

進化するAsakusaの現状と今後



Hadoopエコシステムの情勢

■ Hadoop 1.xの陳腐化

- 最新の1.2.1がすでに1年前 (2013/08/01)のもの
- 2.xでHDFSの安定化・高速化が行われたが、1.xにバックポートなし
 - つまりHadoop1系は今後の更新はない
- エコシステムもYARNを中心に構成されるようになりつつある

■ SQL on Hadoop周りがかかなり成長

- Impala, Drill, PrestoなどHDFSに対するクエリが強化されている
- Hive on Tezがとても高速
- 今後もこの周辺に大規模な投資が行われることが予想される

■ MapReduceの直書きは非推奨？

- HortonworksのリリースノートからMapReduceという単語が削除
- Hive/Pigなどの上位言語で記述することが前提となりつつある
- もともとノーチャラスはMapReduceは非推奨

Asakusa 0.7.0 (2014/09/25)

■ 問題意識

- 大量のデータをHadoopクラスタから移動するのに多大なコストがかかる
 - 当初はHadoopHDFSの信頼性が低い
- 可能な限りHadoopクラスタ上で絞ってから移動したい
 - HDFSの信頼性の向上。特にMapRの登場は大きい

■ SQL on Hadoop向けデータフォーマットに対応

- ParquetやORCFileなどのHive向けデータフォーマットに対応
 - SQL処理
- Asakusaで整形したデータをHive等でアドホック集計
 - バッチ処理をして、クエリーで検索する

AsakusaとSQLの使い分け

■SQL

- アドホックに書ける
- 非常に複雑なデータ処理に弱い
- かつちりとしたデータ処理に弱い
- テストを実施しにくい

■Asakusa

- 業務バッチレベルの複雑なデータ処理に対応
- 設計ありきのDSLでかつちりと開発
- アドホックなデータ処理に弱い
- 強力なテスト機構

■両者を組み合わせることで広いレンジのデータ処理が可能に

Hadoop 2.xに正式対応

- Asakusa DSLで書くとHadoop 1.x/2.xのどちらでも動く
 - ビルドツールを強化し、1.x/2.xのリリースバイナリを同時に生成可能に
 - とくにSQL on HadoopはHadoop 2.xを前提としたものが多い
 - AsakusaのPortabilityの高さの証左
 - Hadoopエコシステム中ではもっとも後方互換性が高い
- YARN (Hadoop 2.x)
 - 「遅すぎ」という感じでもなくなってきた
 - Hadoop 2.xでHDFSの高速化まで見るとYARNのほうが速度向上するケースも
 - ただし、小さなデータを処理する際のオーバーヘッドは増大
 - 小さなデータばかり処理すると、Hadoop 1.xより50%くらい速度低下も
 - 有効性は高いが、宣伝されているほど無敵ではない

Asakusa 0.7.1 (2014/11/20)

■ 問題意識

- Asakusaで書くとバッチの中に「小さなデータ」を処理するステージがどうしてもできる
 - 補助データの処理
 - 異常処理

- 「スモールジョブ問題」
 - ステージの入力データが100MB未満
 - ステージの実行時間が40秒未満
 - 上記のステージ数がバッチ全体の50%以上

- スモールジョブは処理時間のほとんどがHadoopオーバーヘッド
 - 分散処理の立ち上がりに時間が掛かっている
 - 正味の処理時間は数秒程度でも、オーバーヘッドが乗ると数十秒に

- 業務系バッチ処理に顕著で、通常的大量ログ分析では顕在化しない

スモールジョブ最適化

■ スモールジョブ問題のインパクト (参考値)

- データサイズごとの特性
 - 10GBのMapReduce処理: 240秒前後
 - 10MBのMapReduce処理: 30秒前後
- 例: 100ジョブのうち90%がスモールジョブだと
 - $240s * 10 + 30s * 90 = 2400s + 2700s$
 - スモールジョブの処理時間の方が長い

■ スモールジョブ最適化

- スモールジョブを見つけたら自動で独自のエンジンに切り替え
- 非分散モードで高速に動作
 - 10MBの入力で5~10倍高速
- 入力データが増大しても勝手に分散モードで動作する
 - 日次処理に月次データを入れても安全に処理できる

スモールジョブ最適化の効果

● 実績値

種別	未使用	使用	総ジョブ数	スモールジョブ数
ツリー系集計演算 バッチ	約50分	約15分	約184	約180
データ正規化処理	約8分	約3分	約20	約17
BOM系原価計算	約65分	約52分	約60	約10

その他のメリット

– デベロッパーテスト高速化

- 開発環境内でのテスト実行はたいいてい小さなデータ
- ここでもスモールジョブ最適化を利用可能に
- ほとんどのケースで5倍程度高速化

– Windowsでの開発

- スモールジョブ実行エンジンを利用すると、Windows開発環境内でテストも実行可能に
- 近々に正式なドキュメントを公開予定

最新機能まとめ

- Hadoop 2.xに正式対応
 - エコシステムの最新機能を利用可能に
 - Asakusa DSLで書けば1.x/2.xのどちらでも動作
- SQL on Hadoopとの相互運用性が向上
 - お互いの不得意を補い合う
 - AsakusaとSQLを組み合わせて柔軟なデータ処理
- スモールジョブ問題を緩和
 - 複雑なバッチ処理ほどスモールジョブが紛れ込みやすい
 - 事前に予見できなくても自動的に最適化
 - データが増大しても自動的に対応

Asakusaの今後

■ 1.投資可搬性の確保

- **業務系システムでは非常に重要**
- Hadoopエコシステムは変化が早い
 - 昨日書いたプログラムが明日にはもう時代遅れ
 - 特にHadoop distributionのサポートの供給停止は早い
 - 場合によっては次のバージョンがStableにならないうちに現状プラットフォームの打ち切りのアナウンスだけが先に出る
 - また、Hadoopv1.0のように2-3年で既に新規投資がされないということも起きる
- Asakusaは後方互換性を重要視する
 - **「いったんAsakusa作ったプログラムが可能な限り使い続けることができるようにする。」**
 - Asakusaのポリシー
 - Asakusaの責任母体であるノーチラスのポリシーでもある
 - 10年にわたってEDI等の業務系ミドルをサポートし続けた背景

Asakusaの今後

■2.多機能化

- よりAsakusaの使い勝手も向上させていく
- SQL on Hadoopも対応もその一つ
 - Asakusaでのアウトプットを再利用可能にしていく方策
- Smallジョブの実行の高速化もその一つ
 - 実際の構築からのフィードバックを適切に生かす

■3.高速化

- Hadoopが進化していく過程は、高速化の流れが一つある
- プラットフォームを拡大に対応していく
 - Hadoop2系への追従
 - その他の実行プラットフォームも検討していく

検討中の今後の対応

■ Spark

- 分散メモリークラスター
- 機械学習の実行基盤としては、現在もっとも有望
- コミュニティ活動も活発
- Asakusaの対応基盤としての検討
 - Better Hadoopとしての存在価値はある
 - ただし、Asakusaの最適化スタイルに合わないアーキテクチャ
 - Push型ではなくPull型になっているので、Asakusaでの最適化が有効ではない
 - 単発ステージのバッチ処理ならば確実にHadoopより高速だが、複雑なデータフローが起きる業務バッチ処理には向いていない
- ノーチャスとしては、Asakusaでの検討は行いつつも、別の形でのSparkの利用も視野にいれつつある

検討中の今後の対応

■ Tez

- 開発母体はHorton
 - 現状ではHive on Tez
- 現時点でもっとも高速と目されているHadoop実行エンジン
 - DAGベース
 - Asakusaとの親和性が比較的高い
- Asakusaの実行基盤として検討開始
 - 本家のHortonとも連携予定
- ただし、仕様がStableではないので、もう少し時間がかかる

検討中の今後の対応

■ Microsoft

- Azureでの対応検討開始
- MicrosoftとHortonの連携も視野に入れる
- Azure上でのAsakusaの稼働確認
 - 現状でのクラウドプラットフォームの実績
 - AWS・EMR
 - さくらインターネット
- プラットフォームのパフォーマンスから検討している状態

中長期レンジ

■3~5年のレンジでの検討

- ハードウェア環境の大幅な変更が想定される
- 不揮発性メモリー
 - これによりサーバのあり方が変わる
 - これによりDBのアーキテクチャがかなり変わる
 - おそらくRDBでのOracleの時代の終わりが始まる
- 分散クラスターのあり方が変わる
 - より高密度
 - より集約された「クラスター」的なものになる
- おそらく現状のHadoopのアーキテクチャではオーバーヘッドが大きすぎる
 - 別の何かが出てくる
 - そのようなプラットフォームにもAsakusaは対応する予定
 - そのための準備に入る

中長期レンジ

■ 3~5年のレンジでの検討

- 業務系処理については、データサイズは拡大しないと思われる
 - いまのところPBクラスの業務データはない
- 計算は複雑になる
 - いままでギブアップしていた計算に挑戦する気配もある
 - 複雑なデータ制御
 - SQLではメンテできない
 - データの数の増大（サイズではない）
 - 入力データの種類の増大
 - 例) 電力自由化のメータのデータ
 - ビッグデータの最右翼の候補だったが、全然大きくない
 - 計算に利用する種別は増える
 - ただし計算中間結果が増大する
 - 高速化のニーズが強い
 - 高速化のスピードが要求される
 - たぶん、現状の延長線上ではなく別のアーキテクチャで3年後ぐらいに超高速になる
 - 「Asakusaが賭けるに足る」ということではオッズは高いです。などと。