

KLabでのEMR活用

高田敦史

自己紹介

高田敦史(takada-at)

開発本部データ分析グループ

Kラボラトリー所属

KLabソーシャルゲーム最初期からソーシャルゲームの開発にかかわる

→横断的に多くのゲームの開発にコミット

現在はソーシャルゲームのデータ分析が主なミッション

データ分析にかかわる技術

(1)データ抽出(ログ/DB)

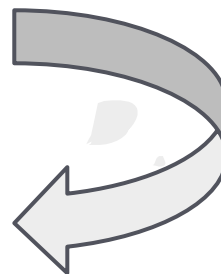


(2)データ蓄積・加工



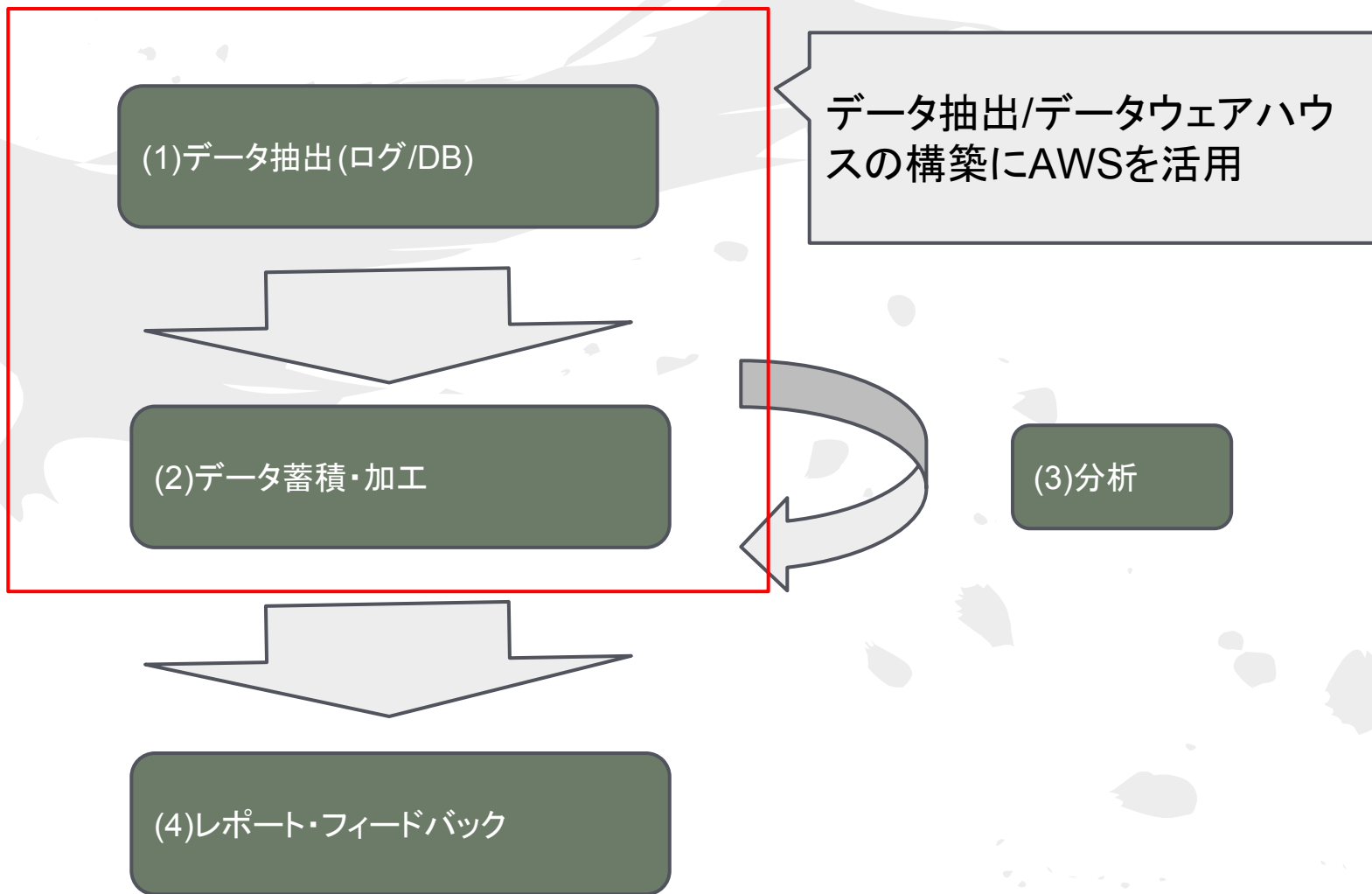
(4)レポート・フィードバック

データが効果を発揮するために必要なプロセス



(3)分析

データ分析にかかわる技術

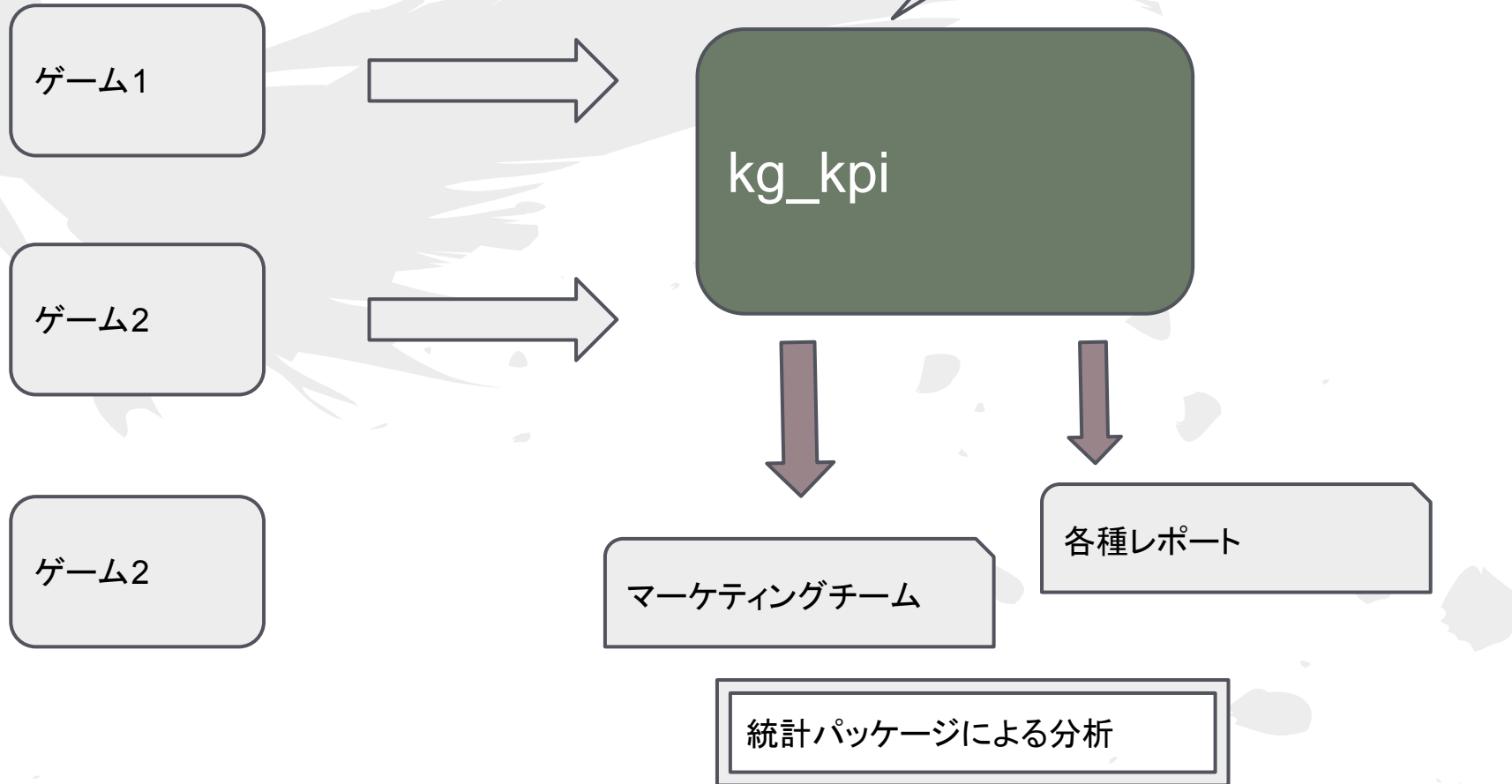


kg_kpi

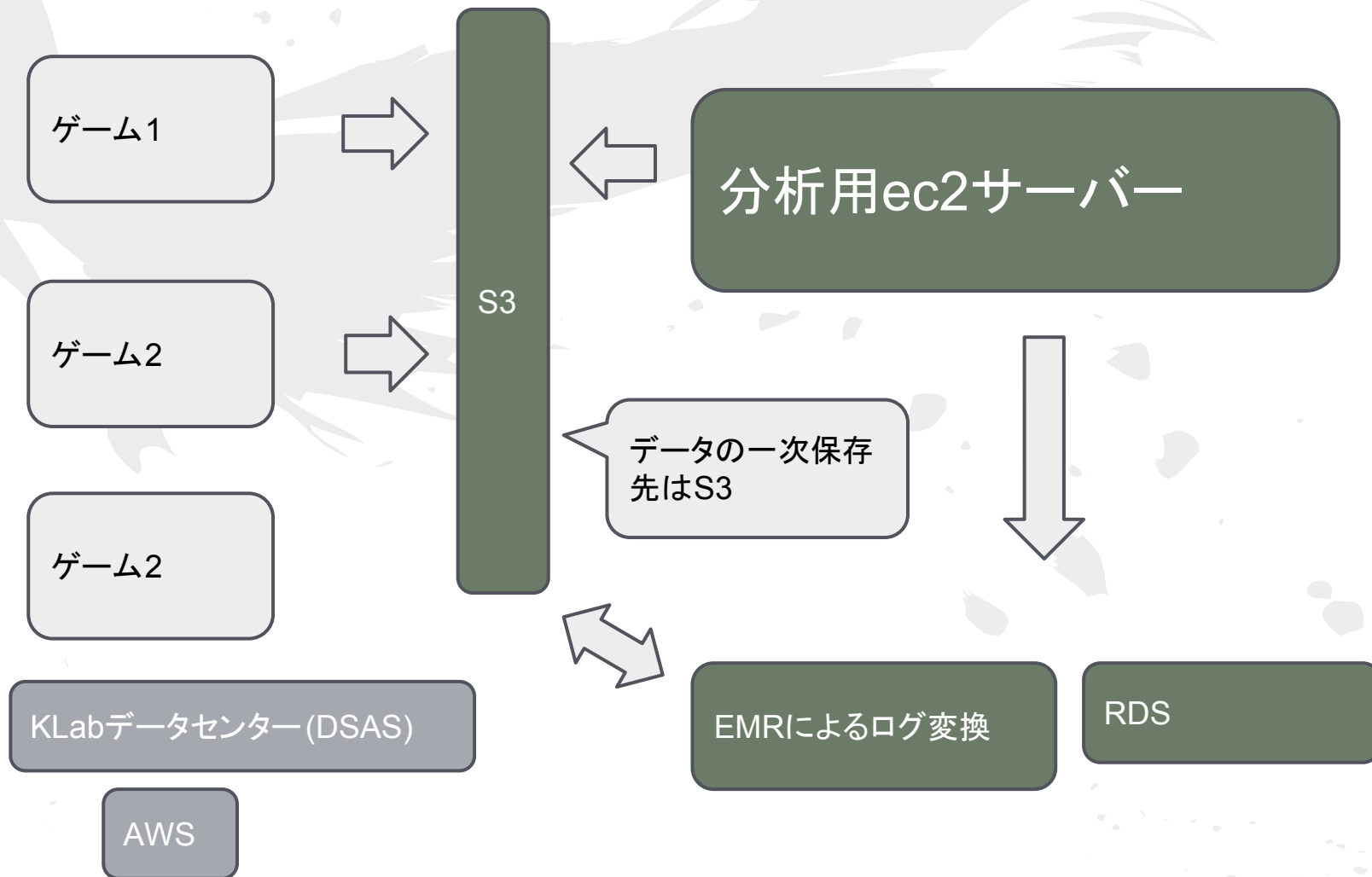
KLabが提供する30以上のソーシャルゲームのデータを蓄積したデータウェアハウス

データウェアハウスとは、直訳すれば「データの倉庫」である。利用者により定義範囲は異なるが、一般に時系列に整理された大量の統合業務データ、もしくはその管理システムを指す。 [wikipedia](#)

システム構成



システム構成



システム構成

- pythonによるバッチスクリプト群で構成される
- kg-kpi-tools
 - appexporttools
 - KLabデータセンターで動作
 - 各ゲームからのデータ抽出・送信
 - awsimporttools:
 - 分析用EC2サーバーで動作
 - データの取り込み・変換
 - awsreporttools
 - 分析用EC2サーバーで動作
 - レポートの作成

Elastic MapReduceの活用

Elastic MapReduce

aws上でHadoopを利用できる仕組み

比較的簡単に分散処理が書ける

apacheのアクセスログの変換などに使用

Elastic MapReduceの活用

メリット:

比較的簡単に分散処理が書ける

→分散処理が必要な重い集計・計算

デメリット:

デバッグがめんどろ

map+reduceの実装には慣れが必要

Elastic MapReduceの活用

細かい集計タスクのたびにMapReduceのコードを書くのは効率が悪い

→その後の集計に都合のよい形式に変換する

EMRの利用例: apacheログ

```
w0411 2012-10-10 23:59:59 V=3 127.0.0.1 - - - - - "GET /main.php/sample/log?  
log_id=36602246&battle_id=17998015&stage_id=&opensocial_app_id=12010023&opensocial_view  
er_id=7142034&opensocial_owner_id=123456 HTTP/1.1" 200 18214 151473 991 18521 "-"  
"SoftBank/2.0/002SH/SHJ001 Browser/NetFront/3.5 Profile/MIDP-2.0 Configuration/CLDC-1.1"  
prod -
```

CSVフォーマット

大量のためパースがボトルネックに

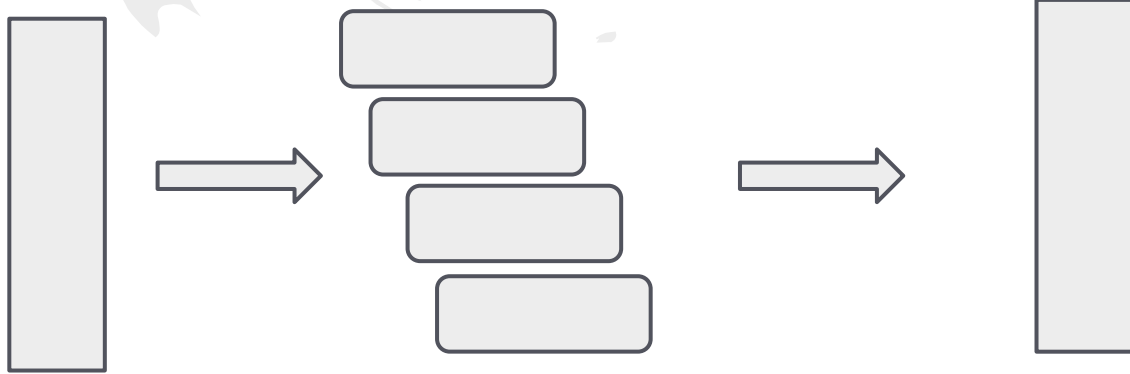
(圧縮しても1GiB以上)

プロジェクトごとにかかなり差異がある

EMRアクセスログ変換

30以上あるゲームのそれぞれ大容量のログを変換しなければならない

→Elastic Mapreduceを利用し、10数台のHadoopクラスタを利用して高速にログの変換を行なう



apacheログのバイナリ化

msgpack:

JSONのような構造化データをシリアライズするバイナリフォーマット

高速に処理できるよう工夫されている

msgpackログ

4カラムのバイナリデータに変換

→ユーザーごとのアクセスや、各時間ごとのアクセスを高速に集計可能

時刻(unixタイム)	user ID	URL	GETパラメータ
-------------	---------	-----	----------

ログ変換プログラムの工夫

コードはすべてpython



mrjob:

Elastic MapReduceに対応したMapReduceのラッパー

Hadoop Streamingを利用

ログ変換プログラムの工夫

appexporttools:

別で利用しているpython製ログ集計スクリプト
アプリごとのログの違いを吸収

→MapReduceの中からも呼び出す

→MapReduceスクリプト自体をいじる必要はほとんど無い

msgpackログから集計

常時集計しているもの

- 個々のユーザーの滞在時間・ログイン回数

たまに取得しているもの

- アワリーUU/PV
- 特定のパスへのアクセス

変換前: 1日分の集計に30分以上

変換後: 1日分は数分程度

hiveの活用



変換したログの集計は高速だが、長期の集計となるとかなり時間がかかる。

(DBから集計するよりはだいぶ遅い)

S3上のmsgpack化ログをhiveで集計できないか?

→現在評価中

msgpack-hadoop

githubで提供されている

<https://github.com/msgpack/msgpack-hadoop>

公式っぽいけど、あまりメンテされていない様子

msgpack-hadoop

外部テーブルとして読み込みできる

```
CREATE EXTERNAL TABLE IF NOT EXISTS mpbin (v string)
ROW FORMAT SERDE 'org.msgpack.hadoop.hive.serde2.MessagePackSerDe'
STORED AS INPUTFORMAT 'org.msgpack.hadoop.mapred.
MessagePackInputFormat'
OUTPUTFORMAT 'org.apache.hadoop.hive.ql.io.
HiveIgnoreKeyTextOutputFormat'
LOCATION 's3://s3logpath/';
```

msgpack-hadoop

hiveの関数でmsgpackの内容も取得可能

ex. 中間テーブルへの読み込み

```
INSERT OVERWRITE TABLE accesslog
SELECT from_utc_timestamp(from_unixtime(cast(accesstime AS INT)),
'JST'), user_id, path
FROM mpbin
LATERAL VIEW msgpack_array(v, 0, 1, 2) m
AS accesstime, user_id, path
```

1日分のS3上のファイルをインポートするのに、6-7
分程度

msgpack-hadoop

msgpack_array関数にバグがある
(0番目の要素がつねにNULLになる)

修正したもの

<https://github.com/takada-at/msgpack-hadoop/>

msgpack-hadoop

```
SELECT hour(cast(accesstime as string)), count(*), count(distinct user_id)
FROM accesslog
GROUP BY hour(cast(accesstime as string))
```

アワリーUUの集計

はまりどころ:

タイムスタンプを文字列に変換しないと、**hour**関数が**0-12時**になってしまう

まとめ

何度もジョブを書くのは非効率

→最初に集計に都合のいい形式に変換

→ポータビリティ(ライブラリ化)

hiveはSerDeさえ用意すれば様々なフォーマットに対応。

ただしhadoop周りのライブラリは不安定なものが多い.....。