

# Hadoop部会

株式会社エイビス	(部長)	小池 翼
株式会社富士通九州システムズ	(副部長)	小畑 智博
九州東芝エンジニアリング株式会社		小原 辰徳
九州東芝エンジニアリング株式会社		郷原 慎之介
株式会社オーイーシー		高熊 大将
株式会社オーイーシー		玉井 達也
大分大学大学院工学研究科		大場 紀彦

# テーマ

## 「実業務へのHadoopの適用」 ～気象データを用いた分散処理の実装～

- Hadoopとは
- Hadoop環境の構築
- ビッグデータ(気象データ)を用いた検証

# ビッグデータとは

- 構造化されていない莫大な量のデータ  
→ 情報通信、特にインターネットの発達に  
ともなって爆発的に増大



# ビッグデータに対する解答 Hadoopとは

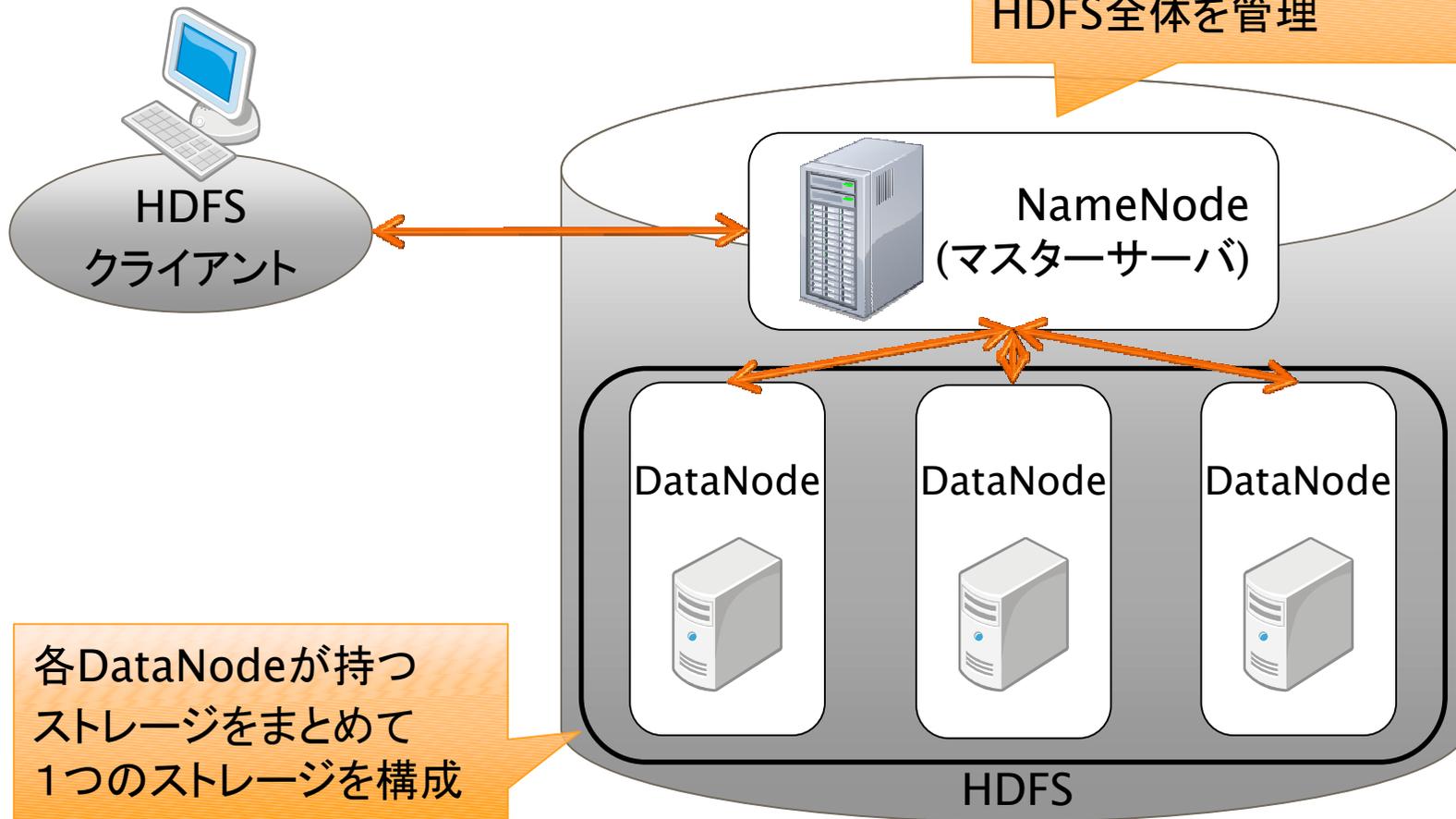
- 「大規模なデータを処理するための  
並列分散処理基盤」
  - 複数サーバを利用し、並列処理を行うのが得意
  - 数千ノードのサーバを使用可能
  - 拡張性に優れる
  - 柔軟なデータ構造に対応



# Hadoopの構成

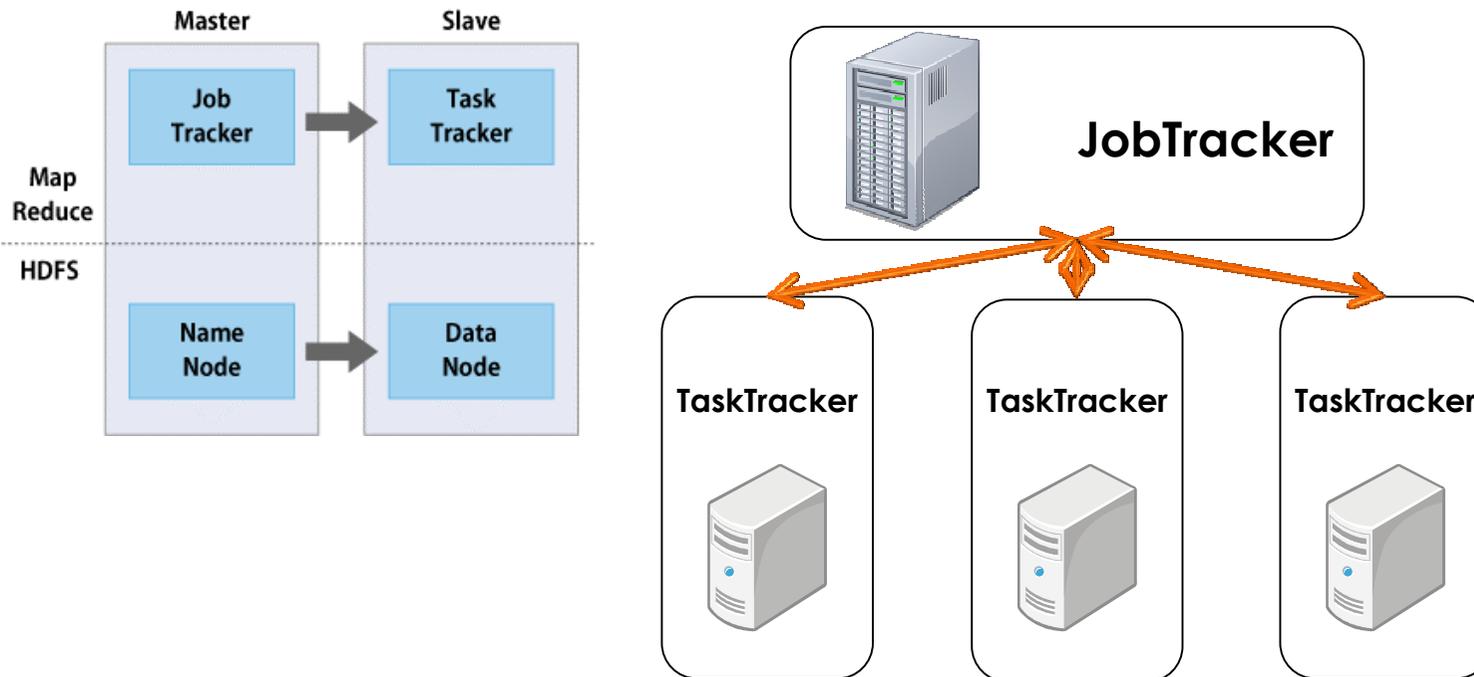
- 「HDFS」と「MapReduce」で構成
  - HDFS（Hadoop分散ファイルシステム）
    - 巨大なサイズのファイルを分割して、複数のサーバに振り分ける
  - MapReduce
    - 分散処理を行う

# HDFS構成図

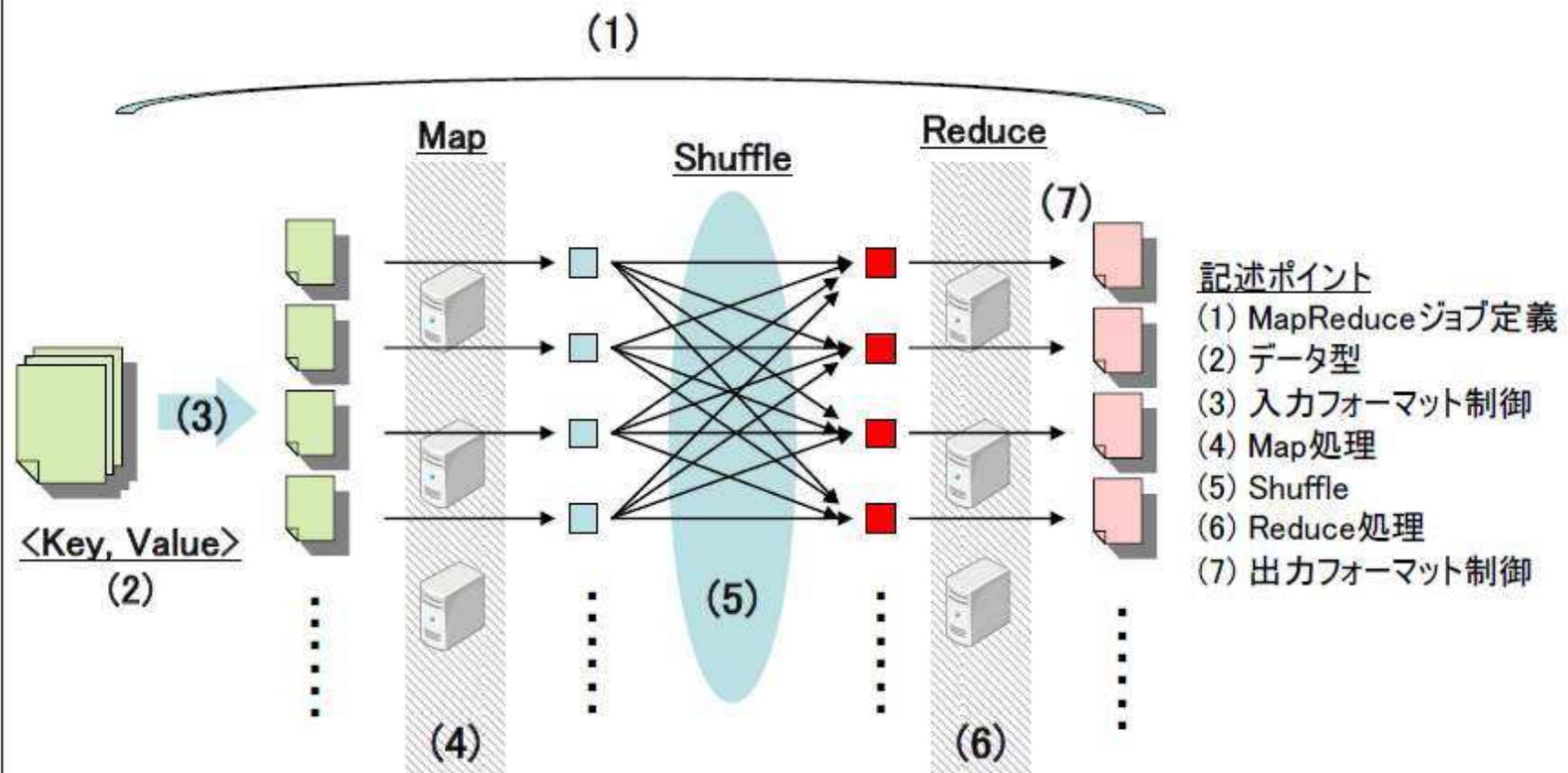


# MapReduce

分散して処理をおこなう



# MapReduce概要図



# MapReduce 例 (WordCount)

THE END OF MONEY IS THE END OF LOVE

Key、Valueのペアを作る「Map」

key	THE	END	OF	MONEY	IS	THE	END	OF	LOVE
Value (出現回数)	1	1	1	1	1	1	1	1	1

同じKeyを持つペアを束ねる「Shuffle」

※キー順にソートし、同じKeyを束ねる

END	IS	LOVE	MONEY	OF	THE
1	1	1	1	1	1

Shuffleの結果を入力として処理を行う「Reduce」

END	IS	LOVE	MONEY	OF	THE
2	1	1	1	2	2

# MapReduceまとめ

## ○ Map処理

- 入力データを分割し、ノードにタスクを割り当てる
- 入力データから<Key, Value>を取り出す

## ○ Shuffle処理

- 中間生成物のデータをソート
- ネットワークを介し、同じキーを持つデータを集める

## ○ Reduce処理

- 集められたデータに対して集約処理を行う

# Hadoopの利点

- 巨大なファイルを分散して持たせて処理ができる
- プログラマがやりたい処理だけに集中出来る  
※ソートやキーの処理を自動で実施してくれる
- 複数のプロセスで計算させることが可能
- 処理を分散している事による耐障害性の実現

# Hadoop環境の構築

●動作環境

Hadoop-0.20.2

●JAVA

jdk-6u35

●OS

CentOS 5.6

## 環境のダウンロード

- OS : CentOS5.6 32bit  
<http://ftp.riken.jp/Linux/centos/5.6/isos/i386/>
- Hadoop : hadoop-0.20.2  
<http://archive.apache.org/dist/hadoop/core/hadoop-0.20.2/>
- java : jdk-6u35  
<http://www.oracle.com/technetwork/java/javase/downloads/jdk6u35-downloads-1836443.html>

# 環境設定

- ① CentOSをインストール
- ② java,hadoopファイルをダウンロード(/tmp)
- ③ javaインストール(rootで実行)

## インストール

```
#chmod 755 jdk-6u35-linux-i586-rpm.bin  
#./jdk-6u35-linux-i586-rpm-bin
```

## パスの設定

```
#export JAVA_HOME=/usr/java/latest  
#tail -n 2 /etc/bashrc
```

# 環境設定

## ⑤ hadoopのインストール(rootで実行)

### ファイルの解凍・ユーザー変更

```
#tar xzf hadoop-0.20.2.tar.gz  
#mv hadoop-0.20.2 /usr/local  
#chown -R hadoop /usr/local/hadoop-0.20.2
```

### パスの設定

```
#ln -s /usr/local/hadoop-0.20.2 /usr/local/hadoop  
#export HADOOP_HOME=/usr/local/hadoop  
#tail -n 3 /etc/bashrc
```

# 環境設定

## ⑥ 設定ファイルの変更(core-site.xml)

```
/usr/local/hadoop/conf/core-site.xml

<configuration>
  <property>
    <name>hadoop.tmp.dir</name>
    <value>/hadoop</value>
  </property>
  <property>
    <name>fs.default.name</name>
    <value>hdfs://localhost:54310</value>
  </property>
</configuration>
```

# 環境設定

## ⑦ 設定ファイルの変更(hdfs-site.xml)

```
/usr/local/hadoop/conf/hdfs-site.xml
```

```
<configuration>
  <property>
    <name>dfs.name.dir</name>
    <value>${hadoop.tmp.dir}/dfs/name</value>
  </property>
  <property>
    <name>dfs.data.dir</name>
    <value>${hadoop.tmp.dir}/dfs/data</value>
  </property>
</configuration>
```

# 環境設定

## ⑧ 設定ファイルの変更(mapred-site.xml)

```
/usr/local/hadoop/conf/mapred-site.xml
```

```
<configuration>  
  <property>  
    <name>mapred.job.tracker</name>  
    <value>localhost:54311</value>  
  </property>  
  <property>  
    <name>mapred.local.dir</name>  
    <value>${hadoop.tmp.dir}/mapred</value>  
  </property>  
</configuration>
```

# 環境設定

## ⑨Master、Slaveの設定

Master  
Slave

```
172.0.0.1 localhost.localdomain localhost
192.***.***.*** master master.localdomain
192.***.***.*** slave slave.localdomain
```

## 環境構築の問題点

- 動作環境を設定する上でつまずいた設定
  - ファイル設定の正解が不明
  - 正確に動作しているかの確認がしにくい
  - HDFS化に時間が要する

# Hadoopの利用事例

- 利用用途（例）
  - アクセスログ解析
  - Web検索の索引作成
  - POSデータの傾向解析
- 利用企業
  - 米Yahoo! （25000台規模）
  - Facebook （10000台規模）
  - 楽天



## 使用したデータ

- 気象データ (CSV)
  - 地点:九州全域
  - データ:気象庁

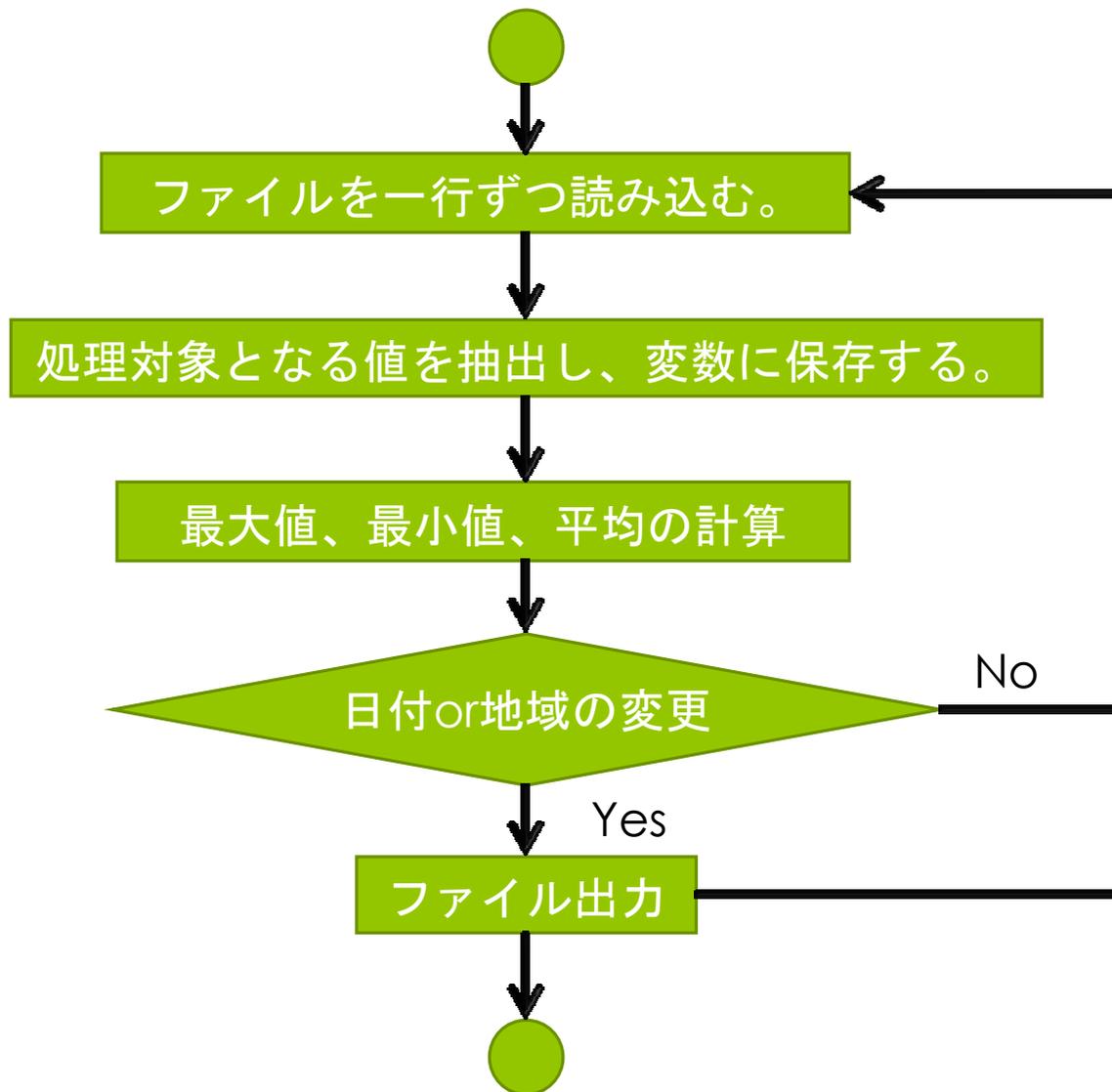
県番号	地点	日付	時刻	気圧_現地	気圧_海面	降水量	気温	湿度	風速	風向
82	47807	20080625	0:10	1003.3	1005	0.5	20.5	92	0.9	北
82	47807	20080625	0:20	1003.3	1005	0	20.6	92	1.5	北北西
82	47807	20080625	0:30	1003.3	1005	0.5	20.5	92	2	北北西
82	47807	20080625	0:40	1003.2	1004.9	0.5	20.6	92	1.9	北北西
82	47807	20080625	0:50	1003.3	1005	0	20.6	92	1.8	北北西

Key

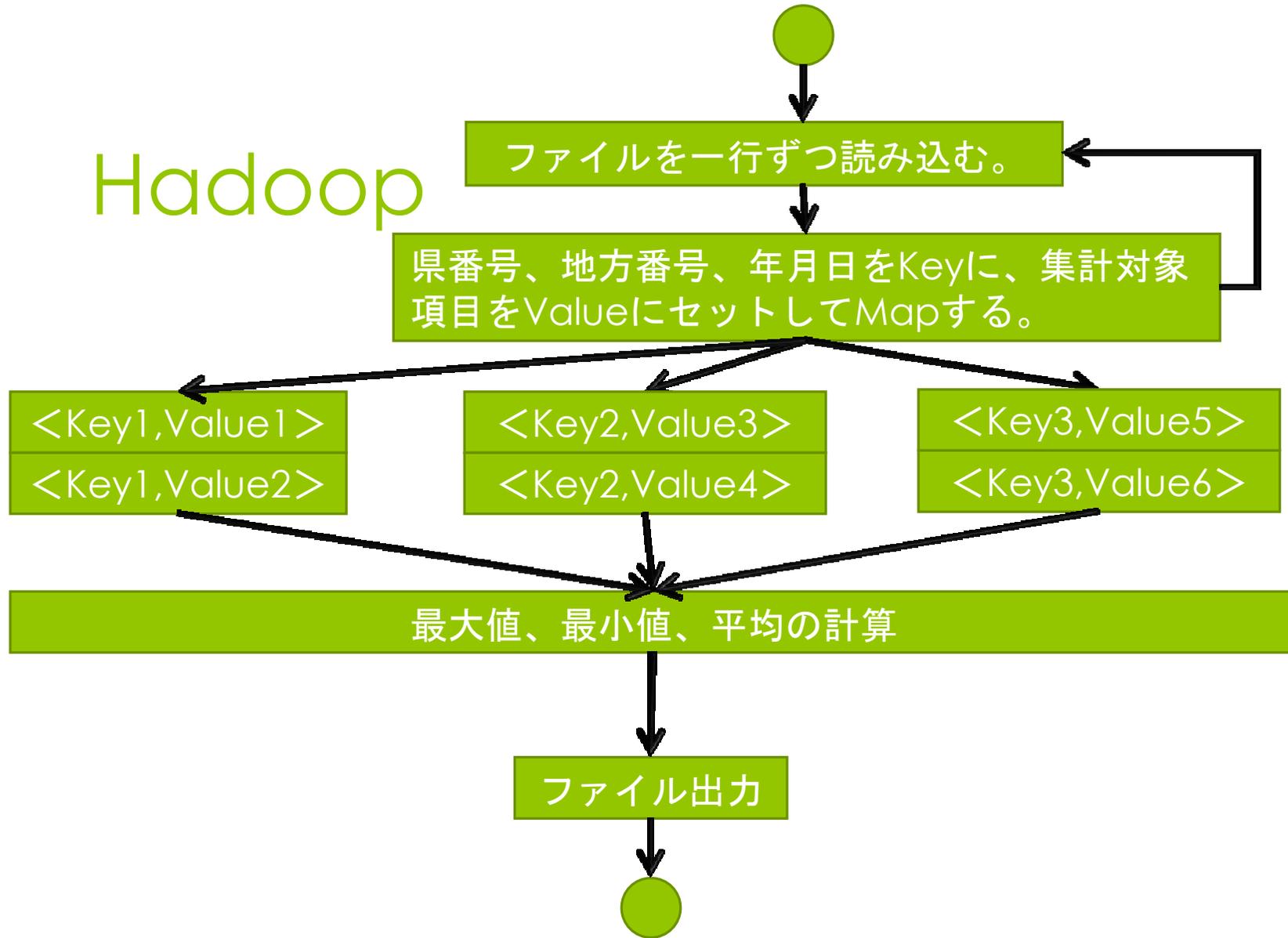
Value

各項目を、日付別、地域別で集計（最大値、最小値、平均値）する。

Java



# Hadoop



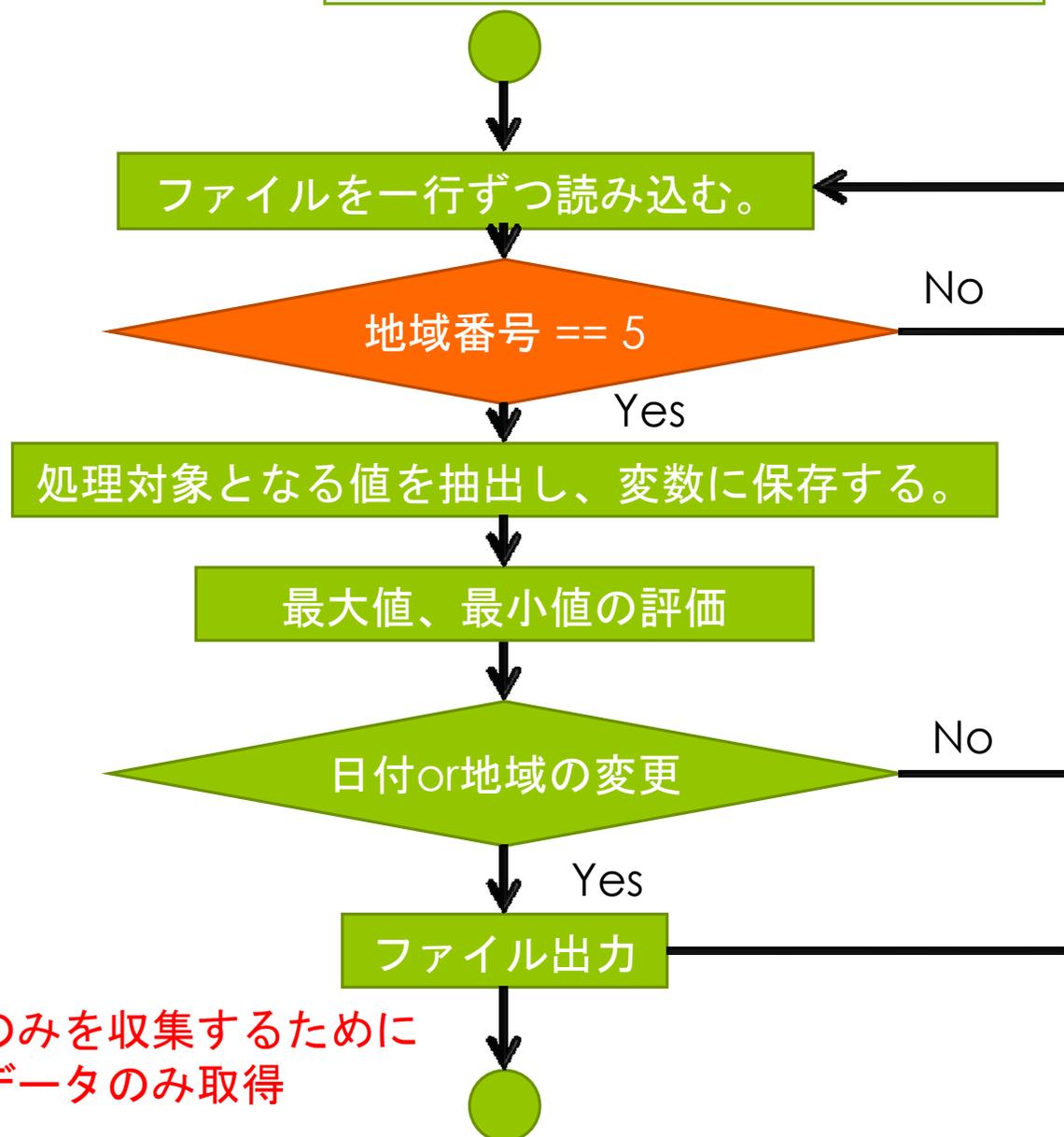


# データ設定

県番号	地点	日付	時刻	気圧_現地	気圧_海面	降水量	気温	湿度	風速	風向	最大風速	風向	日照時間
82	47807	20080625	0:10	1003.3	1005	0.5	20.5	92	0.9	北	1.8	北	0
82	47807												0
82	47807												0
⋮													
83	795	20080326	0:10			0	10.9		7.3	西	11.9	西	
83	795	20080326	0:20			0	10.7		8.2	西	11.9	西	
83	795	20080326	0:30			0	10.6		8.5	西	11.6	西	

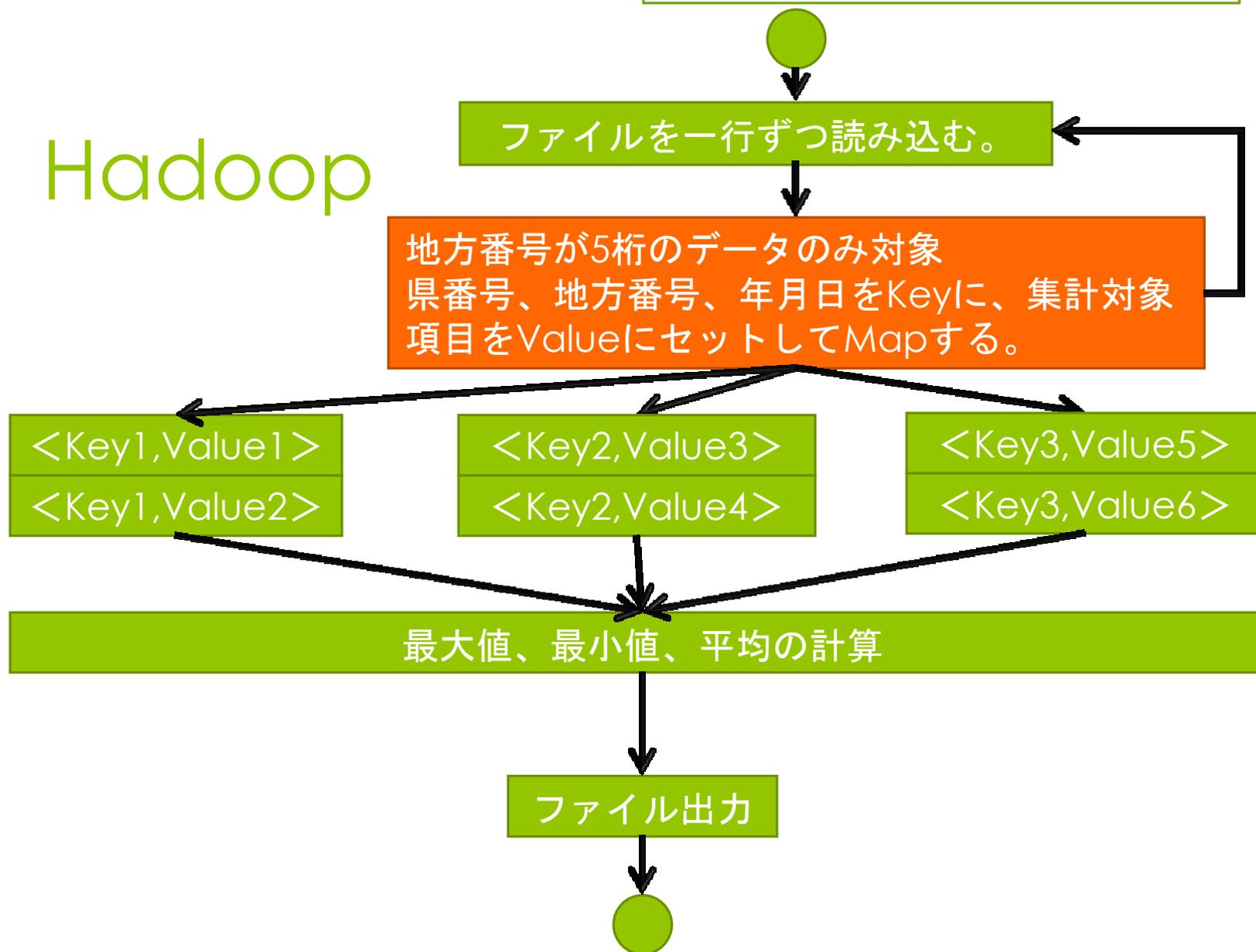
地域番号によって、データの形式が異なる。

Java



※同一形式のデータのみを収集するために  
地域コード5桁のデータのみ取得

## Hadoop



# 評価の準備

- 1. 評価データの規模
  - 0.5GB
  - 1GB
  - 2GB
  - 8GB
  - 16GB
- 2. 分散方法の選定
  - 完全分散モード（2台）

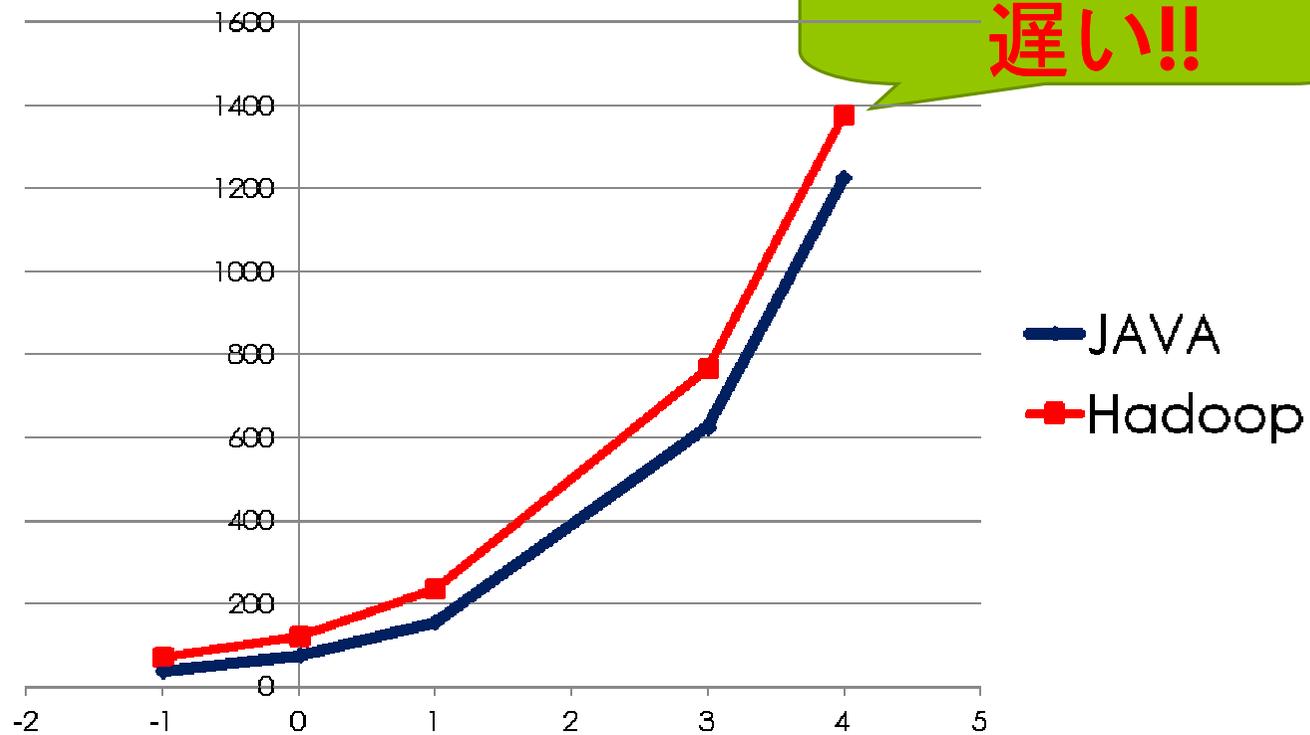
## 評価の方法

- 気温の平均・最大・最小を求める
- 処理時間を測り比較を行う
- 完全分散にて同じ処理を実施し処理時間を評価

	通常処理 (JAVA)	完全分散処理 (Hadoop)
0.5GBデータ	37	1' 10
1GBデータ	1' 13	2' 00
2GBデータ	2' 35	3' 57
8GBデータ	10' 24	12' 45
16GBデータ	20' 24	22' 56

# 評価結果

## ○ 処理時間の比較



**Hadoop**  
**遅い!!**

## 評価結果

- 並列化のためのオーバーヘッド
- 通信による影響
- ノード数の不足
- アルゴリズムの違い(ソート処理の有無)



### 「Hadoop」

- ・ メモリではソート処理できない  
セグメントをディスクへ書き込む
- ・ セグメントのマージ処理

# Combiner処理

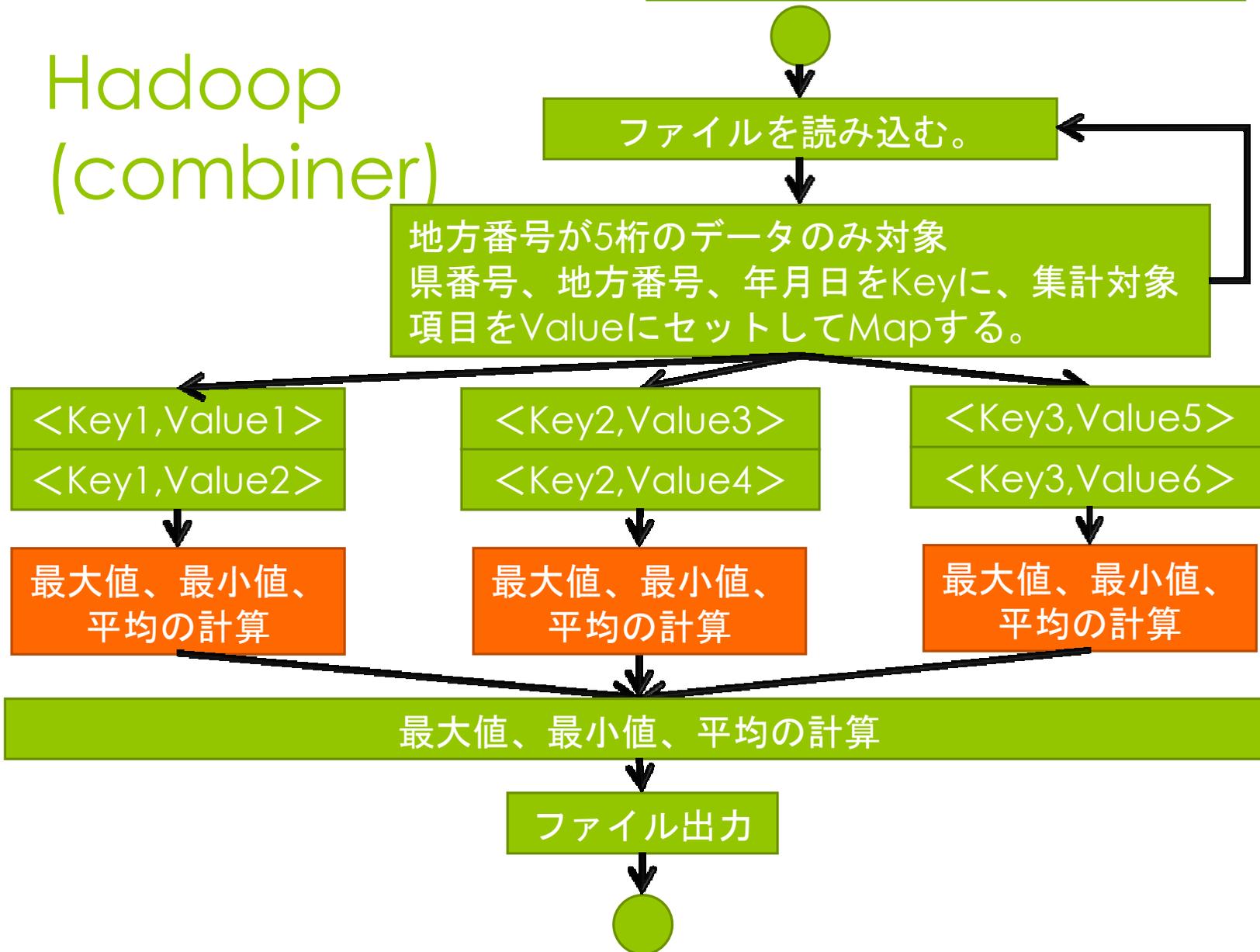


- マップの一時集計を担う処理

メモリの使用が抑えられる

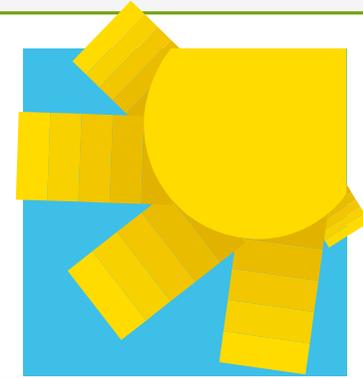
Reduceに渡す処理が減る(通信量の削減)

# Hadoop (combiner)

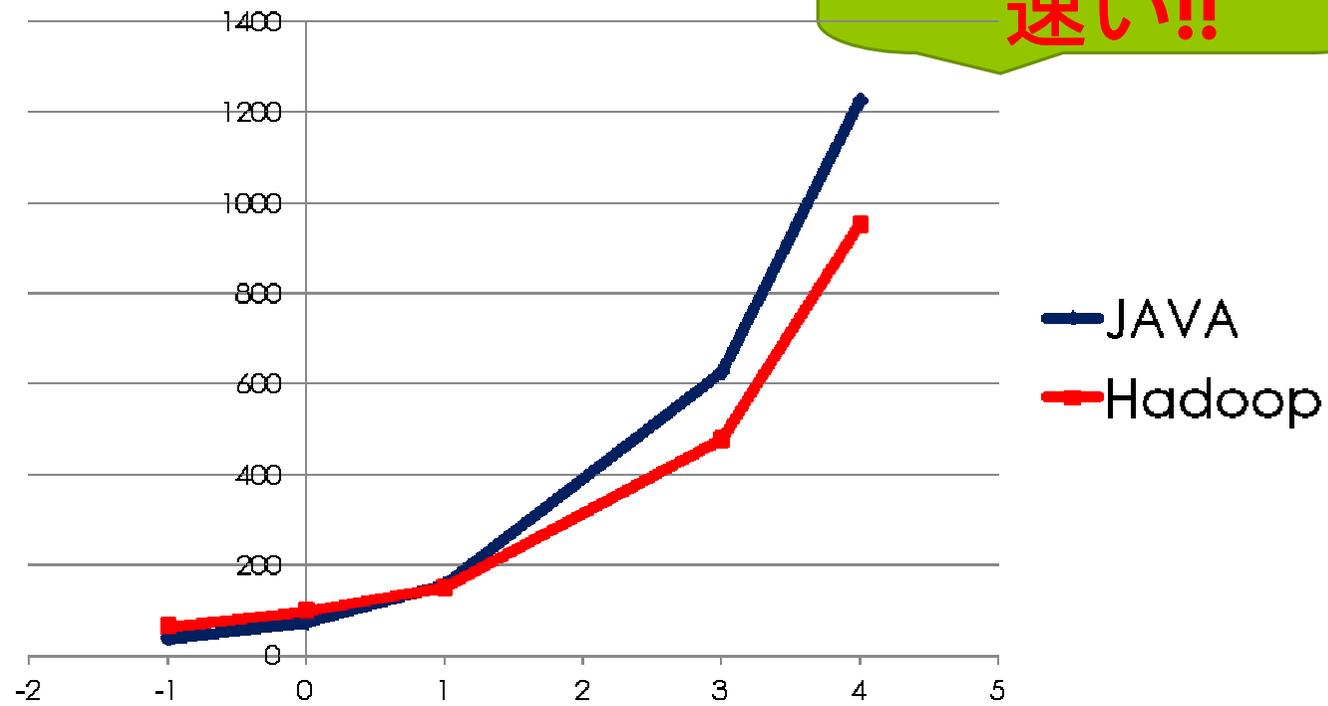


# 評価結果

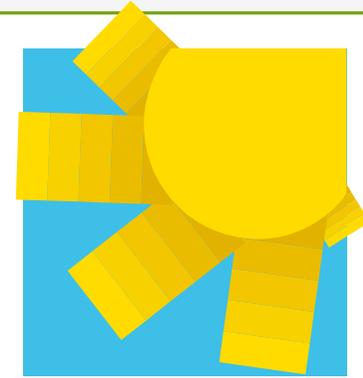
- Combiner導入後



**Hadoop  
速い!!**



# 評価結果



## ○ Combiner導入後

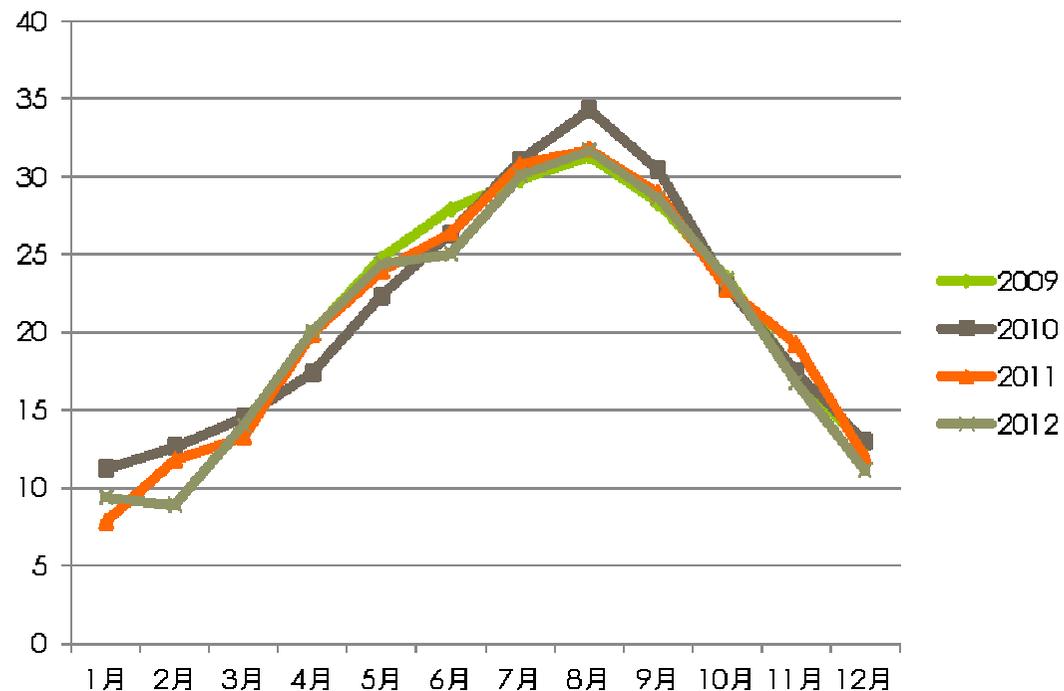
	通常処理 (JAVA)	完全分散処理 (Hadoop)
0.5GBデータ	37	1' 03
1GBデータ	1' 13	1' 38
2GBデータ	2' 35	2' 30
8GBデータ	10' 24	7' 57
16GBデータ	20' 24	15' 52

271秒(4分30)短縮!!

# 評価結果

## ○ 気象データの分析（処理結果）

集計例：月別最大平均気温の推移（大分市）



## 今後の課題

- 実運用環境の構築作業簡易化
  - ・ Hadoop環境の構築が複雑である。
- セキュリティの強化
  - ・ ファイアーウォールの設定
  - ・ ユーザ認証機能などが不完全である。
- MapReduce実装の簡易化
  - ・ サブプロジェクト（Hive、Pigなど）による、MapReduce処理記述の簡略化
  - ・ Hadoop StreamingやApache Thriftを用いた、Java以外の言語での開発

# 今後の展望

- 今回の実装による実運用への適合

- ・ ログファイルの解析

- 例：ホームページのWebアクセスログを分析し、レコメンド  
システムログを分析し、システムエラー発生時の状況を分析

- その他の可能性

- ・ さまざまなデータの組合せによる発見
  - ・ 大量データによる機械学習の実装
  - ・ DBに代わる新たなデータ管理方法

## 参考資料

- Hadoop徹底入門（翔泳社）
- Hadoop（O'Reilly Japan）
- Apache Hadoop  
(<http://hadoop.apache.org/>)
- CodeZine  
～複数マシンへHadoopをインストールする  
(<http://codezine.jp/article/detail/2699>)
- 気象庁「気象統計情報」HP  
(<http://www.jma.go.jp/jma/menu/report.html>)

ご清聴ありがとうございました。