

# Key-Valueストアを用いた GPVデータ閲覧サービスの構築(改)

2012年4月20日

株式会社テクリエ  
岡村 和英

- 気象情報をクラウド上で容易に利用できるようにしたい。
  - 特定のパラメータ値に基づいて事前に作成された画像を取得するのではなく、要求に応じて任意の画像を生成・表示するようにできないか？
  - パブリッククラウド等を用いることで急激なアクセス増加に応じて可変的にスケールの変更可能なシステムに出来るのでは？
  - 過去のデータへも容易かつ迅速にアクセスしたい。

- GPV = Grid Point Value : 格子点値 ▶
  - 地球の大気の状態を格子状に区切り、各点で計算された気象要素(気温、風、湿度など)の予測値を格納したものの。
- 2つの数値モデル
  - 全球数値予報モデルGPV(GSM) ▶
    - 主目的:短期、週間予報など
    - 日本域 0.2度×0.25度(20Kmメッシュ)、全球域 0.5度×0.5度 ▶
  - メソ数値予報モデルGPV(MSM) ▶
    - 主目的:防災情報
    - 日本域のみ(北緯22.6~47.6度、統計120~150度)
    - 地上面 0.05度×0.0625度(5Kmメッシュ) ▶

参考:

気象庁ホームページ - 天気予報・台風 - 数値予報

<http://www.jma.go.jp/jma/kishou/known/yohou.html>

(財)気象業務支援センター - MSM、GSMデータ概要

<http://www.jmbasc.or.jp/hp/online/f-online0a.html>

<http://www.jmbasc.or.jp/hp/online/f-online0c.html>

- 現在、気象庁が作成し気象業務支援センターから配信されているGPVデータファイルはGSM、MSMとも **GRIB2** 形式に統一されている。
  - 詳細な内容及び形式については、気象業務支援センター発行の技術資料に記載あり。
- GRIB形式とは、
  - WMO (World Meteorological Organization: 世界気象機関)が策定したGPVデータの為のファイル格納方式。
  - 大量のGPVデータをまとめて格納することを目的としたフォーマット。

## 参考:

World Meteorological Organization: WMO International Codes  
<http://www.wmo.int/pages/prog/www/WMOCodes.html>

(財)気象業務支援センター – 刊行資料

配信資料に関する技術情報(気象編)第205号 (書籍)

「平成18年3月からの数値予報もでのるGPV等の変更について」  
気象通報式及び国際地点番号表(平成22年版) (CD-ROM)

- MSM気圧面予報データの場合

Indicator Section (GRIB版番号、全体サイズ等)

Identification Section (作成者識別、参照時刻等)

Grid Definition Section (開始緯度経度、格子点数、間隔等)

Product Definition Section (データ種別、予報時間、気圧等)

Data Presentation Section (データ点数、データ形式等)

Data Section (圧縮エンコードされたデータ)

End Section (終端マーカ)

繰り返し

- MSM気圧面予報データの場合

参照時刻における

1000hPa等圧面の高度、風速、気温、上昇流速度、相対湿度  
(緯度方向0.1度、経度方向0.125度間隔)

950hPa等圧面の高度、……

900hPa等圧面の……

∩

100hPa等圧面の……

3時間後の…

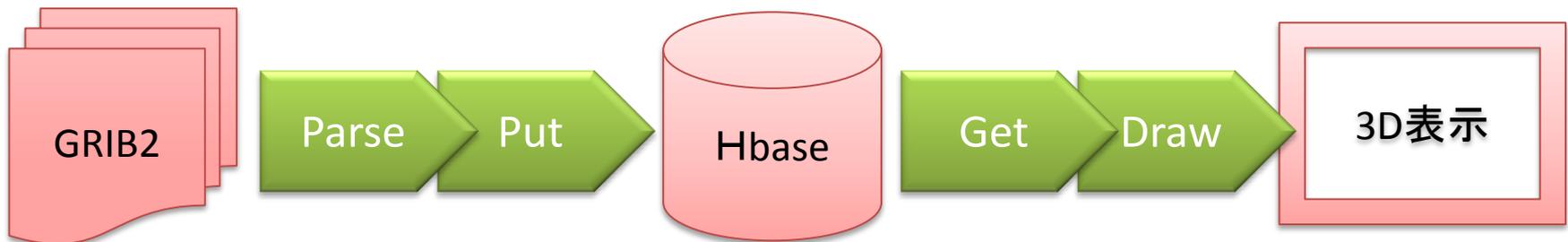
6時間後の…

∩

15時間後の…

# 今回作成したもの

- GRIB2データパーサー
  - GRIB2形式のバイナリファイルからGPVデータを読み込む。
- Key-ValueストアAPI
  - GRIB2ファイルから読み込んだGPVデータをエリア分割し、Hbaseに書き出す。
  - 指定されたエリアのGPVデータをHbaseから読み込む。
  - 今回はMSM(日本周辺)のデータのみに対応。
- GPVビューアー
  - GPVデータを3D表示する。



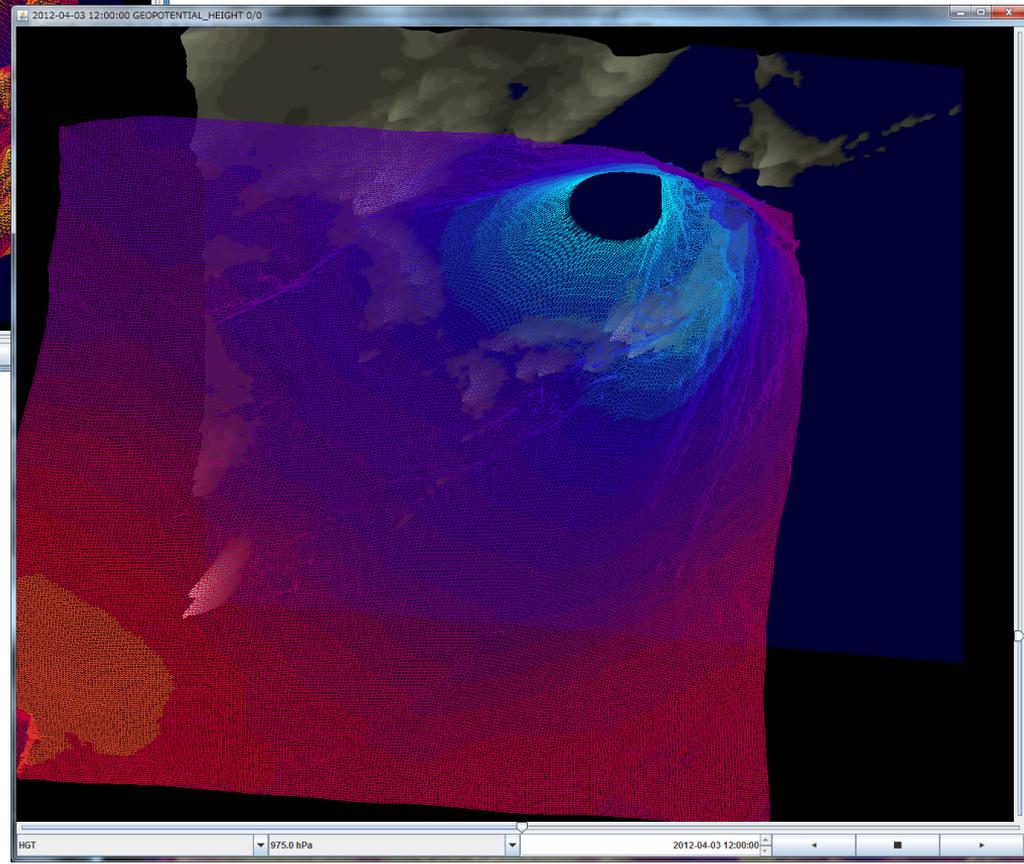
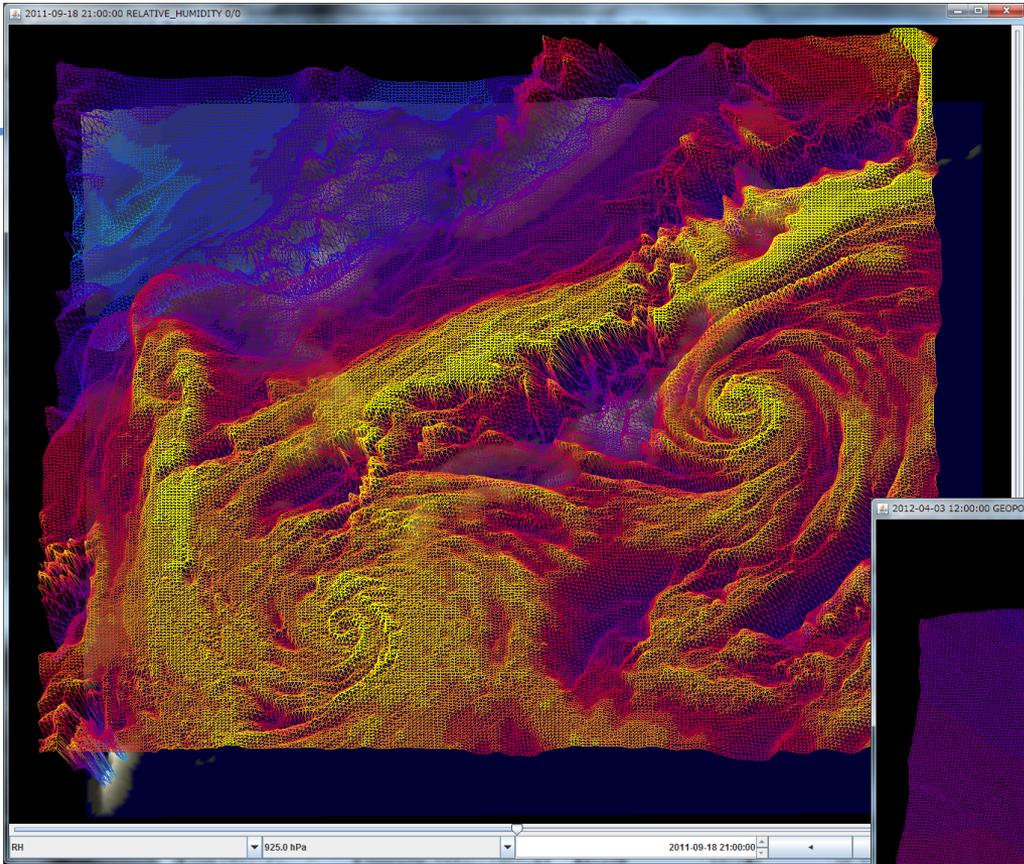
- Windows7
- MacOS X v10.6
- Amazon EC2 (new)
  - Amazon Linux 64bit × 5 smallインスタンス  
(namenode & hmaster) + (4 × (datanode & regionserver))
- Java SE ~~7~~ → 6  
Swing APIをMacOS側に合わせるため。
- Hbase ~~0.90.4~~ → 0.92.1   
~~今回はHDFSではなくローカルディスクを利用~~  
(0.20.203がWindows上で動作しなかった為)
- Hadoop 1.0.2 (new) 
  - Amazon EC2上でHadoop + Hbaseを分散稼働。  
(Windows 及び MacOS ではローカル稼働で構築)
- JOGL (Java OpenGL)
  - JogAmp JOGL ~~2.0 RC2~~ → 2.0 RC5 

- 単一テーブル”gpv”に全データを格納
- キー

<del>対象日時</del> → データ種別	<del>データ種別</del> → 対象日時	緯度(5度単位)	経度(5度単位)
----------------------------	----------------------------	----------	----------

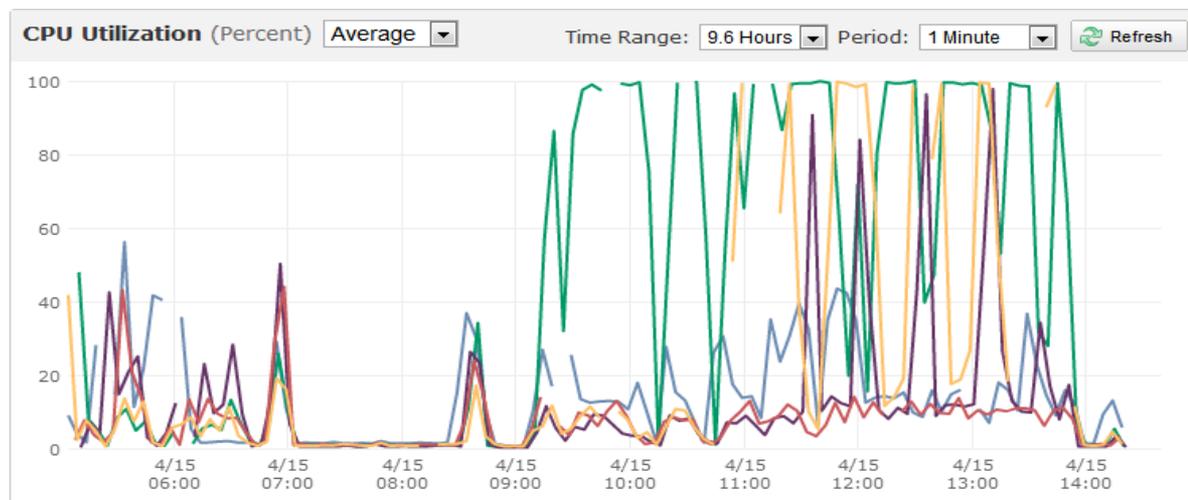
- 同一種別で異なる日時のデータを連続して取得することを想定し、データ種別と対象日時を入れ替えた。
  - 指定された緯度経度を中心とした±5度(10度)分を1エリアとして取り扱う(隣のエリアと半分重なっている)。
  - 古い予報値を新しい予報値で順次上書きしていき、最終的には解析値(観測値から導出された値)に置き換わる。
- データ
    - ”info”ファミリー: GRIBファイルの参照時刻、ファイル名。
    - ”griddef”ファミリー: データの開始緯度・経度、間隔、データ点数など。
    - ”data”ファミリー: `気圧値`をカラム名としてデコード済みのデータを格納する。

# << Demonstration >>



# やってみてわかったこと(1)

- データ投入に非常に時間がかかる。
  - local のHbaseクライアントアプリからEC2へ投入したところ、1ファイルの登録に25分もかかってしまった。
    - 前回作成環境 (localhost 内、HDFS未使用) では90秒弱。
  - GRIBファイルをアップロードし、EC2側で登録してみた。
    - ファイル転送 20秒 + 登録 8分 ... HDFSのオーバーヘッド？
    - CPU利用率が100%に到達しているので、largeインスタンスに変更すればもう少し早くなるかもしれない。
  - 領域分割しなければ60秒／ファイル、1分半～2分／日程度。
    - エリアの重ねあわせによってデータ量が4倍に増えているのに対して、時間が8倍かかっている。



# やってみてわかったこと(2)

- 投入中にregionserverが異常終了（microインスタンス利用時）
  - Splittingが発生したタイミングで落ちている？メモリ不足？
  - Smallインスタンスに変更したら投入後も落ちなくなった。
    - やっぱりメモリ不足？
- HDFSは大喰らい。
  - HDFS上に置かれているHbase関連ファイルサイズが計9GBであるのに、各HDFSデータノードが消費しているディスク容量が計25GB。（replication=1）
    - 増えたり減ったりなので回収されているような感じもするが・・・
  - HDFSがいっぱいになったらregionserverが異常終了。
    - HBaseの.logsファイルをロストしたらしく、再起動もできなくなりました。
    - HBaseだけではなく、HDFSを再起動したところで復帰。
    - DataNodeを追加することでHDFSへの容量追加ができるはずだが、今回はここまでで時間切れ。
    - 1ヶ月分登録する予定だったが半月分しか投入できなかった。
- TZ設定忘れに注意。
  - 古いデータで上書きしてしまった。

- ~~パブリック・クラウドの利用~~
- GPVデータの加工表示
  - 特定高度面における等圧線の表示
  - 特定経路線における高度、等圧面の表示
  - 異なる種類のデータの重ね合わせ表示
    - ~~地形、気圧、風向き、温度など~~
- GSM(全球モデル)データへの対応
  - 気圧面によってデータ点数が異なる。
  - テーブルレイアウトの再検討
- モバイルデバイスからの利用



まだまだ、当初の目標にはほど遠い状況

クラウド部会に参加して一緒に作りませんか？

## 今回作成したもの



- GRIB2データパーサー
  - GRIB2形式のバイナリファイルからGPVデータを読み込む。
- Key-ValueストアAPI
  - GRIB2ファイルから読み取り、Hbaseに書き出す。
  - 指定されたエリアのGPVデータを取得する。
  - 今回はMSM(日本周辺)のデータを取得する。
- GPVビューアー
  - GPVデータを3D表示する。

githubで公開しています。

<https://github.com/techlier/wgpv>

Key-ValueストアAPI、GPVビューア部についても順次公開の予定です。

