2014.11.28
2014 Asakusa Framework Day

Asakusa Frameworkの実利用状況と評価結果

三菱UFJインフォメーションテクノロジー株式会社
ITプロデュース部
土佐鉄平

Overview

1. 当社ご紹介
2. Asakusa Framework とは
3. 当初の印象
4. 実利用状況
5. 評価結果
6. 印象の変化
7. 課題
8. まとめ

私たちは、三菱東京UFJ銀行をはじめとする
三菱UFJフィナンシャル・グループの総合金融サービスを支える
金融×ITのリーディングカンパニーです。

2．Asakusa Frameworkとは

Hadoop上で稼動する業務バッチアプリケーションを開発するためのフルスタックフレームワーク

- **MapReduceをラップし直感的な業務ロジック実装が可能**
- **業務バッチアプリ開発に有用な周辺機能が充実**
- **並列分散処理環境を最大限活かす最適化機能**

3．当初の印象

当初はネガティブな印象が先行

業務バッチをHadoopで稼動させるニーズがあるのか？
⇒ 非構造データの分析処理がメインのイメージ

独自言語としてのDSLを、習得しなければならないのか？
⇒ ガラパゴス技術の懸念

国産OSSは普及するのか？
⇒ 普及例が見当たらない

４．実利用状況

① 単発集計処理の通常バッチからの移行

② ログからの情報抽出バッチ

③ 大量計算処理を伴う業務バッチ

（参考）スペック

- CDH 4.1.2
- 8Nodes (5 Slave Nodes)
- 7 TB Available(3 replications)
- 10GB Memory per server
- Xeon® CPU X5650 @ 2.67GHz
  (6core x 2 socket)

４．実利用状況
①単発集計処理の通常バッチからの移行

単発で依頼された大量データ集計処理

- **IN　　：約6700万件、数カラム**
- **OUT　　：約170万件、数カラム**
- **TIME　　：約7分**

※ 当初、Ruby＋RDBMS で開発していたが、
10時間弱かかる状態でチューニングにも苦慮。
Asakusaの試行も兼ねて移行。

４．実利用状況
②ログからの情報抽出バッチ

複数種類のログからの情報抽出バッチ（月次稼動）

- **IN　　：約6億件（約200GB）10カラム**
- **OUT　　：約1億件（約20GB）10カラム**
- **TIME　　：約2.5時間**

４．実利用状況
③大量計算を伴う業務バッチ

大量の列間計算、行間計算を行う業務バッチ処理（月次稼動）

- **IN　　：約2千万件（約3GB）約7百カラム**
- **OUT　　：約130万件（約3GB）約6百カラム**
- **TIME　　：約10分**

5．評価結果

MapReduceをラップし直感的な業務ロジック実装が可能	一般的なJavaスキルで十分開発可能な直感性
業務バッチアプリ開発に有用な周辺機能が充実	充実したテスト機能とバッチ実行/ログ機能
並列分散処理環境を最大限活かす最適化機能	高度な最適化機能と便利な視覚化機能

即利用可能な充実したバッチ開発支援機能

5．評価結果

通常のプログラム

```
// リスト内のキーと値に
// ついてキー毎に
// 合計値を計算
for(rec in list){
  result[rec.key]
   += rec[key].value
}
```

MapReduceプログラム

```
// リスト内の情報をキーと値に分割
map(key,value,output){
  output(value.key,
         value.value)
}

// キー毎にグルーピングされた
// リスト内で合計値を算出
reduce(key,values,output){
  for(value in values){
    sum += value
  }
  output(key,sum)
}
```

Asakusa DSL

```
// キー毎にグルーピング
// されたリスト内で
// 合計値を算出
groupSort(
  @group("key")list,
  result){
  for(value in list){
    sum += value
  }
  result.add(sum)
}
```

MapReduceをラップし直感的な業務ロジック実装が可能

一般的なJavaスキルで十分開発可能な直感性

5．評価結果

バッチのIn/OutをExcelで定義し、自動テスト

JOB管理ツールからの呼出を受けるAPIシェル

Stage情報やデータの行情報等が詳細に出力されるログ

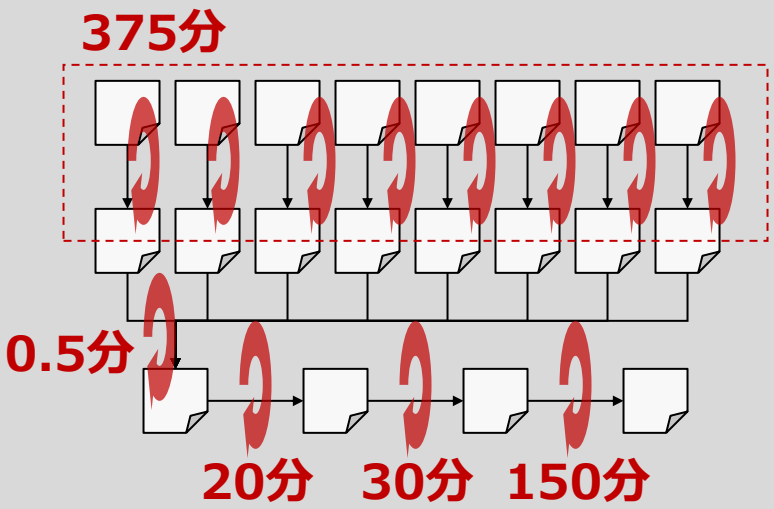
業務バッチアプリ開発に
有用な周辺機能が充実

**充実したテスト機能と
バッチ実行/ログ機能**

5．評価結果

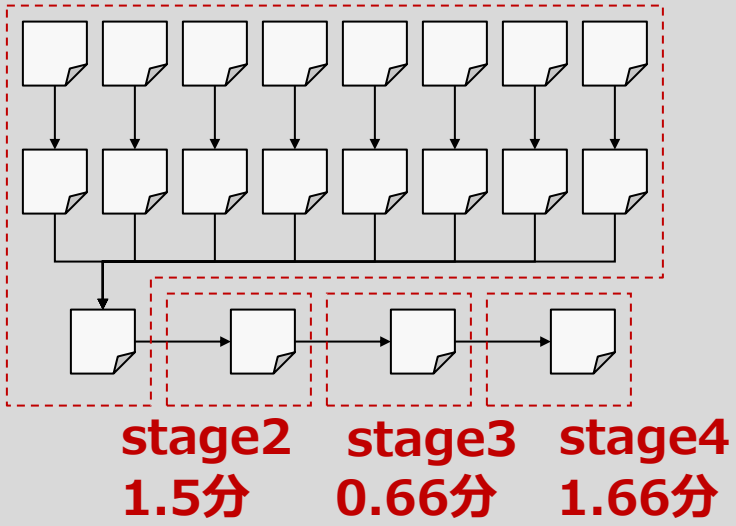
Ruby + RDBMS

合計：575.5分



Asakusa Framework

合計：7.3分



並列分散処理環境を
最大限活かす最適化機能

**高度な最適化機能と
便利な視覚化機能**

5．評価結果

MapReduceをラップし
直感的な業務ロジック実装
が可能

→ 一般的なJavaスキルで
十分開発可能な直感性

業務バッチアプリ開発に
有用な周辺機能が充実

→ 充実したテスト機能と
バッチ実行/ログ機能

並列分散処理環境を
最大限活かす最適化機能

→ 高度な最適化機能と
便利な視覚化機能

即利用可能な充実したバッチ開発支援機能

6．印象の変化

業務バッチをHadoopで稼動させるニーズがあるのか？	課題はあるが確かにメリットは大きい
独自言語としてのDSLを習得しなければならないのか？	DSL習得の欠点は少ないJavaのFWとして扱える
国産OSSは普及するのか？	OSSコミュニティを盛り上げるのはユーザー

当初の印象は大きく改善し、課題を具体的に把握

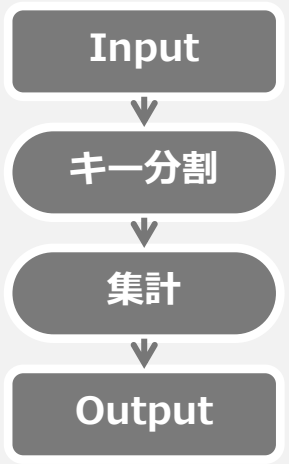
（参考）AsakusaDSL から MapReduce への変換

プログラムからDAGを中間生成 → MapReduceプログラム生成
⇒ Apache Spark と同様の内部アーキテクチャ

Asakusa DSL

```
// キー毎にグルーピング
// されたリスト内で
// 合計値を算出
groupSort(
  @group("key")list,
  result){
  for(value in list){
    sum += value
  }
  result.add(sum)
}
```

DAG(Directed Acyclic Graph)



MapReduceプログラム

```
// リスト内の情報をキーと値に分割
map(key,value,output){
  output(value.key,
         value.value)}

// キー毎にグルーピングされた
// リスト内で合計値を算出
reduce(key,values,output){
  for(value in values){
    sum += value
  }
  output(key,sum)}
```

グローバルスタンダードな内部アーキテクチャ
⇒ DSLを利用していてもガラパゴス化の懸念は少ない

7．課題

学習コストが高い 独習での習得は困難	情報は充実しつつある 勉強会・トレーニングも 活用可能
Hadoopへデータ転送コストで Asakusaのメリットを 失いかねない	WindGate、Sqoopの活用 や様々な工夫が可能
そもそも企業システムの中で Hadoop全体をどう活用するか	Hadoopへの正確な理解と メリット/デメリットへの 正しい判断が必要

Hadoopをどう活用するかに行き着く

7. 課題

Asakusa Framework 情報の入手方法

- 公式ドキュメント http://asakusafw.s3.amazonaws.com/documents/latest/release/ja/html/index.html
- メーリングリスト http://www.asakusafw.com/community/mlinfo.html
- 勉強会 http://asakusafw.connpass.com/
- トレーニング http://www.nautilus-technologies.com/service/training.html
- ブログ http://www.adventar.org/calendars/200

学習コストが高い
独習での習得は困難

情報は充実しつつある
勉強会・トレーニングも
活用可能

7．課題

- **WindGate**

  RDBやHDFSファイル等のデータインターフェースとAsakusaバッチをシームレスに連携

- **Sqoop**

  RDBとHDFS間のデータ転送を実現

- **工夫案**

  バッチに利用するカラムだけをHDFSに取り込む、等

Hadoopへデータ転送コストで Asakusaのメリットを 失いかねない

WindGate、Sqoopの活用や様々な工夫が可能

7．課題

- **RDBMSとは何が違うのかの理解**

  大きなIO処理単位を扱うアーキテクチャのため、ディスクシステムをシンプルにでき、ひいてはデータマートもシンプルにできる

- **誤った固定概念の払拭**
  - 特性をよく理解すればクエリエンジンも有用
  - ディストリビューションによってはバックアップ機能も充実
  - 自動テスト環境をバックアップ環境とする工夫案もある
- **「ビッグデータOS」としての存在感**

  クエリエンジンではなく、多様なデータ活用手段を提供するプラットフォーム

そもそも企業システムの中でHadoop全体をどう活用するか

Hadoopへの正確な理解とメリット/デメリットへの正しい判断が必要

7．課題

学習コストが高い 独習での習得は困難	情報は充実しつつある 勉強会・トレーニングも 活用可能
Hadoopへデータ転送コストで Asakusaのメリットを 失いかねない	WindGate、Sqoopの活用 や様々な工夫が可能
そもそも企業システムの中で Hadoop全体をどう活用するか	Hadoopへの正確な理解と メリット/デメリットへの 正しい判断が必要

Hadoopをどう活用するかに行き着く

8．まとめ

Hadoop業務バッチを開発する手段として唯一の選択肢

Hadoop自体をどう活用するかの検討が肝要

OSSはユーザー自身が盛り上げていくことができる

Hadoop + Asakusa で
新しい企業情報システムのあり方を考える

ご清聴ありがとうございました