ノーチラス・テクノロジーズ
会社概要

NAUTILUS

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

会社紹介

会社概要

- 株式会社ノーチラス・テクノロジーズ
  - 住所　：〒107-0051 東京都港区元赤坂1-5-12 住友不動産元赤坂ビル7F
  - 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
  - 事業内容：
    - SI事業
      - エンタープライズシステムにおけるシステム構築支援
        - システム導入検討から適用技術の選定・PoCの実施
        - 要件定義から開発までの一連の請負開発
        - 運用設計から本番の保守までの保守サービス提供
      - 分散処理のミドルウェア提供
        - Asakusa Framework/Asakusa on M³BPの開発
        - Asakusa Framework/Asakusa on M³BPを用いたシステム開発
        - Asakusa Framework /Asakusa on M³BPの保守サポート
        - 製品の代理販売（MapR/Databricksなど）
    - 研究開発事業
      - 分散処理に関連する新技術を用いた製品開発
  - 関連会社
    - オーシャンブリッジ
  - 加盟団体
    - OSSコンソーシアム、日本情報システムユーザ協会
  - パートナー認定
    - Databricks、AWS ISVパートナー

現在の大きな環境変化にどう対応するか？

- 外部環境の変化
  - コロナ対応
  - DX化
  - EC/モバイル強化
  - 物流連携
  - 個人情報対応
  - 5G対応
  - IoT導入
  - AI/機械学習活用
  - セキュリティ

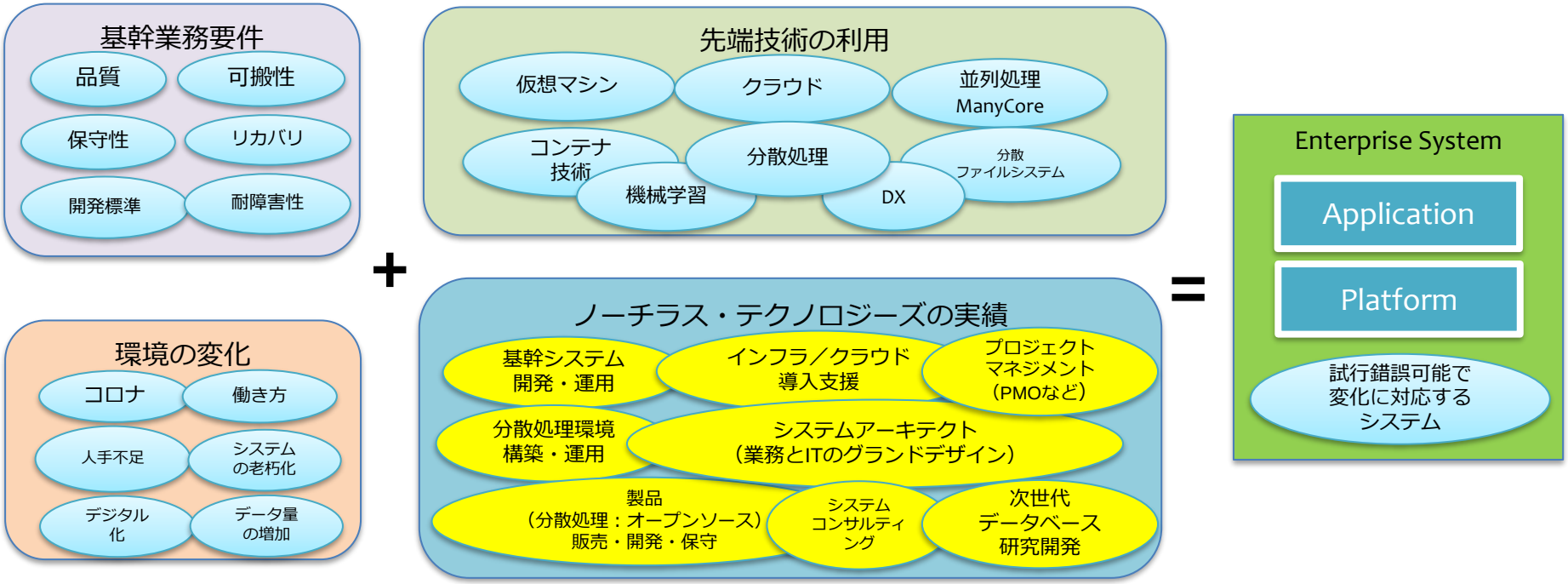
- 正解がない？
  - 顧客サービス強化？
  - 物流 x IT？
  - ECサイトの再構築？
  - マスター連携？
  - ゼロトラスト？
  - N/W環境への対応？

- 正解はだれもわからない
  - 試行錯誤の回数「だけ」が生存確率を上げる

- 何が足らないか？
  - 人がいない
    - 相談相手がいない
    - 中に人がいない
  - 技術がない
    - 最新技術？
    - 本当の情報はインターネットには書かれない
  - 試したいアイデアはあるがどうしてよいかわからない

「変化の大きい環境を下支えするプラットフォーム」を作る

- エンタープライズ（業務）システムを新しい環境に合わせて進化させる
  - 状況に応じて変化を受け止めるような試行錯誤できるプラットフォームやシステムが必要になる
  - 当社は、新しい技術とこれまでのエンタープライズシステムの実績から、時代の変化に合わせたシステムの提供と、人員のサポートを行うことができます。



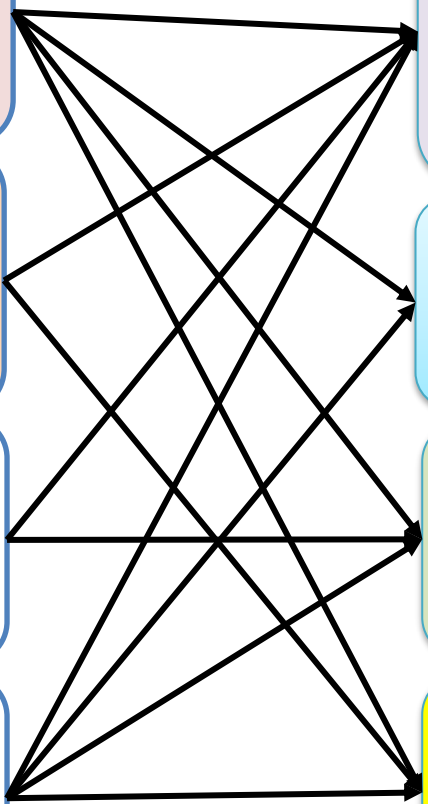
ノーチラス・テクノロジーズができること

DXを推進したいという話があり、業務システムを見直したいがどこから手をつけていいかわからない

実現したいことはあるのだが、どういう技術がよさそうで、具体的にどう進めればよいかわからない

これまでのシステムをオンプレミスからクラウドにのせかえたい。どうすれば実現できるかわからない

業務システムの夜間処理が長く、朝からの業務がすぐに始められない。業務に影響のでないシステムにしたい



コンサルティングサービス / システムアーキテクト
- お客様での業務要件を調査・ヒアリングし、ITに落とし込んだ場合の解決策やシステム化の方向性などを提案します
- システムの要件定義や設計や、導入計画の立案等も実施します
- お客様の課題に摘要できる技術を検討し提案します。必要に応じてPoCを実施し適用の可否も判定します

システム開発
- 要件定義から始まるシステム開発を行います。スクラッチでの業務システム開発経験も豊富です。
- 実績：管理会計システム、原価計算システム、バッチアプリケーション開発、業務システムの画面開発

インフラ環境構築
- 要件に合わせて環境設計やその後の構築、クラウドへのシステム移行など、インフラ環境に関する対応を実施します
- 実績：業務システムのクラウド（AWS/Azure）環境構築、オンプレミスからクラウドへの移行、など

ノーチラス・テクノロジーズの得意分野
- 分散処理を用いたデータ処理高速化（バッチ処理・前処理の高速化）
- 分散処理を組み込んだ業務システムの開発
- クラウドを利用した環境の設計・構築・運用

ノーチラス・テクノロジーズが提供するプロダクト

- エンタープライズ（業務系）システムに分散処理を適用して、データ処理の高速化・効率化を目指す製品の研究・開発を実施

Asakusa Framework

- 分散処理のアプリケーション開発と運用をサポートするフレームワーク
- 分散処理エンジンのM³BPとHadoop/Sparkにも一部対応している
- 開発や運用をサポートしている

Asakusa on M³BP

- 単ノードのメニーコアで並列処理を行うエンジン
- 他の分散処理プラットフォームより、高速に処理を行い、運用負荷も軽い
- Asakusa Frameworkでアプリケーション開発を実施可能
- 開発・運用のサポートを提供

劔

- 現在開発中の次世代型データベース
- 開発メンバーすべて日本企業及び団体（大学等）で国産データベースとなっている
- NEDOの研究委託事業として採択されており、一部支援をいただきながら開発中
- 既存RDBよりもかなりのパフォーマンスの向上が見込まれる

すべてをOSSとして公開

- 多くの方に、まずは使ってみていただくことで効果を試して欲しい
- 今後の成長し更新し続けるシステムに対する取り組み方を検討する一助にしてほしい

Asakusa Framework / Asakusa on M³BPの紹介

業務バッチ処理の課題

従来のバッチ処理の問題

DBサーバ頼み

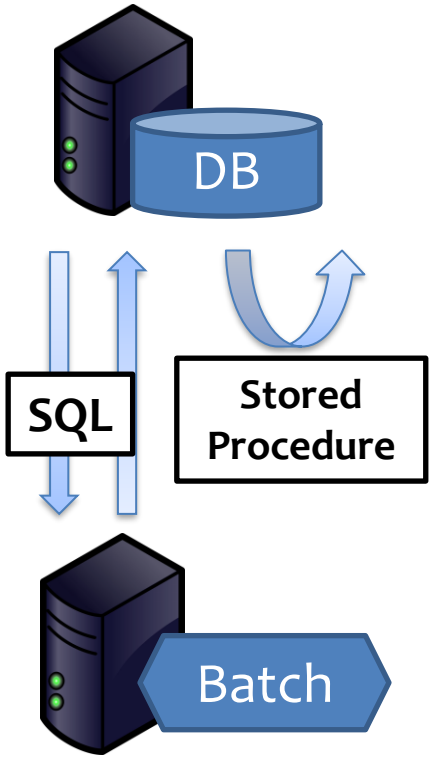
データ入出力SQLやストアドの処理のために、DBサーバ／ストレージにも多額の投資が必要

処理がスケールしない

処理の設計段階から並行処理を意識しなければ、サーバ性能も生かせず、後から性能をあげることも困難

職人芸

バッチ処理の方法論がなく、個社／個別バッチで作りが異なり、保守もバラバラ



基幹業務の複雑化
処理量の増大に
耐え切れない

Asakusa Frameworkがバッチ処理の問題を解決

DBサーバ頼み

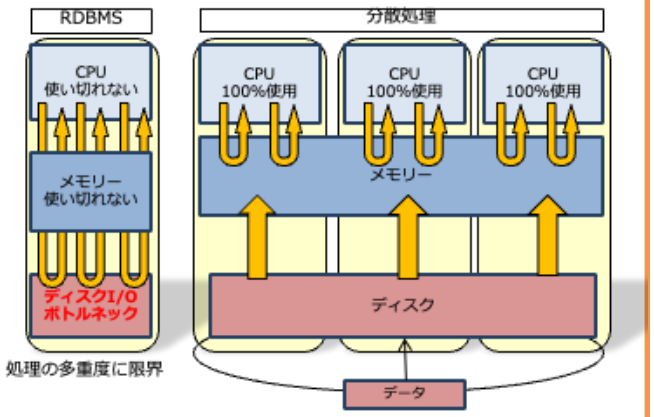
処理がスケールしない

分散処理で高速化を実現

複数のコアやノードで処理を分散するので圧倒的に高速化

～数時間を数分に～

M³BPは、単ノード複数コアで高速化を実現可能

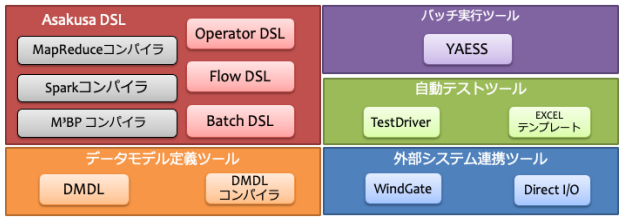


職人芸

開発しやすく保守しやすい

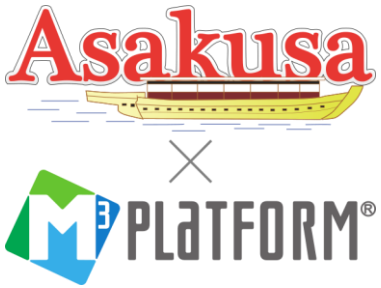
難しい分散アプリケーションの開発をJavaエンジニアなら誰でもできる

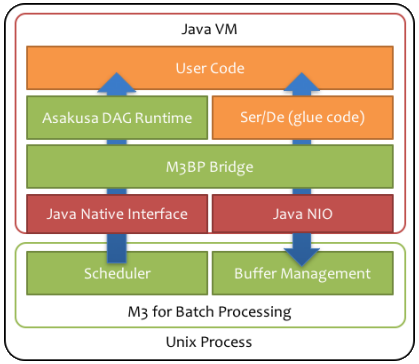
運用をサポートするツールも備え、基幹業務でも利用可能



Asakusa on M³BP

- 分散処理エンジン（M³ for Batch Processing）をFixStars社と共同開発
  - マルチコア環境でDAG形式で並列処理をするインメモリエンジンを提供
    - 小規模データでの複雑な処理を、単一ノード上のマルチコア用に最適化
  - C++でネイティブアプリケーションで動作させて高速化
  - M³ for BPの実行環境には、Hadoop/Sparkは不要
    - クラスタの構築・運用する場合の課題が解消され、高い費用対効果を生む
- 単一ノード・マルチコア環境用のAsakusa on M³BP
  - Asakusa on M³BPにてCPUコアの制御やメモリ管理はC++でM³ for BPで処理し、アプリケーション部分はJVMに振分け高速化
  - Asakusa DSLには一切変更を加えることなくM³ for BPと、Hadoop/Sparkの処理基盤を切り替えて使うことが可能

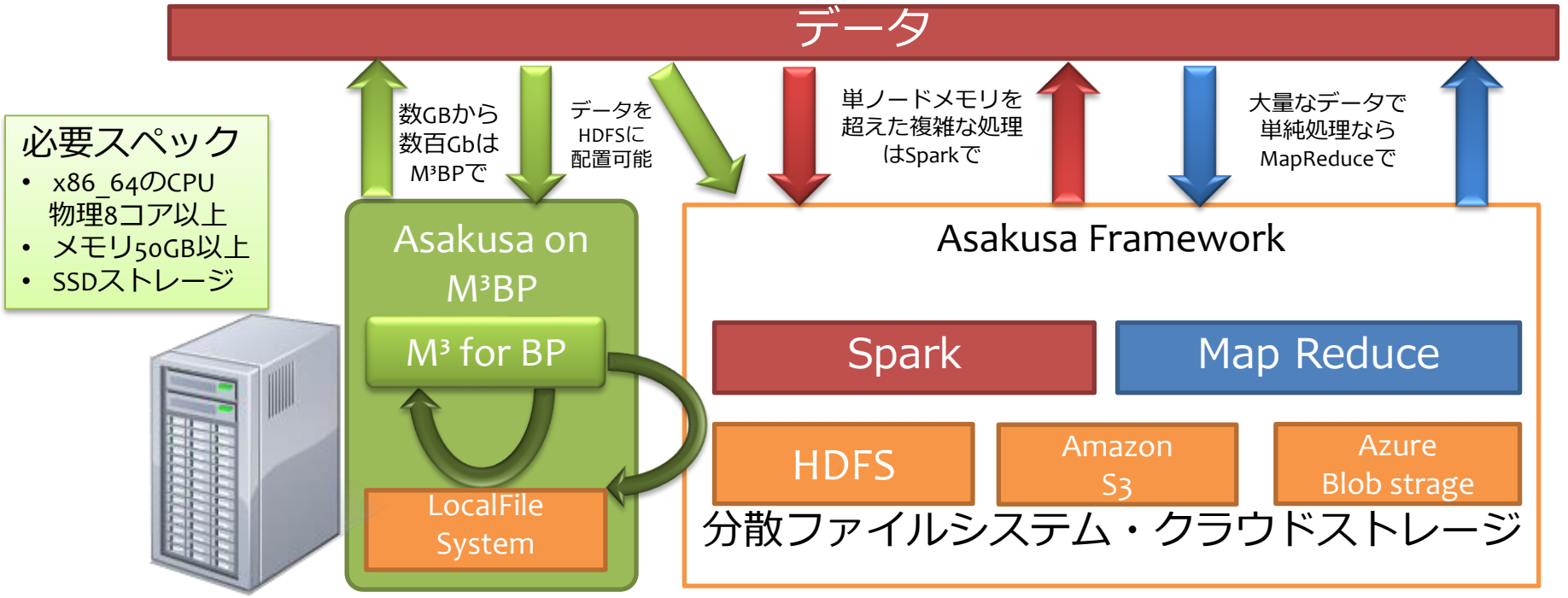




単一ノード・マルチコア・大量メモリにて、小規模バッチ処理をSparkよりも高速処理するM³ for Batch ProcessingとAsakusa on M³BP

Asakusa on M³BPの動作環境

- ローカルと分散の両方にデータアクセス可能
  - 単ノードで、Linuxのローカルファイルシステムやマウント可能なストレージにアクセス
  - データだけ分散ファイルシステム上に配置し、M³BP単ノードで実行をすることも可能
    - クラウドも利用可能だが、性能面で物理コンピュータを推奨
    - Hadoop/Spark上で、Asakusa on M³BPを動作させるのは非推奨(リソースの干渉)
- 数GBから～数百GBのデータについてM3 for BPを利用し、更にデータ量が増えたらアプリケーションを改変せずに、Hadoop/Sparkにエンジンを切り替え複数ノードにスケールアップ可能



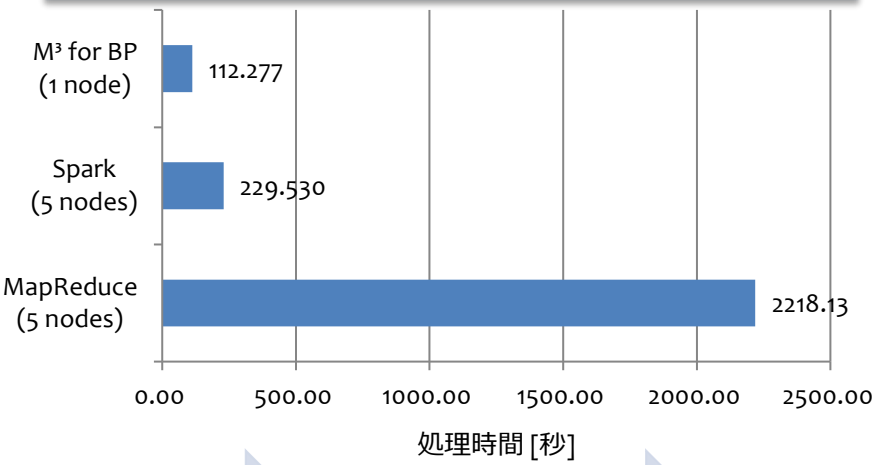
Asakusa on M³BPの性能

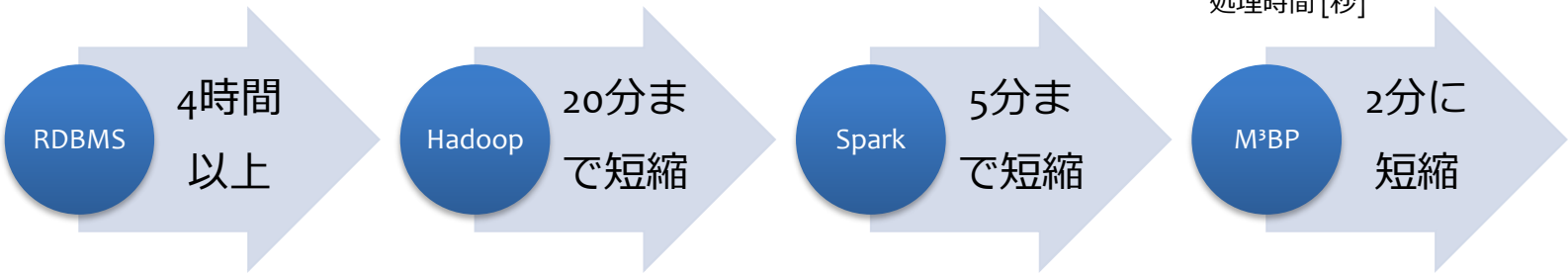
- Hadoop/Spark 5 ノードとM³ for BPにて性能比較
  - Microsoft Azureを利用し、実際の業務バッチで計測
  - 入力データサイズは、約 5 GB

表1　評価環境　　* 計算ノードとは別に管理用ノードが1台必要

	MapReduce	Spark	M³ for BP
バージョン	2.7.2	1.6.1	0.1.0
Java処理系	Java SE Development Kit 8 Update 74		
C++コンパイラ	N/A		GCC 4.8.5
OS	CentOS 7.1		
インスタンスタイプ	Microsoft Azure Virtual Machines D5 v2 16CPUコア、メモリ56GB		
計算用ノード数	5 (*)		1

表2 評価結果
- Sparkの5ノードに対し、M³ for BP の1ノードで約2倍の性能

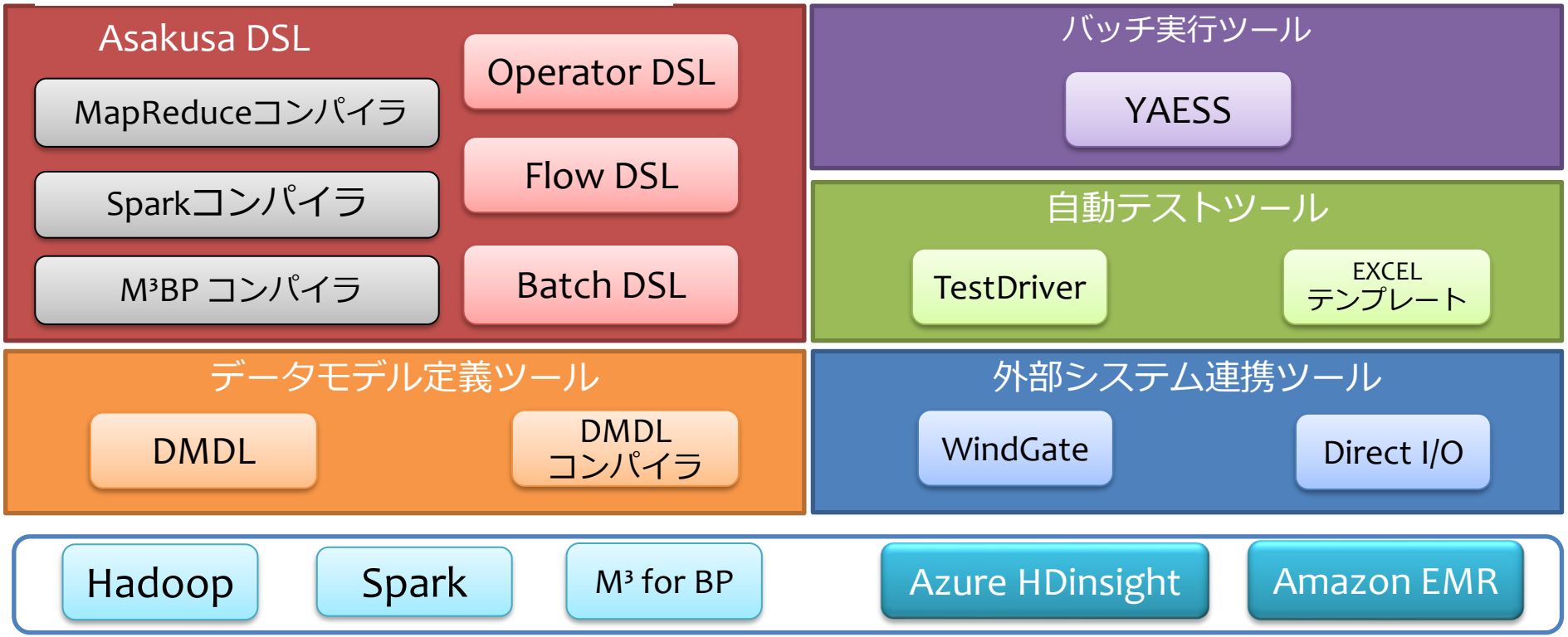




Asakusa Frameworkとは

- **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

- **分散処理のアプリケーション開発を容易にし、更に高速化**
  - 分散処理のプログラムは、Asakusaのコンパイラが自動生成し最適化を実施
  - MapReduce/Spark/M³ for BPのコードを、１つのAsakusaDSLで生成可能
  - Hadoop/Sparkのバージョンアップを、Asakusa　Frameworkのバージョンアップで追従（後方互換性の担保）

３つの特徴

開発容易性	性能	ポータビリティ
• HadoopやSparkの開発方法を覚えなくても開発可能 • ローカル環境でのテストが可能 • 開発時のテスト、チェック機能充実	• ユースケースに応じ最適な実行基盤の選択が可能 • Asakusaのコンパイラや、内部エンジンによる高速処理の実現	• 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 • オンプレミスでもクラウドでも動作

Asakusa Frameworkに関連したサービス

- 弊社が提供する主要なサービス

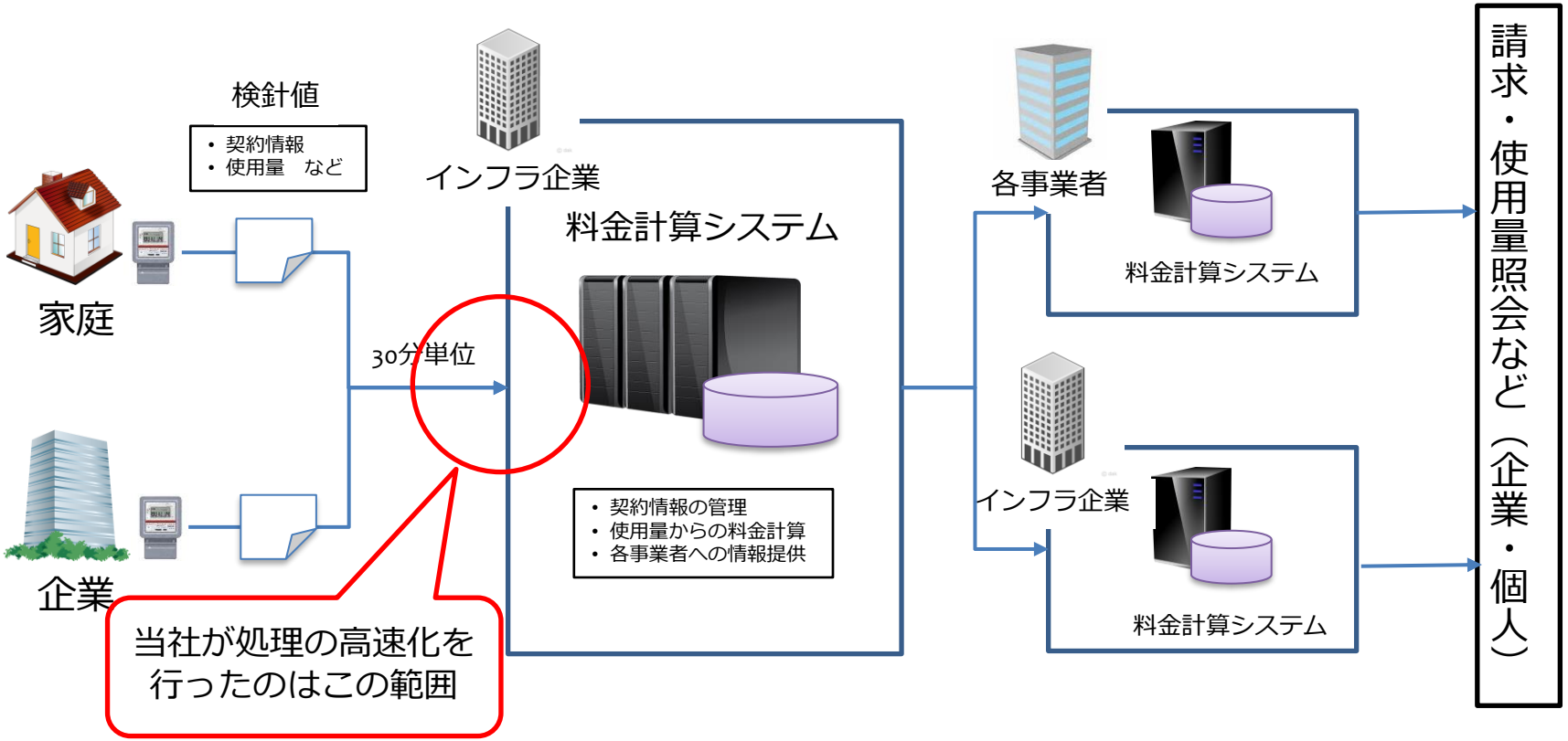
サービス	内容
Asakusa Framework サブスクリプションサポート	Asakusa Frameworkを、有償サブスクリプションとして提供し問合せ対応やV-up版の提供をします
Asakusa Framework プロフェッショナルサービス	• Asakusa Frameworkでの導入を前提に、要件定義やAsakusa DSLでの設計の支援を行います • Asakusa Frameworkでのアプリケーション開発をお客様で実施される場合に、問合せ対応、レビュー等の技術支援を行います
Asakusa Framework PoC/ アプリケーション開発	Asakusa Framework/Hadoopでの検証作業や、アプリケーション開発を実施します
システム分析 コンサルティング	パフォーマンスが出ないシステムの分析や、システムの改善点に関してのコンサルティングを実施します
トレーニング/PaaS	• Asakusa Frameworkトレーニング(入門編) • クラウドプラットフォームサービス(Node0 DBR)

Asakusa Framework 、Asakusa on M³BP導入事例

事例. 某社会インフラ企業
基幹システムの入り口でのデータ処理

使用量請求システムでの活用事例（1/3）

- 全体の概要図
  - 30分単位で送られてくる検針値から、料金請求や使用量照会までを行えるようにするシステム



使用量請求システムでの活用事例（2/3）

- 課題
  - スマートメータの設置数の増加により、30分単位で送られてくるデータ量が増えていくのに合わせ、処理が遅くなり決められた時間内で処理ができない
- 対象となった処理
  - 30分単位でスマートメーターから送られてくるデータの加工処理
    - データベースに取り込むための加工処理
    - データの突き合わせや変換処理
  - DBに取り込まれる処理と取り込まれたあとにDBで実施される処理のいずれも対象
- 解決の方法
  - DBで行う処理を分散処理を使い高速化
  - できるだけDBに書き戻す回数を減らし、時間の掛かりそうな場所を減らす

使用量請求システムでの活用事例（3/3）

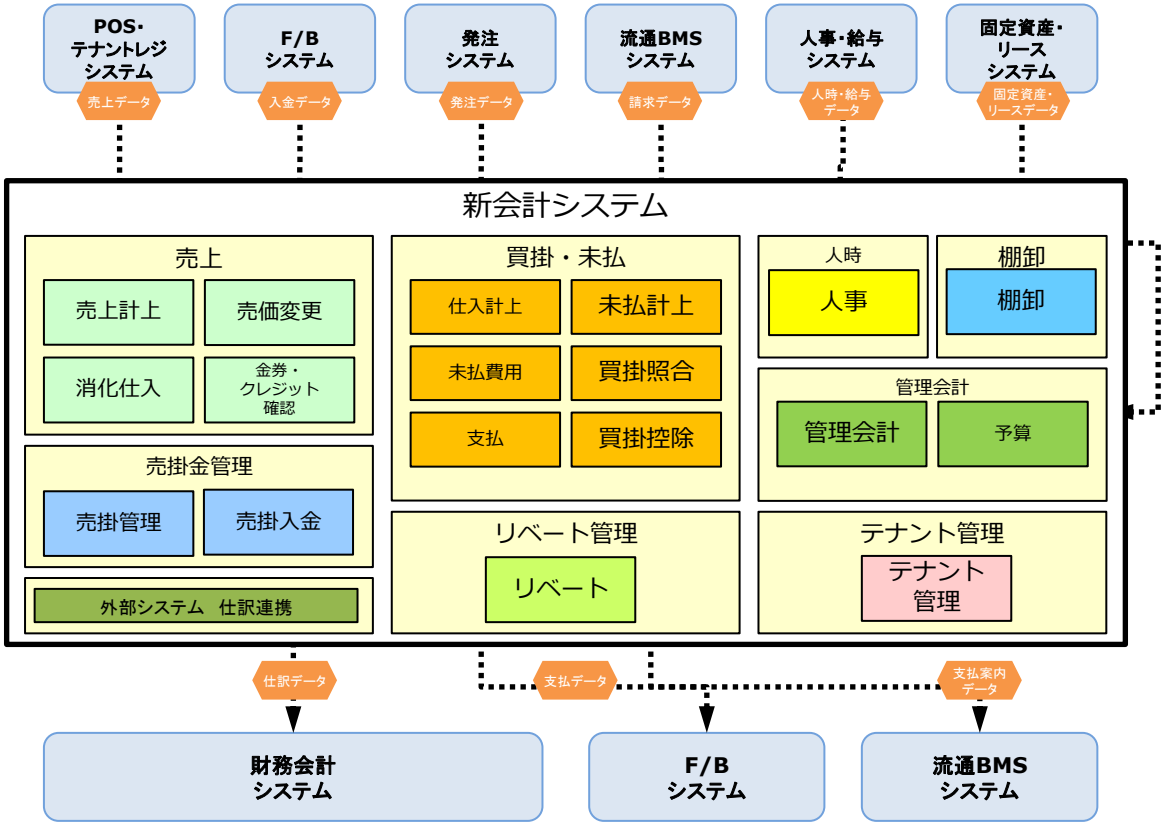
- どういう効果があったのか（成果）
  - 30分単位に集まるデータを5分で処理
    - DBでは、**200万件で25分**かかっていた
    - 総数約**1,000万件**（すべての契約がスマートメーターになったときの最大値）を**5分**で処理
  - 主な処理は ”結合処理”
    - スマートメータからくる1,000万件ほどのデータと契約者のマスタを突き合わせ、更新する処理
    - 突き合わせの結果、更新された契約者マスタと使用量の情報が組み合わされ、料金の請求が行われる

Asakusa Framework 導入事例

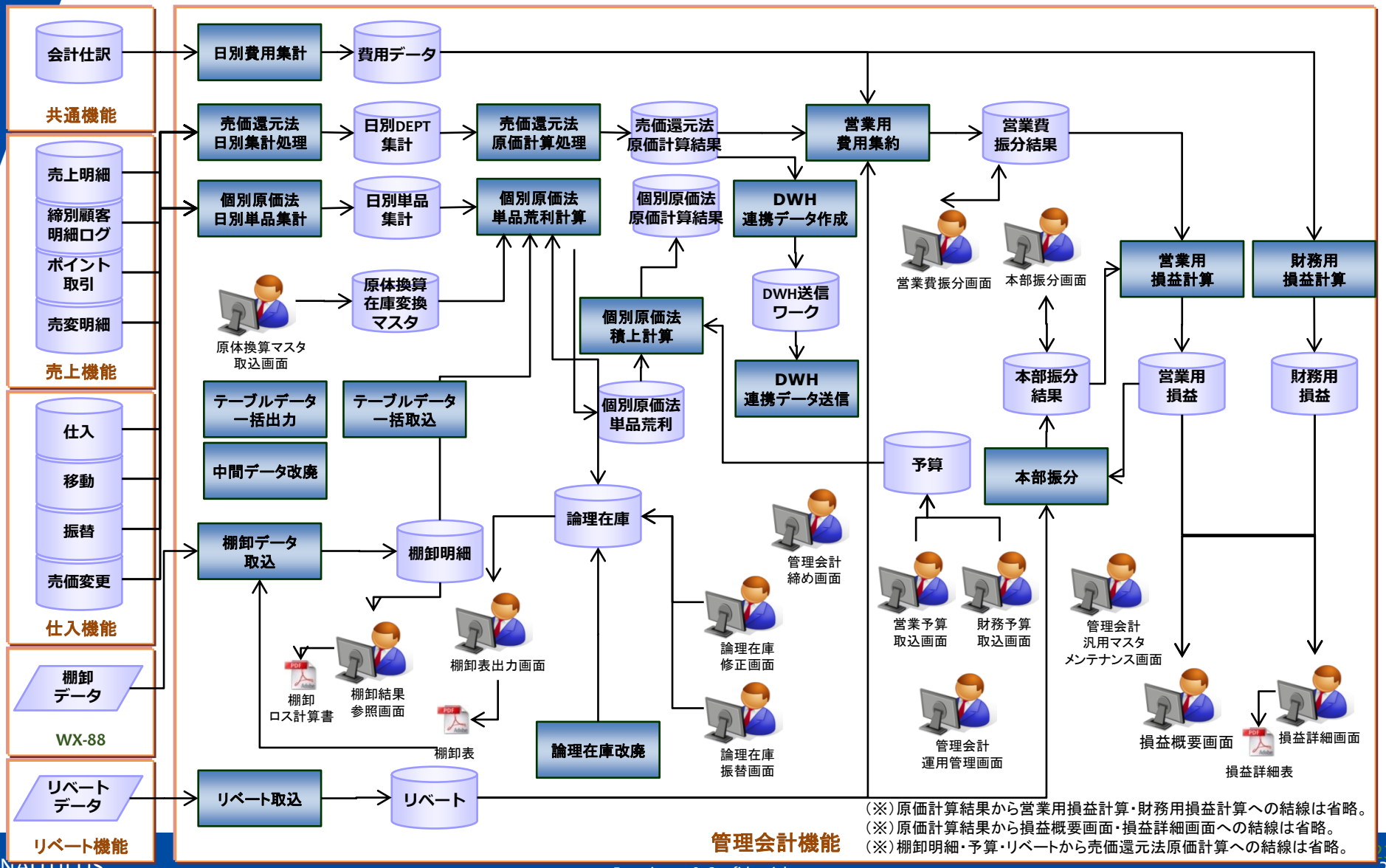
活用事例.株式会社西鉄ストア様
本部基幹システム（管理会計システム）への導入

西鉄ストア様　新会計システム

- システム機能(導入時点)
  - 売上
    - 最大で1億件/per dayの締め処理
  - 債権管理
    - 900万件程度の集計処理
  - 仕入
    - 2700万件の取り込みクレンジング処理
  - 債務処理
    - 日別で300万件のマッチング処理
  - 原価計算
    - 5億4千万件の明細処理
- データサイズ
  - 導入時：100GB～500GB
  - 現在時：２～３TB
- バッチ処理時間
  - 導入時：8時間（管理会計部のみで3時間）
  - 現時点：データ量の増加から13時間に延伸したため、M3BPへ移行し5時間まで処理時間を短縮する
- AWS上にて、24時間365日の運用　2013年から本番稼動
  - 運用・監視・バックアップ・リストア



管理会計のバッチ処理フロー



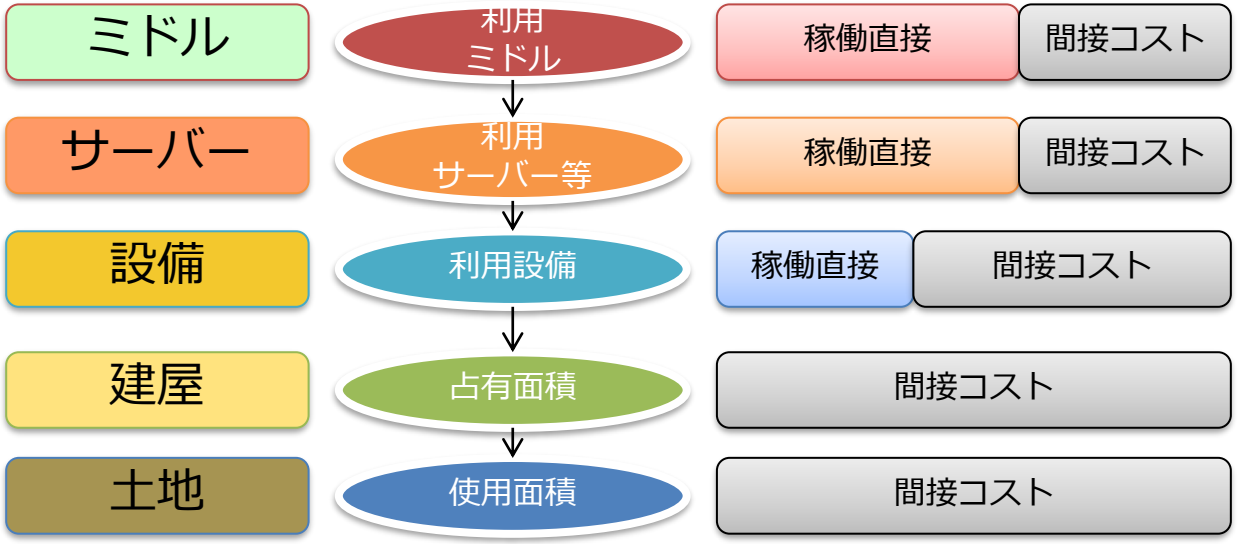
Asakusa Framework 導入事例

活用事例.さくらインターネット株式会社様 データセンター原価分析ソリューション

さくらインターネット様

原価分析/ IoTソリューション事例

- **データセンターの原価処理の課題と対策**
  - 土地、建屋、設備、サーバー、ソフトに電力料金を積み上げて原価を計算してきた
  - サービス改善、新サービスの追加するために、ユーザ単位の原価把握が必要
    - どのユーザが、どの程度利用をしているのか？が不明で、追加投資ができるかどうか不明
- **データセンターの管理会計に挑戦**
  - 電力量、トラフィック量などの100数十万のログデータをユーザ単位で集計する
    - 各構成コストの原価ドライバーの因果関係のモデル化する
    - ハードウェア/ソフトウェア/人件費/修理費の会計データを積み上げ、ユーザ毎に利用量を振り分ける
    - 各ファクターの限界コスト（マージナルコスト）を算出
      - 例）サーバーを一単位増やした場合の追加的に発生するコスト



発生コストを
稼働ベースの
因果関係で
モデル化する

さくらインターネット様

原価分析/IoTソリューション事例

■ バッチシステム概要

- 電力量やトラフィック量等の膨大なログデータと、ユーザ毎の会計データを紐つけ原価を算出するシステムを構築し、最初に石狩データセンターで提供されるVPSと専用サーバの原価計算を実施
- RDBMSで想定20時間かかるバッチをMapReduceで１時間まで短縮。更に、Sparkを使って約10分まで短縮

■ 本ソリューションの成果

- データセンター毎の原価が把握でき、データセンター別、サービス別の利益率にも差があり、対外接続先が多くあるデータセンターほど原価が多く発生していることが明確になった