、標準のスタティックなハフマンテーブルを使用して出力を生成

インテル® IPP zlib の訓練済みハフマンテーブルを利用する最速圧縮

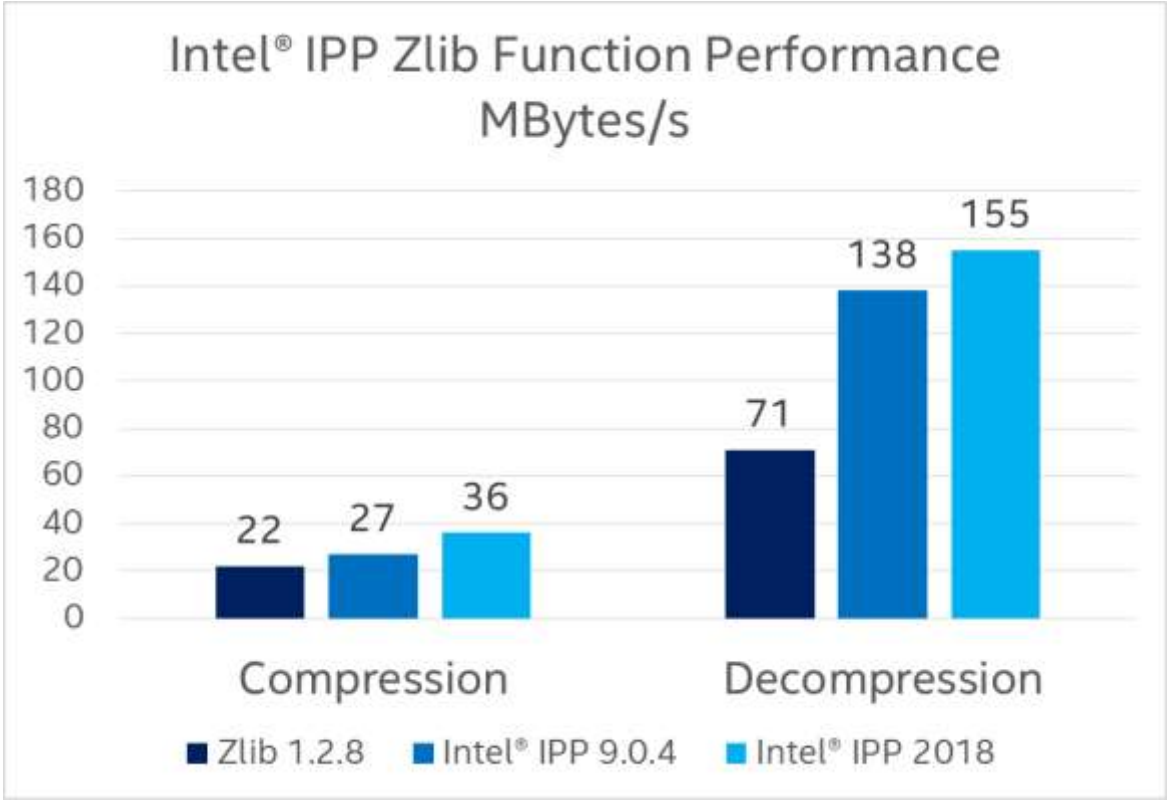
- ステップ 1: デフォルトのインテル® IPP zlib の最速圧縮で同様のデータ (HTML、テキスト、データベース・クエリー、...) の統計を収集
- ステップ 2: 生成されたハフマンテーブルを保存
- ステップ 3: 保存したテーブルを使用するように指定して、インテル® IPP zlib の高速圧縮向けに新しい "z_stream" を作成

File	Default Ratio	Default Compression Performance	Ratio with Table	Performance with Table
bib	2.15	176	2.60	179
book1	1.71	126	2.19	158
book2	2.06	177	2.54	186
geo	1.10	93	1.46	96
news	1.97	130	2.37	148
obj1	1.77	206	1.97	222
obj2	2.31	178	2.59	177
paper1	2.06	179	2.50	186

システム構成 – インテル® Core™ i5-4300U プロセッサー @ 2.50GHz、Windows* 8.1 Enterprise (64 ビット)

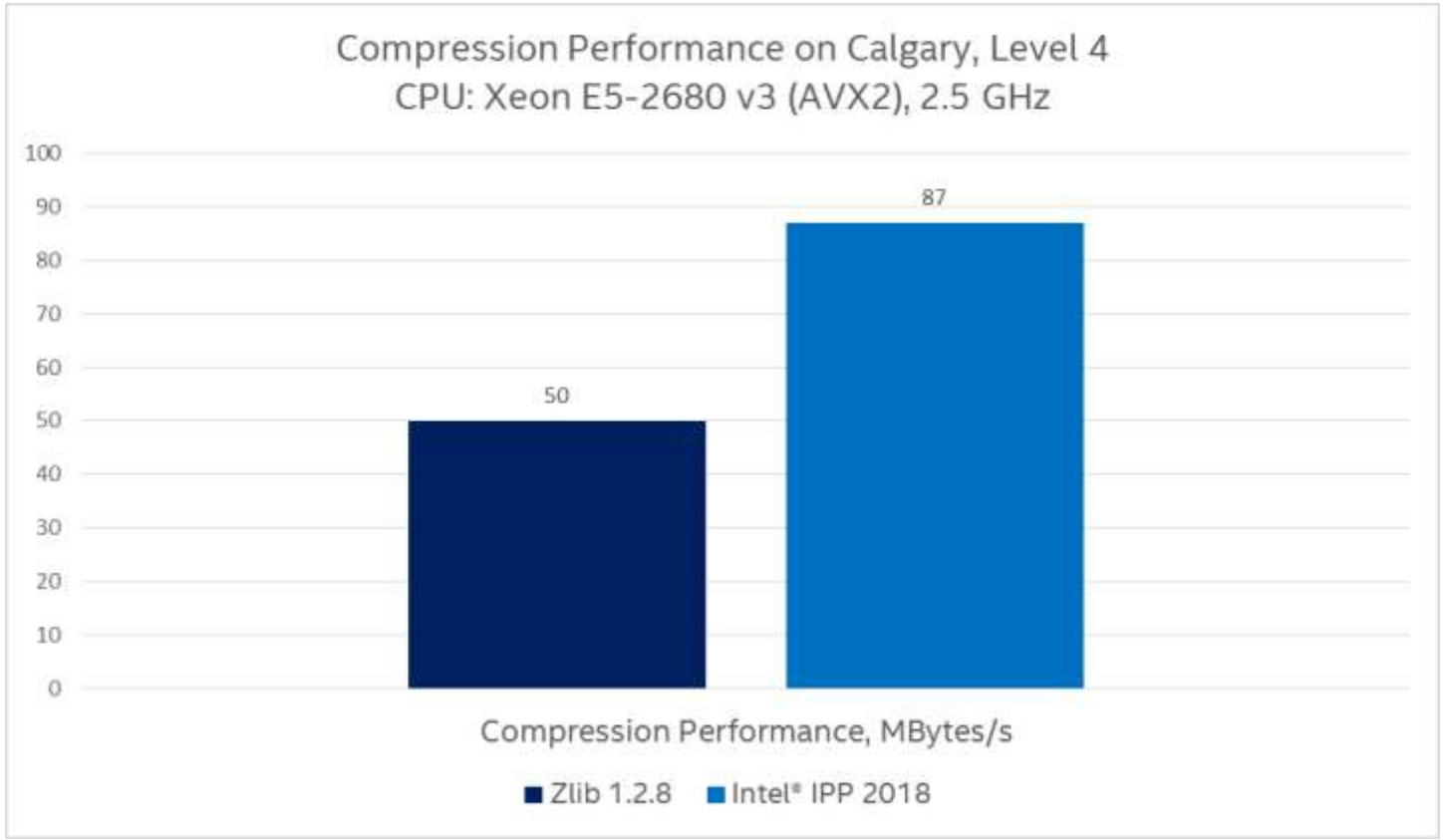
インテル® IPP zlib パフォーマンスの向上

- 基本 zlib アルゴリズムを作成し直し – すべての CPU アーキテクチャーで利点が得られる
- いくつかの新しい命令を使用 – インテル® SSE4.x - インテル® AVX2 対応 CPU で利点が得られる固定ハフマン符号を利用する新しい最速圧縮レベル (レベル -2)
- 訓練済みハフマンテーブル – 最速モードの圧縮率を向上
- zlib オープンソース・ディストリビューションとの統合を改善



システム構成 – ソフトウェア: Red Hat* Enterprise Linux* Server 7.1 (Maipo)、インテル® IPP 2018。
ハードウェア: インテル® Xeon® プロセッサー E5-2680 v3、2.50GHz、2x12 コア + HT、30MB SmartCache、264GB RAM。
データ: Calgary の平均、圧縮レベル 6。

インテル® IPP zlib パフォーマンスの向上 (続き)

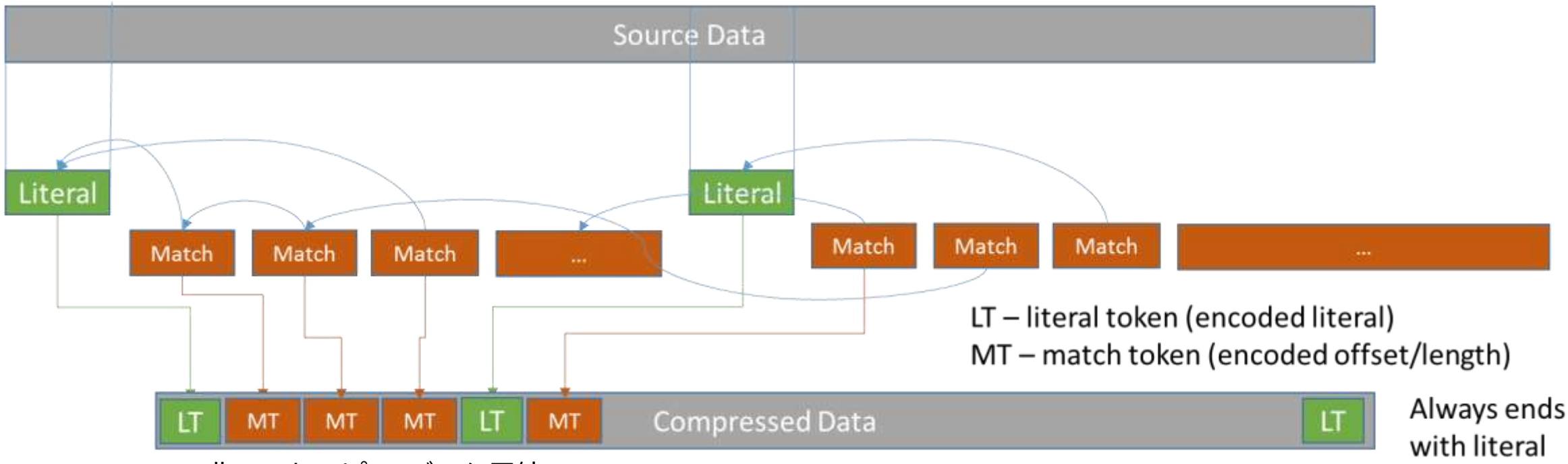


システム構成 – ソフトウェア: Red Hat* Enterprise Linux* Server 7.1 (Maipo)、インテル® IPP 2018。
ハードウェア: インテル® Xeon® プロセッサー E5-2680 v3、2.50GHz、2x12 コア + HT、30MB SmartCache、264GB RAM。
データ: Calgary の平均、圧縮レベル 4。

LZ4 向けのインテル® IPP 関数

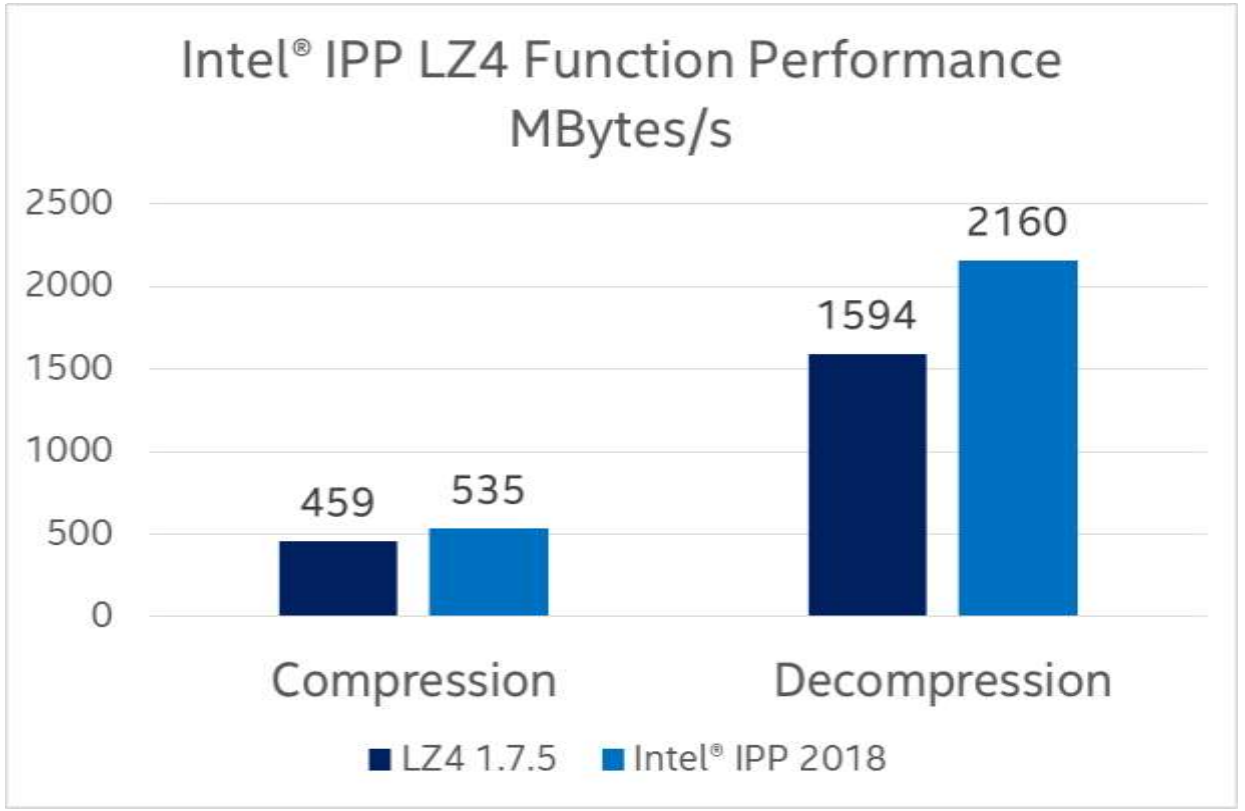
- Google* LZ4 と互換性のある圧縮データ。非常に高速なデータ圧縮に使用
- インテル® IPP 2018 では、LZ4 は圧縮/展開関数をサポート
  - LZ4 高速圧縮 (辞書とデスティネーション・サイズの制限を含む)
  - LZ4 高圧縮 (HC)
- 将来のリリースでサポート予定の機能
  - 辞書とデスティネーション・サイズの制限を含む LZ4 HC (セーフモード)
  - インテル® IPP LZ4 とオープンソース LZ4 の統合の改善。コードを変更しなくても済むようにオープンソース LZ4 と同じ API の提供

LZ4 の内部アルゴリズム



- LZ4 – 非エントロピー・データ圧縮
- リテラル (一意の部分文字列、初出、...) とオフセット長のペアのみ符号化される
  オフセット = 現在のポインターと以前にエンコードされた同様の部分文字列の差異
  オフセットはデコード中に部分文字列のコピーに使用される
- 非エントロピー圧縮法の違いは、主に圧縮データ符号化の違いに依存
  例: LZ4 – バイト境界でアライメントされた符号化、高速デコード、LZO – ビット境界でアライメントされた符号化、高圧縮率と低速
- LZ4 FC/HC (高速圧縮/高圧縮) は一致検索の精度が異なる

インテル® IPP 2018 の LZ4 パフォーマンス



システム構成 – ソフトウェア: Red Hat* Enterprise Linux* Server 7.1 (Maipo)、インテル® IPP 2018。
ハードウェア: インテル® Xeon® プロセッサー E5-2680 v3、2.50GHz、2x12 コア + HT、30MB SmartCache、264GB RAM。
データ: Calgary の平均。

LZO 向けのインテル® IPP 関数

- LZO は辞書ベースの圧縮法のポピュラーな改良
- 高速圧縮で、辞書や追加のメモリーを必要としない非常に高速な展開
- さまざまな圧縮レベル (メソッド) があり、圧縮率と圧縮パフォーマンスのバランスが良い LZO1X メソッドが最も良く使用されている
- インテル® IPP の LZO API

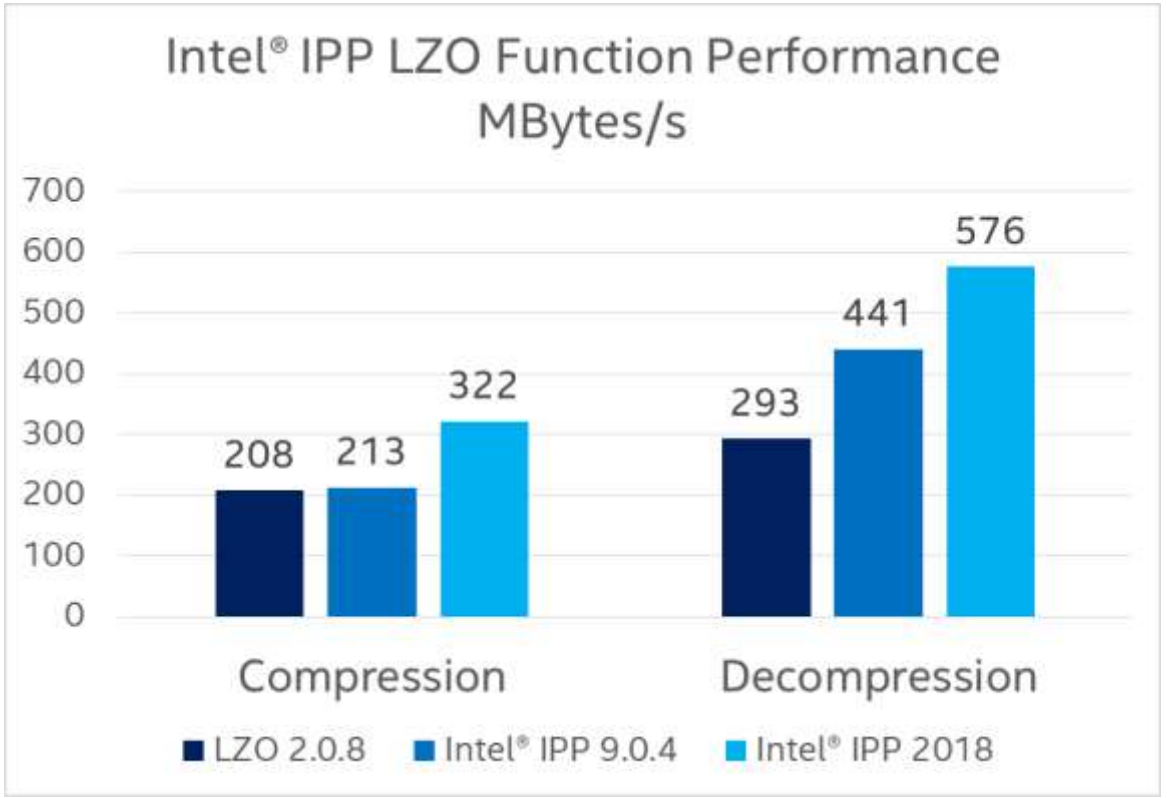
```
ippsEncodeLZOGetSize(IppLZO1XST, BUFSIZE, &LZOSize);

pLZOState = (IppLZOState_8u*)ippsMalloc_8u(LZOSize);

ippsEncodeLZOInit_8u(IppLZO1XST, BUFSIZE, pLZOState);
for(.....){
     ippsEncodeLZO_8u(src, srcLen, dst, &dstLen, pLZOState);
}
```

インテル® IPP LZO のパフォーマンス

- 基本 LZO アルゴリズムを作成し直し – すべてのインテル® CPU で利点が得られる



システム構成 – ソフトウェア: Red Hat* Enterprise Linux* Server 7.1 (Maipo)、インテル® IPP 2018。
ハードウェア: インテル® Xeon® プロセッサー E5-2680 v3、2.50GHz、2x12 コア + HT、30MB SmartCache、264GB RAM。
データ: Calgary の平均。

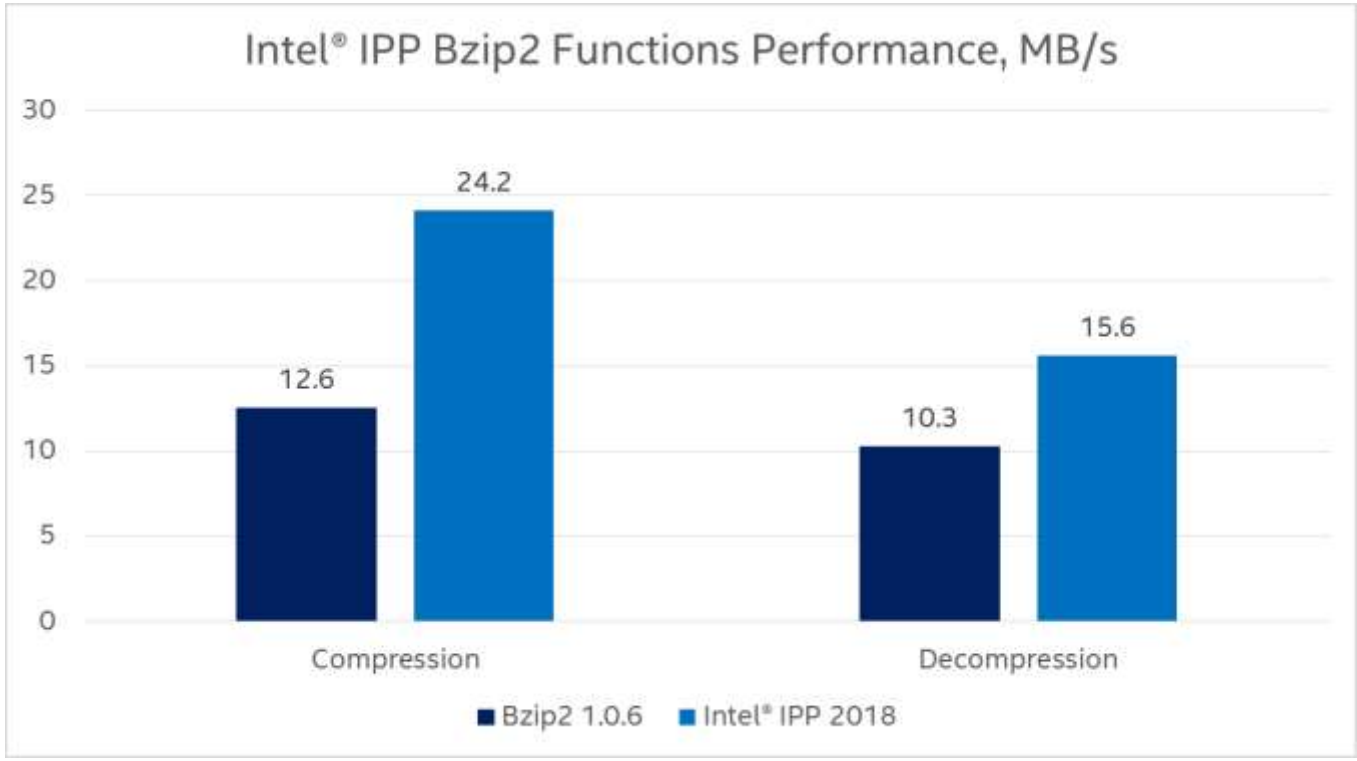
bzip2 向けのインテル® IPP 関数

インテル® IPP bzip2 の機能

- bzip から簡単に移行できる bzip v.1.0.6 向けの最適化パッチを提供
- 完全なソースコードを提供、bzip とのバイナリー互換性

典型的な使用例

サーバー側で 1 回データを圧縮し、クライアント側で複数回展開する



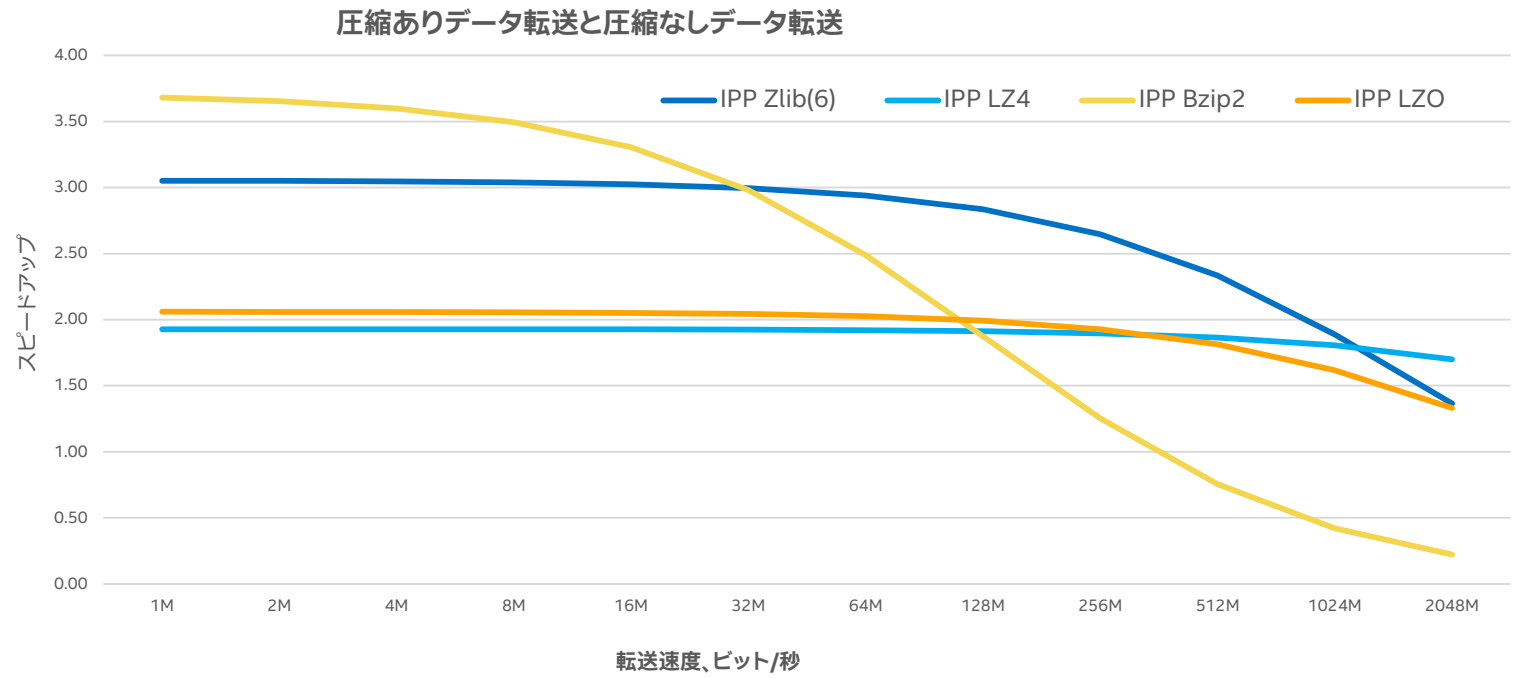
システム構成 – ソフトウェア: Red Hat* Enterprise Linux* Server 7.1 (Maipo)、インテル® IPP 2018。
ハードウェア: インテル® Xeon® プロセッサー E5-2680 v3、2.50GHz、2x12 コア + HT、30MB SmartCache、264GB RAM。
データ: Calgary の平均。

intel Software

データ転送のためのデータ圧縮

圧縮した Calgary データをネットワーク経由でサーバーからダウンロードするのに適した圧縮アルゴリズムは?

合計時間 = 転送時間 + 展開時間



システム構成 – ソフトウェア: インテル® IPP 2018、インテル® C++ コンパイラー 16.0。
ハードウェア: インテル® Core™ i7-6700K プロセッサー、8MB キャッシュ、4.20GHz、16GB RAM、Windows Server* 2012 R2。

関連情報

- インテル® IPP の入手方法
  - 無料のインテル® IPP (フォーラムによるサポート付き)
  - インテル® ソフトウェア開発製品スイートの一部として利用可能
    - インテル® Parallel Studio XE
    - インテル® System Studio

- 製品に関する詳細
  - インテル® IPP 製品情報: https://www.isus.jp/intel-ipp/
  - インテル® IPP のベンチマーク: https://software.intel.com/en-us/intel-ipp/benchmarks# (英語)
  - プライオリティー・サポート: https://supporttickets.intel.com/servicecenter?lang=en-US (英語)
  - ユーザーフォーラム: https://software.intel.com/en-us/forums/intel-integrated-performance-primitives (英語)

法務上の注意書きと最適化に関する注意事項

- 本資料の情報は、現状のまま提供され、本資料は、明示されているか否かにかかわらず、また禁反言によるとよらずにかかわらず、いかなる知的財産権のライセンスも許諾するものではありません。製品に付属の売買契約書『Intel's Terms and Conditions of Sale』に規定されている場合を除き、インテルはいかなる責任を負うものではなく、またインテル製品の販売や使用に関する明示または黙示の保証 (特定目的への適合性、商品性に関する保証、第三者の特許権、著作権、その他、知的財産権の侵害への保証を含む) をするものではありません。
- 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。
- © 2018 Intel Corporation. 無断での引用、転載を禁じます。Intel、インテル、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Core、Xeon、Intel Xeon Phi は、アメリカ合衆国および / またはその他の国における Intel Corporation の商標です。

最適化に関する注意事項

インテル® コンパイラーでは、インテル® マイクロプロセッサーに限定されない最適化に関して、他社製マイクロプロセッサー用に同等の最適化を行えないことがあります。これには、インテル® ストリーミング SIMD 拡張命令 2、インテル® ストリーミング SIMD 拡張命令 3、インテル® ストリーミング SIMD 拡張命令 3 補足命令などの最適化が該当します。インテルは、他社製マイクロプロセッサーに関して、いかなる最適化の利用、機能、または効果も保証いたしません。本製品のマイクロプロセッサー依存の最適化は、インテル® マイクロプロセッサーでの使用を前提としています。インテル® マイクロアーキテクチャーに限定されない最適化のなかにも、インテル® マイクロプロセッサー用のものがあります。この注意事項で言及した命令セットの詳細については、該当する製品のユーザー・リファレンス・ガイドを参照してください。

注意事項の改訂 #20110804