第一回 分散コンピューティング勉強会
Asakusa Frameworkの活用事例

2018年6月14日

株式会社　ノーチラス・テクノロジーズ
http://www.nautilus-technologies.com/
mailto:contact@nautilus-technologies.com
Tel: 03-6712-0636

NAUTILUS

Agenda

- 会社紹介
  - ノーチラス・テクノロジーズとは
- Asakusa Framewrokの活用例
  - 最近の活用例
  - 機械学習における活用事例
    - 販売予測での活用
    - 為替予測での活用　※非公開
- Asakusa Frameworkの紹介
  - Asakusa Frameworkでの開発時におけるメリット

会社紹介

ノーチラス・テクノロジーズ

ノーチラスをひとことでいうと

- 分散技術を業務システムに適用し、業務をよりよくすることを目指す会社です
- Hadoopを中心とした分散技術から取り組みを開始
  - Sparkやその他の新しい技術を取り込みながら発展
- その他、新しい分散技術を追いかけながら、業務システムに活かせるミドルウェアの開発を行っています
- すべてOSSとして公開しており、皆様にもご活用いただける形で公開しています

会社概要

- **株式会社ノーチラス・テクノロジーズ**
  - 住所　：東京都品川区北品川1-19-5コーストライン品川ビル
  - 代表　：代表取締役会長　神林　飛志
    代表取締役社長　目黒　雄一
  - 設立　：2011年10月3日
  - 資本金：45百万円

Asakusa Framework事業	Asakusa Frameworkの開発 Asakusa Frameworkを用いたシステム開発 Asakusa Framework保守サポート
ビッグデータ基盤事業	Hadoopディストリビューションの販売 Hadoopエコシステムのコンサルティング
流通BMS事業	消費財流通のEDI標準である流通BMS準拠の EDIソフトウェアパッケージの開発・販売・保守

*1 ：Cloudera社はISVパートナー契約

直近での活用例

Asakusa Frameworkを利用した活用事例

活用事例（1/2）

- **請求システムの前処理**
  - スマートメータからの使用量データと契約者データの加工処理
  - データ入力は30分毎。
  - M³BPで加工処理を1/10に高速化。

- **発電量予測**
  - スマートメータからの実績データを元に地域、電信柱単位等様々な単位での
    発電量を予測する

- **DWHの前処理**
  - BIからのSQL処理遅延
  - M³BPで1/3に高速化

- **別システムへの展開　M³BP**
  - 本番稼働済システムでの高速化,安定稼働実績を評価
    割賦計算システム=>請求システムへ適用拡大

活用事例（2/2）

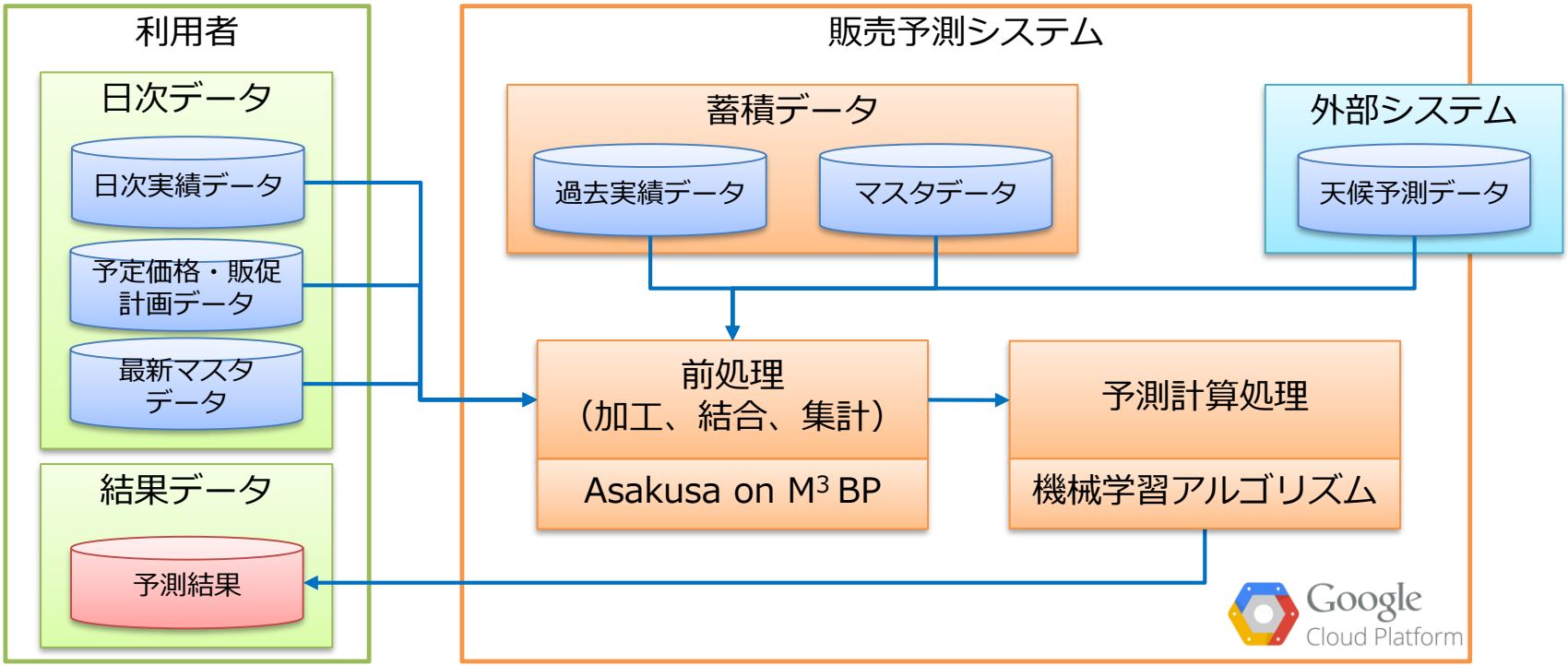
- **他行口座名寄せ**
  - 法人営業向け One to Oneマーケティング
  - 仕向情報と外部情報提供企業データを紐付けることにより法人営業向けターゲット情報を提供
- **DWHからの加工,集計,バッチ処理のオフロード**
  - データフローと分析処理の分離
  - データ増に伴う処理時間増に対しデータが揃うまでの時間を短縮し情報の鮮度アップを実現
- **MCIF分析基盤(Taradata)でのバッチ処理**
  - 処理遅延が顕在化しつつあったため高速化対策ソリューション候補としてバッチ処理のオフロードを検討
- **取引審査**
  - 取引審査の既存バッチ処理遅延とデータ増のよる処理の限界
  - HFTの売買審査開始による取込データ量の更なる増加(5倍)
  - 処理時間を1/180に短縮

機械学習のデータ処理

販売予測での活用

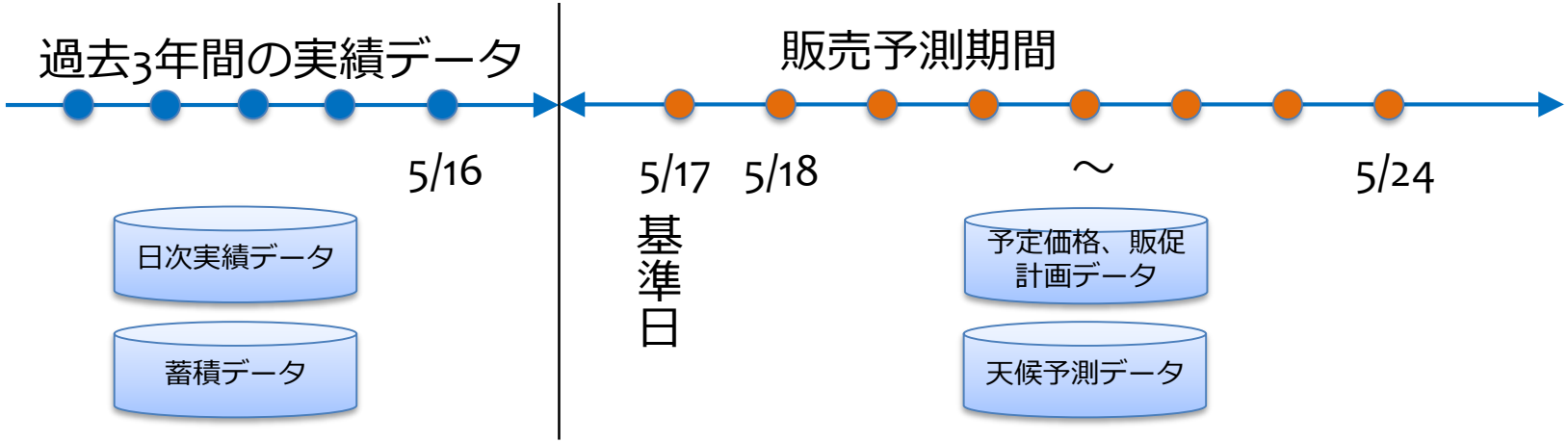
全体処理フロー

- 販売予測システム概要
  - 利用者は、日々の売上実績や商品・特売情報を店別に準備し、販売予測システムへアップロードする
  - 販売予測システムは前処理として日次データと蓄積データ、外部システムから連携した天候予測データを結合して予測に必要なデータを準備する
  - 複数のアルゴリズムによる機械学習処理を行い、より最適となる未来の販売数を予測する
  - 利用者は、未来の販売予測数をダウンロードする



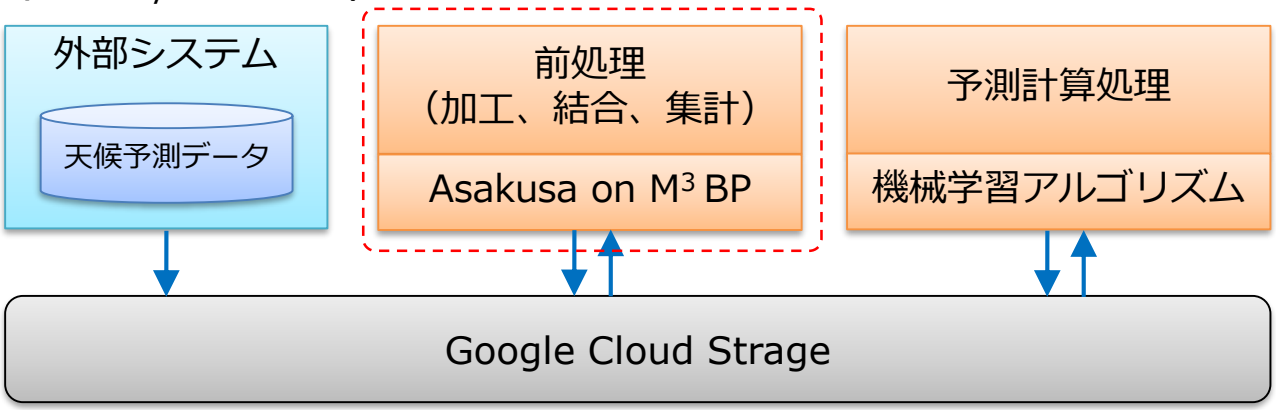
運用の流れ

- 日々の運用は次のように行われる
  - 基準日前日（例:5/16）
    - 利用者は以下をシステムにアップロードする
      - 基準日前々日までの日次実績データ
      - 基準日から最大7日後の予測対象期間（5/17～5/24まで）の予定価格、販促計画データ
    - 販売予測システムは前処理・予測計算処理を行い、販売予測期間の販売予測数を算出する
  - 基準日（例:5/17）
    - 利用者は予測結果を取得し、前日と同様に日次データ・予定価格、販促計画データをアップロード



前処理について

- 前処理の概要
  - 以下のような処理を行う
    - 日次実績データと過去実績データをマージ
    - 最新マスタの取込と実績データへの結合
    - 後継商品へ実績の紐付け
    - 予測対象店舗、商品の絞り込み
    - 日別に店舗・商品毎に集計
    - 天気予報データの表記揺れを訂正
  - Asakusa on M^3 BPで動作
    - Google Cloud Platform上の１インスタンスのみ
    - 天候データ及び機械学習処理とのデータ連携でGoogle Cloud Storageと直接アクセス（DirectI/Oで入出力）する



予測計算処理について

- 予測計算処理の概要
  - 予測結果
    - 営業日 + 店舗 + 商品単位の予測売上点数をCSVで出力
    - データの内容にエラーがあった場合は別ファイルで原因を含めて出力
  - 精度評価の実施
    - 販売予測計算は複数のアルゴリズムを用いて計算しており、それぞれのアルゴリズムで専用のインスタンス上で並行で処理を行う
    - 予測結果と実績の突き合わせを行い精度評価を行う
    - 商品別に予測精度の高いアルゴリズムを選択して予測結果を出力する

Asakusa Framework導入メリット

- Asakusa Frameworkを採用することによるメリット
  - 高性能
    - インメモリ処理
    - CPUコア毎に並列実行する
    - 前処理は過去データのジャーナルを扱うため、データ量は比較的大きくなるが、数GB程度のデータ量であればM^3 BP 1 ノードで十分な性能が出せる
  - 拡張性
    - データ規模に応じてスケールアップで対応可能。ただし、メモリに乗り切らないほどデータが増加した場合はHadoop（Asakusa on Spark）へ移行できる
  - 保守性
    - 多種のデータモデルを扱う複雑な処理であるため、SQL等ではメンテナンス性が低くなりがち（改修コスト高）
    - Cloud Storageに蓄積したデータを直接扱うことができるため、Asakusaのみで加工・蓄積ができる（DBは不要）

機械学習のデータ処理

為替予測での活用

※本セクションは非公開です

NAUTILUS

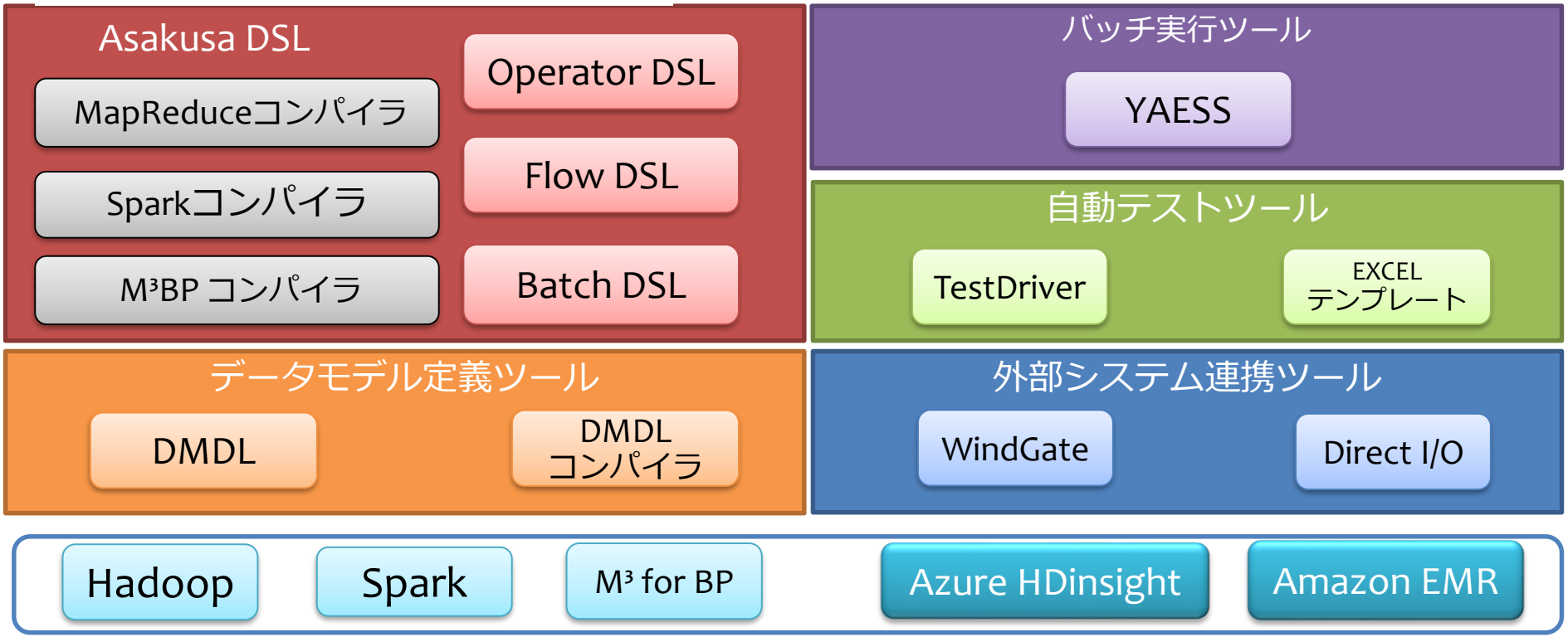
Hadoop用バッチ処理開発・実行・運用
フレームワーク

Asakusa Frameworkの紹介

Asakusa Frameworkとは

- **分散環境用高速バッチ開発フレームワーク**
  - Asakusa Frameworkは業務システムのバッチ処理に、分散並列処理の能力を活用する開発用フレームワーク
  - Hadoop/Spark/HPC等の分散処理に対応し、分散環境向けアプリケーション開発に必要な開発環境・実行環境・運用環境を用意

Asakusa Frameworkコンポーネント



Asakusa Frameworkとは

- **並列／分散環境向けバッチ開発フレームワーク**
  - Asakusa Frameworkは、基幹業務システムのバッチの高速処理を目的とし、複数の並列／分散基盤に対応した業界唯一のフレームワーク
  - 基幹バッチ開発に求められるテスト・デプロイをサポート

3つの特徴

開発容易性

- 特定のAPIに依存せず、学習が容易
- ローカル環境でのテストが可能
- 開発時チェックの充実

ポータビリティ

- 一つのソースから複数環境向けの実行コードを生成
- バージョンアップ時の互換性を重視
- オンプレ/クラウド対応

低学習コスト

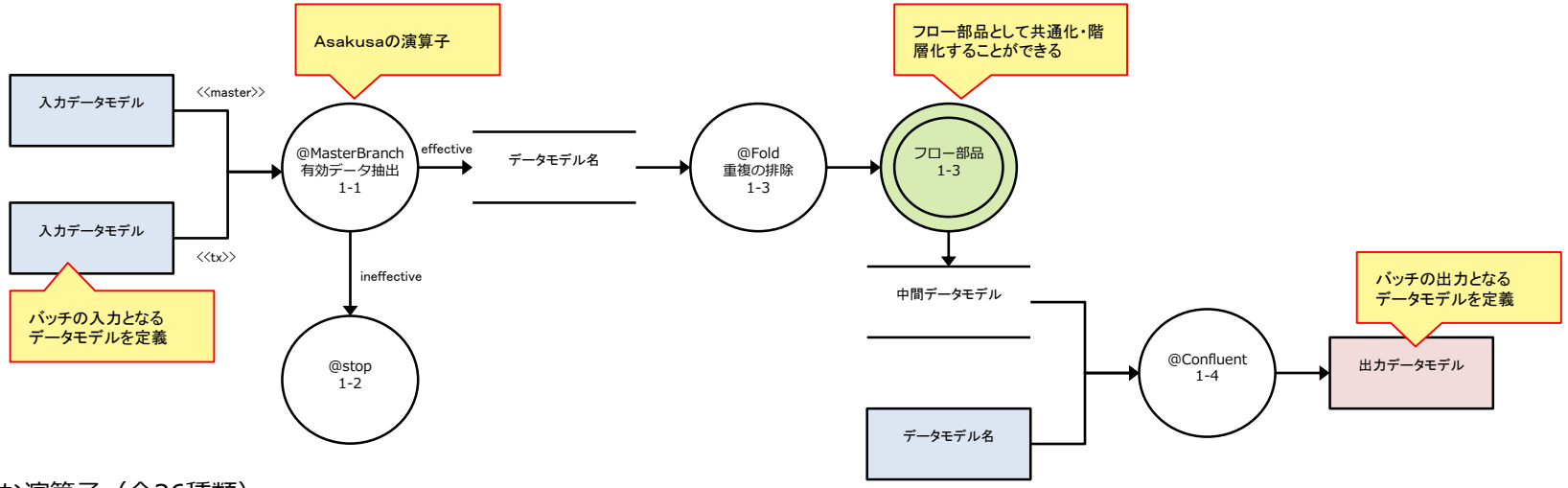
- 2日間の講習を受講すれば2週間程度でプログラムが書けるレベル

AsakusaFrameworkの開発容易性

- 1.特に設計にウェイトをおいている
  - 「設計という資産の維持」を目的としている
    - 「実装は陳腐化するが、設計は残る」という考え方
    - バッチ処理設計から素直な実装が可能
- 2.分散処理特有のスキルは不要
  - MapReduceやSparkの内部構造を意識する必要がない
    - MapReduceやSparkのAPIは隠蔽しているので、Javaの知識があれば開発が可能
    - Asakusaのコンパイラが最適化を行い高いパフォーマンスを実現する

Asakusa Frameworkの開発効率

- Asakusa DSLの設計と開発
  - バッチ処理をデータフロー形式で記述する
  - 入力モデル、中間モデル、出力モデルとその処理内容を記述したデータフロー設計よりそのまま実装が可能
  - 設計において、MapReduceやSparkを意識する必要はない



主な演算子（全26種類）

分岐 Branch	レコードを内容に応じた出力に振分	単純集計 Summarize	Keyでグループ化して集計
マスタ分岐 MasterBranch	レコードとマスタデータの内容に応じて振分	畳み込み Fold	Keyでグループ化して畳み込んで集計
マスタ付き更新 MasterJoinUpdate	マスターデータをKeyで結合して更新	更新 Update	レコード内容の更新
合流 confluent	複数の入力を合流して出力	変換 Convert	モデルを変換

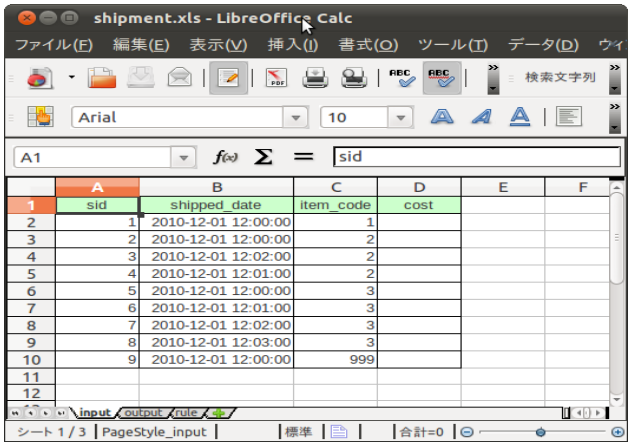
AsakusaFrameworkテスト容易性

- Hadoopでの大規模な開発の為の品質確保が可能
  - **OperaterDSL(演算子)でのテスト**
    - 演算子に対して通常の Java クラスと同様に、Junitでテスト可能
  - **データフロー、バッチのテスト**
    - Hadoop や外部入出力と自動的に連携したテストが可能
    - 一連の処理を自動的に行う**『テストドライバ』**を提供
    - テストデータのひな形を Excel ファイル・JSONで定義

テストドライバの処理

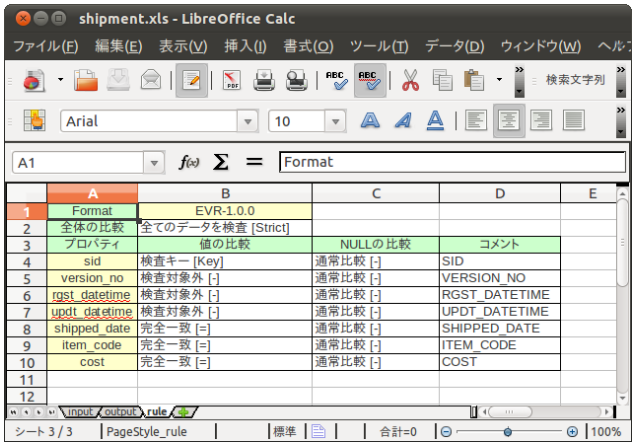
1. 入力データを初期化する
2. 入力データを流し込む
3. 対象のプログラムをテスト実行する
4. 出力結果を取り込む
5. 出力結果と期待データを検証する

入力データ・期待データシート



	A	B	C	D	E	F
1	sid	shipped_date	item_code	cost		
2	1	2010-12-01 12:00:00	1			
3	2	2010-12-01 12:00:00	2			
4	3	2010-12-01 12:02:00	2			
5	4	2010-12-01 12:01:00	2			
6	5	2010-12-01 12:00:00	3			
7	6	2010-12-01 12:01:00	3			
8	7	2010-12-01 12:02:00	3			
9	8	2010-12-01 12:03:00	3			
10	9	2010-12-01 12:00:00	999			
11						
12						

比較条件シート



	A	B	C	D	E
1	Format	EVR-1.0.0			
2	全体の比較	全てのデータを検査 [Strict]			
3	プロパティ	値の比較	NULLの比較	コメント	
4	sid	検査キー [Key]	通常比較 [-]	SID	
5	version_no	検査対象外 [-]	通常比較 [-]	VERSION_NO	
6	rgst_datetime	検査対象外 [-]	通常比較 [-]	RGST_DATETIME	
7	updt_datetime	検査対象外 [-]	通常比較 [-]	UPDT_DATETIME	
8	shipped_date	完全一致 [=]	通常比較 [-]	SHIPPED_DATE	
9	item_code	完全一致 [=]	通常比較 [-]	ITEM_CODE	
10	cost	完全一致 [=]	通常比較 [-]	COST	
11					
12					

Asakusa Frameworkの投資可搬性(ポータビリティ)

- Spark1.xから2へバージョンアップを実施する場合
- Hadoop <-> Sparkへ基盤を変更する場合

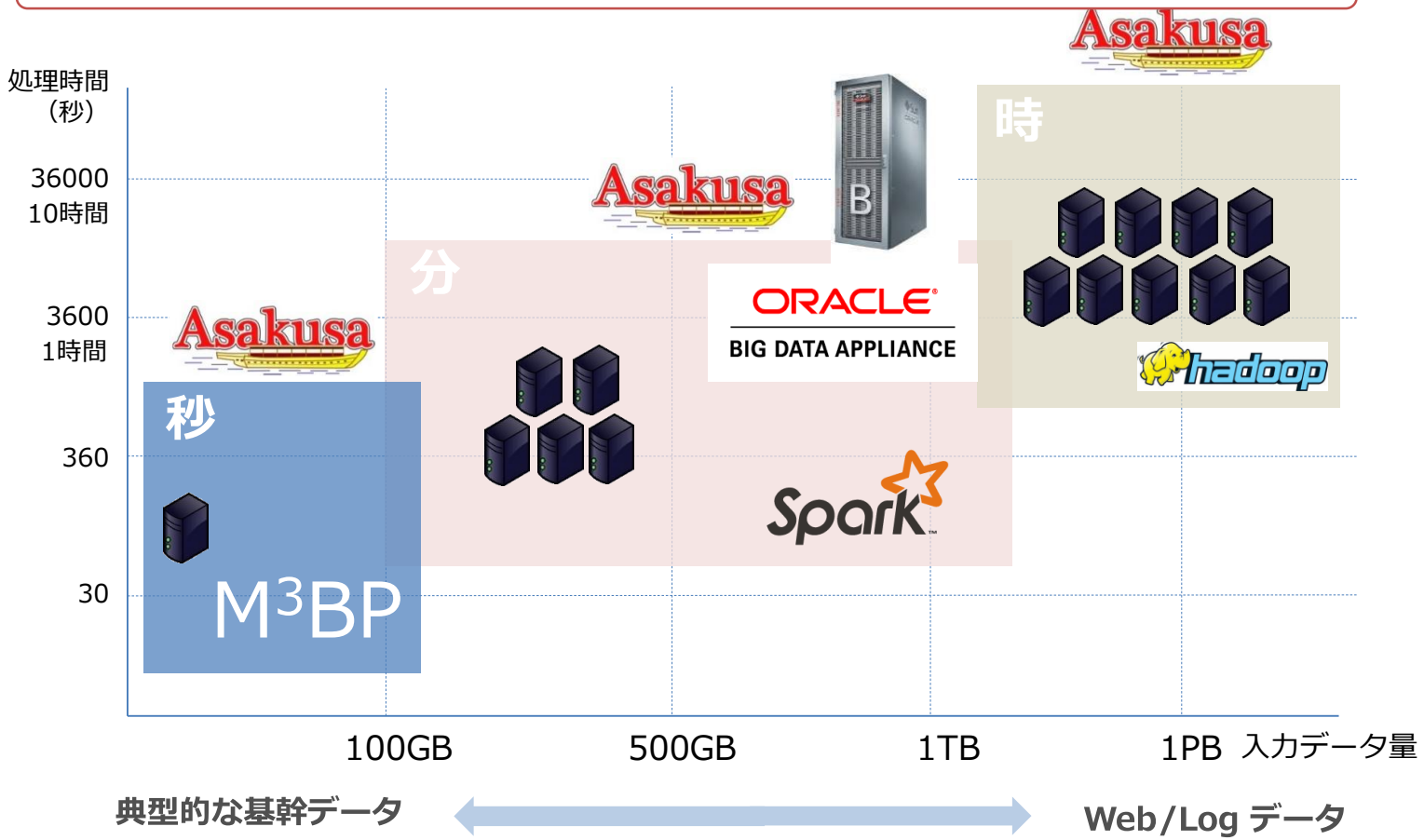
=> プログラムのソースコード変更が必要

- **AsakusaFrameworkでは投資可搬性を確保**
  - 後方互換性100%が基本ポリシー
  - Asakusaで開発したプログラムが可能な限り使い続けることができるようにする
    - 分散環境は今後の変化・進化していくので、ユーザは追随して行く必要があるがその都度開発するのは投資回収が困難
    - Asakusa開発資産は「常に最新の分散環境を利用できる」(原則3か月毎のリリース)

      AsakusaFrameworkがHadoop/Sparkのバージョン差異を吸収
    - **実行環境がMapReduceからSparkに変わってもAsakusaはソース変更の必要なし**

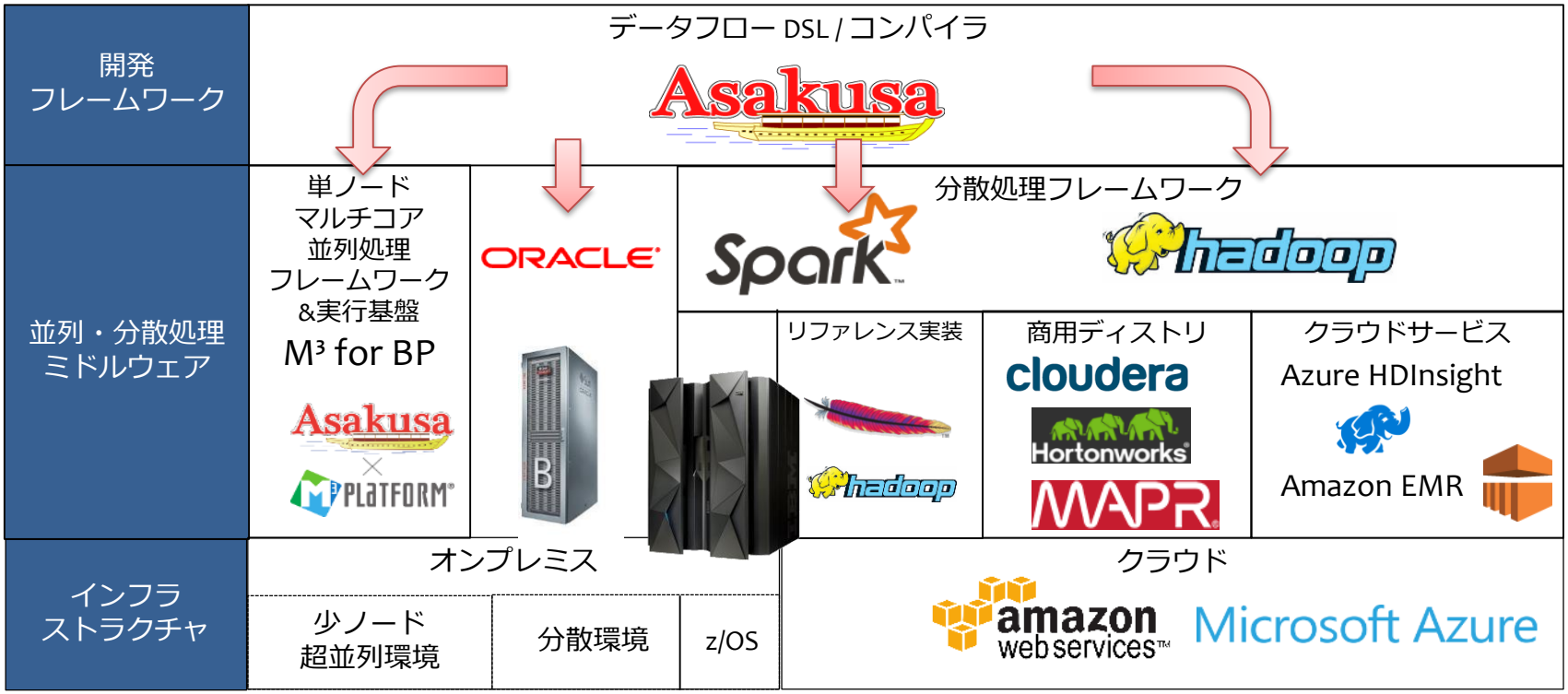
Asakusa Frameworkの投資可搬性

ユースケースに応じて、最適な実行環境で動作させることが可能



Asakusa Frameworkの投資可搬性（ポータビリティ）

様々な分散・並列処理基盤をAsakusa Frameworkにより選択可能に



最後に

- 今日は一部の導入事例を紹介しましたが、採用事例が増えてきています
  - 流通業、金融業、電力系企業、サービス系企業 etc
- 利用に関すること、事例紹介などご要望がありましたらご連絡ください
- お問い合わせ先
  - 株式会社ノーチラス・テクノロジーズ
    - Tel: 03-6712-0636
    - Mail : contact@nautilus-technologies.com