FUJITSU

shaping tomorrow with you

Hadoopのキホンと活用術のご紹介

株式会社 富士通研究所
人工知能研究センター
人工知能基盤プロジェクト
主管研究員 上田 晴康

自己紹介

FUJITSU

- 大学卒業時は、
  - 第2の人工知能ブームが来た頃
  - 機械学習の研究してました
- 人工知能の冬の時代は、並列処理や組合せ最適化の研究
- 「ビッグデータ」の時代
  - Hadoop、Sparkの研究を経て
  - 機械学習をもっともっと便利にする技術開発に携わっています

本日のアウトライン

FUJITSU

1. Apache Hadoopの基本から最新動向まで

2. Hadoopをもっと簡単・便利に活用する技術のご紹介

3. ビッグデータを活用する人工知能技術のご紹介

FUJITSU

1. Apache Hadoopの基本から最新動向まで

- Hadoopとは
- 技術紹介

何故Hadoopが必要になったのか？

FUJITSU

データ量はどんどん増えるのに、処理速度はそんなに上がらない

2002年⇒2012年の向上率

＜容量＞

価格性能比の向上

ストレージ量：33倍

メモリ：30倍

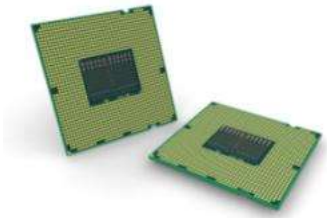
VS

＜処理速度＞

データ処理技術の向上

ストレージ読み込み：数倍

CPU：20倍



ネットワーク：13倍

Hadoopとは

FUJITSU

- 大量のデータを手軽に複数のマシンに分散してバッチ処理できるオープンソースのプラットフォーム

  - Hadoop（ハドゥープ）という名前は開発者の子供が付けた象のぬいぐるみの名前が由来

- hadoopの特徴は、大量高速処理と劇的なコストダウン
  - hadoopを用いれば、安価なコモディティサーバ（IAサーバ）をそろえるだけで、大量データ処理を非常に高速に行うことができます。
    - 数千台並べることも可能！
  - 耐故障性が高く（機材故障時に処理を自動リスタート）、運用工数が低くなります。
  - hadoopの実行環境としてクラウドコンピューティング(AmazonEMRなど)を利用することで、機材の購入すら省くことも可能です。

Hadoopの歴史

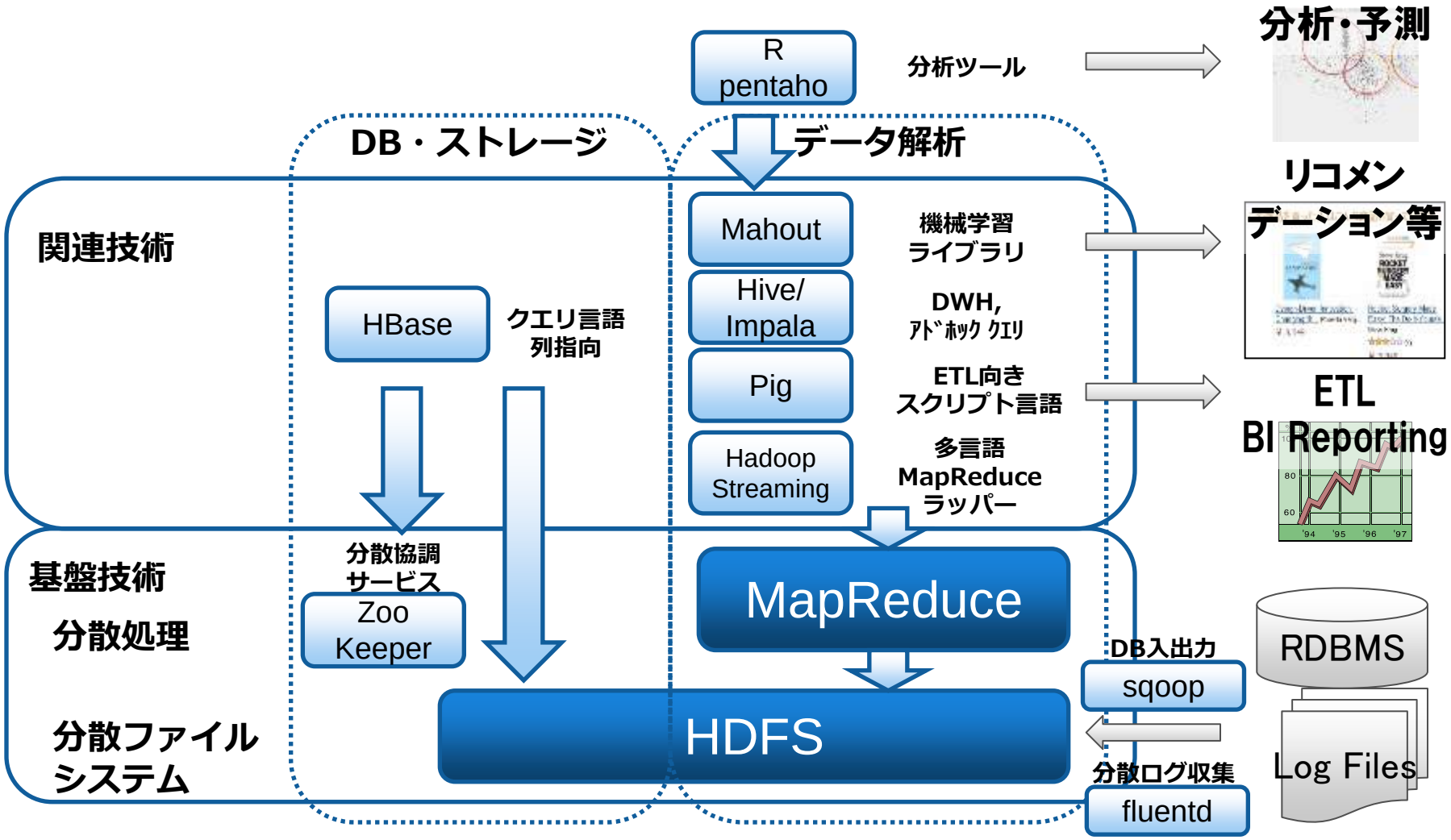
FUJITSU

- 2004: GoogleがHadoopの基となるMapReduceの論文公開
- 2005: Hadoopプロトタイプ作成
  - オープンソースのテキスト検索エンジンLuceneを利用したWeb検索エンジンNutchの中心的な開発者、Doug Cuttingらが、Nutchを数十億のWebページに対応させる
- 2006: Yahoo!が主要投資開始
  - Yahoo!が興味を持ち、Nutchから分散バッチ処理システムとして汎用的に利用できる部分を切り離して、独立したHadoopプロジェクトとして立ち上げる
  - Doug CuttingもYahoo!へ
  - このような経緯から、米Yahoo!は現在もHadoopの最大のユーザーの一つ
- 2008:大規模なソート処理のTerasortベンチマークで新記録を達成
- 2009: Cloudera事業開始（Hadoopの商用利用）
  - 創立者はMike Olson（元Oracle副社長）、Christophe Bisciglia（Google）、Dr.Amr Awadallah（VivaSmart）、Jeff Hammerbacher（Facebook）など
  - 2009年9月: Doug CuttingもClouderaへ
- 2011：Hortonworks事業開始（Yahoo!からスピンアウト）
  - コミュニティを重視し、YARNを開発してコミュニティに寄付

Doug Cutting

エコシステムの全体像

■高速バッチ基盤を活用する関連技術群が充実している



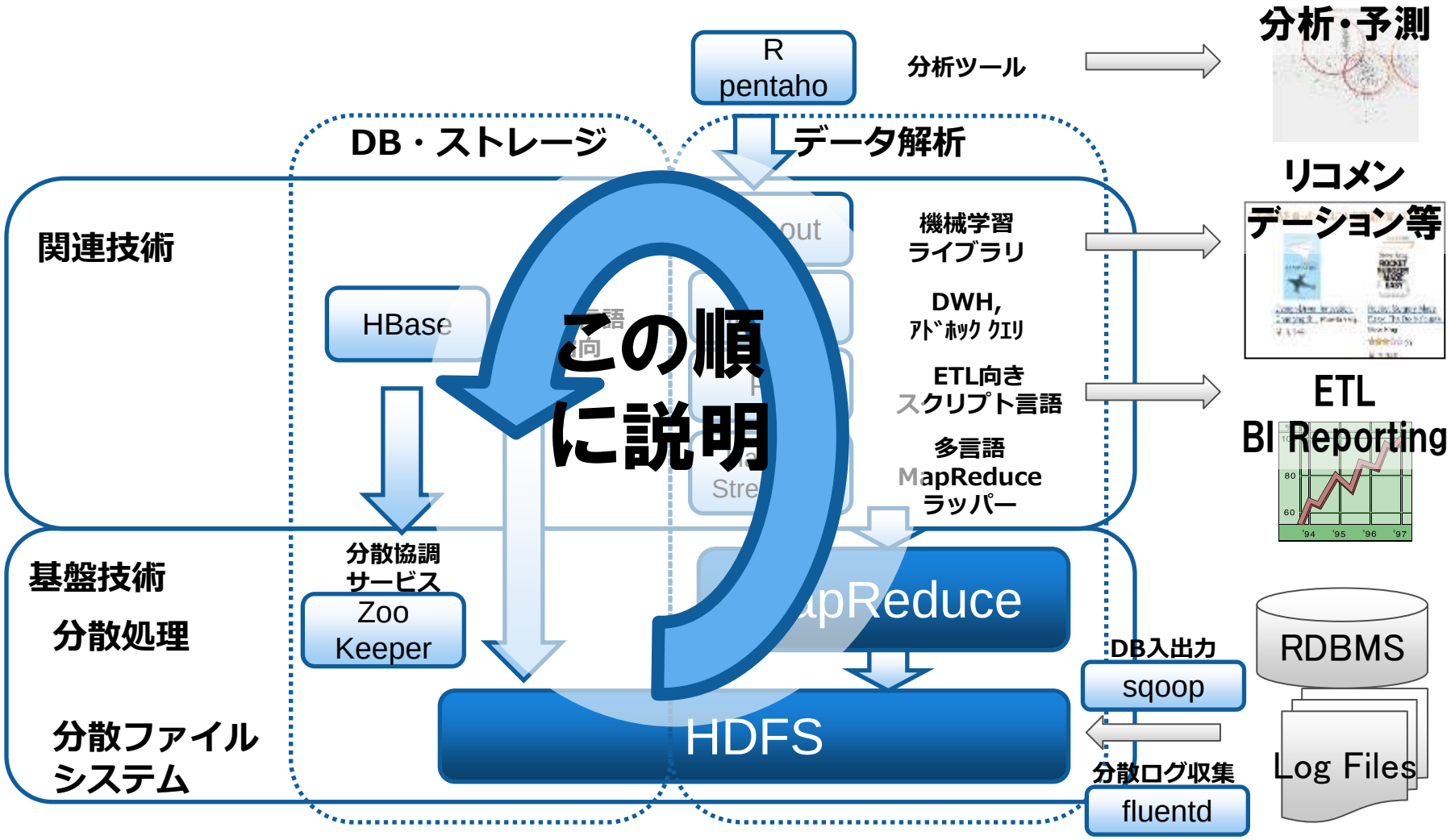
FUJITSU

1. Apache Hadoopの基本から最新動向まで

- Hadoopとは
- 技術紹介

エコシステムの全体像

■高速バッチ基盤を活用する関連技術群が充実している



Hadoopのコア技術(1)
分散ファイルシステムHDFS

FUJITSU

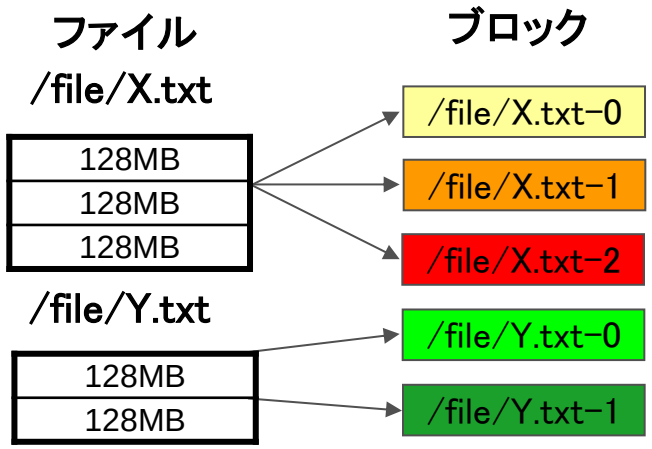
- クラスタ全体でひとつの大きな仮想ファイルシステムを形成
- ファイルをブロック(128MB)に分け、メタデータDBで管理
  各ファイルのどのブロックがどのDataNodeに保存されて
  いるかを記録
- 同じブロックを複数のDataNodeに保存（レプリカ）

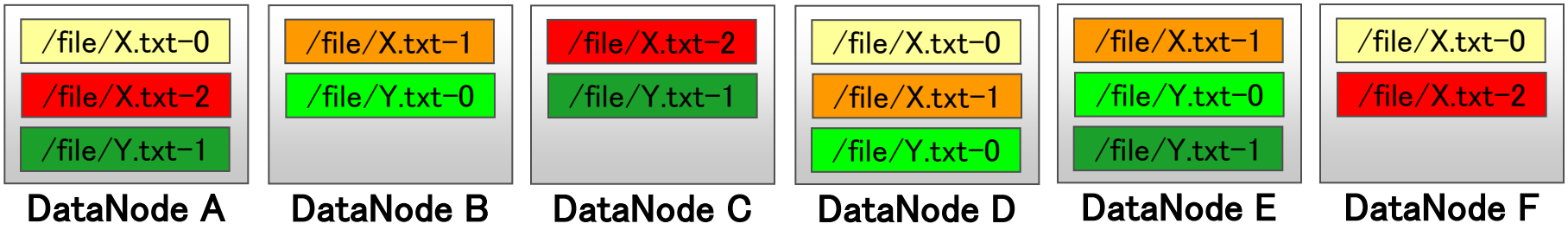
シーク回数を減らすことに特化して高速アクセス

NameNode

HDFS メタデータDB

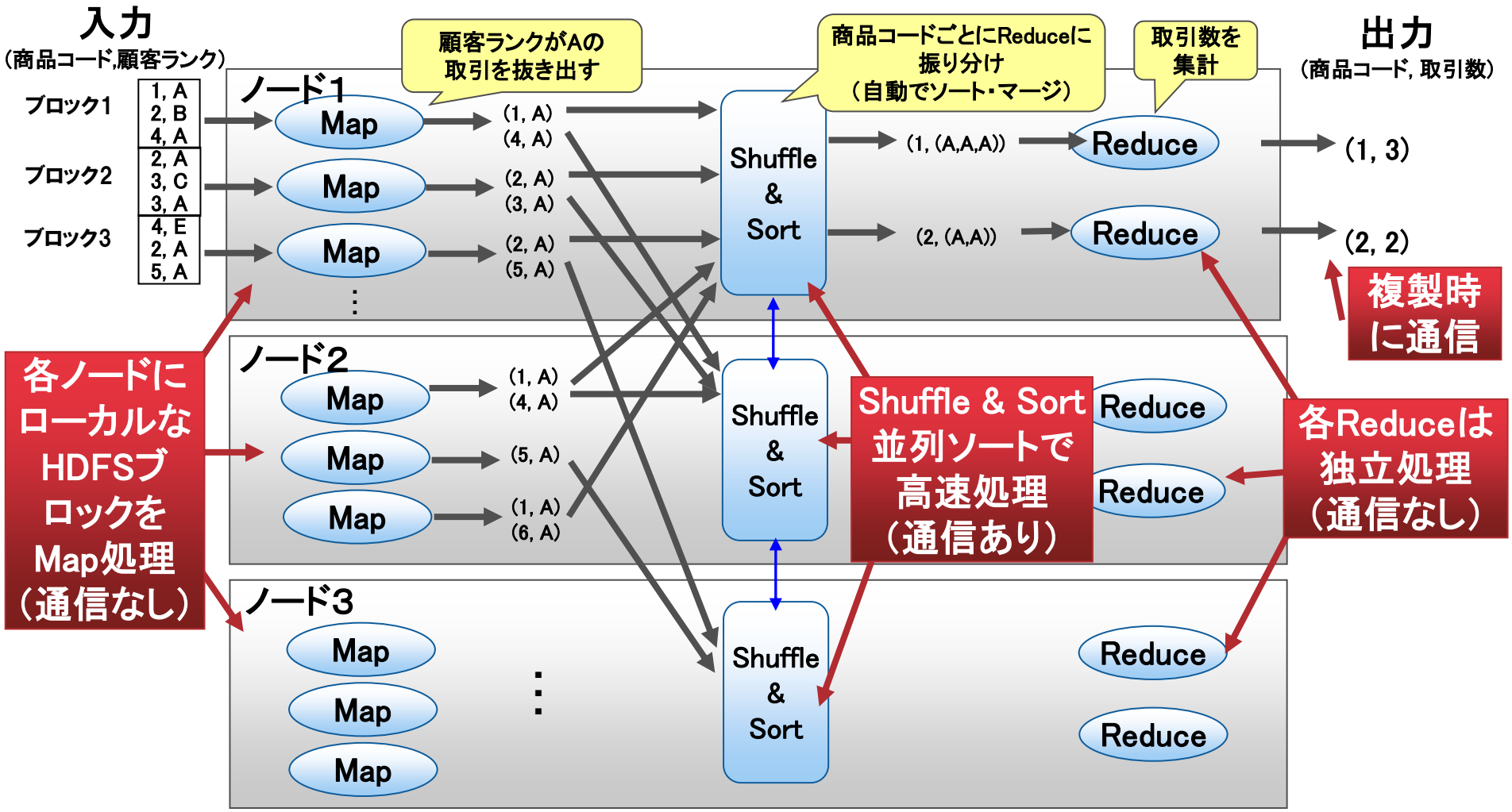
ファイル名	ブロック番号	DataNode
/file/X.txt	0	A,D,F
	1	B,D,E
	2	A,C,F
/file/Y.txt	0	B,D,E
	1	A,C,E





Hadoopのコア技術（2）MapReduce

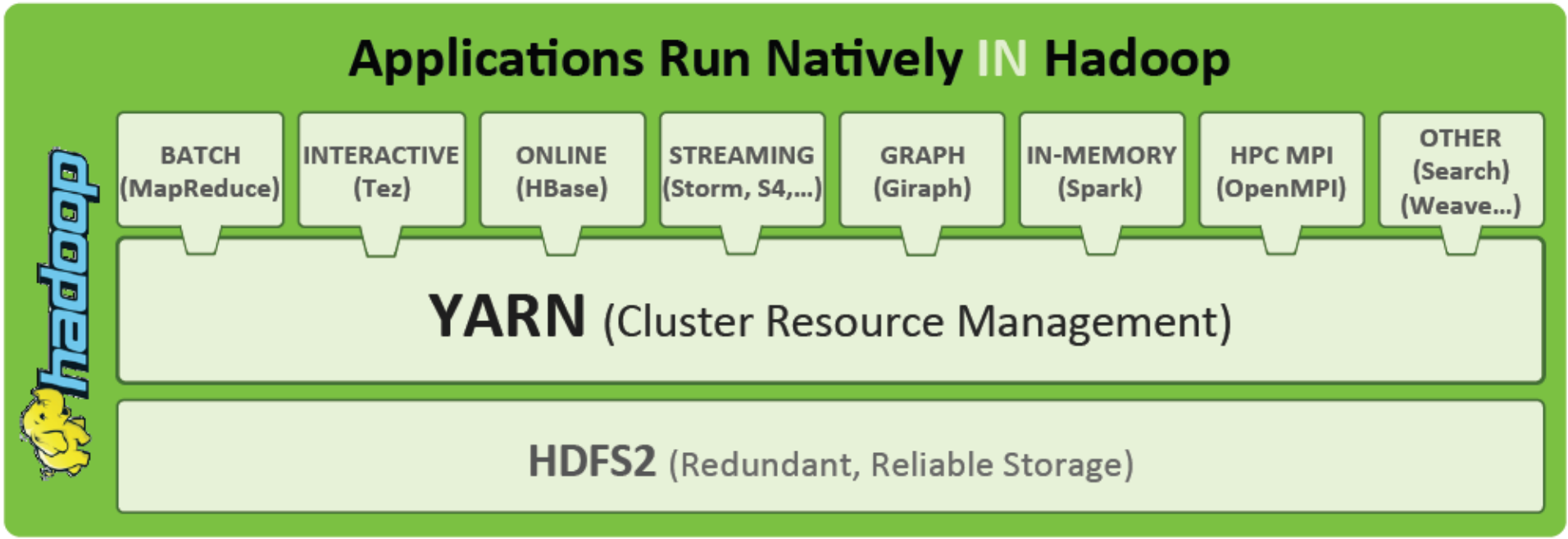
■ランクA顧客が取引した商品の数を集計する例



Hadoop最新動向
YARN: 複数の処理フレームワーク向け基盤

FUJITSU

- 計算資源の管理を、処理の管理から分離
- MapReduce以外に、準リアルタイム処理や複数回データ処理する用途も可能に
- 数万サーバまでスケール可能になった



Apache Hadoop YARN - Enabling Next Generation Data Applications by Hortonworks on Aug 12, 2013
http://www.slideshare.net/hortonworks/apache-hadoop-yarn-enabling-nex

Hadoopエコシステムにある(準)リアルタイム処理フレームワーク

FUJITSU

- HBase: Key Value Store
- Apache Tez: 複数のMap Reduce連携を高速化
- ストリーム処理/CEP: Apache Storm、Apache S4、Apache Samza
  - 計算資源をもらうだけで、リアルタイムなデータ処理・通信は各フレームワークが独自に実行
- リアルタイムクエリ: Impala(Cloudera)、Presto(Facebook)、Drill(MapR)…
- Apache Spark: インメモリ処理

Hadoopの最新動向

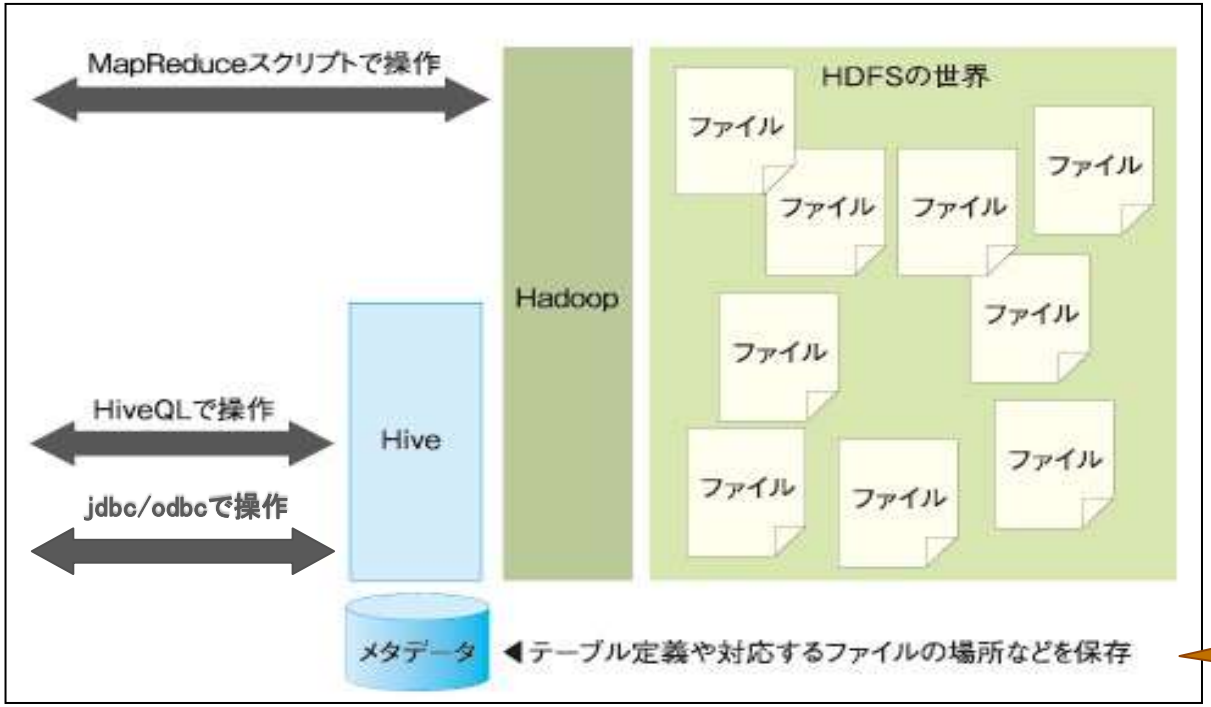
FUJITSU

- Hadoop 3.0に向けてHDFSを改良中
  - イレイジャーコーディング (Erasure Coding)
  - レプリカをやめて、誤り訂正符号を使って耐障害性、可用性を確保

  ⇒必要ディスク容量が減らせる
  - 従来は3レプリカだと3倍のHDDが必要だった

技術紹介（3）Hive

- Hadoop上で動作するデータウエアハウス
- HDFSなどの分散ファイルシステム上に存在するCSVなどのファイルに対して、HiveQLというSQLライクな言語で分類集計操作ができる。



Facebookが開発し、2008年Hadoopプロジェクトに寄付された

デフォルト設定では、メタデータとしてカレントディレクトリのderbyDBを利用

HiveQL

FUJITSU

- HiveQLはMap Reduceのラッパーになっている。
- SELECT文などを実行すると裏でMap&Reduceのジョブが走り出して、分散処理されて結果を得る。

●テーブルの結合

```
hive> SELECT z.zip, p.pref, z.city, z.town FROM zip z
    > LEFT OUTER JOIN pref p ON (p.id = z.pref)
    > WHERE z.ver = '2008-12-26' AND z.town REGEXP '銀座';
Total MapReduce jobs = 1
…略…
Ended Job = job_200901011221_0005
OK
0691331 北海道 夕張郡長沼町 銀座
3670052 埼玉県 本庄市 銀座
1040061 東京都 中央区 銀座
3950031 長野県 飯田市 銀座
4480845 愛知県 刈谷市 銀座
7450032 山口県 周南市 銀座
8040076 福岡県 北九州市戸畑区 銀座
Time taken: 69.588 seconds
```

Hiveの最新動向

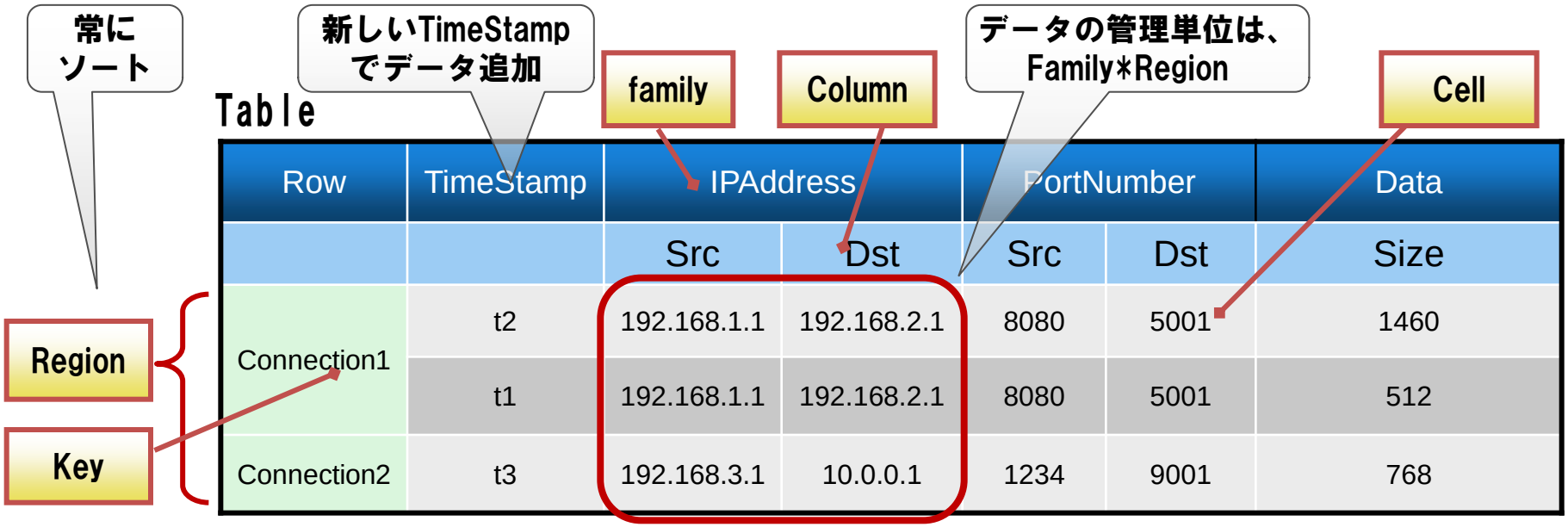
FUJITSU

- データを保持するフォーマットは、CSV/TSVではなくカラムベースのORCFileフォーマットやParquetフォーマットが推奨に
- Hive 2.0がリリース（2016年2月）
  - 秒未満のクエリ実行に向けて高速化
- バックエンドはMap Reduceでなく、TezまたはSparkが推奨になった
  - 従来は30秒程度のオーバヘッドがあった
- MapやReduceタスクを実行する JavaVMは立ち上がりっぱなしにできるようになった
  - JITコンパイラが良く効く
  - データをメモリにキャッシュし続けられる
- 普通のRDB/DWH同様にクエリの最適化がきっちり効くようになった
  - Cost Base Optimization

技術紹介(4) HBase

FUJITSU

- 列指向のKVS(Key Value Store)。書き換え可能。ランダムアクセス可能
  - GoogleのBigTableのOSSクローン(Javaで実装)
  - 数十億行/数百万列の疎なデータを格納可能
- 内部はRegion単位で分散管理。自動的に負荷分散(Sharding)。
- 書き込み時はジャーナルおよびRegionの更新を定期的にHDFSに保存
  - Map Reduceを用いるためのJava APIもある



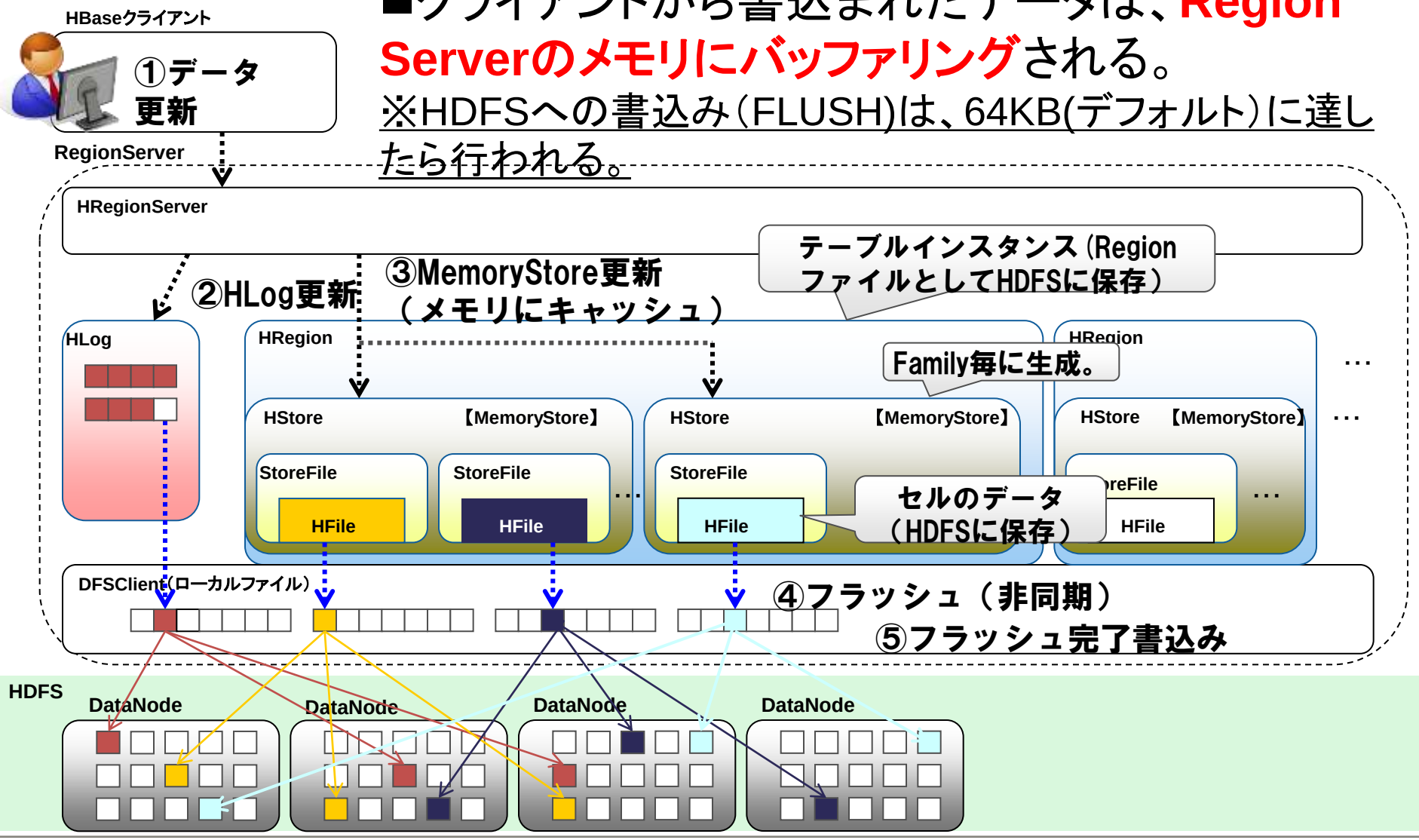
Row	TimeStamp	IPAddress		PortNumber		Data
		Src	Dst	Src	Dst	Size
Connection1	t2	192.168.1.1	192.168.2.1	8080	5001	1460
	t1	192.168.1.1	192.168.2.1	8080	5001	512
Connection2	t3	192.168.3.1	10.0.0.1	1234	9001	768

(Table, Row_Key, Family, Column, Timestamp) = Cell (Value)

HBaseの動作（更新処理）

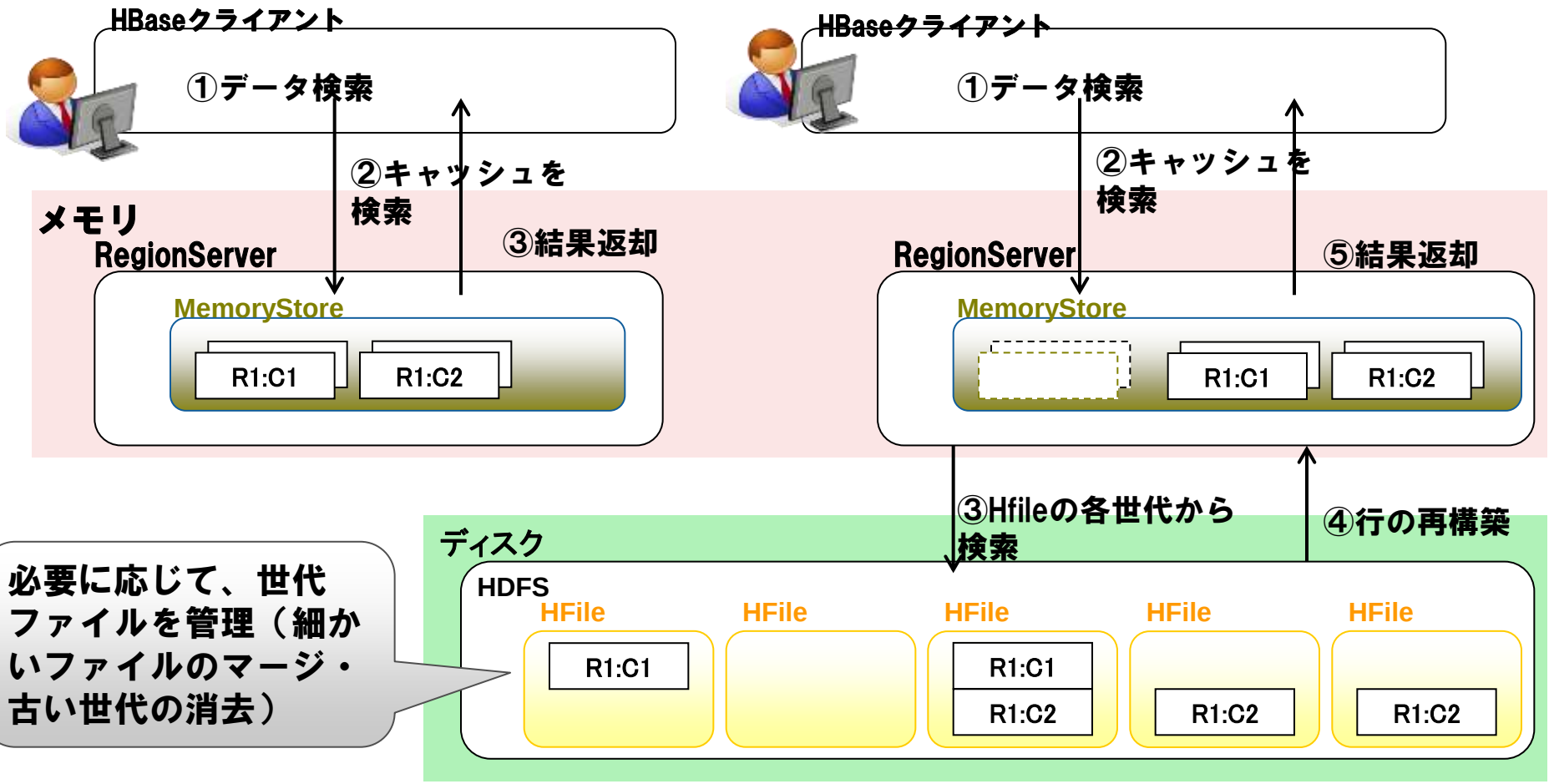
FUJITSU

■クライアントから書込まれたデータは、**Region Serverのメモリにバッファリング**される。
※HDFSへの書込み（FLUSH)は、64KB(デフォルト）に達したら行われる。



HBaseの動作(検索処理)

- RegionServerキャッシュを調べ、キャッシュにない場合は、ディスク(HDFS上のHFile)を検索する。



FUJITSU

2. Hadoopをもっと簡単・便利に活用する技術の紹介

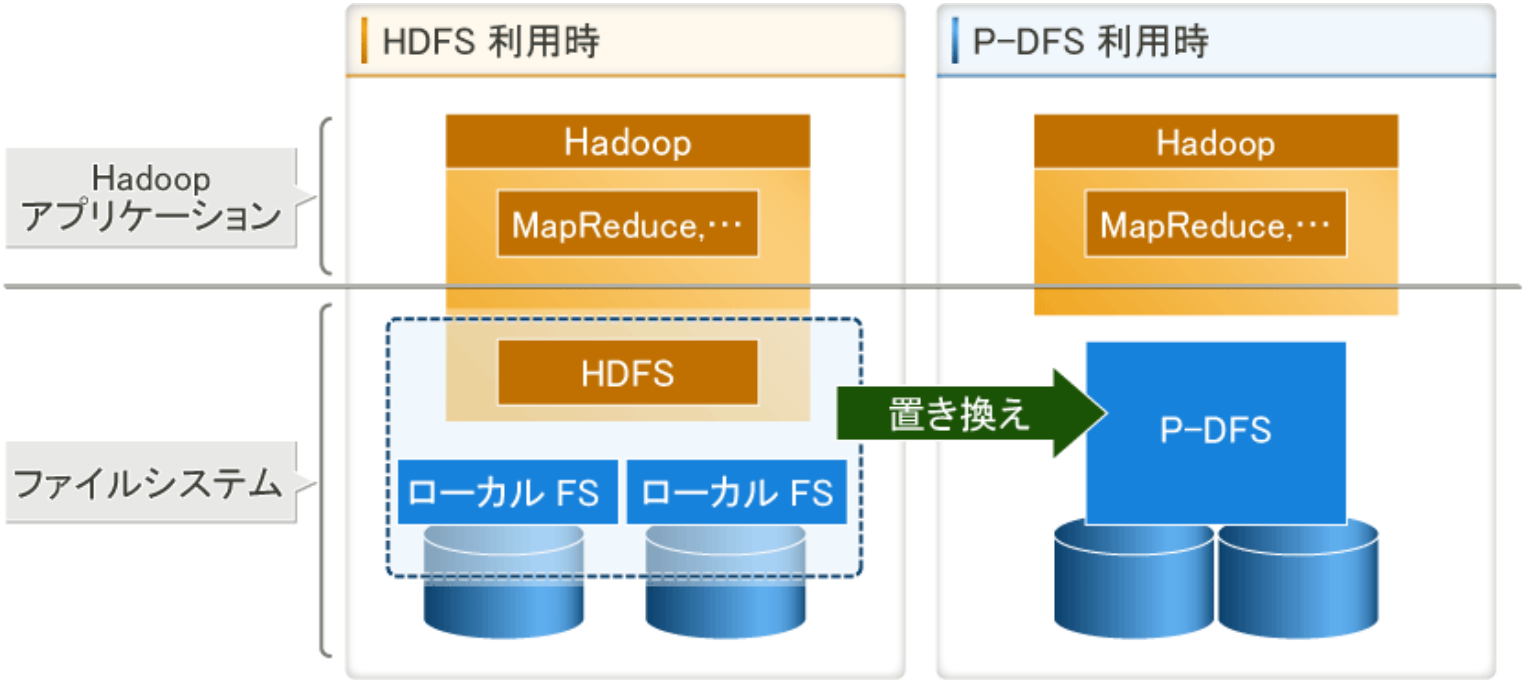
- PDFS
- Hadoopマルチプレクサ

富士通の取り組み

FUJITSU

Hadoopをエンタープライズ向けに強化

- エンタープライズ向けビッグデータ活用
  - HDFSを富士通ファイルシステム”P-DFS”に置き換え
  - 格納先はローカルディスクではなく、共用ディスク装置を使用



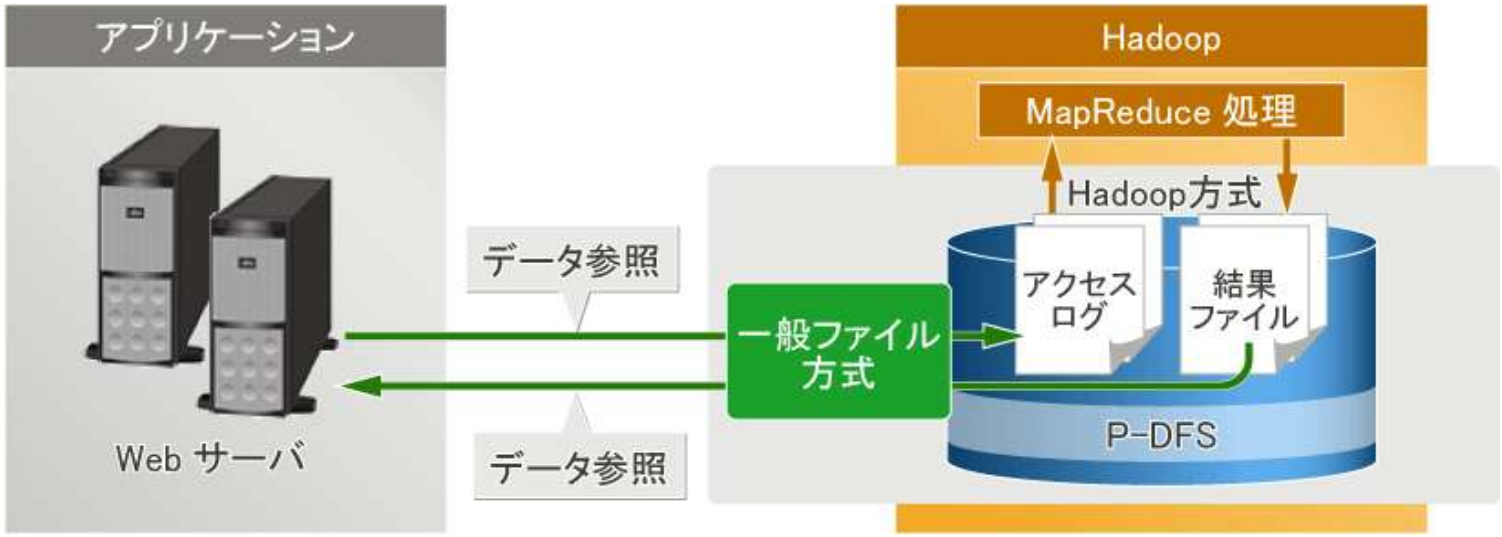
P-DFSによる課題解決1

FUJITSU

Hadoop方式と一般ファイル方式の両方をサポート

HDFSの課題	
1	既存システムとのデータ連携性が悪い
2	データの保全性が低い
3	設計・チューニングに高度なノウハウが必要

- 前準備をする必要はなく、既存システムのファイルをそのままHadoopで処理可能
- 処理結果ファイルは、従来使用の一般アプリケーションからPOSIXインタフェースで参照可能
- 一般アプリケーションもそのまま利用できるため、データ転送の処理や新しくアプリケーションを開発する必要はない
- POSIXサポートにより、市販セキュリティソフトも使用可能



P-DFSによる課題解決2

FUJITSU

従来のバックアップソフトが利用可能

HDFSの課題	
1	既存システムとのデータ連携性が悪い
2	データの保全性が低い
3	設計・チューニングに高度なノウハウが必要

- データのバックアップが**容易**
  - 共用ディスク装置の機能を使用可能
  - 既存システムで利用しているバックアップソフトをそのまま使用可能
- 管理サーバ(Namenode)ダウンに備えたバックアップが**不要**
  - 共用ディスク装置は、二重化されている
  - **MDS冗長化で、業務継続も可能**

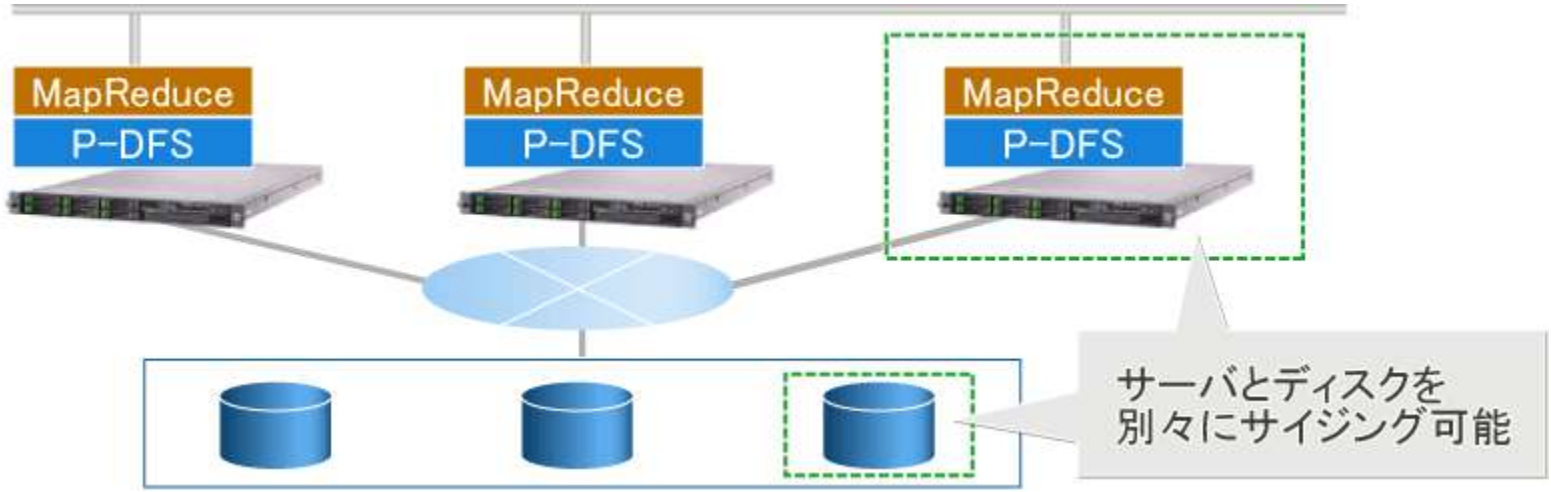
P-DFSによる課題解決3

FUJITSU

事前設計不要／サーバとディスクを個別にサイジング可能

HDFSの課題	
1	既存システムとのデータ連携性が悪い
2	データの保全性が低い
3	設計・チューニングに高度なノウハウが必要

- HDFSが各サーバのローカルディスクを使用するのに対し、P-DFSは共用ディスク装置を使用
- サーバ／ストレージを別々にサイジング » 必要な資源のみ追加可能
- データ量やファイルサイズの事前設計は不要
- サーバだけの追加／切り離しが可能 » データのリバランス不要



P-DFS導入効果

■ 測定モデル

200GBのデータを既存システムから転送してHadoopで分析処理を実行し、
その結果を取り出す業務フローの処理時間を比較

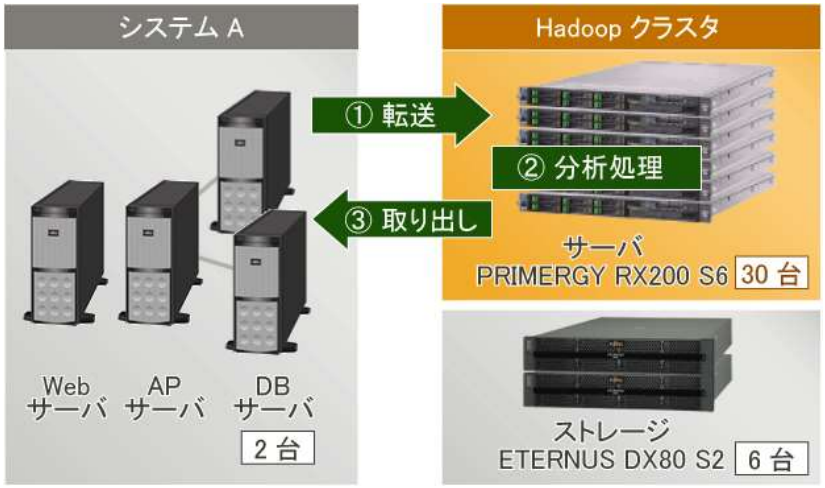
■ 測定環境

サーバ	PRIMERGY RX200 S6 30台	メモリ：48GB CPU：Xeon E5606（2.13GHz/4コア）
ストレージ	ETERNUS DX80 S2 6台	接続方式：iSCSI 転送速度：10Gb/s

■ 測定結果

	Hadoop（HDFS）	P-DFS for Hadoop
① 転送／ロード	70分	―
② 分析処理※	51分30秒	35分
③ 取り出し／転送	70分	―
合計処理時間	191分30秒	**35分**

※Hadoopでの分析処理時間はモデルにより異なります。



FUJITSU

2. Hadoopをもっと簡単・便利に活用する技術の紹介

- PDFS
- Hadoopマルチプレクサ

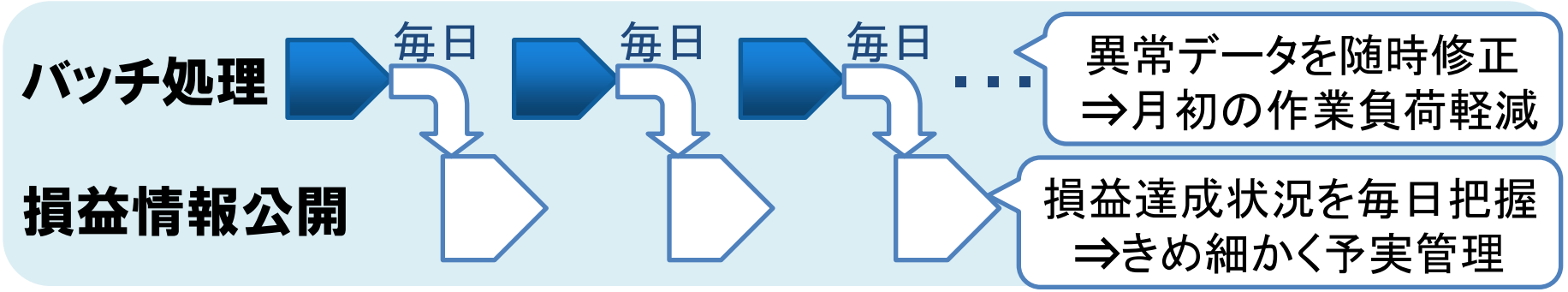
【製造】売上原価計算バッチの高速化

FUJITSU

デイリーな原価計算による損益管理の精度向上と見える化

■ 現状：月末締めで第4営業日に損益情報を公開 **1回/月**

■ 今後：毎日の原価計算で損益情報を公開 **1回/日**

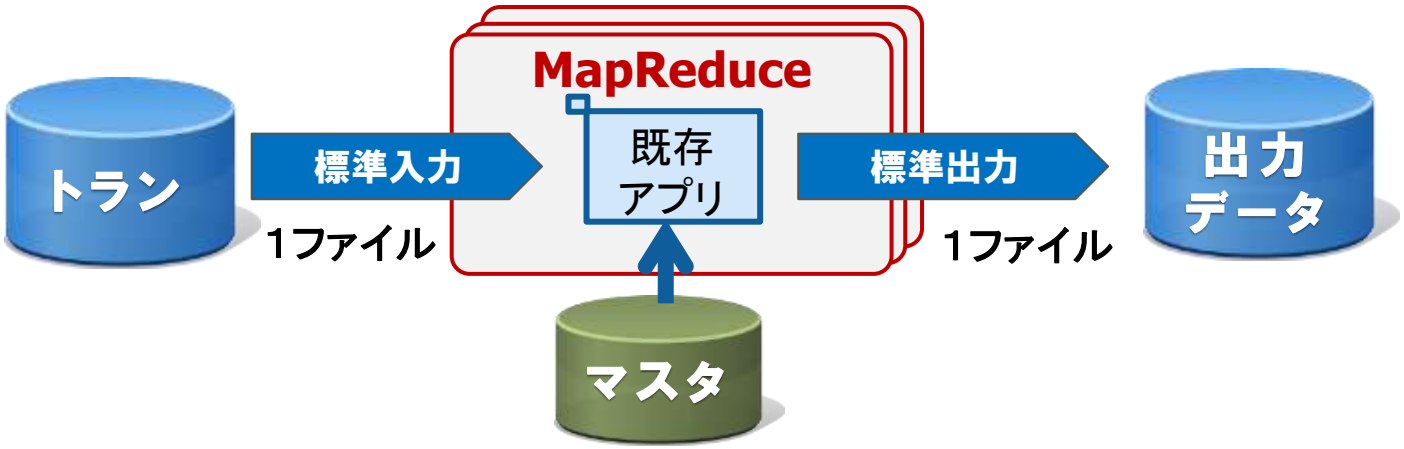


従来業務からの移行性が課題

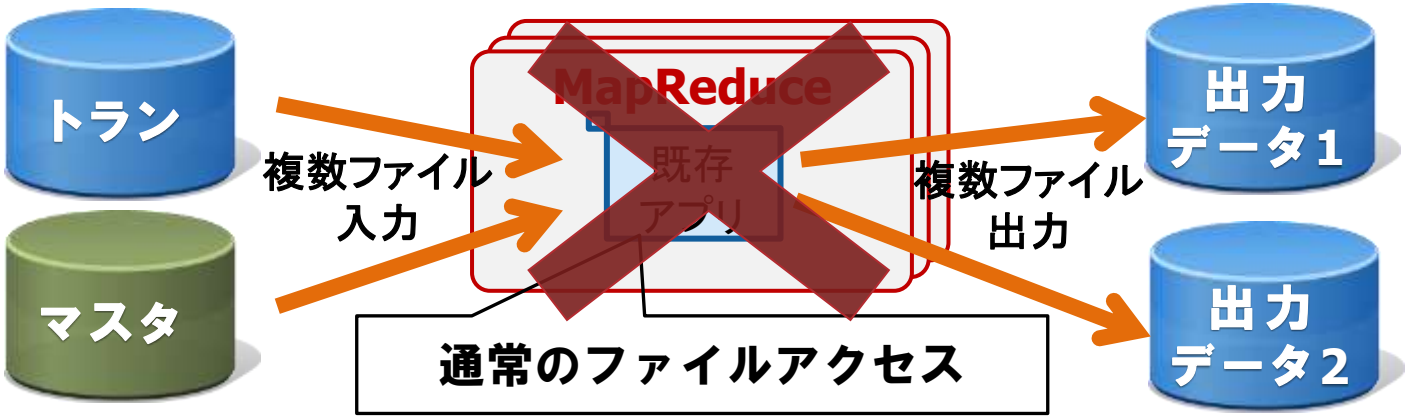
FUJITSU

突き合わせ処理など複数ファイルの入出力を行うアプリは Hadoopで実行できなかった

Hadoop Streamingでは、一つのファイルを標準入出力を通じて処理



既存のバッチアプリは複数の入出力ファイルをアクセス

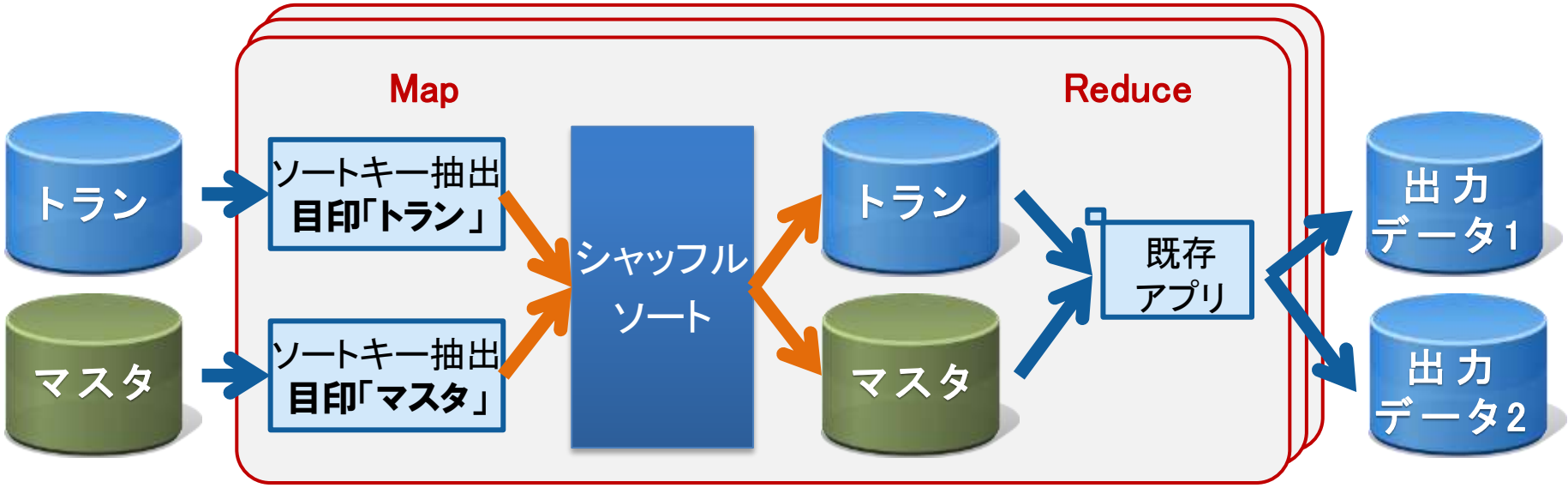


Hadoopマルチプレクサ技術

FUJITSU

Hadoopの高速並列ソートを活用しつつ 既存アプリの並列実行ができる

- 課題: **Hadoop**の基本原理の**MapReduce**は、1データ列に対する並列ソートであるため、複数ファイル処理には複雑な実装が必要
- 解決技術：入力ファイルごとに異なる目印をつけて**Hadoop**のソート機構に渡し、ソート後に元の入力ファイルフォーマットで復元。ファイルごとに位置の異なるソートキーの抽出もサポート



既存COBOLアプリケーションの活用

FUJITSU

+

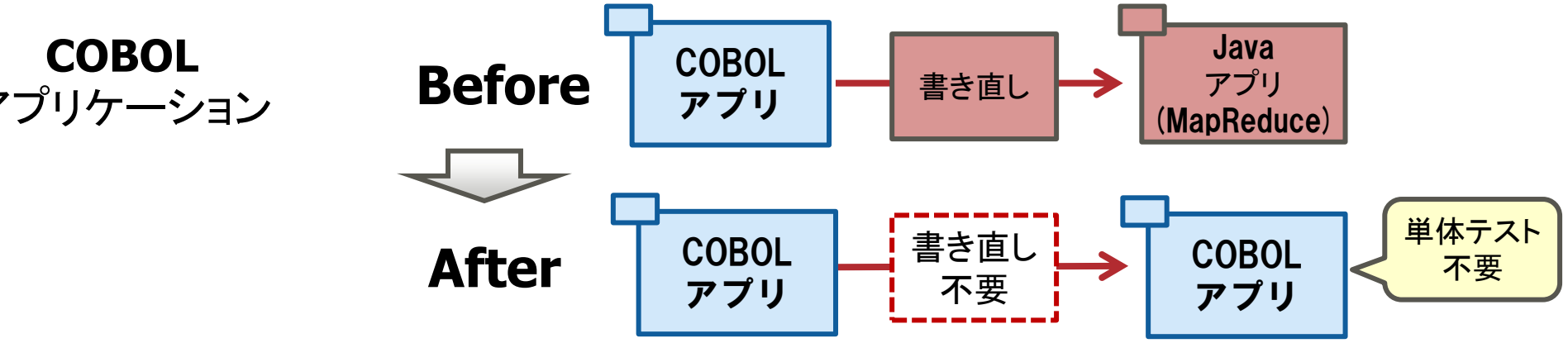
hadoop

従来業務からの移行性を確保

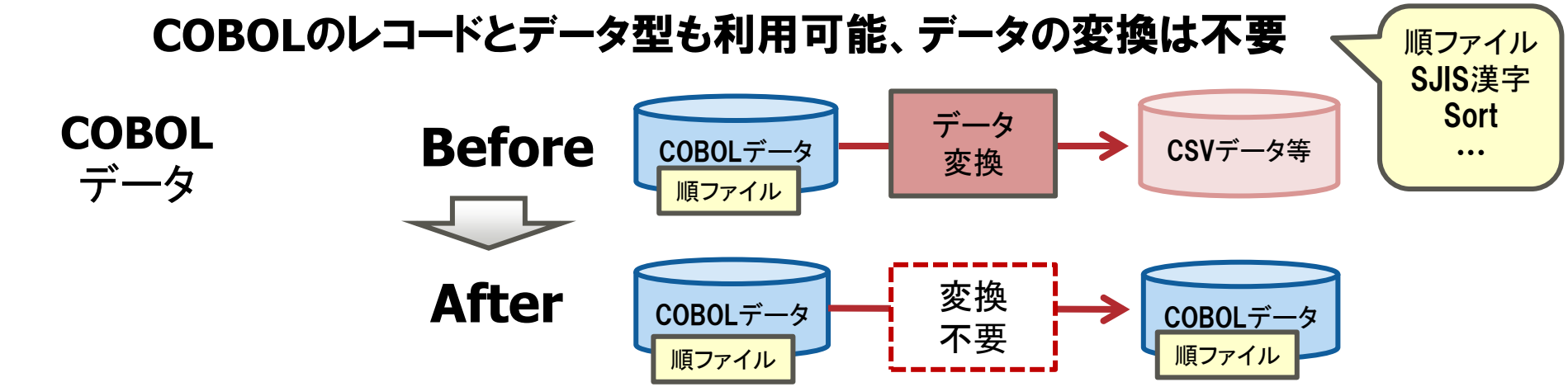
FUJITSU

既存のCOBOL資産を修正することなく、そのまま活用

Hadoop適用のため必要であった、JavaやMapReduceプログラミングは不要



COBOLのレコードとデータ型も利用可能、データの変換は不要

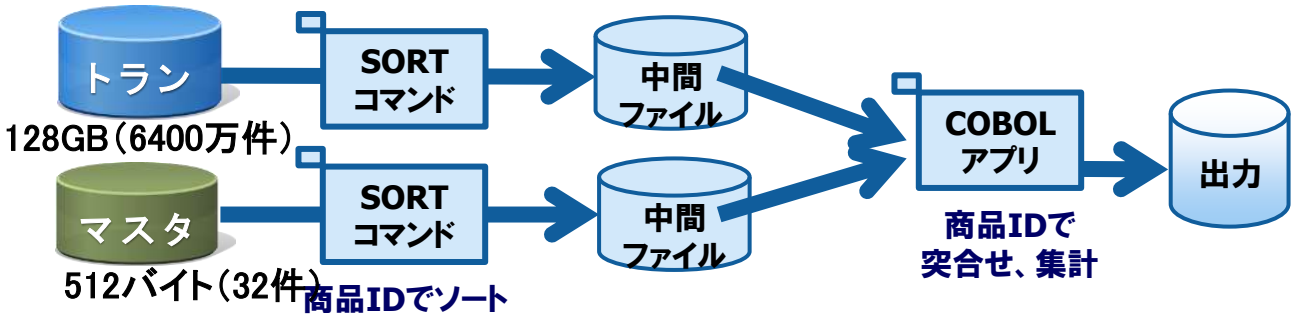


バッチ処理の並列処理による効果実測例

FUJITSU

■ 従来のバッチアプリケーション

例）トランザクションデータをマスタデータと突き合わせ集計する処理の場合



150分

■ Apache Hadoop+NetCOBOLで
16多重で並列処理

50分

■ Interstage Big Data Parallel Processing Server
+NetCOBOLで
16多重で並列処理

8分

2時間半がわずか8分。約18分の1 に短縮

FUJITSU

3. ビッグデータを活用する人工知能技術の紹介

- 富士通の人工知能技術 zinrai
- 機械学習自動化技術

富士通が目指すAIの方向性

FUJITSU

人と協調する、人を中心としたAI

継続的に成長するAI

AIを商品・サービスに組み込んで提供

Human Centric AI

富士通のAI技術のブランド

FUJITSU

Human Centric AI

ジンライ

Zinrai

- 語源
  疾風迅雷（すばやくはげしいこと。）
- 名前に込めた想い
  人の判断・行動を“スピーディ”にサポートすることで、
  企業・社会の変革を”ダイナミック”に実現させる。

富士通が保有するAI技術を体系化

センシング

人・企業・社会

アクチュエーション

Human Centric AI Zinrai

知覚・認識	知識化	判断・支援
画像処理	自然言語処理	推論、計画
音声処理	知識処理・発見	予測、最適化
感情・状況認識	パターン発見	対話、推薦

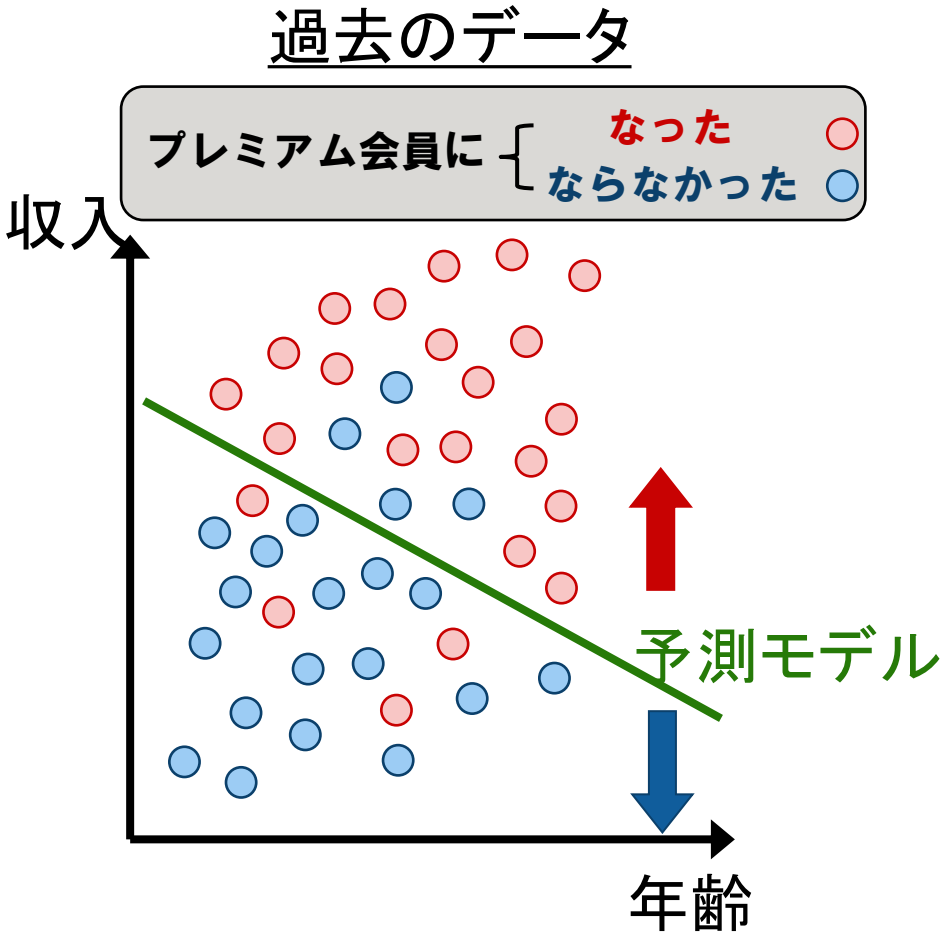
学習	Deep Learning	機械学習	強化学習
先端研究	脳科学	社会受容性	シミュレーション

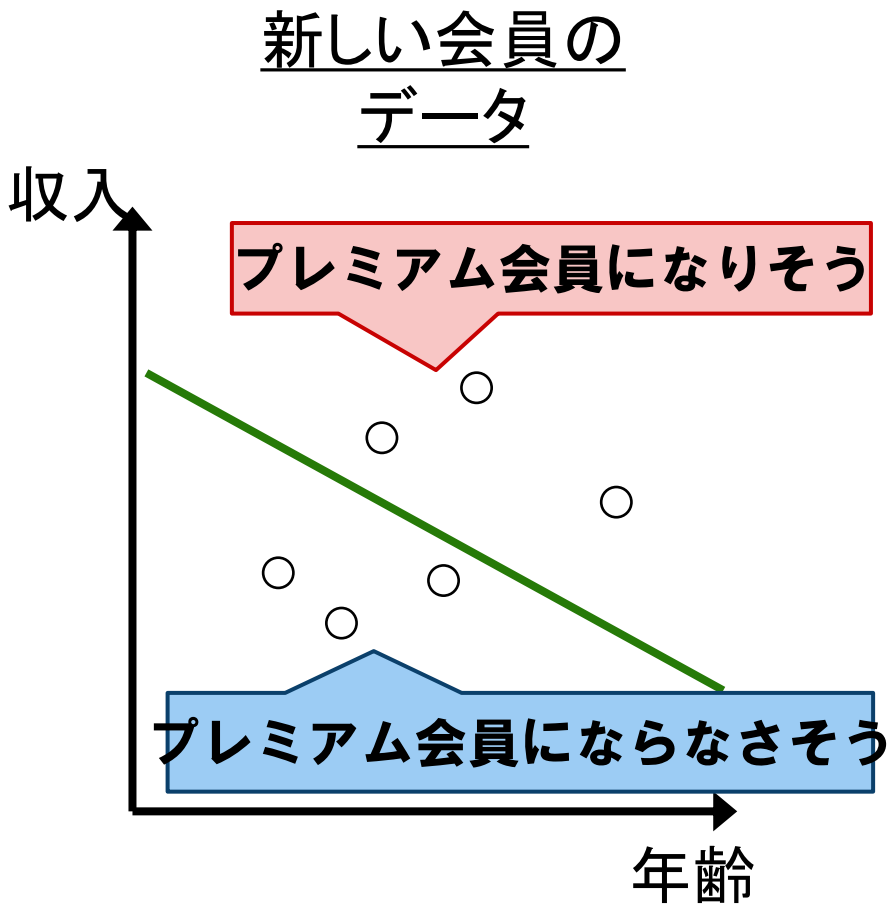
機械学習とは？

FUJITSU

機械学習とは？

過去のデータから隠れた法則性（予測モデル）を見つけだし
その法則で新しいデータの予測をする技術





機械学習アルゴリズムの選び方？

FUJITSU

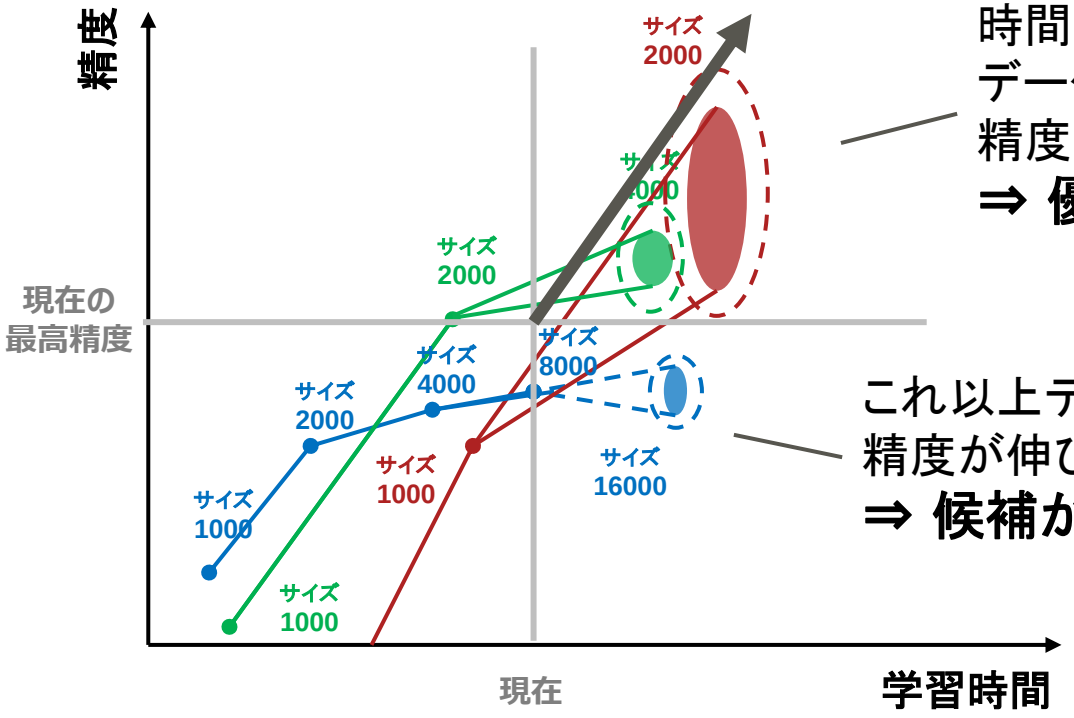
■アルゴリズムによって予測モデルの作り方が異なる

10種類以上も候補があるから

どれが良いかわからない！！！

全部のアルゴリズムを試すために一工夫しました

- データ量を少しずつ大きくしながら、各アルゴリズムの精度向上を見積もります
- 予測精度が上がる可能性が高く、短時間に実行が終わる候補を選定して優先実行します
- 見込みのない候補はすばやく除外します

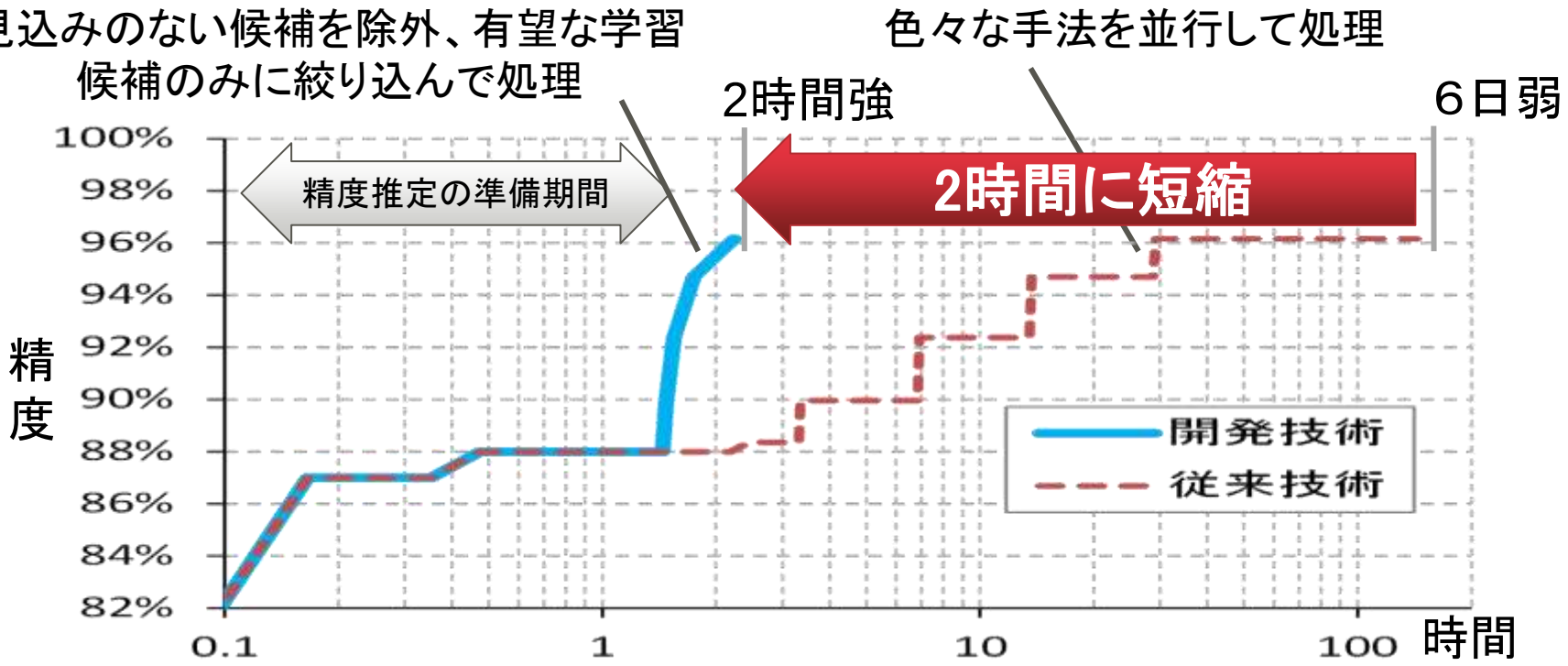


時間はかかるが
データを増やせばまだ
精度向上
⇒ **優先的に実行**

これ以上データを増やしても
精度が伸びない
⇒ **候補から除外**

ロジスティック回帰
SVM（RBF）
ランダムフォレスト

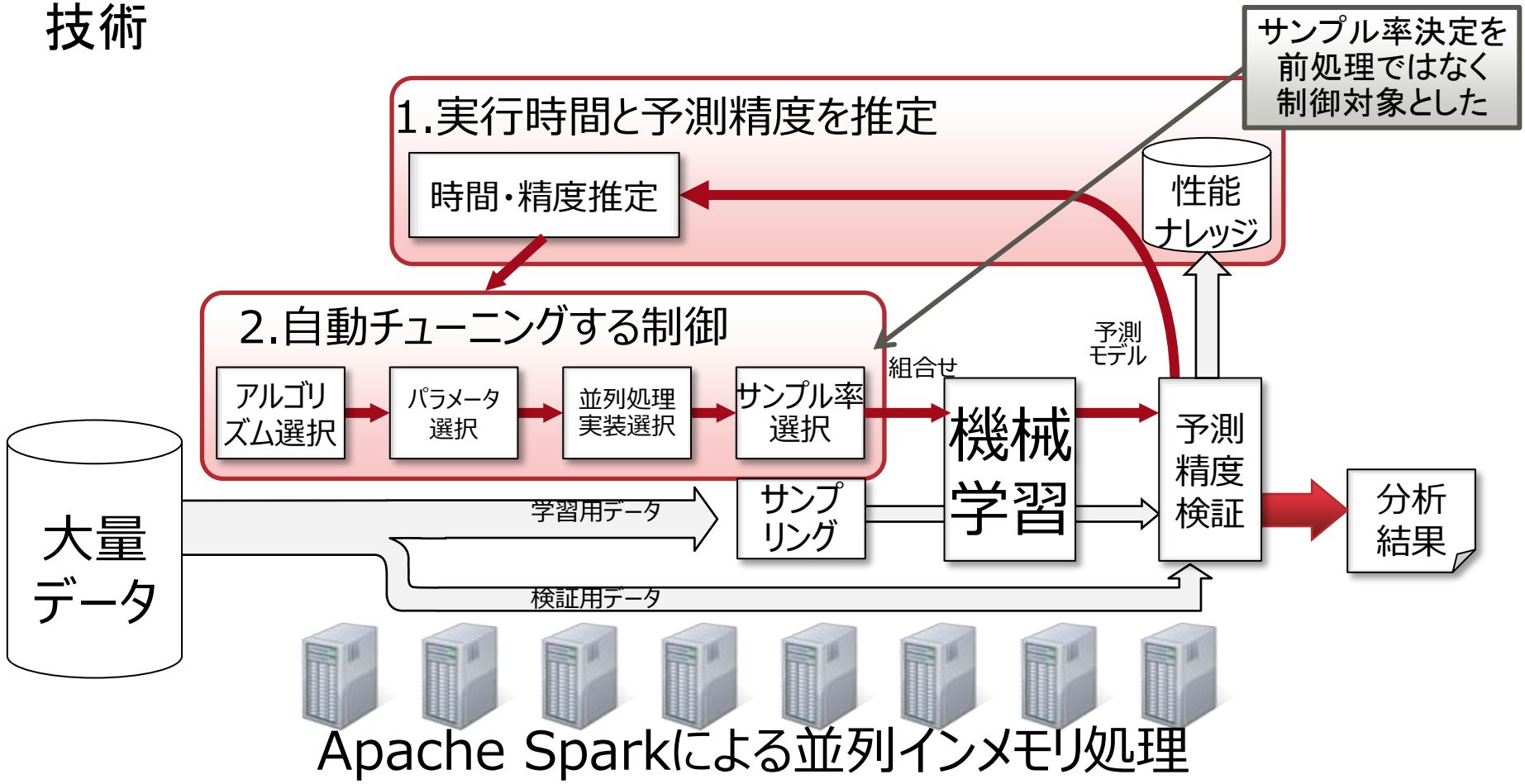
性能：網羅的処理だと6日⇒2時間で完了



アルゴリズム	データ量				…		
	10万	20万	40万	80万		2500万	5000万
Random Forest [並列バギング]	51秒 76%	52秒 80%	69秒 81%	60秒 84%		1760秒 96%	†4338秒 97%
Random Forest [Spark]	38秒 76%	49秒 76%	78秒 76%	114秒 76%		†1590秒 76%	†2695秒 76%
Gradient Boosting [並列バギング]	96秒 76%	97秒 78%	119秒 81%	113秒 83%		1420秒 88%	3679秒 88%
Gradient Boosting [Spark]	434秒 88%	475秒 88%	544秒 88%	691秒 88%		†5221秒 88%	†7933秒 88%
Support Vector Machine (RBF kernel) [並列バギング]	529秒 73%	609秒 73%	815秒 79%	1,348秒 81%		†1.3日	†約3日

こんな技術でできています

- サンプリングした小さなデータの学習履歴から、動的に実行時間と学習した予測モデルの精度を推定する技術
- 見込みのあるアルゴリズム・動作条件に素早く絞り込み実行する技術

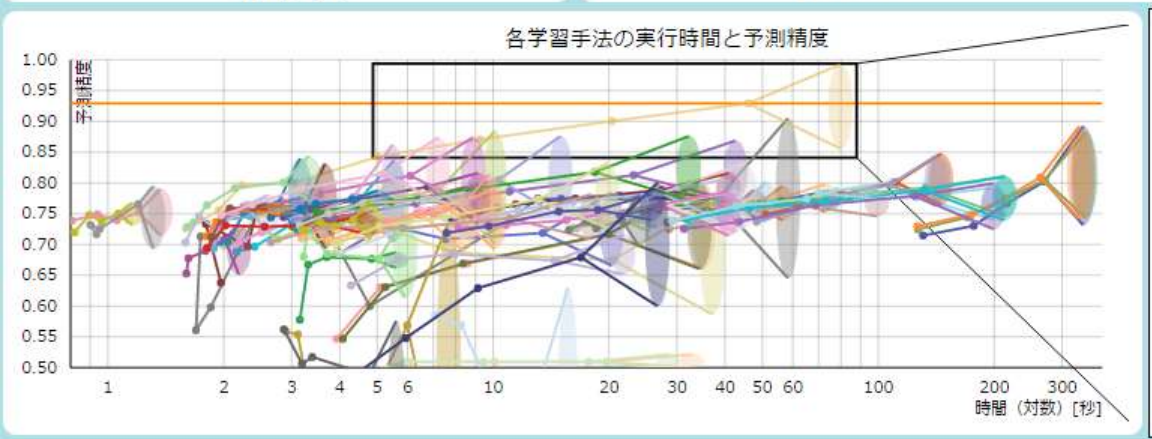


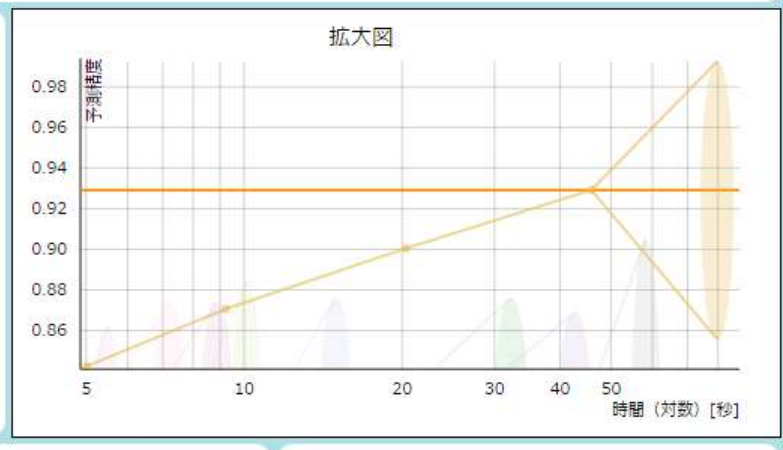
デモンストレーション

FUJITSU

wizz

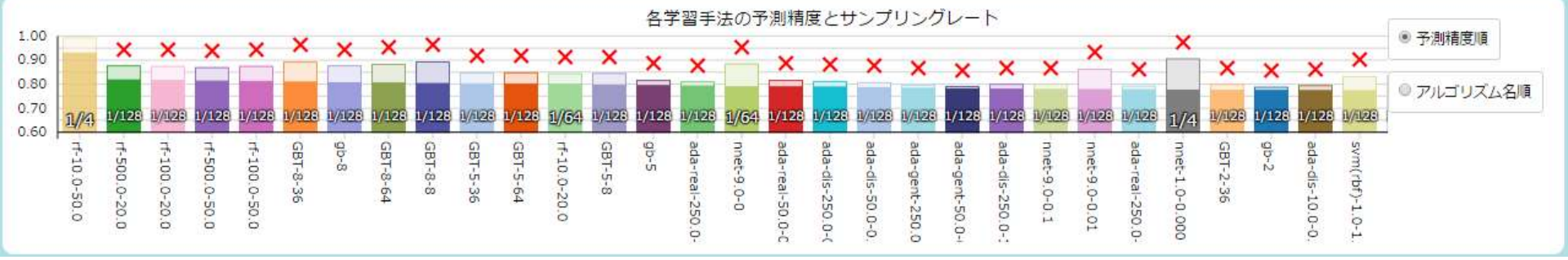
データ精言：560,000行、54属性
機械学習アルゴリズム：9、全候補モデル数：82





最高予測精度：0.9292
経過時間：7,487 秒

時間送り
コマ送り



FUJITSU

最後に

Hadoopはビッグデータ処理の基盤

FUJITSU

- 大量に蓄積されたデータを加工するときには欠かせない技術
  - ✓ 数十GBを超えるデータを扱うときには試してみましょう
  - ✓ MapReduceは基礎として知っておくべきですが、通常使うのはHiveなどがお勧めです
  - ✓ ランダムアクセスが必要なら HBaseやelastic searchなどを使いましょう
- 分析や他との連携をする際には他のOSSと連携しましょう
  - ✓ BIツールの多くはHiveのインターフェースを持っています
  - ✓ ログファイルなどの収集はfluentdなどが便利です
- 商用パッケージも成熟してきています
  - 富士通製品もご検討下さい

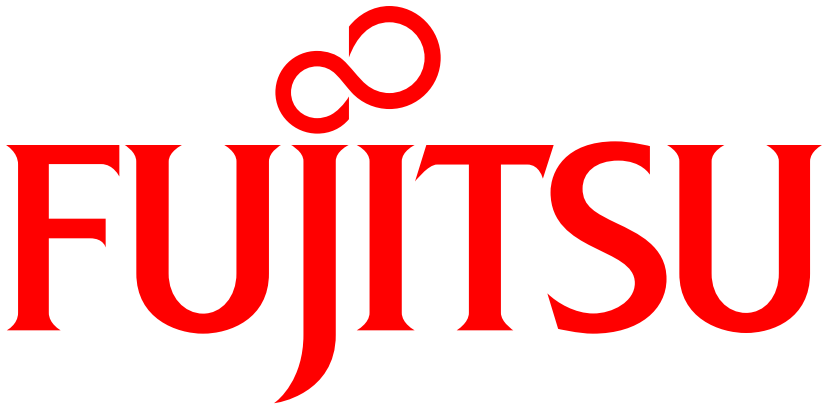
富士通研究所で一緒に働きませんか？

FUJITSU

- 熱い思いを持つ方大歓迎です
  - 自分が考えたアルゴリズムを極めたい！
  - ビッグデータを分析して社会に貢献したい！
  - 特許を出してお金持ちになりたい！
  - HadoopやSparkに貢献したい
- 大量のサーバ、GPU、そしてなにより一緒に議論する仲間がいます
- もちろん、富士通の福利厚生がもれなくついてきます 

いつでもコンタクトしてきてください

email: hal_ueda@jp.fujitsu.com
twitter: @halbon_ueda
facebook.com/haruyasu_ueda.1



shaping tomorrow with you