分散処理アプリ演習　第4回
MapReduceによるレコメンデーションエンジンの実装

（株）NTTデータ

講義内容

1. 導入
   - MapReduceの代表的な適用領域（集計・統計）
2. MapReduceによる集計・統計処理の応用例～レコメンデーションエンジン～
   - レコメンデーションとは、協調フィルタとは、確率による相関の度合いの算出
3. レコメンデーションエンジンの実装
   - 確率モデルによるレコメンデーションエンジンの実装

1. 導入

MapReduceの代表的な適用領域とは？

MapReduceの代表的な適用領域には
どんなものがあるか？

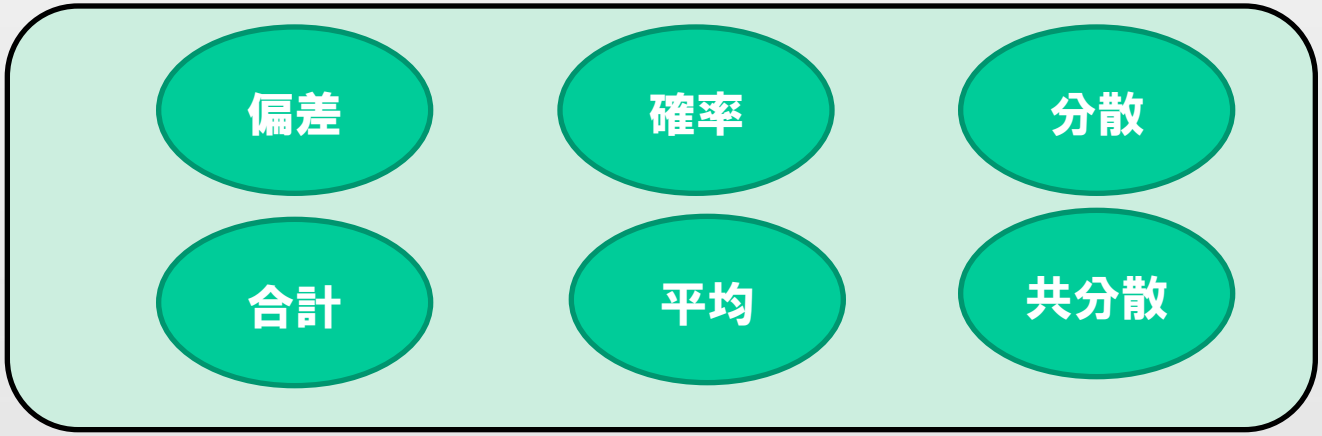
MapReduceの代表的な適用領域（集計・統計）

- MapReduceは大量データの集計・統計処理に適した計算モデルである

**統計処理 = 大量のサンプルを必要に応じてフィルタ/加工し、
それらから全体の要約を求める処理**

**Mapタスクで大量のデータをフィルタ/加工。
Reduceタスクでデータ間の関係から全体の要約を求める**

集計・統計処理の例



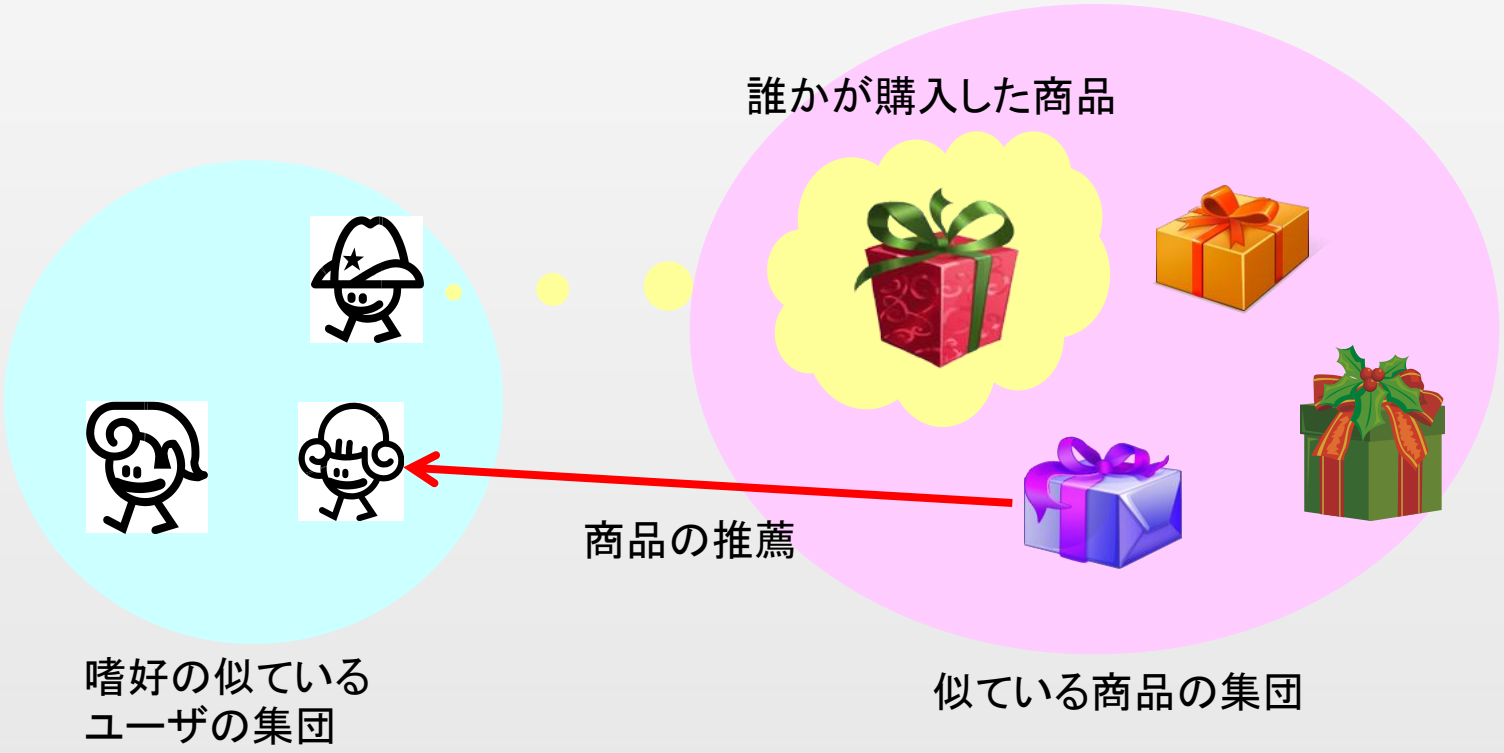
本講義で実施する内容

本講義では集計・統計処理の応用例として、MapReduceでレコメンデーションエンジンを実装する

2. MapReduceによる集計・統計処理の応用例 ～レコメンデーションエンジン～

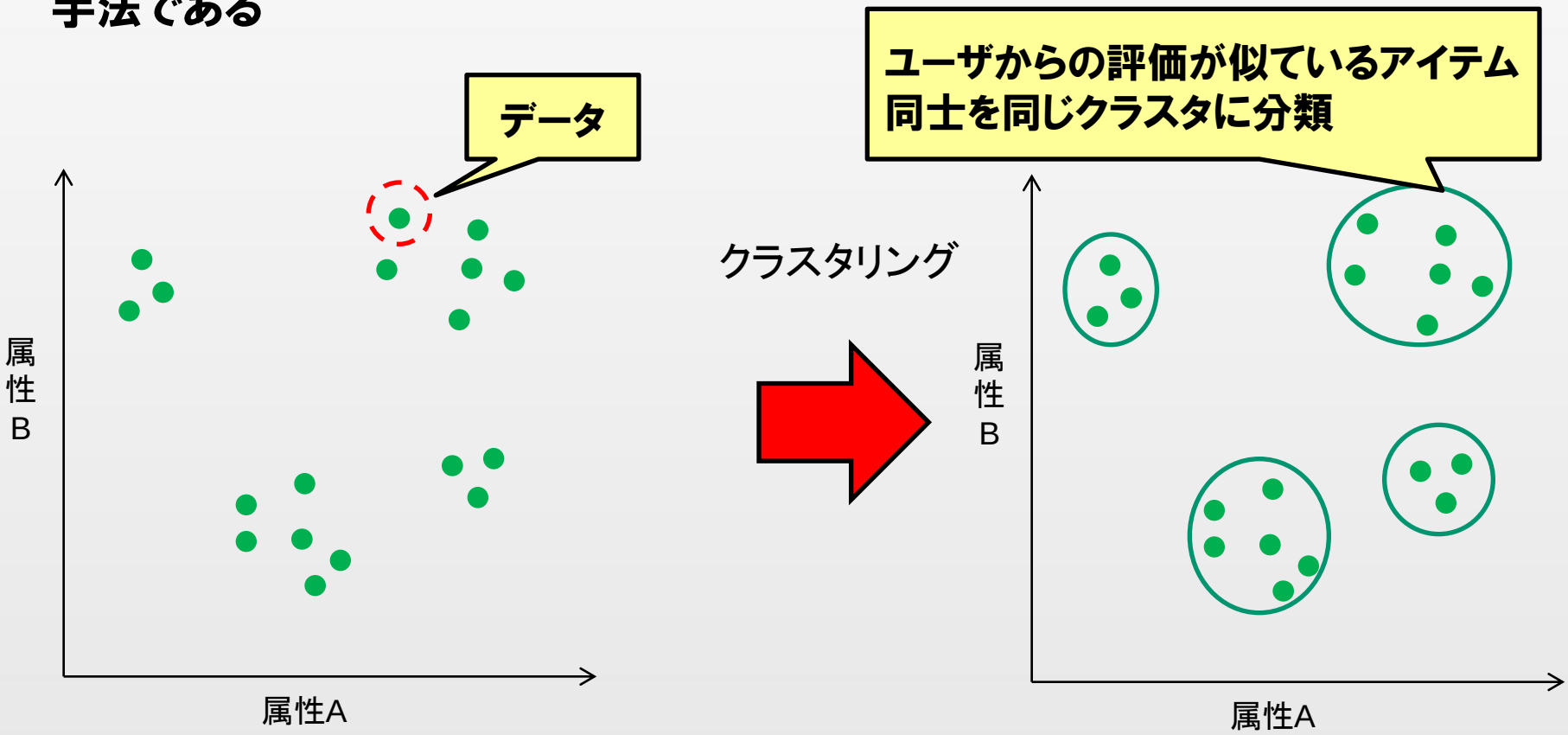
レコメンデーションとは？

- ECサイトなどで、利用者ごとの嗜好を分析して、商品やコンテンツを推薦する仕組である
- 似たような傾向を持つ利用者や、似た性質を持つアイテムをクラスタリングする「協調フィルタ」などが代表的な手法である



協調フィルタとは？

- 協調フィルタとはアイテムやユーザを表すデータを、ユーザとアイテム間のインタラクションをもとに計算した相関や類似度と呼ばれる相関の度合いを表す指標を用いて、関連性のあるユーザ同士、アイテム同士を分類（クラスタリング）する手法である



相関の度合いを求める方法

- レコメンデーションではユーザとユーザ、またはアイテムとアイテム同士の「相関の度合い」を計算する。相関の度合いを求める方法としては「確率」、「ピアソンの相関係数」や「コサイン距離」、「ユークリッド距離」といった指標が用いられる

- 如何に精度の高い相関を見つけられるかがレコメンデーションのポイント

確率による相関の度合いの算出

- 確率モデルを協調フィルタに応用する場合は、例えば商品Aが購入されたとき、一緒に商品Bが購入される確率が高ければ、2つの商品は相関の度合いが大きいというモデルを構築することができる
- 全ユーザの購入履歴から商品Aを購入したユーザが、一緒に商品Bを購入する確率を求める場合、次のように求める

$$\frac{\text{商品Aと商品Bが一緒に購入された回数}}{\text{商品Aとほかの商品が一緒に購入された回数}}$$

- 逆に、商品Bを購入したユーザが商品Aを購入する確率は次のように求められる

$$\frac{\text{商品Aと商品Bが一緒に購入された回数}}{\text{商品Bとほかの商品が一緒に購入された回数}}$$

相関の度合いを求めてみよう

- **例えば図のように、一緒に購入された商品のペアのデータが与えられたとき、商品Aを購入した人が商品Bを購入する確率は次の通り**

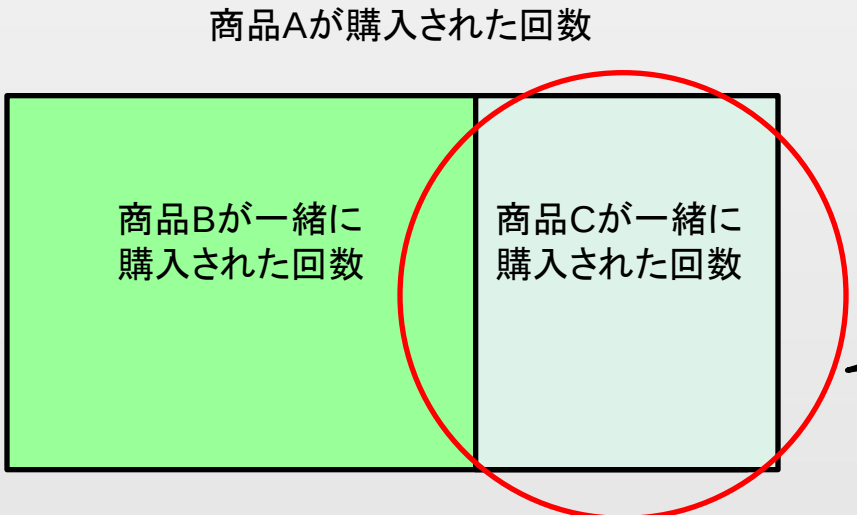
一緒に購入された商品のペア

(商品A，商品B)
(商品B，商品C)
(商品A，商品C)
(商品A，商品B)

$$\frac{\text{商品Aと商品Cを購入した回数}}{\text{商品Aと商品Bを購入した回数}+\text{商品Aと商品Cを購入した回数}}$$

$$=\frac{1}{2+1}$$

$$=\frac{1}{3}$$



商品Aが購入された全体のうち、この部分を求める

3. レコメンデーションエンジンの実装

演習内容

- 別紙「レコメンデーションシステム仕様」と「レコメンデーションシステムデータ仕様」をもとに、レコメンデーションエンジンを完成させる
- 問題設定は次の通り
  - ある菓子店では、顧客が同時に購入する商品に着目し、商品間の潜在的な関係を見つけ、商品を購入しようとしているユーザにほかの商品を推薦するシステムの導入を検討している。
  - レコメンデーションエンジンはユーザに商品をレコメンドするフロントエンド部と、商品間の相関を計算するバックエンド部から構成され、フロントエンドはすでに完成している
  - バックエンドは3つのMapReduceジョブから構成され、ひとつは実装済みで、残り2つのジョブは途中まで実装している。残りの部分を実装し、完成させる
  - 3つのジョブはドライバプログラムから起動する。ドライバプログラムはすでに実装済みである

作成するプログラム

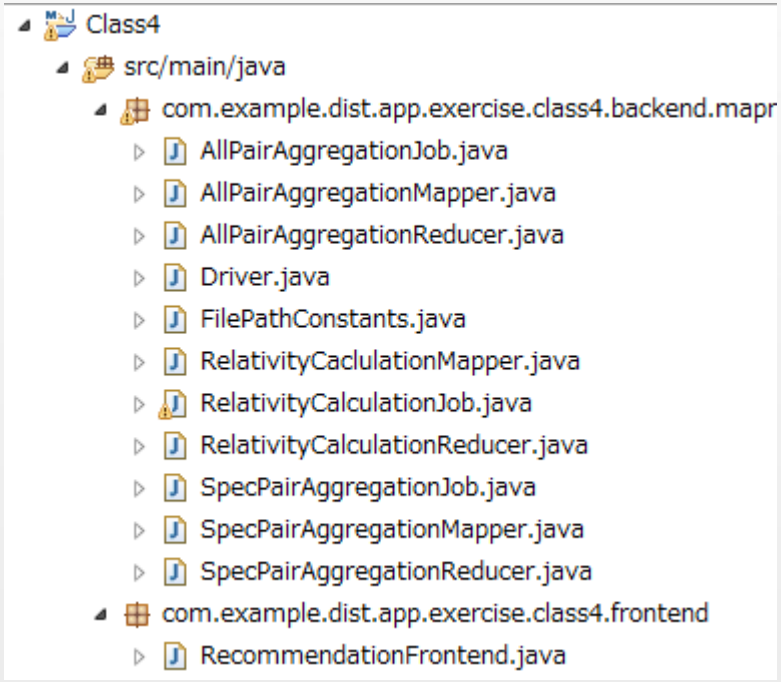
- 前スライドの通り、3つのジョブを完成させる。未完成の2つのジョブを完成させる。3つのジョブの役割は次の通り
  - ペア総数集計ジョブ
    - AllPairAggregationMapper.java（Mapperクラスのソースファイル）
    - AllPairAggregationReducer.java（Reducerクラスのソースファイル）
    - AllPairAggregationJob.java（Jobクラスのソースファイル）
  - 特定ペア集計ジョブ
    - SpecPairAggregationMapper.java（Mapperクラスのソースファイル）
    - SpecPairAggregationReducer.java（Reducerクラスのソースファイル）
    - SpecPairAggregationJob.java（Jobクラスのソースファイル）
  - 関連度計算ジョブ（実装済み）
    - RelativityCalculationMapper.java（Mapperクラスのソースファイル）
    - RelativityCalculationReducer.java（Reducerクラスのソースファイル）
    - RelativityCalculationJob.java（Jobクラスのソースファイル）

赤字で表示されているソースファイルを完成させる

※未実装部分にはソースファイルにコメントで//TODOと書かれている

演習環境

- EclipseのClass4プロジェクト内に、必要な資材がそろっている

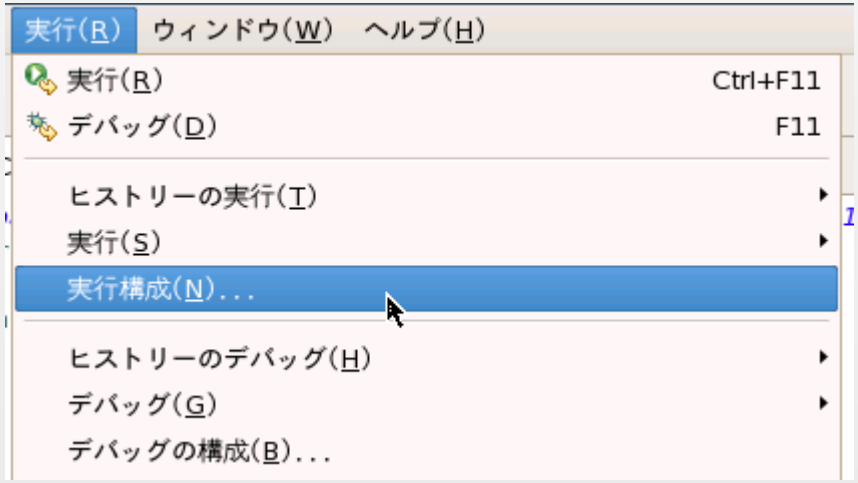


「Class4」プロジェクト内の
「src/main/java」ディレクトリ内に
演習対象のソースコードが格納されている

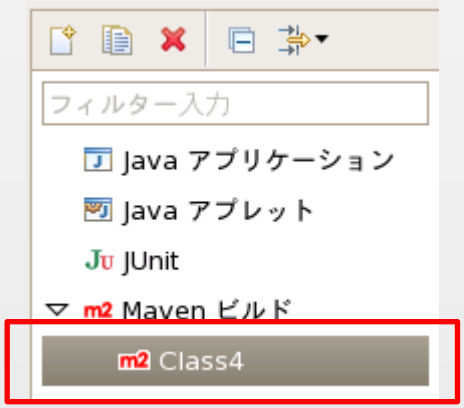
実行バイナリ(Jarパッケージ)の作成方法

- 作成したジョブをコンパイルし、実装済みのドライバプログラム(Driver.class)をまとめてreccomendation-0.1.jarという名前でjarパッケージにまとめる

Eclipseのメニューから
[実行]→[実行構成…]を選択



出現したダイアログボックスの左側にある「Maven ビルド」から、「Class4」を選択し、ダブルクリック



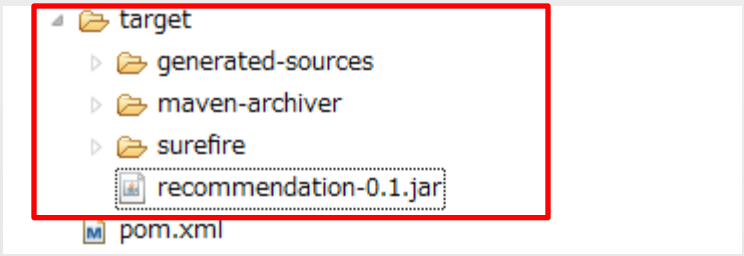
※ Eclipse上での操作です

実行バイナリ（Jarパッケージ）の作成方法［つづき］

- コンパイル／パッケージングが始まる。ここでコンパイルエラーなどがある場合はEclipseのコンソールに表示される

```
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 2.109s
[INFO] Finished at: Mon Mar 05 14:20:34 JST 2012
[INFO] Final Memory: 7M/17M
[INFO] ------------------------------------------------------------------------
```

- ビルド成功すると、「Class4」プロジェクト内の「target」ディレクトリに、recommendation-0.1.jarが作成される



実行方法（バックエンド部）

① ペア総数集計ジョブを実行する前に、コマンドラインで、ローカルファイルシステム内の購入ペアデータ（/root/hadoop_exercise/4/data/goods_pair）をHDFSに格納する

```
# # これはコマンドライン上での操作です
# hadoop fs -put /root/hadoop_exercise/04/data/goods_pair ¥
 hadoop_exercise/04/data/goods_pair
```

② 3つのジョブはそれぞれ別々に実行できるようにドライバプログラムが実装されている。ドライバプログラムの実行時の引数で、実行するジョブを選択できる。ドライバプログラムは下図のようにコマンドラインから実行する。**関連度計算ジョブはほかの2つのジョブが終了しないと、必要なファイルが生成されず、実行できないので注意すること。**

引数	説明
allpair	ペア総数集計ジョブを実行
specpair	特定ペア集計ジョブを実行
relativity	関連度計算ジョブを実行
なし	一連のジョブを実行

```
# # これはコマンドライン上での操作です
# #ペア総数集計ジョブを実行する場合
# hadoop jar ~/workspace/Class4/target/recommendation-0.1.jar ¥
 com.example.dpap.class04.backend.Driver  allpair
```

実行方法（バックエンド部）［つづき］

- 各ジョブが実行されると、HDFS上の/user/root/hadoop_exercise/04/data以下に次のファイルが作られる

ジョブ	説明
ペア総数集計ジョブ	denomination
特定ペア集計ジョブ	numerator
関連度計算ジョブ	related_goods

- コマンドラインから、これらのファイルを確認することができる

```
# # これはコマンドライン上での操作です
# 特定ペア集計ジョブ実行後、生成されたファイルの中身を確認する場合
# hadoop fs -cat "/user/root/hadoop_exercise/04/data/numerator/part-*"
```

「/part-*」を忘れないこと

バックエンド部実行時の注意

- HDFS上のファイルは上書きができない。よって、以前にジョブが生成したファイルが残っている状態で連続してジョブを実行すると、ジョブが失敗する
- ジョブを再実行する際には、コマンドラインから、各ジョブが生成したHDFS上のファイルを削除すること

```
# # これはコマンドライン上での操作です。

# #ディレクトリ内に、ジョブが作成したファイルがあるかどうか確かめる
# hadoop fs -ls hadoop_exercise/04/data/

# #ペア総数集計ジョブが作成したファイルを削除する
# hadoop fs -rmr hadoop_exercise/04/data/denomination

# #特定ペア集計ジョブが作成したファイルを削除する
# hadoop fs -rmr hadoop_exercise/04/data/numerator

# #関連度計算ジョブが作成したファイルを削除する
# hadoop fs -rmr hadoop_exercise/04/data/related_goods
```

実行方法（フロントエンド部）

- 全てのジョブを実行した後に生成される「関連商品データ（related_goods）」をもとに、フロントエンドがレコメンデーションを実行できる
- フロントエンドの実行前に一度だけ、HDFS上に「商品リスト（goods_list）」を格納する

```
# # これはコマンドライン上での操作です
# hadoop fs -put ~/hadoop_exercise/04/data/goods_list  ¥
 hadoop_exercise/04/data/goods_list
```

- 商品リストの格納後、フロントエンドはコマンドラインで次のように起動する

```
# # これはコマンドライン上での操作です
# java -cp .:/root/workspace/Class4/target/recommendation-0.1.jar:¥
/usr/lib/hadoop/conf:¥
/usr/lib/hadoop/hadoop-core.jar:¥
/usr/lib/hadoop/lib/commons-logging-1.0.4.jar:¥
/usr/lib/hadoop/lib/log4j-1.2.15.jar:¥
/usr/lib/hadoop/lib/guava-r09-jarjar.jar ¥
  com.example.dpap.class04.frontend.RecommendationFrontend
```

実行方法（フロントエンド部）［つづき］

- フロントエンドの起動後、商品番号の入力を求められるので、リストの中から商品のいずれかを入力すると、関連する商品がレコメンドされる

```
商品番号          商品名
0         セサミクッキー
1         ホワイトチョコレート
2         ミルクチョコレート
3         ビターチョコレート
4         いもようかん
5         みたらし団子
6         水ようかん
7         ねりようかん
8         うぐいすあんぱん
9         あんドーナツ
10        チョコチップクッキー
11        きんつば
12        えびせん
13        イチゴのショートケーキ
14        ミルフィーユ
15        リンゴのタルト
16        いちごのタルト
17        フルーツロール
18        ごま団子
19        醤油団子
20        塩あんみつ
21        クリームあんみつ
22        みつまめ
23        ところてん
24        マカロン
25        マスクメロンシャーベット
26        オレンジピールチョコレート
27        マンゴーシャーベット
28        杏仁豆腐
29        マンゴープリン
30        ミルクプリン
31        カスタードプリン
32        焼きプリン
33        カスタードシュークリーム
34        生シュークリーム
35        和栗のモンブラン
36        抹茶ロール
商品番号を入力し、商品を選択してください[-1 = 終了 , -2 = 商品リストの再表示] >36
抹茶ロールとよく一緒に購入されている、おすすめの商品はこちらです。
醤油団子
生シュークリーム
水ようかん
```

解答例

- Eclipseプロジェクトの「Class4-Answer」内のソースコード参照

まとめ

本講義で学んだ内容

- 統計・集計処理はMapReduceの適用が有効な領域のひとつである
- 統計・集計処理の応用例としてのMapReduceの適用事例として、レコメンデーションエンジンがある
- レコメンデーションに利用されるアルゴリズムの一つ「協調フィルタ」において、ユーザ間/アイテム間の関連の度合いの指標を計算する部分に統計処理を適用できる