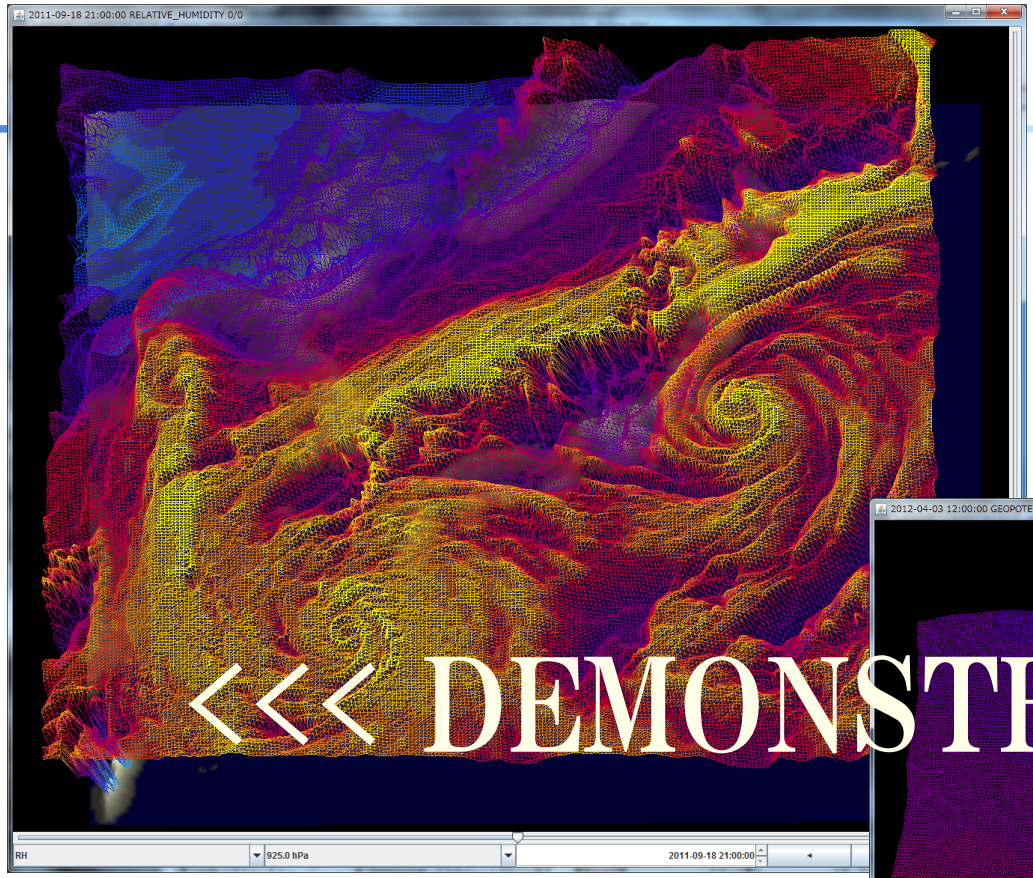
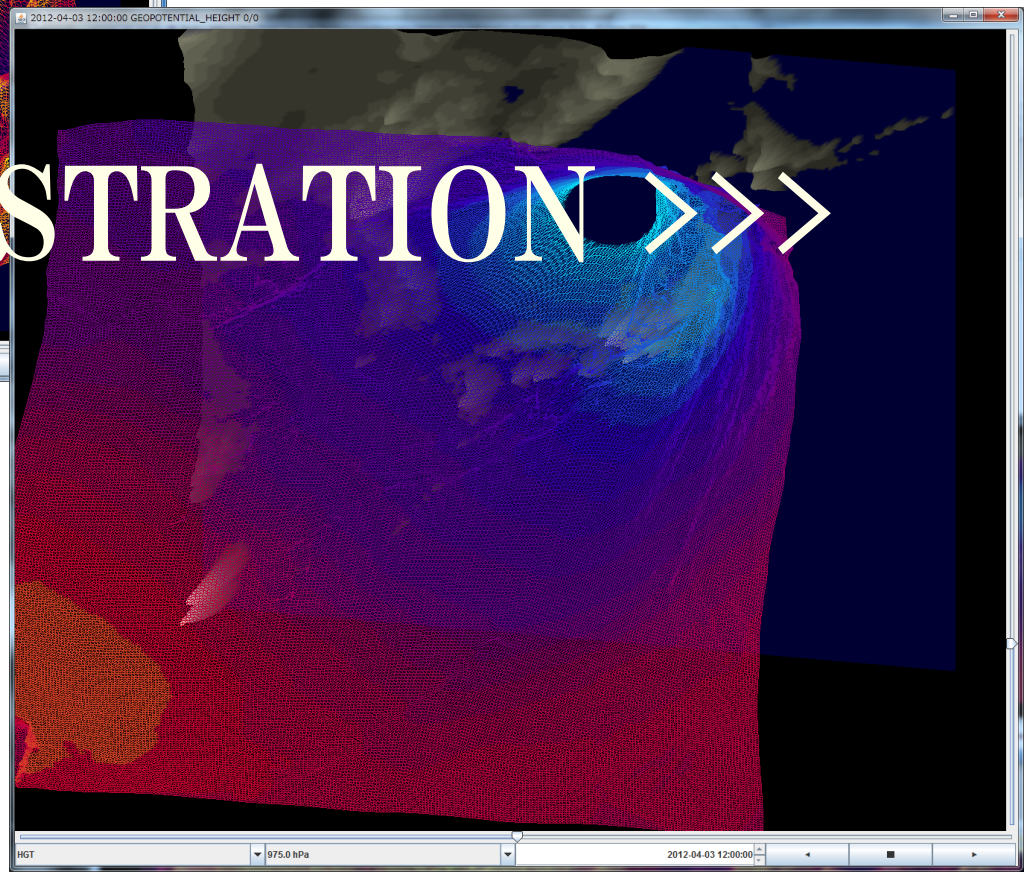
SoftwareJapan 2014 ITフォーラムセッション

気象予報データ(数値予報GPV)を用いたデータビジュアライゼーション

2014年2月4日

先端IT活用推進コンソーシアム
クラウドテクノロジー活用部会
岡村 和英
(株式会社テクリエ)





<<< DEMONSTRATION >>>

<<< 数値予報GPVとは？ >>>

数値予報GPVとは

- GPV = Grid Point Value：格子点値
  - 観測気象データを基に、地球の大気の状態を格子状に区切り、スーパーコンピュータで様々な物理現象のシミュレーションを行なうことで、各点の気象要素（気温、風、湿度など）の推移予測（数値予報）を行い、その値を格納したもの。
  - 天気予報、注意報、警報などを支える重要なデータ

参考：
気象庁ホームページ – 数値予報とは
http://www.jma.go.jp/jma/kishou/know/whitep/1-3-1.html

主な数値予報モデル

- 全球モデル（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
- メソモデル（MSM）
  - 主目的：防災情報
  - 日本域のみ（北緯22.6～47.6度、東経120～150度）
  - 地上面 0.05度×0.0625度（5Kmメッシュ）気圧面 0.1度×0.0625度

参考：
気象庁ホームページ –数値予報モデルの種類、全球モデル、メソモデル
http://www.jma.go.jp/jma/kishou/know/whitep/1-3-4.html
http://www.jma.go.jp/jma/kishou/know/whitep/1-3-5.html
http://www.jma.go.jp/jma/kishou/know/whitep/1-3-6.html
(財)気象業務支援センター – MSM、GSMデータ概要
http://www.jmbsc.or.jp/hp/online/f-online0a.html
http://www.jmbsc.or.jp/hp/online/f-online0c.html

GPVデータファイルの形式

- 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。

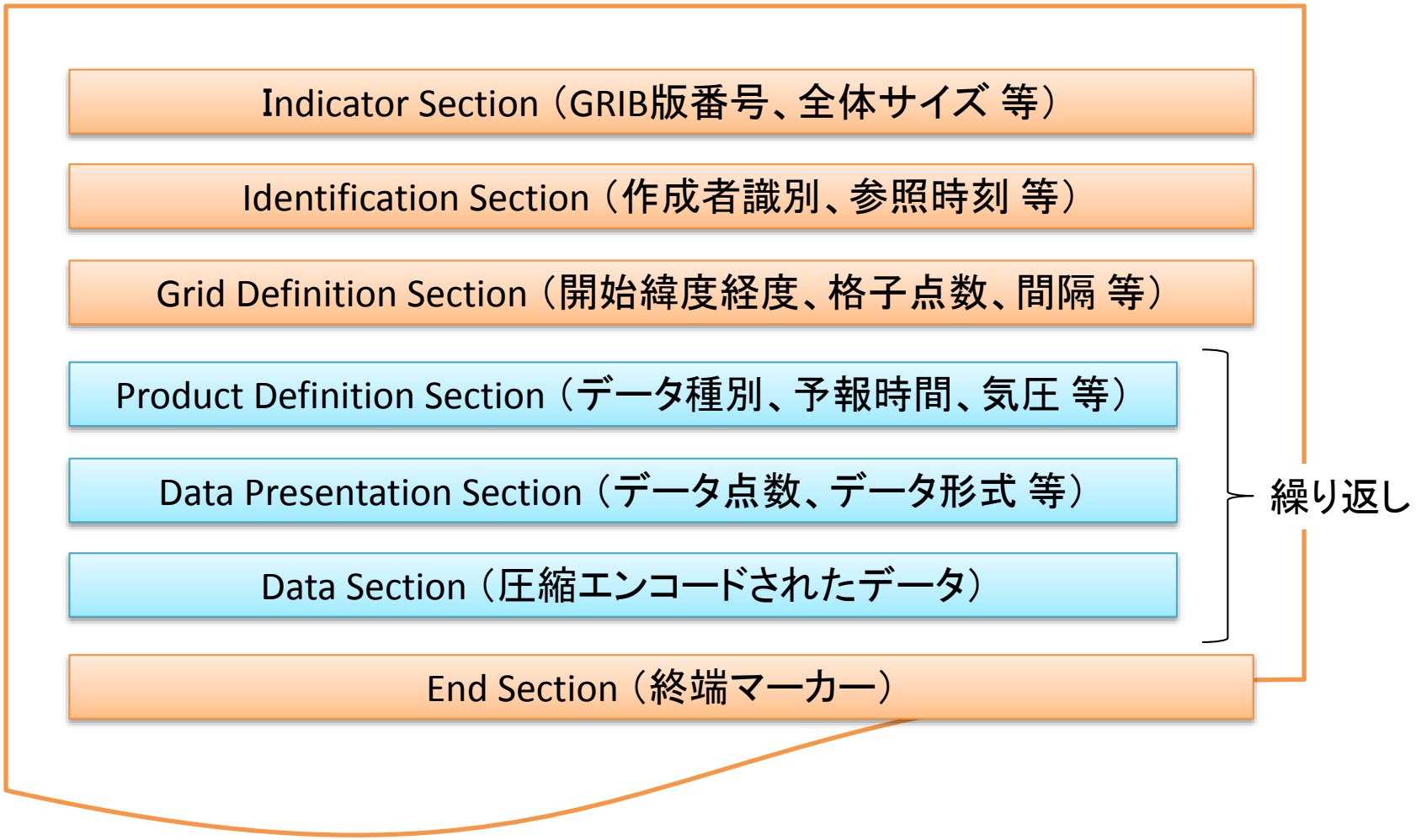
参考：

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)

GRIBファイルの構造

- MSM気圧面予報データの場合



GRIBファイルの内容

- 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・・・

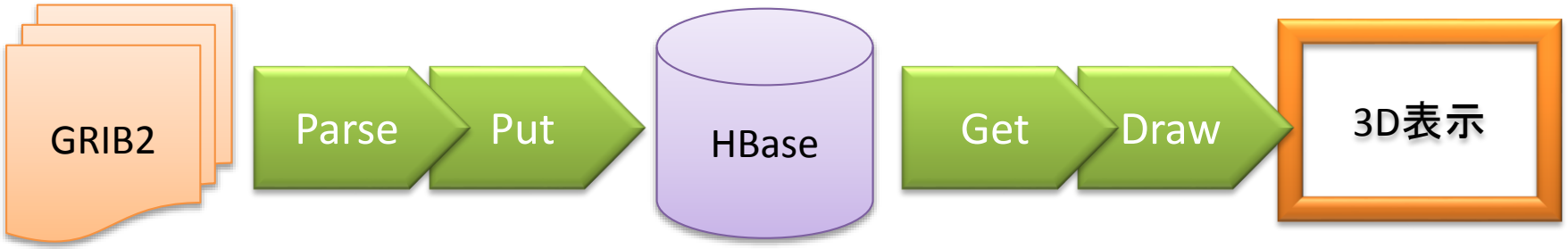
<<< 開発内容 >>>

開発の背景

- 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。

システム概要

- 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。
- KeyValue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。



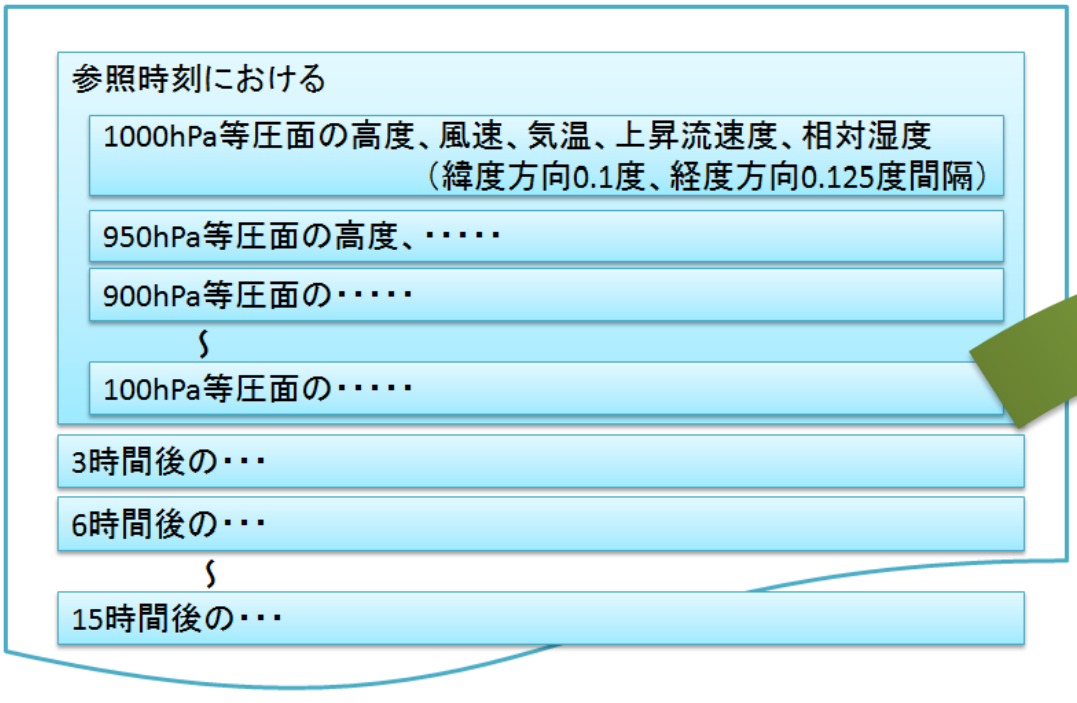
動作環境

- MacOS X または Windows
- Amazon EC2
  - Amazon Linux 64bit × 5 smallインスタンス

    (namenode & hmaster)
    + (4 × (datanode & regionserver))
- Java SE 7
- HBase 0.94.15
- Hadoop 1.2.1
  - Amazon EC2上でHadoop＋HBaseを分散稼働。
- JOGL (Java OpenGL)
  - JogAmp JOGL 2.1.3

設計の趣旨

- 地上面データテーブル “MSM_JP_SURF”
- 気圧面データテーブル “MSM_JP_PALL”
- キー値（SURF 3時間単位、PALL 1時間単位）
  - 「対象日時」+「緯度（5度単位）」+「経度（5度単位）」
  - 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。
- カラム構造
  - “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。
  - データファミリー：‘データ種別識別値’をファミリー名、‘パラメータ値（気圧面であれば気圧値）’をカラム名として1エリア分のGPVデータグリッドを実数値に展開したものをバイナリデータとして格納。



Family	HGT					TMP			VVEL			…
Qualifier Key	1000 hPa	950 hPa	900 hPz	…	100 hPa	1000 hPa	950 hPa	…	1000 hPa	950 hPa	…	…
00:00-25-125	….	….	….	….	….	….	….	….	….	….	….	….
00:00-25-130	….	….	….	….	….	….	….	….	….	….	….	….
00:00-25-135	….	….	….	….	….	….	….	….	….	….	….	….
…	….	….	….	….	….	….	….	….	….	….	….	….
00:00-30-125	….	….	….	….	….	….	….	….	….	….	….	….
00:00-30-130	….	….	….	….	….	….	….	….	….	….	….	….
00:00-30-135	….	….	….	….	….	….	….	….	….	….	….	….

横になが～～～～い
テーブル構造

できたこと

- HBaseにデータを格納することで、GRIB2から都度読みだすより迅速に目的のデータを取得することが出来る様になった。
- 複数のGRIB2ファイルに跨っていた過去データも容易に取得できるようになった。

しかし・・・

- GRIB2データに比べてかなり大きなリポジトリが必要になってしまうのでは？
  - エンコードされた予報値(12bit)を実数値(64bit)に展開・格納したことによる増加（約5倍）。
  - エリアの重ね合わせによる増加（約4倍）。
  - HBaseの仕組みによるオーバーヘッド。

リポジトリサイズの比較

- 気圧面15時間予報データ
  - GRIB 48MB／ファイル → HBase 1GB

> - エンコードされた予報値(12bit)を実数値(64bit)に展開・格納したことによる増加（**約5倍**）。
> - エリアの重ね合わせによる増加（**約4倍**）。
> - HBaseのデータ保持方法によるオーバーヘッド。

予想通り！

- その他に更新ログ等が加わる。
- 圧縮オプション（COMPRESSION=GZ）を有効にすれば、130～170MB程度にまで縮小可能。
  - ・・・しかし、圧縮を行なうとHBaseが非常に遅くなる。
  - CPUとIOのバランスが取れていない。
  - クライアント側でデータ圧縮を行なった方が良さそう。（未対応）

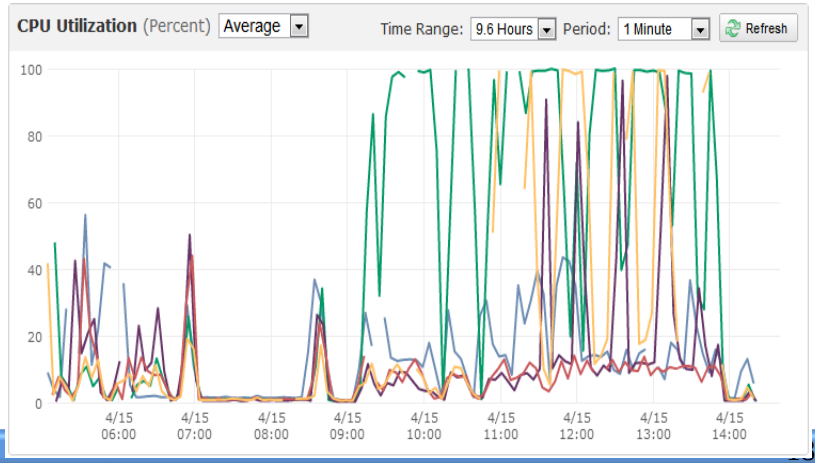
リポジトリサイズの比較

- 1日分格納後・・・
  - 気圧面15時間予報(48MB)
    18～33時間予報(48MB)
    36～39時間予報(16MB)
    各3時間毎×8回＝900MB → HBase 3.4GB
  - 地上面15時間予報(61MB)
    18～33時間予報(69MB)
    36～39時間予報(23MB)
    各3時間毎×8回＝1.2GB→ HBase 4.4GB
- 3ヶ月分格納後・・・
  - GRIB 190GB → HBase 330GB

テーブル作成時にmaxVersions=1に設定し、古い予報値を新しい予報値で上書きした分だけ容量が削減されている。

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。
    - 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インスン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。
    - 現在とテーブル設計が違うので、今はもう少し速くなってるかもしれない。



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ってしまった。
    - HBaseだけではなく、HDFSを再起動したところで復帰。

やってみてわかったこと(3)

- TZ設定忘れに注意
  - 古いデータで上書きしてしまった。
- maxfilesに注意
  - サーバOSではあまり問題にならなかったが、デモ用にデスクトップOS(Mac)上での環境構築を行ったところ、ファイルオープン数が超過してHBaseが落ちてしまった。
  - 以前はMacでも問題なく動いていたが、今回、テーブルレイアウトを変更し、カラムを増やしたため、同時にオープンするファイルが増えてしまった。
- 圧縮は遅い
  - テーブル構造やデータ内容にも依存するが、圧縮があると目に見えて遅くなる。
  - クライアント側の方が処理能力に余裕があるので、圧縮済みのデータを格納する方がよさそうに思える。

まだまだやりたいこと

- 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値の重ね合わせ表示
    - 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
- GPV以外のデータとの組み合わせ
  - 気象警報、生物季節観測、災害ハザードマップなど
- モバイルデバイスからの利用
  - WebGL、three.jsでWebアプリ化？

ソースコードについて

システム概要

- 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。
- KeyValueストアAPI
  - GRIB2ファイルから読み込んだGI
    Hbaseに書き出す。
  - 指定されたエリアのGPVデータを
  - 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。

githubで公開しています。
https://github.com/techlier/wgpv