AITC
先端ＩＴ活用推進コンソーシアム

クラウド・テクノロジー研究部会

Mahoutを使ったプロトタイプ

2012/02/01

クラウド・テクノロジー研究部会 リーダー
アドソル日進株式会社
荒本道隆

AITC
先端ＩＴ活用推進コンソーシアム

クラウド・テクノロジー研究部会のご紹介

背景

AITC
先端ＩＴ活用推進コンソーシアム

- 多種多様なクラウド関連技術がオープンソースや製品として出てきているが、「クラウド」という言葉の定義自体が幅広いため、企業内システムでの適用できる領域・効果が分かりづらい。そこで、クラウド関連技術の幅広い技術の情報収集をおこないつつ、HadoopやNoSQLなどの中で利用可能なものを実際に試用し、それらを使って**プロトタイプを試作**してみることで、クラウドを企業内システムで活用するヒントや具体的なイメージを得るための活動を行う。

活動目的

- クラウド技術を企業内で利用するためのモデルケースやノウハウを調査、検討、プロトタイプ開発を通して習得し、そこで得た知見を勉強会・調査報告書を通して展開する。

活動内容

AITC
先端ＩＴ活用推進コンソーシアム

- 情報収集、事例研究
  - クラウド関連の実装や技術に関する情報を収集
- プロトタイプ作成
  - 企業内システムを前提としたプロトタイプを作成
  - Hadoop以外も対象にする
- 勉強会の開催
  - プロトタイプを実際に触ってみて、知見を共有

とにかく手を動かして試してみる

活動方法

- 月例ミーティングの開催
  - Face to Face での情報交換
  - 作ったものを持ち寄って、評価
- SNS上での情報交換
  - 2010年度はあまり活用できなかった
  - より活性化していく
- ハンズオン形式での勉強会を開催
  - 2010年度は、１回開催
  - 2011年度は、２～３回開催予定
  - 教える側の方が、よりスキルアップできる

クラウド技術を習得する上での壁

AITC
先端ＩＴ活用推進コンソーシアム

- 本を読むと、そんなに難しそうじゃないですが、
  - 経験がある人ほど、ここで止めてしまう
  - 実際に動かしてみると、色々と苦労するが、見えてくることも多い
    - Hadoop 0.20.2, Mahout 0.5, Hbase 0.92.0
- クラウドが必要になる大量データが社内に無い
  - あったとしても、扱いがデリケートなデータばかり
  - AITCで、大量データを提供してくれる先を探してみましょう
    - 気象庁、
- テーマ探し
  - とりあえず動かしてみる
  - 協働プロジェクト
- モチベーション
  - 実は、「発表会に向けて」が一番大きい
  - 勉強会の講師をしてみませんか？

2010年度の活動成果

AITC
先端ＩＴ活用推進コンソーシアム

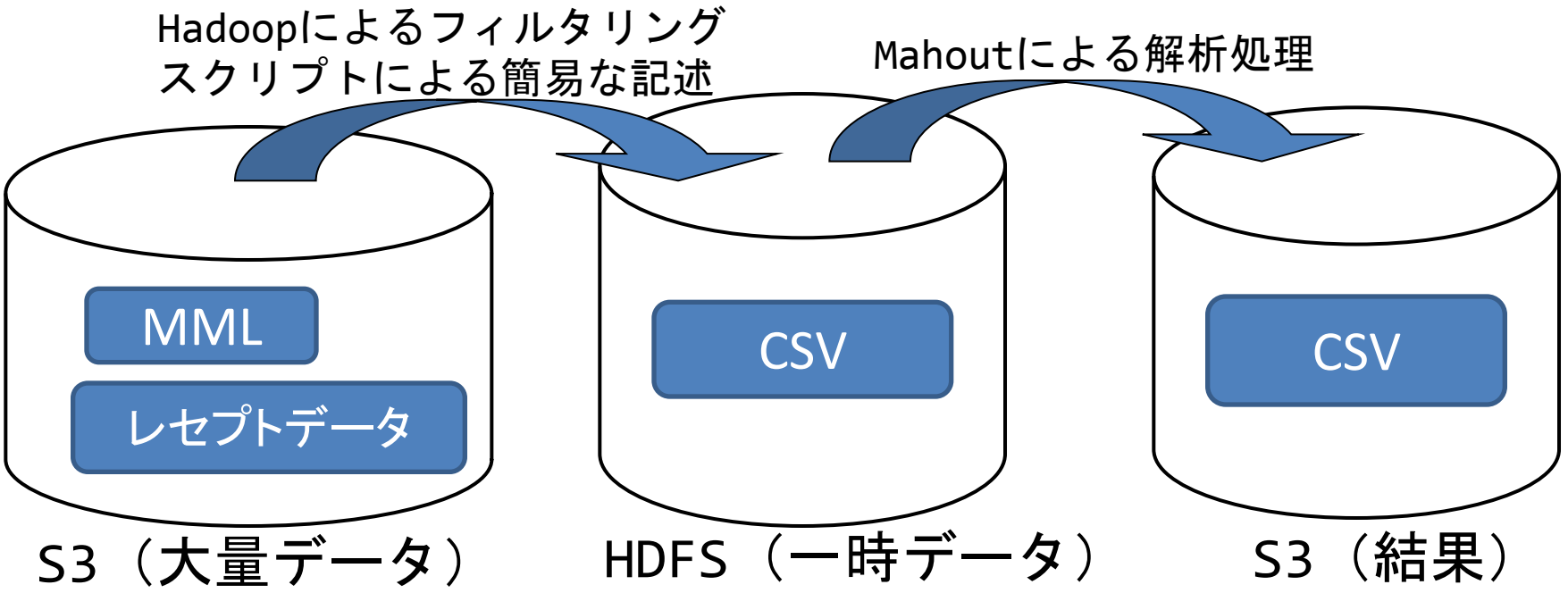
- ガイドラインの調査報告
  - クラウドセキュリティガイドラインと医療情報システムガイドラインの調査
  ～クラウド上で医療情報を活用するために～
- MMLを使ったプトロタイプ
  - 医療情報（MML3.0） Hadoopプログラミング
- Mahoutを使ったレコメンデーションのプロトタイプ
  - 本日紹介します
- 数値予報GPV
  - 気象資料（数値予報GPV）の活用検討
- GPVを使ったプロトタイプ
  - Key-Valueストアを用いたGPVデータ閲覧サービスの構築

AITC
先端ＩＴ活用推進コンソーシアム

Mahoutを使ったプロトタイプ

ゴールイメージ

- 蓄積されたデータを使って、Mahoutで解析
- まずは、お医者さんが利用することを想定
  - 患者側が利用することはできないだろうか？



Mahoutとは

- 機械学習・データマイニングのライブラリ
- Java＆オープンソース
  - Apache2.0ライセンス
- Hadoop上で動作
  - hadoop-coreが含まれているので、単体でも動作
- AmazonEMR上でも動作
  - MahoutDriver を指定して実行

Mahout機能一覧

- Clustering : クラスタリング : 似ているものをグループ化する
- Classification : 分類 : 分類・判別を行う
- Pattern Mining : パターンマイニング : 頻出パターンを抽出する
- Recommender : 推薦 : 興味に合うものを推薦する
- Regression : 回帰 : 数値予測を行う
- Dimension Reduction : 次元縮約 : 説明変数を重要な変数へ縮約
- Evolutionary Algorithms : 進化的アルゴリズム

参考URL： http://sites.google.com/site/mahoutjp/

- さらに詳細な情報
  - https://cwiki.apache.org/MAHOUT/algorithms.html

開発方法

- ローカルHadoopのスタンドアロンモードで開発
  - Linuxがあれば、hadoopのファイルを展開するだけ
- Amazon EMR（Elastic MapReduce）＋S3で実行
  - コマンド一発でHadoop実行環境が準備できる
  - S3の使い勝手も検証
- Streamingモードをできるだけ使用
  - フィルタリングをRubyで記述
    - Rubyは初心者です
      - Rubyらしくないコードでスイマセン

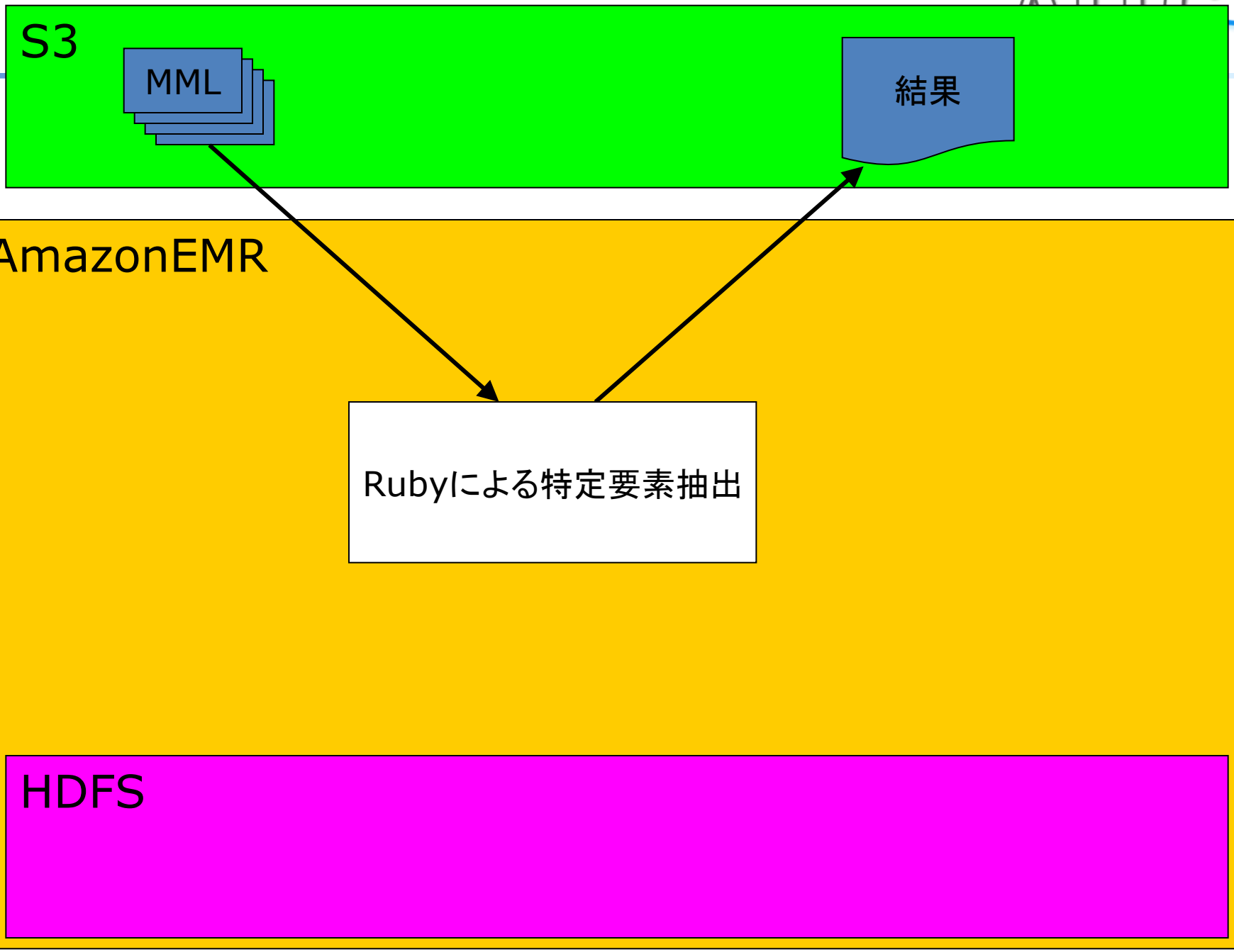
AITC
先端ＩＴ活用推進コンソーシアム

まずはMMLを使って AmazonEMRに慣れる

MMLから特定要素を抽出

AITC
先端ＩＴ活用推進コンソーシアム

- MMLとは
  - Medical Markup Language
  - 診察情報をXMLで記述したもの
- 処理概要
  - S3上に大量のMMLが保存されていると仮定
  - 特定の範囲のMMLだけ処理したい
  - MMLの中から特定の要素を抽出
  - 結果をS3上に保存
- まずはAmazonEMRに慣れる＆性能測定



Rubyによる特定要素抽出(xml)

```
#!/usr/bin/env ruby
require "rexml/document"
include REXML

string = ""
while line = STDIN.gets
    string += line
end
doc = Document.new string
# print doc


birthday = XPath.first( doc, "//mmlPi:PatientModule/mmlPi:birthday" )
print birthday.text.split("-")[0] + "¥t"

bodyWeight = XPath.first( doc, "//mmlFcl:birthInfo/mmlFcl:bodyWeight" )
print bodyWeight.text + ","
bodyHeight = XPath.first( doc, "//mmlFcl:birthInfo/mmlFcl:bodyHeight" )
print bodyHeight.text
```

実行方法－１

- Rubyによる特定要素抽出
  - ローカルHadoop

```
hadoop jar $HADOOP_HOME/contrib/streaming/hadoop-0.20.2-streaming.jar ¥
 -mapper "ruby mapper1.rb" ¥
 -input ./input/*/*/ ¥
 -output ./output1
```

  - AmazonEMR

```
elastic-mapreduce  -j $JOB_ID ¥
  --stream  --step-name “mapper1.rb" ¥
  --mapper s3://bucket名/mml/mapper1.rb ¥
  --reducer s3://bucket名/mml/red_dummy.rb ¥
  --input    s3://bucket名/mml/input/*/*/ ¥
  --output  s3://bucket名/mml/output1
```

やってみて気付いた事－１

- Javaによる実装より簡単だった
- AmazonEMRは「Reduceなし」だとエラーになる
  - 何もしない red_dummy.rb を作成
- inputにワイルドカードを指定できる
  - s3://bucket名/mml/input/2011/09/
  - s3://bucket名/mml/input/2011/*/
  - s3://bucket名/mml/input/*/09/
  - s3://bucket名/mml/input/*/*/
  - S3上へデータを格納
    - 「hadoop distcp」を使用

やってみて気付いた事－２

- ファイル数＝map数
  - インスタンス数を増やせば、単純計算で時間短縮
    - AmazonEMRのsmallインスタンスで測定
    - データはS3に格納
  - 100ファイル（4,517,102Byte）
    - 1インスタンス：16分
    - 1+1インスタンス：10分
    - 1+5インスタンス：3分
    - 1+10インスタンス：2分
  - 1,000ファイル（45,166,848Byte）
    - 1インスタンス：227分
    - 1+1インスタンス： 100分
    - 1+5インスタンス：23分
    - 1+10インスタンス：12分

参考値－1

- AmazonEMRのsmallインスタンスで測定
  - データはS3に格納
  - 100ファイル（4,517,102Byte）
    - 1インスタンス：16分
    - 1+1インスタンス：10分
    - 1+5インスタンス：3分
    - 1+10インスタンス：2分
    - EMR上でHadoop、データはS3：9分
    - EMR上でHadoop、データはローカル：9分
    - ローカルHadoop（PC使用）、データはローカル：2分
    - ローカルシェル（PC使用）、データはローカル：2分

参考値－2

- AmazonEMRのsmallインスタンスで測定
  - データはS3に格納
  - 1,000ファイル（45,166,848Byte）
    - 1インスタンス：227分
    - 1+1インスタンス： 100分
    - 1+5インスタンス：23分
    - 1+10インスタンス：12分
    - EMR上でHadoop、データはS3：84分
    - EMR上でHadoop、データはローカル：77分
    - ローカルHadoop（PC使用）、データはローカル：19分
    - ローカルシェル（PC使用）、データはローカル：17分

コスト比較:もしも10万ファイルだったら

- AmazonEMRのsmallインスタンス(0.115$/時)
  - 参考値－2の数値を100倍にして計算
  - 100,000ファイル(4,516,684,800Byte)の予想値
    - 1インスタンス:22700分(379時間)
      - 0.115x1x379=43.585$
    - 1+1インスタンス: 10000分(167時間)
      - 0.115x2x167=38.41$
    - 1+5インスタンス:2300分(39時間)
      - 0.115x6x39=26.91$
    - 1+10インスタンス:1200分(20時間)
      - 0.115x11x20=25.3$

参考値－3（頑張ってチューニング後）

AITC 先端ＩＴ活用推進コンソーシアム

- AmazonEMRのsmallインスタンスで測定
  - Ruby+DOM+XPath → grep+sed+awkでSAXもどき
  - 1,000ファイル（45,166,848Byte）
    - 1インスタンス：227分 → 83分
    - 1+1インスタンス：100分 → 51分
    - 1+5インスタンス：23分 → 10分
    - 1+10インスタンス：12分 → 8分
    - EMR上でHadoop、データはS3：84分 → 50分
    - EMR上でHadoop、データはローカル：77分 → 50分
    - ローカルHadoop（PC使用）、データはローカル：19分 → 5分
    - ローカルシェル（PC使用）、データはローカル：17分 → 0.5分

AITC
先端ＩＴ活用推進コンソーシアム

レセプトデータを使った
レコメンデーション

レセプトデータを使ったレコメンデーション

AITC
先端ＩＴ活用推進コンソーシアム

- S3上にレセプトデータがあると仮定
  - DPC履歴ファイル
  - 処方履歴ファイル
  - などなど
- Mahoutによるレコメンデーションを実行
- 結果をS3に保存

レセプトデータについて

- DPC履歴ファイル
  - 施設CD,患者ID,入院年月日,退院年月日,分類番号,....
- 処方履歴ファイル
  - 患者ID, 医師ID,処方年月日,医薬品コード,...
- この2ファイルを患者IDでJOIN
  - JOIN結果を入力データとする

レコメンデーション

- Mahoutのレコメンデーションの入力フォーマット
  - ユーザーID(long),アイテムID(long),評価値(double)
    - 患者ID→ユーザーID
    - 医薬品コード→アイテムID
    - 評価値はどう付けるか？→とりあえず1
- 入力データから、特定の要素だけ取り出す
  - 患者ID,医薬品コード,評価値
  - 分類番号の前6文字（傷病名の分類）,医薬品コード,評価値
- Webサーバのアクセスログなどにも適用できる
  - 127.0.0.1 – **aramoto** [26/Sep/2011 11:00:00 +0900] “GET /hogehoge/**index.html**” 200 564 – “Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1…)
  - 127.0.0.1 – **aramoto** [26/Sep/2011 11:00:05 +0900] “GET /hogehoge/download?uuid=**1234-5678-9abcd**” 200 24271 – “Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1…)

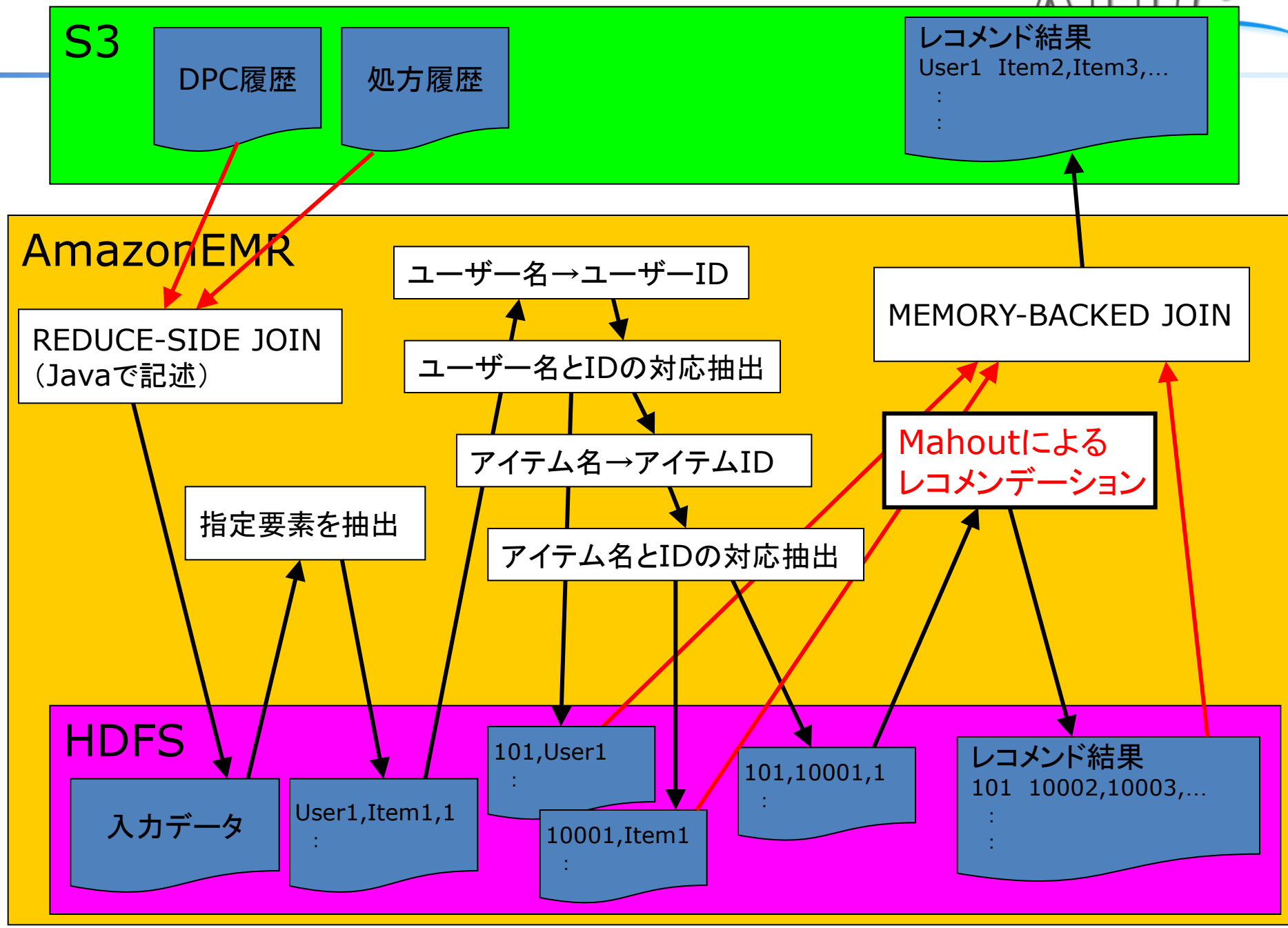
技術的な課題と解決方法

- 複数の入力ファイルをJOIN
  - REDUCE-SIDE JOIN, MAP-SIDE JOIN
    - 参考URL：http://code.google.com/p/try-hadoop-mapreduce-java/source/browse/trunk/try-mapreduce/src/main/java/jp/gr/java_conf/n3104/try_mapreduce/JoinWithDeptNameUsingReduceSideJoin.java
    - キーが**int**なので、**long**に修正
  - MEMORY-BACKED JOIN
    - HDFS上のファイルをメモリに取り込む

      userfile = `hadoop fs -cat  hdfs:///work/output_uid/part-*`

      users = Hash[*(userfile.gsub("¥t","").gsub("¥n",",").split(","))]
- 「ユニークなID」をどう生成するか？
  - mapred.task.id から数字部分だけ抽出し、そこから採番
    - 値：attempt_local_0001_m_000017_0

    id = (ENV["mapred_task_id"].scan(/[0-9]/).join("").to_i % 1000000000000) * 2 * 1000000



Rubyによる特定要素抽出(csv)

```
#!/usr/bin/env ruby

while line = STDIN.gets
        line = line.gsub("¥n","")
        if line.length == 0 then next end
        if line[0,1] == "#" then next end

        params = line.split(",")
        print params[0] + "," + params[5] + "," + "1" + "¥n"
end
```

必要に応じて変更

実行方法－１

- Streamingを使用
  - ローカルHadoop

```
hadoop jar $HADOOP_HOME/contrib/streaming/hadoop-0.20.2-streaming.jar ¥
 -mapper "ruby map_uid.rb" ¥
 -reducer "ruby red_uid.rb" ¥
 -input ./input ¥
 -output ./output1
```

  - AmazonEMR

```
elastic-mapreduce  -j $JOB_ID ¥
 --stream  --step-name "map_uid.rb" ¥
 --mapper s3://com.gol.aramoto.aitc/mml/map_uid.rb ¥
 --reducer s3://com.gol.aramoto.aitc/mml/red_uid.rb ¥
 --input  hdfs:///input ¥
 --output hdfs:///output1
```

実行方法－2

- Mahoutによるレコメンデーション
  - ローカル

```
mahout recommenditembased ¥
 --input ./output3 ¥
 --output ./output4
```

  - AmazonEMR

```
elastic-mapreduce -j $JOB_ID ¥
 --jar s3://bucket名/jars/mahout-core-0.5-job.jar ¥
 --main-class org.apache.mahout.driver.MahoutDriver ¥
 --step-name "Recommend ItemBased" ¥
 --arg recommenditembased ¥
 --arg --input  --arg hdfs:///output3 ¥
 --arg --output --arg hdfs:///output4
```

実行結果

- Mahoutによるレコメンデーションの結果

```
073xxxxx56          [62xxxxx22:1.0,62xxxxx80:1.0,62xxxxx72:1.0,64xxxxx43:1.0,64xxxxx018:1.0,
64xxxxx017:1.0,64xxxxx01:1.0,64xxxxx01:1.0,62xxxxx01:1.0,62xxxxx01:1.0,]
077xxxxx83          [62xxxxx41:1.0,62xxxxx22:1.0,62xxxxx80:1.0,64xxxxx62:1.0,64xxxxx43:1.0,
64xxxxx18:1.0,64xxxxx17:1.0,64xxxxx01:1.0,64xxxxx01:1.0,62xxxxx01:1.0,]
086xxxxx91          [62xxxxx41:1.0,62xxxxx80:1.0,62xxxxx72:1.0,64xxxxx62:1.0,64xxxxx43:1.0,
64xxxxx18:1.0,62xxxxx01:1.0,64xxxxx01:1.0,64xxxxx54:1.0,64xxxxx25:1.0,]
130xxxxx58          [62xxxxx41:1.0,62xxxxx80:1.0,64xxxxx62:1.0,64xxxxx43:1.0,64xxxxx18:1.0,
64xxxxx01:1.0,64xxxxx01:1.0,62xxxxx01:1.0,64xxxxx01:1.0,64xxxxx70:1.0,]
:
```

- 各種マスターファイルがWeb上で入手可能

  http://www.iryohoken.go.jp/shinryohoshu/downloadMenu/

マスター			対象		
マスター名	件数	最終更新日	医科	歯科	調剤
医科診療行為マスター（279KB）	5,705件	平成23年10月14日	○	○	
医薬品マスター（701KB）	18,230件	平成23年 9月30日	○	○	○
特定器材マスター（34.4KB）	971件	平成23年 9月 9日	○	○	○
傷病名マスター（1.00MB）	23,523件	平成23年 9月30日	○	○	
修飾語マスター（43.1KB）	1,979件	平成23年 9月30日	○	○	
コメントマスター（8.46KB）	319件	平成22年 4月 1日	○	○	○
歯科診療行為マスター		平成23年 9月12日		○	
歯式マスター（6.33KB）	915件	平成23年10月 3日		○	
調剤行為マスター（4.68KB）	97件	平成22年 9月22日			○

やってみて気付いた事

- ローカルのHadoopとAmazonEMRの違い
  - ログがすぐに確認できない
  - IDの素とした mapred.task.id のパターンが違う
    - ローカル：attempt_local_0001_m_000017_0
    - EMR：attempt_201109241353_0004_r_000000_0
  - EMRは「Reduceなし」だとエラーになる
- 変更したrbファイルのS3へのアップロード
  - s3sync を使用すると楽だけど
- Mahoutによるレコメンデーション
  - 出力先とtempを毎回削除しないと、エラーになる

    ローカル：rm -rf temp

    EMR：elastic-mapreduce -j $JOB_ID

    --ssh "hadoop fs -rmr hdfs:///user/hadoop/temp”

感想－1

AITC 先端ＩＴ活用推進コンソーシアム

- Hadoopは、何をしても分単位の時間がかかる
  - ローカルで効率良く開発
  - ちょっとしたミスがあると、数十分後に後悔
  - 「自動的に実行状況を表示」するツールを作成
  - 「終わったら、EMRをシャットダウン」するツールを作成
  - あいた時間に何をするか考えてから測定
- S3にファイルを置くのに時間がかかる
  - Hadoop distcp, s3sync, Firefoxプラグイン
- １台での実行に限定すれば
  - ローカルPCで実行した方が早い
  - データ量が爆発した時に必ず困る
  - → 事前にHadoopにしておけば安心

感想－2

- Amazon EMR+S3の組み合わせは便利
  - S3上に公開データがあれば、とても利便性が高い
  - 複数のEMRを起動し、同時に別条件で測定
  - EMR上のファイルは、EMRを落とすと自動で消える
- 様々なデータをS3に置いて公開して欲しい
  - 防災情報XML、GPV、…

今後の展開について

AITC
先端ＩＴ活用推進コンソーシアム

- このモデルに、レコメンデーションは合わない？
  - 実データから抽出したテストデータで実行してみたけど、
    - あまり良い結果は得られなかった
  - お医者さんは、薬の知識がちゃんと頭に入っているはず
    - このシステムを必要とする医者には、診察して欲しくない
  - データ検索や社内文章システムのログ解析では使えそう
- 次は、クラスタリングにトライする予定
  - 例えば、請求書に書く病名は、
    - 病名によって薬を処方するのではなく、
    - 薬の処方によって、請求書の病名が決まるらしい
    - 過去のレセプト情報を元に、薬の処方から病名の候補を出す
  - さまざまなデータの分類に転用できそう

次回活動予定

- 次回のクラウド･テクノロジー研究部会は
  - 2012/02/17（金）14:00～18:00
  - アドソル日進株式会社（品川駅から徒歩15分）
  - 内容
    - Hadoop
    - OpenSocial
    - クラウド時代のセキュリティ
    - AmazonWSのミニ勉強会
    - 協働プロジェクトについての検討

- 部会への参加をお待ちしております
  - AITC会員企業の方：参加申し込みをしてください
  - AITC会員企業以外の方：AITC事務局にご連絡ください