

# ホストCOBOL資産を活用した 大規模バンキングシステムでの Hadoop適用検証事例

**2013年4月19日**

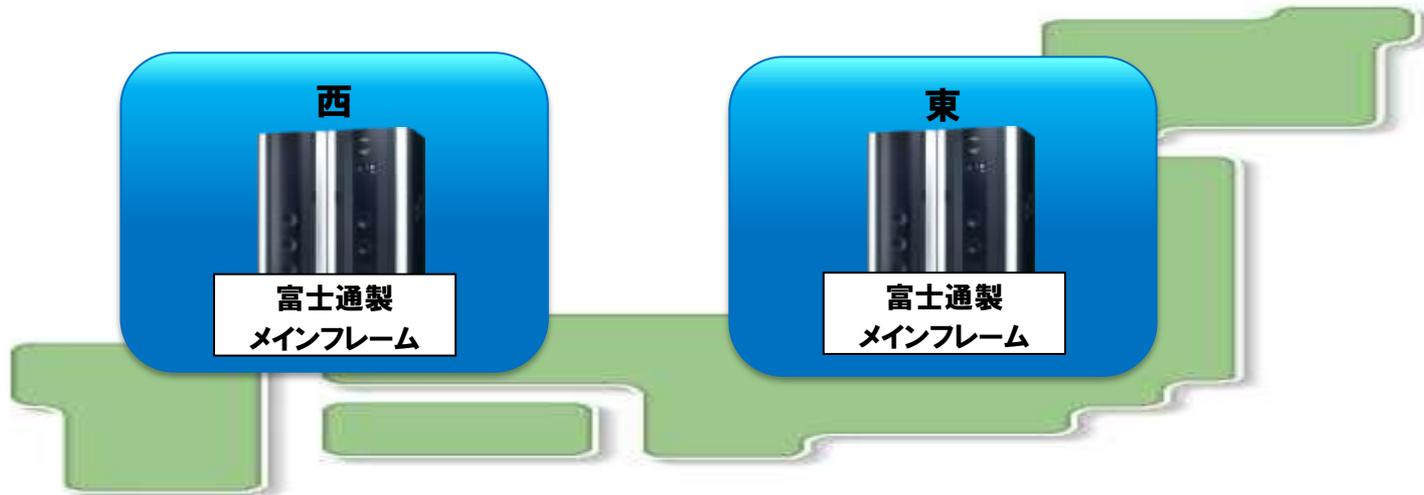
**株式会社富士通ミッションクリティカルシステムズ**

**先端ビジネス本部**

**仙名 弘昌**

# 1. お客様システム概要

- お客様                      某金融機関様
- 保有システム              勘定系・情報系システム  
(ホスト／オープン)
- ホスト資産                オンライン系 5M step  
                                    バッチ系        6M step
- 特徴                         東西でのシステム配置  
                                    勘定系システムは富士通製メインフレーム



## 2. Hadoop検証に至った経緯

### お客様システムの特徴

勘定系のホストシステムは  
バッチ処理がオンラインより重い  
(CPUのサイジングを実施)

現行でもホストシステムの機能を  
サブシステム(オープン)へ切り出  
している

以下の背景から、バッチ処理にHadoopを適用して有効性を検証

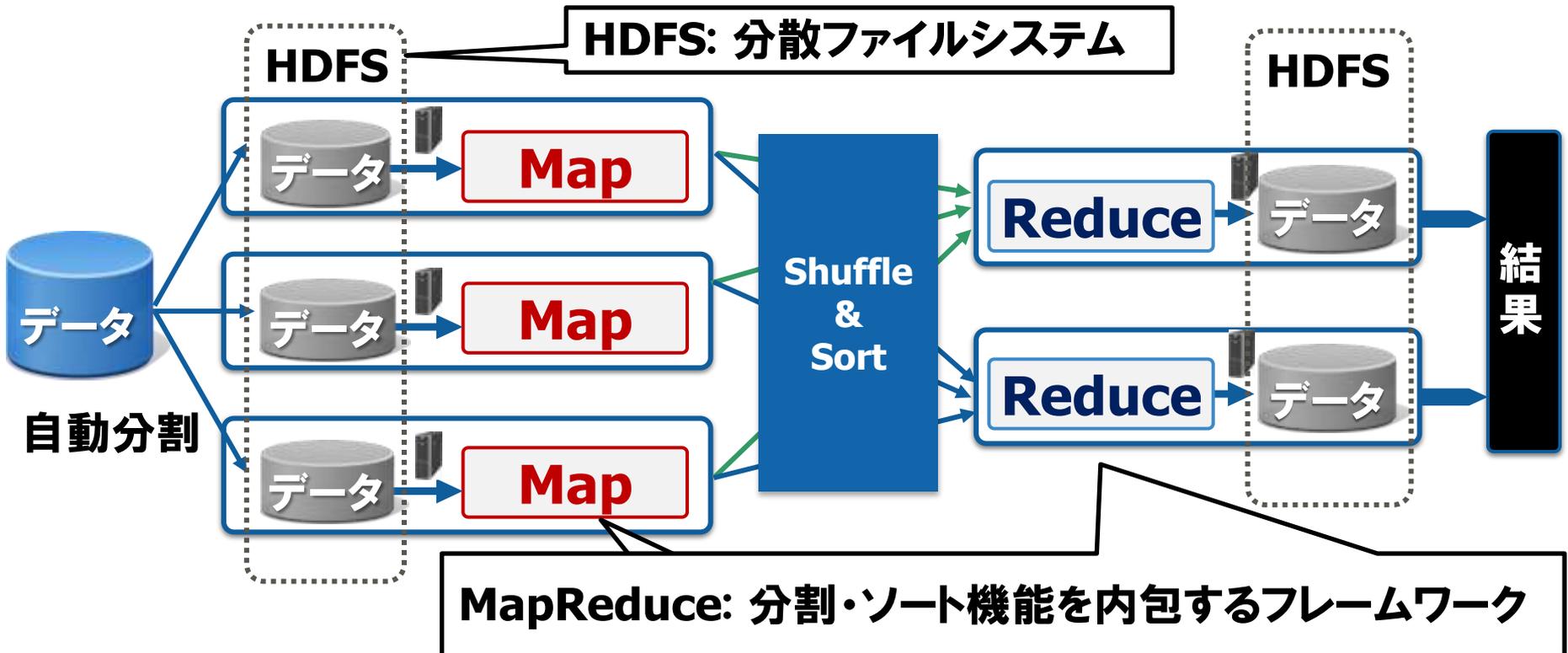
ユーザ側で意識せずに多重処理を  
容易に実現 (低コストで性能確保)

NetCOBOLのHadoop対応  
(既存COBOL資産が活用可能)

# (参考)Hadoopとは

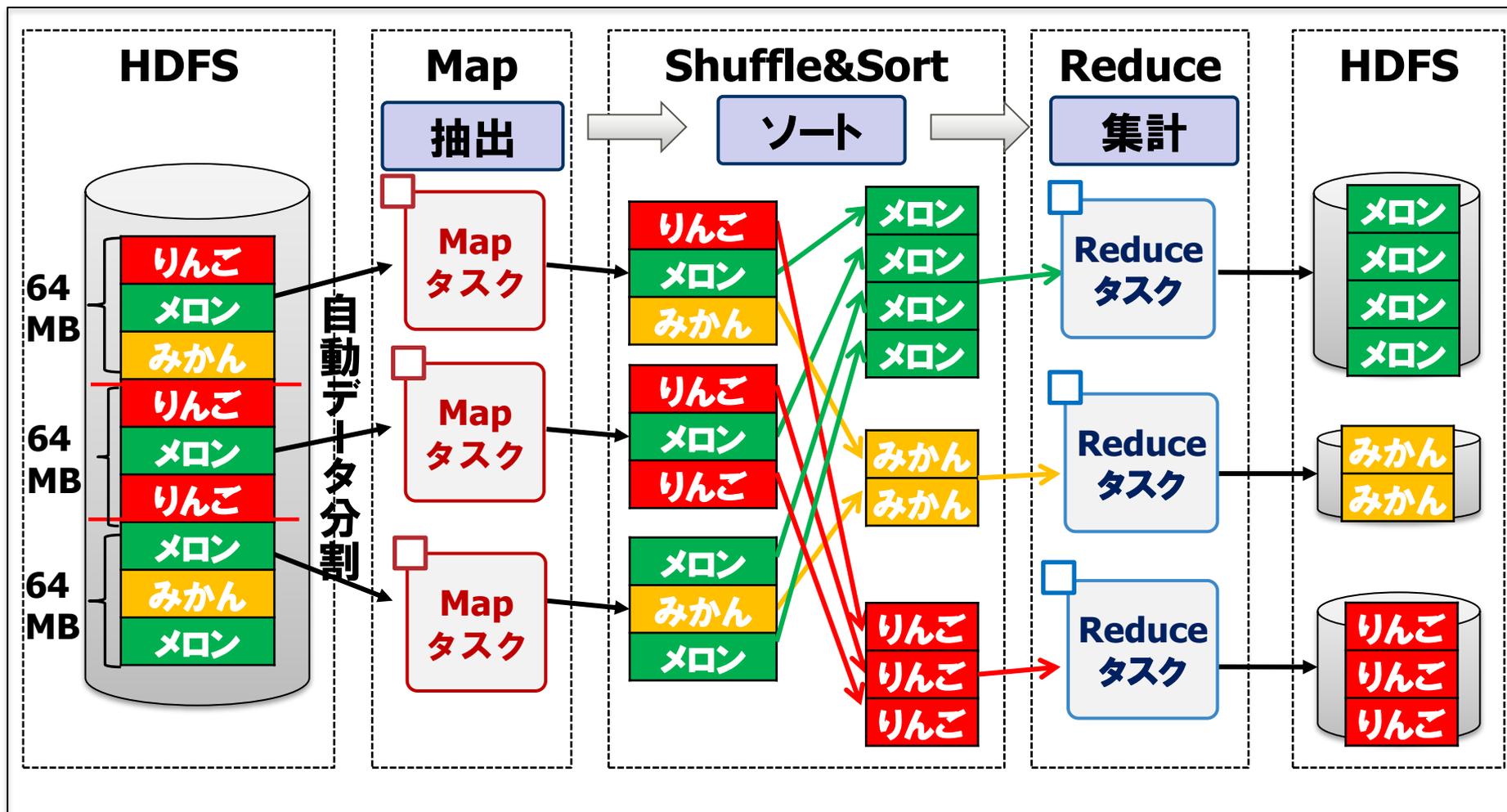
Hadoop:大規模データを並列分散処理するためのソフトウェア基盤

- スケールアウト(サーバ追加)による高速化
- 並列分散処理技術の活用による自動並列実行



# (参考) Hadoopの並列分散処理

Map → Shuffle&Sort → Reduceの3ステップで実現



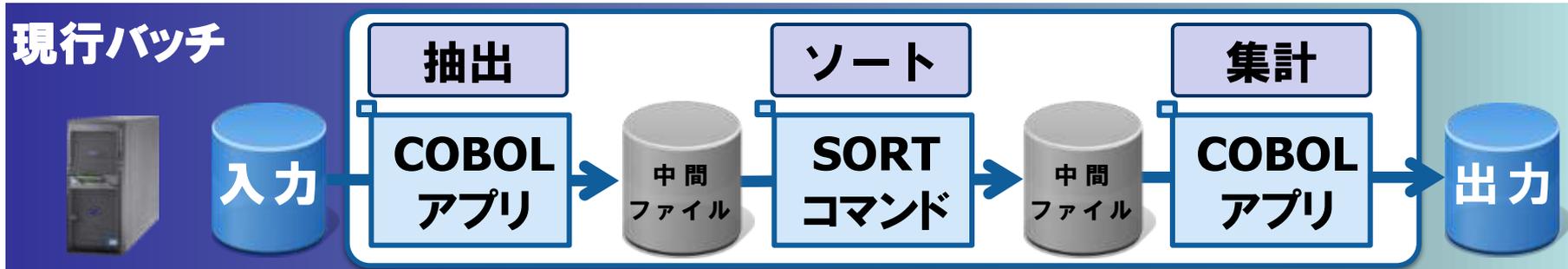
# (参考) NetCOBOLのHadoop対応

既存のCOBOLバッチアプリをHadoop上で実行可能

バッチ処理時間を並列分散処理により大幅短縮

- COBOLアプリに手を入れずにHadoop上で実行可能
- COBOLデータも利用可能であり、データの変換は不要

現行バッチ



Hadoop適用後



# 3. 検証概要

## ■ 検証の目的

勘定系システムのホストバッチ業務以下の観点で検証を実施

### Hadoop適用によるバッチ処理性能確認

抽出したバンキングシステムのバッチ処理モデルに対して、ホスト側で実行した場合と比較して性能面で優位性があるかを確認

### 現行システムのCOBOL資産活用による移植性確認

現行システムのCOBOL資産を活用してHadoopを適用することで、オープンシステムへ移行する際の移植性が高いかを確認する

次期システムでのHadoop適用範囲の見極めとコストメリットを評価する

# 4. 検証スケジュール(実績)

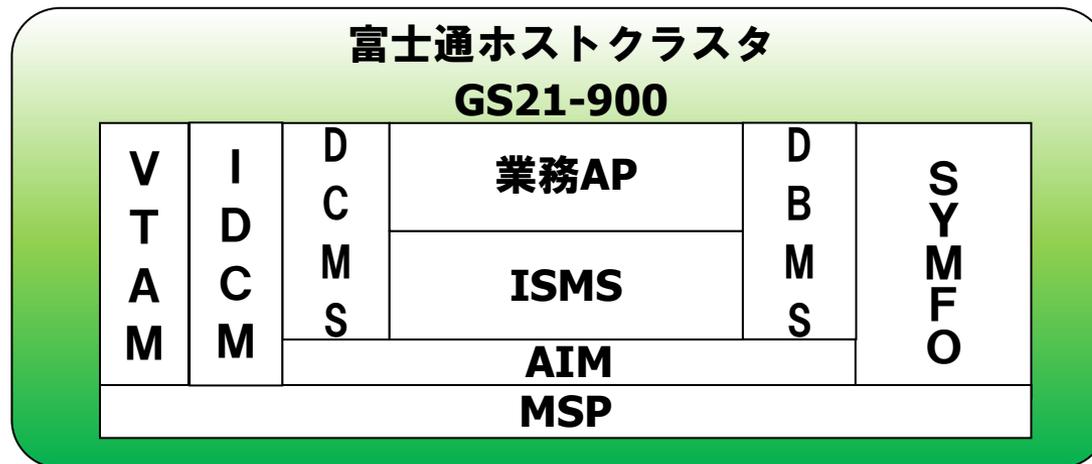
検証計画～検証測定までを2012年7月～10月の期間で実施

	2012年			
	7月	8月	9月	10月
検証モデル検討	→			
インフラ基盤整備	明細作成/機器手配	→	設置現調	→ 環境構築
テスト用アプリ設計/作成			→ ホスト環境	→ オープン環境 → テストデータ作成 → 試験項目/手順作成
検証作業			→ 検証パターン抽出	→ ホストCOBOL NetCOBOL
評価				→ 取りまとめ

# 5. 検証マシン環境 (1/3)

## ■ 検証用ホスト機器

ホストシステム側の検証環境構成を以下に示す



**FCLS**  
※ホストクラスタ向けに2パス  
※各ETERNUS筐体向けに8パスずつ

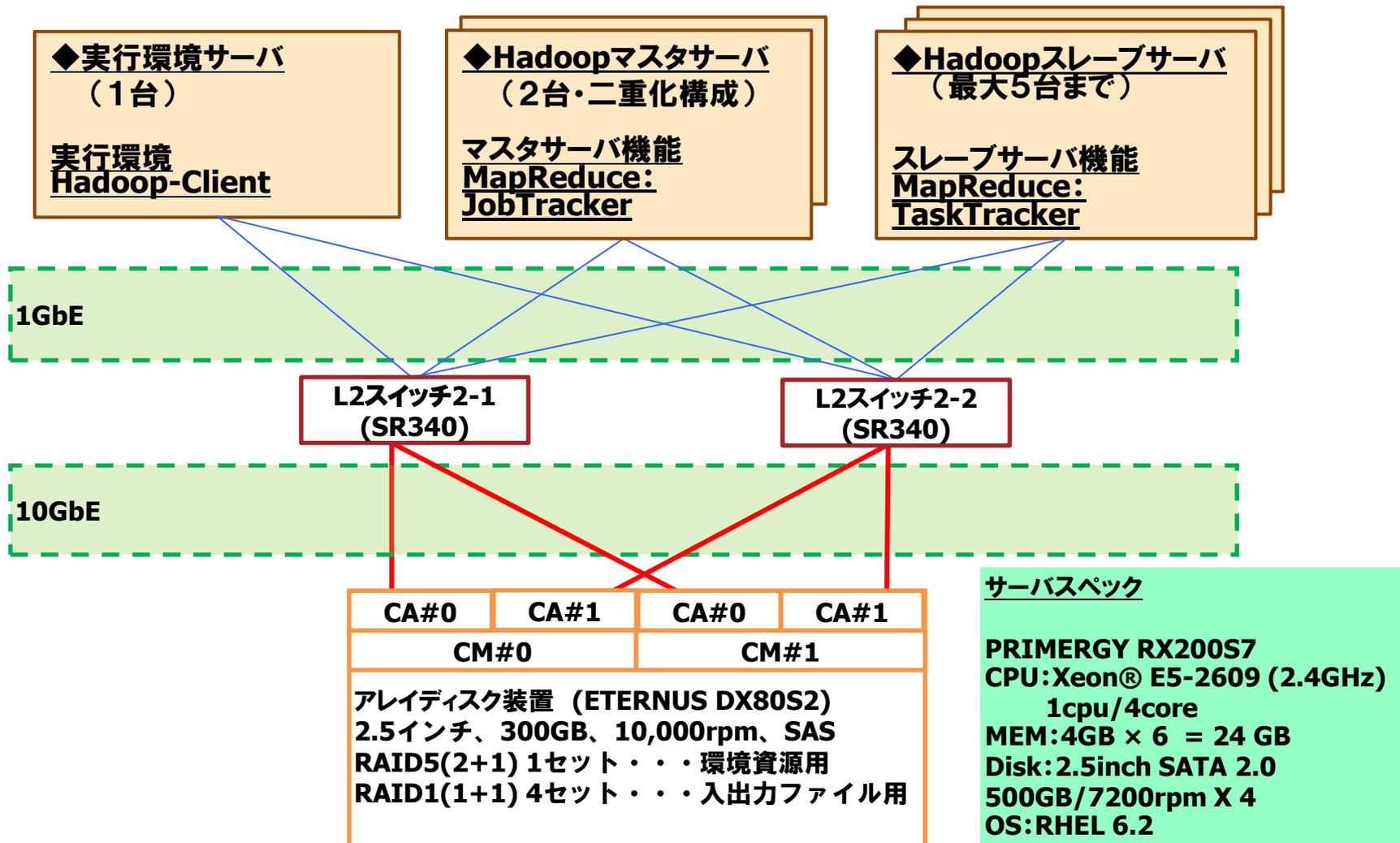
**FCLS**  
※ホストクラスタ向けに2パス  
※各ETERNUS筐体向けに8パスずつ



# 5. 検証マシン環境 (2/3)

## ■ 検証用オープン機器(ハード構成)

オープンシステム側検証環境構成を以下に示す



# 5. 検証マシン環境 (3/3)

## ■ 検証用オープン機器(ソフト構成)

オープンシステム側の主な搭載コンポーネントを以下に示す

搭載製品		Hadoop マスタ サーバ	Hadoop スレーブ サーバ	実行環境 サーバ
OS	Red Hat Enterprise Linux	●	●	●
言語	NetCOBOL		●	●
Java実行環境	JRE	●	●	●
ファイルシステム DFS※	管理サーバ (MDS)	●		
	クライアント (AC)	●	●	●
分散処理基盤 Hadoop	NameNode	●		
	JobTracker	●		
	DataNode		●	
	TaskTracker		●	
	Hadoop-Client			●

※DFS:富士通独自の分散ファイルシステム

# 6. 検証バッチ業務と測定条件

## ■ 検証バッチ業務の概要

取引ログデータの目的別振分

⇒ 条件ファイルをもとに取引ログデータを複数の取引種別に分類

## ■ 測定条件

### (1)スレーブサーバ数

1、3、5台と変動させ測定を実施

### (2)タスク数

- Mapタスク数: 4タスク/スレーブサーバ  
(BDPPS※推奨値「コア数」を採用)
- Reduceタスク数: 3タスク/スレーブサーバ  
(BDPPS推奨値「コア数-1」を採用)

### (3)測定アプリケーション

- ホストCOBOLソース: そのまま使用
- ホスト固有資産(アセンブラ等): 同等の擬似アプリを作成

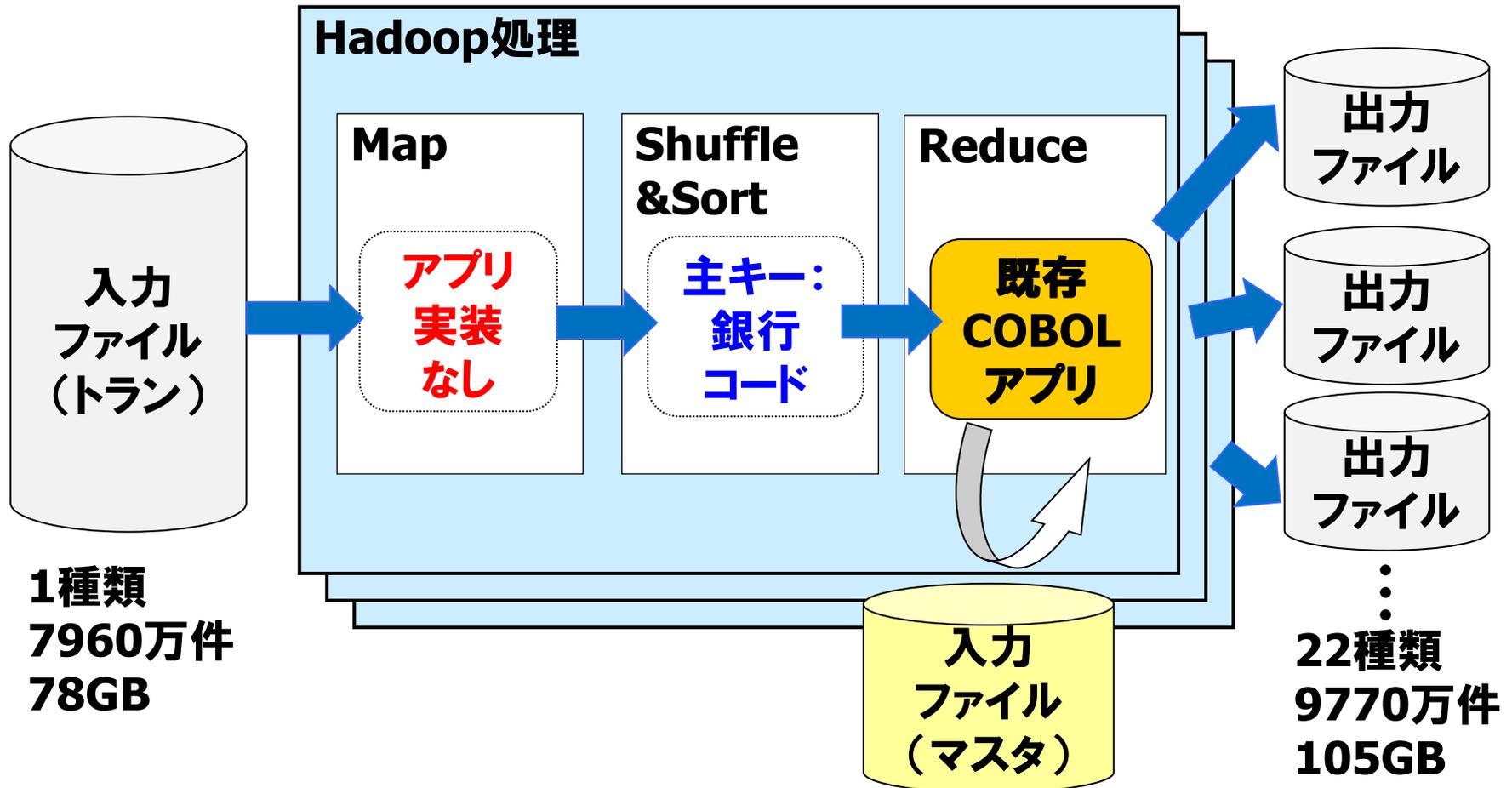
※BDPPS: Interstage Big Data Parallel Processing Server

# 7. 処理方式概要

## ■ 処理方式

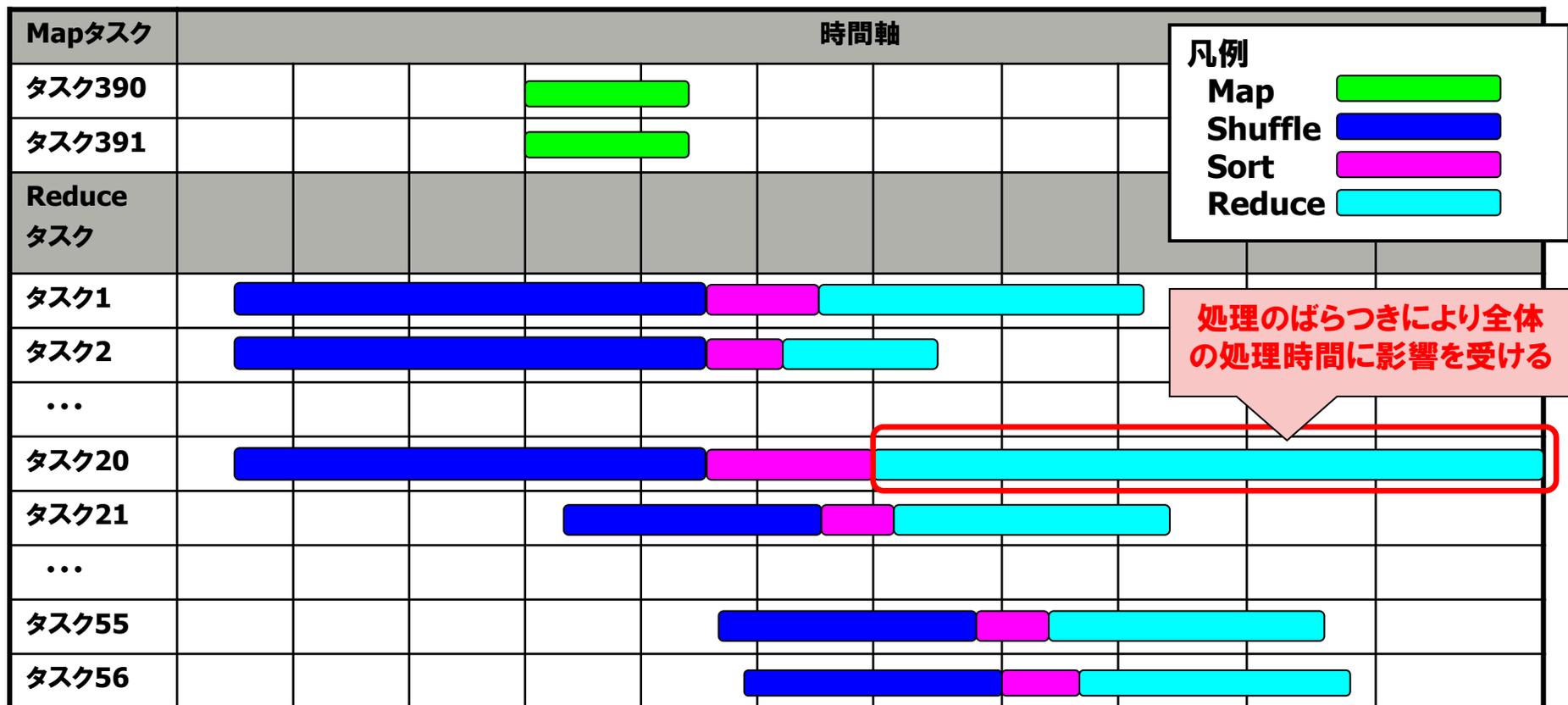
業務の処理概要と入出力データ量は以下の通り

### オープンサーバ(Hadoop)



# 9. 検証結果考察 (1/2)

大規模な銀行の処理が長時間化し全体の処理時間に影響

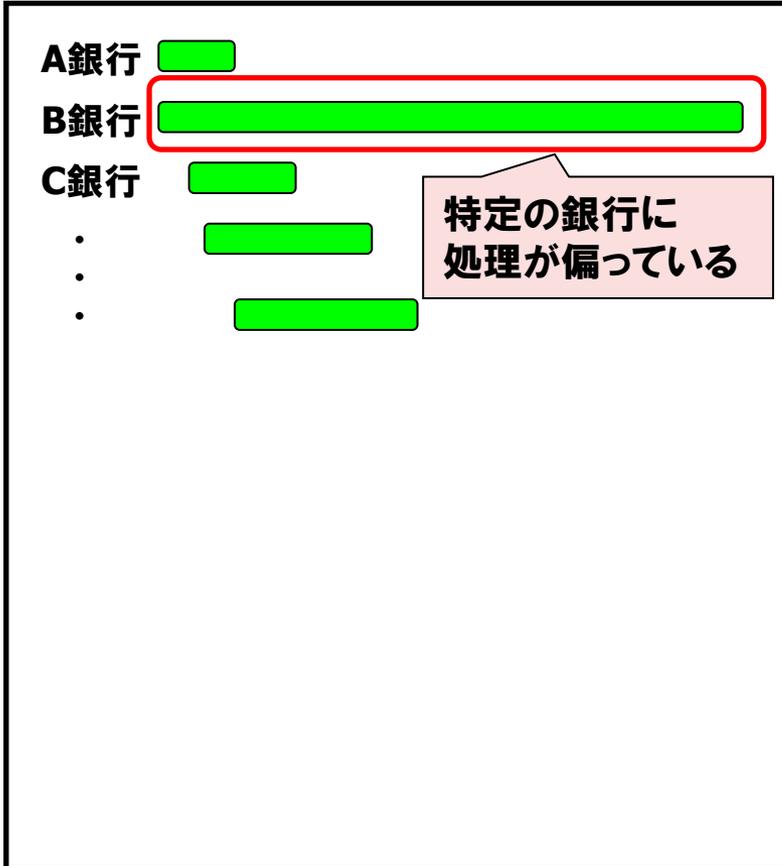


【ガントチャートの特徴(Reduce)】

# 9. 検証結果考察 (2/2)

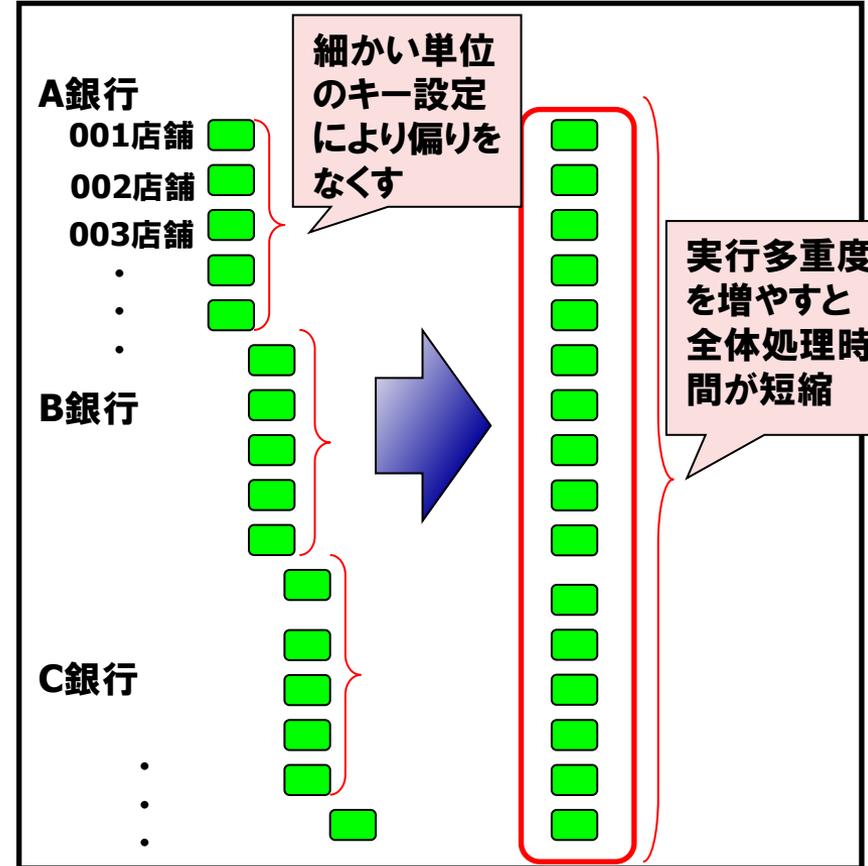
## 銀行単位の並列分散処理

特定の銀行に処理が偏り、サーバを追加しても、最大の処理時間以上の短縮はできない



## 店舗単位の並列分散処理

サーバ (CPUコア) 追加により性能向上 (処理時間短縮) の余地がある



## 性能

- 15多重度でホストの2倍性能を実現  
(構成ファイルの設定だけで多重度を変更)
- 処理単位の細分化により性能向上
- Map処理で対象データを絞り込めば  
さらに性能向上

## 移植性

- 既存COBOLアプリの改修なし

## Hadoop適用のポイント

- 処理単位を細分化する  
ex) 銀行単位 → 店舗単位
- Shuffle & Sortは効果大  
既存のソート処理は狙い目

- ・ Linuxは、Linus Torvalds氏の米国およびその他の国における登録商標または商標です。
- ・ Red Hat、RPMおよびRed Hatをベースとしたすべての商標とロゴは、Red Hat, Inc. の米国およびその他の国における登録商標または商標です。
- ・ Apache Hadoop、Hadoop、HDFSはApache Software Foundation の米国およびその他の国における登録商標または商標です。
- ・ Javaは、Oracle Corporationおよびその子会社、関連会社の米国およびその他の国における登録商標です。
- ・ その他の会社名または製品名は、それぞれ各社の商標または登録商標です。



**FUJITSU**

shaping tomorrow with you