

2025年の崖を乗り越える！

バッチ処理のモダナイゼーション

株式会社アシスト
情報基盤技術統括部 技術3部
宮本 玲

自己紹介



宮本玲

- 1997年アシスト新卒入社
- メインフレーム性能管理製品技術担当
- ETL製品技術担当/マネージャ 2005~
- データ連携製品技術マネージャ 2014~
- 趣味：将棋、落語、手品、青春18きっぷ

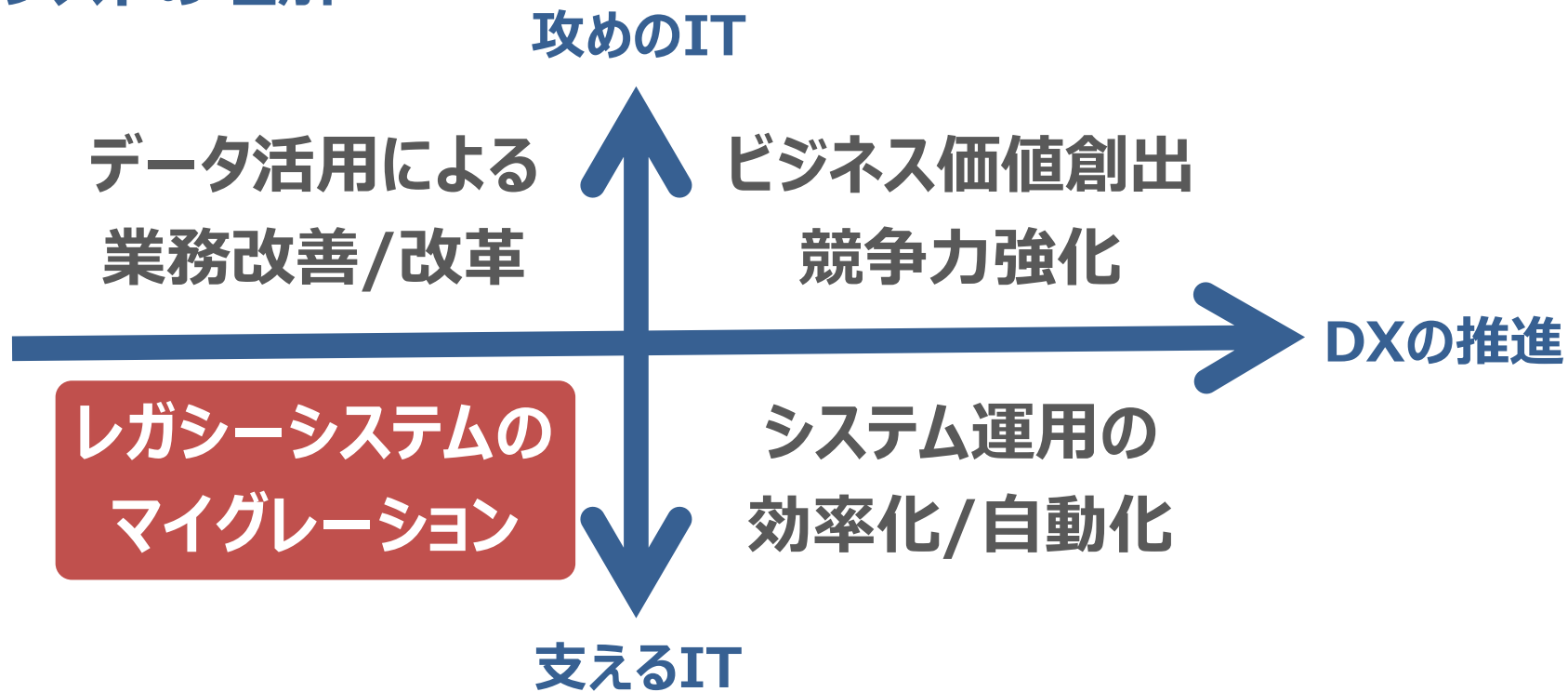
アジェンダ

1. バッチ処理のモダナイゼーション
2. バッチ処理基盤を考える
3. モダナイゼーション事例をまとめる
4. エンジンの特性

1. バッチ処理のモダナイゼーション

2025年の崖 (DXレポート)

● アシストの理解

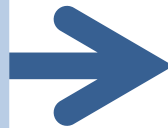


レガシーシステムの問題点と対応策

※出展元：DXレポート～ITシステム「2025年の崖」の克服とDXの本格的な展開～」（経済産業省）を加工して作成
https://www.meti.go.jp/shingikai/mono_info_service/digital_transformation/pdf/20180907_01.pdf

傾向

- 利用技術の老朽化
- システムの複雑化
- ブラックボックス化



問題点

- 変化に対応できない 56%
- 属人化により継承が困難 67%
- 保守運用のコストが高い 52%



期待

- 変化への対応力
- 人財の活用
- 運用コストの削減



対応策

レガシーシステムの資産を活かして
最新の製品や設計で置き換える
(レガシーモダナイゼーション)

レガシーモダナイゼーション

● 3つの対応方式

対応方式	① リホスト インフラのみ刷新	② リライト アプリを書き換え	③ リビルド 再構築
インフラ 特徴 アプリ 業務仕様	サーバ化 変更無し 変更無し	サーバ化 新言語・パッケージ 変更無し	サーバ化 新言語・パッケージ 変更
方式別の課題	変化に対応しにくい	技術習得の期間・コストが高い	
共通課題	メインフレームと同等の処理性能を担保できるか		

レガシーモダナイゼーション

● システム構成の比較

メインフレームシステム



オープンシステム



アプリケーション

オンライン

バッチ

オンライン

バッチ

ミドルウェア

データベース

データベース

ジョブ管理

ジョブ管理

帳票

帳票

統合監視

セキュリティ

データ連携

OS

メインフレームOS

Win/Linux/UNIX

ハードウェア

メインフレームH/W

オープン系H/W

バッチ処理のモダナイゼーション

レガシーシステムの バッチ処理課題

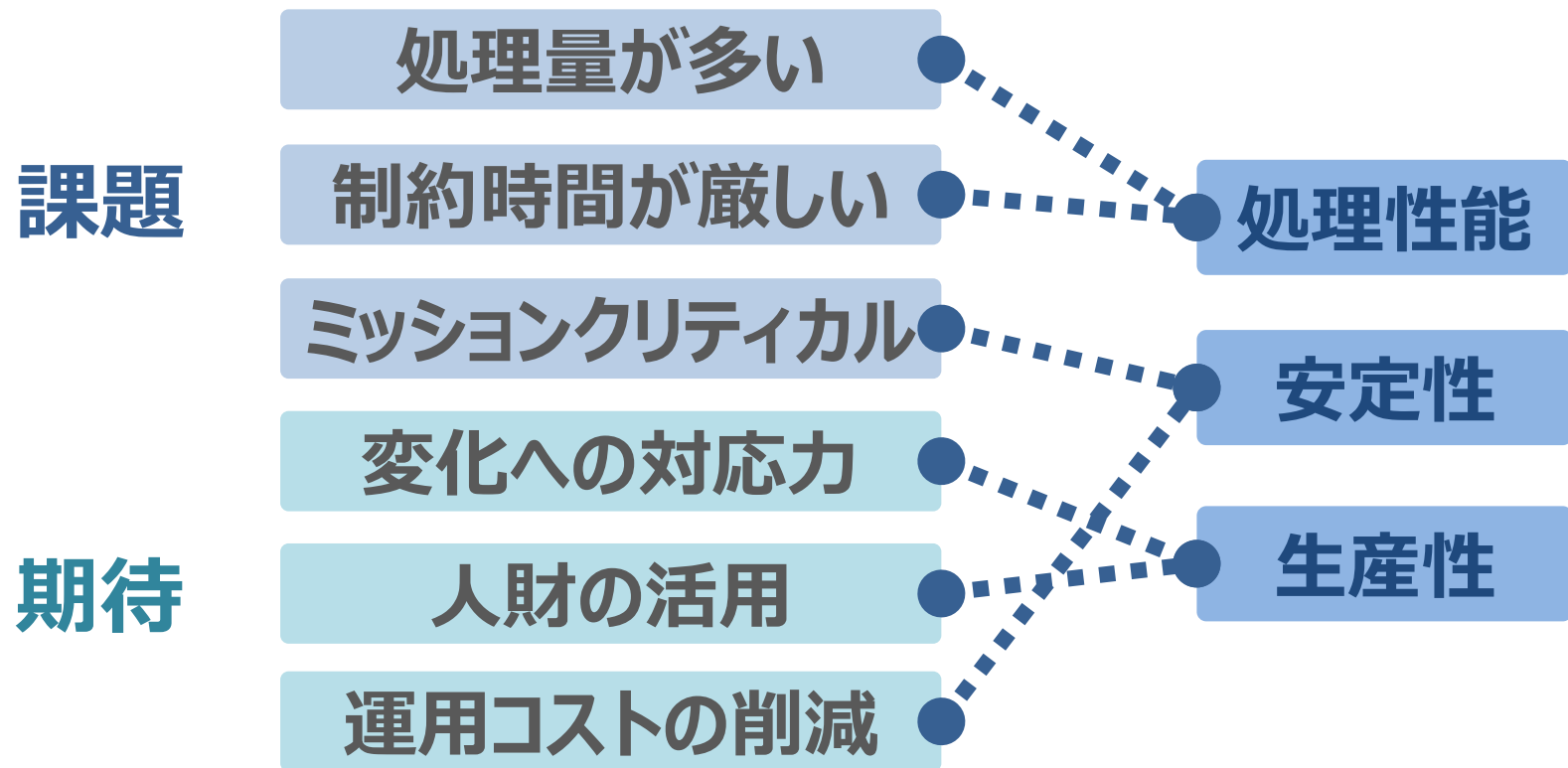
- 処理量が多い
- 制約時間が厳しい
- ミッションクリティカル

モダナイゼーション に期待する成果

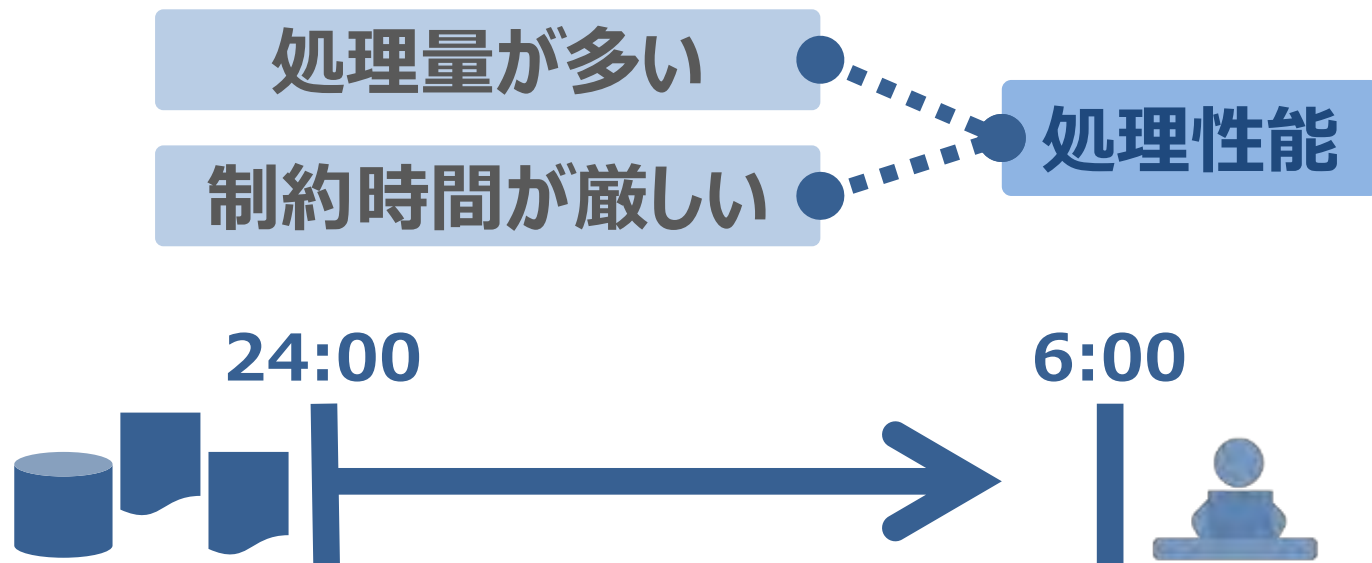
- 変化への対応力
- 人財の活用
- 運用コストの削減

- **課題をクリアし、期待を実現する
バッチ処理基盤の構築**

バッチ処理の課題解決のポイント



処理性能



- データ量が多い処理でも、メインフレームと同様の制約時間内で処理を完了できる基盤が必須

安定性

ミッションクリティカル

運用コストの削減

安定性

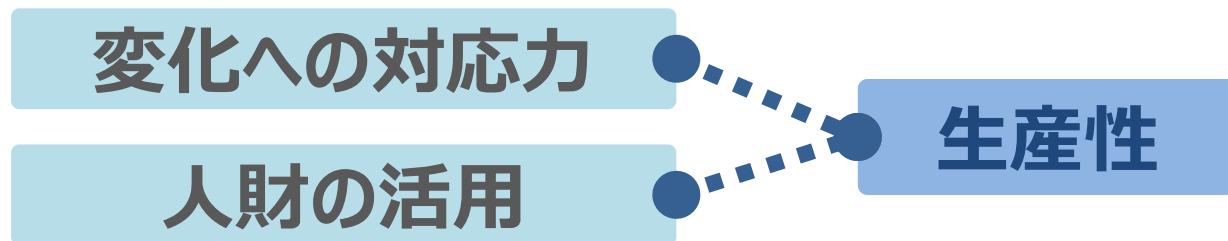
24:00



6:00

- 業務に密接に関わる基幹バッチ処理を、オープン化後も継続した安定運用できる仕組みが必須

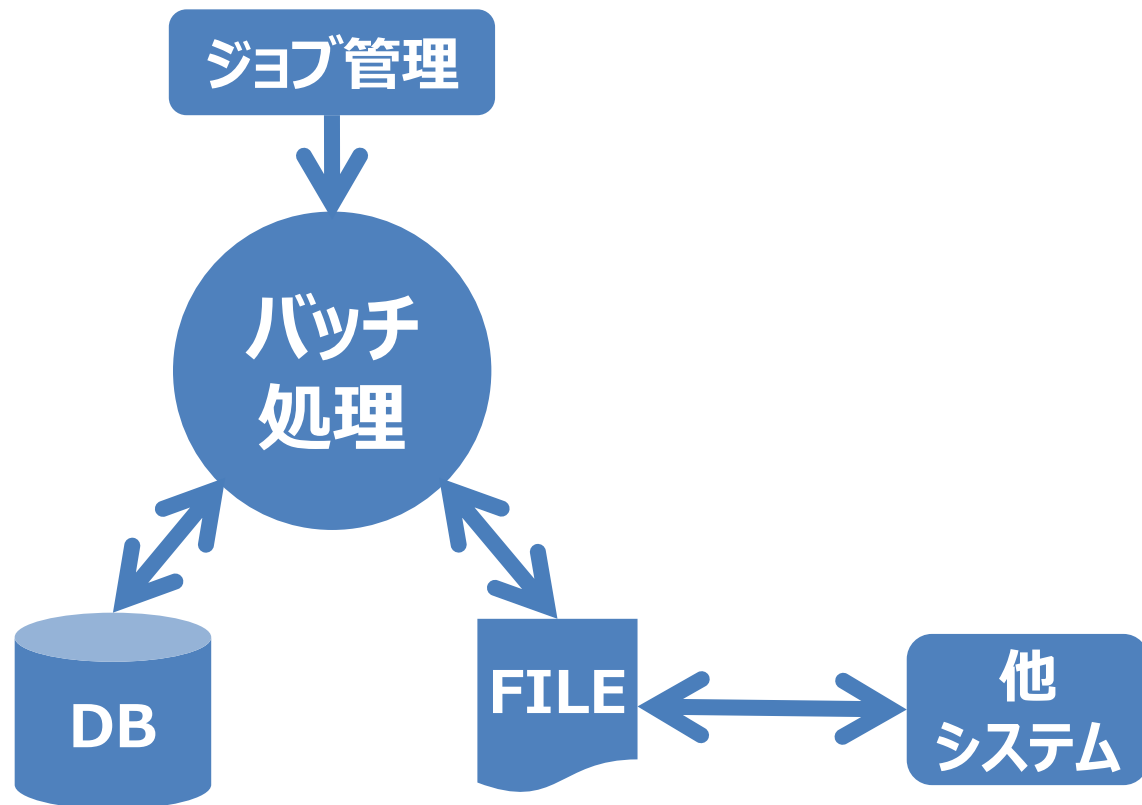
生産性



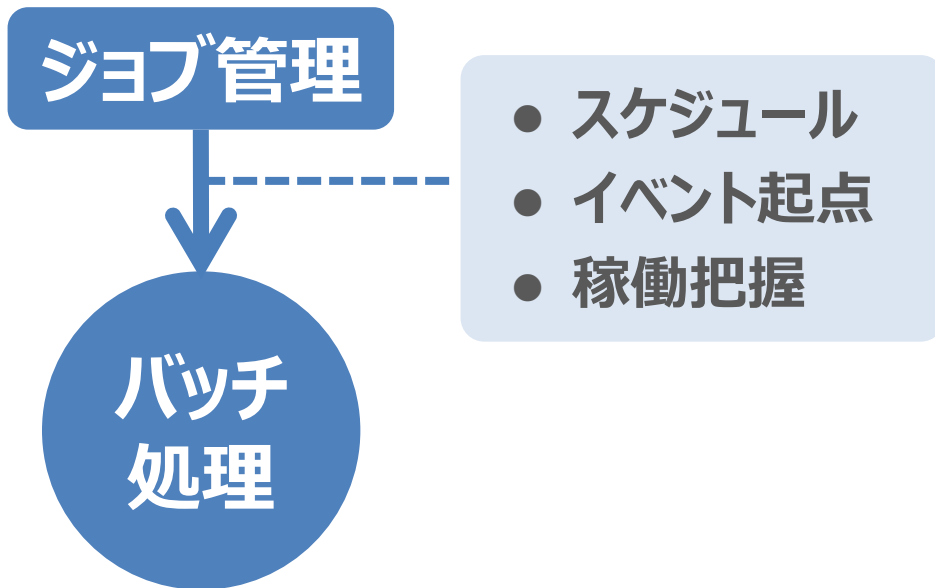
- **誰でもかたんに習得でき、属人化を防ぎ、ITチームとして変更要件に柔軟に対応できる仕組みが必要**

2. バッチ処理基盤を考える

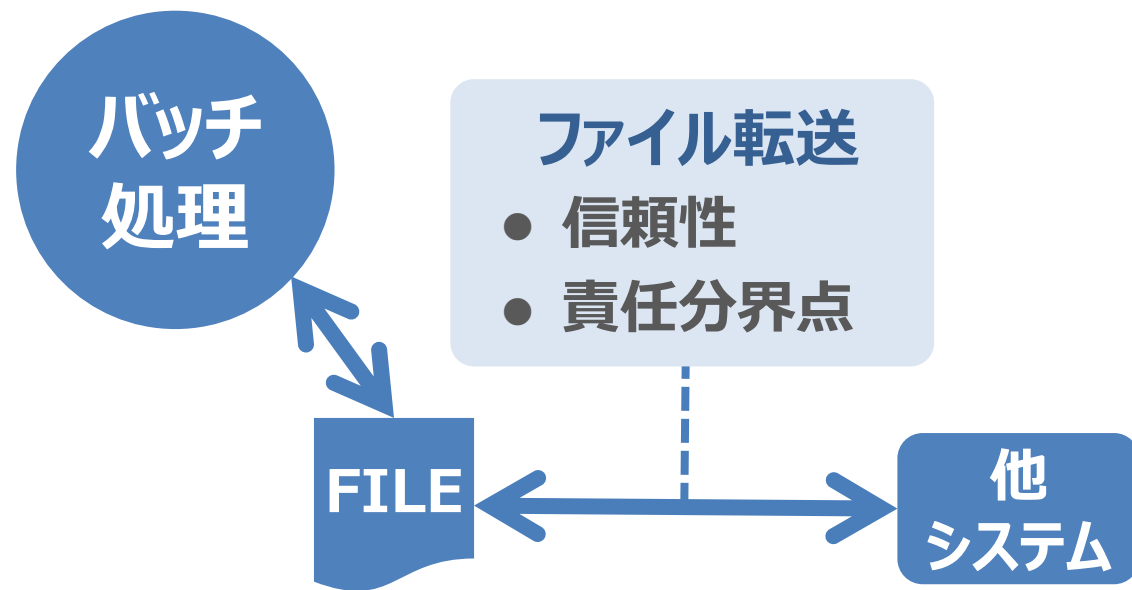
バッチ処理のシステム構造



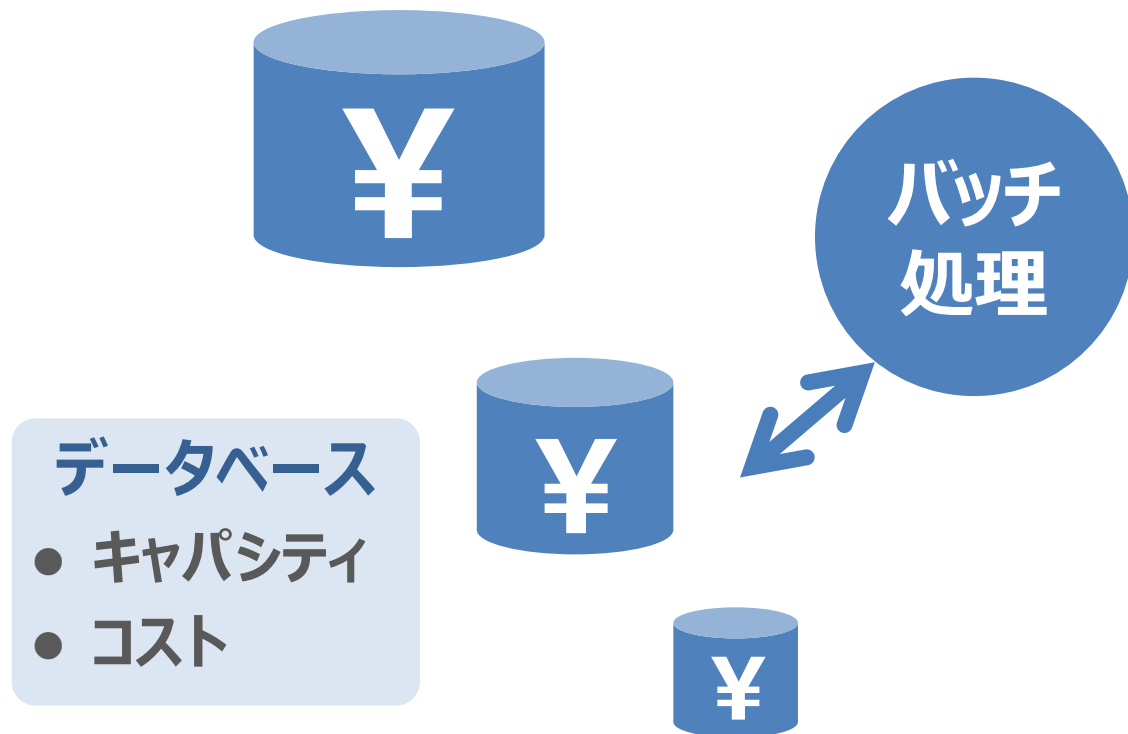
ジョブ管理



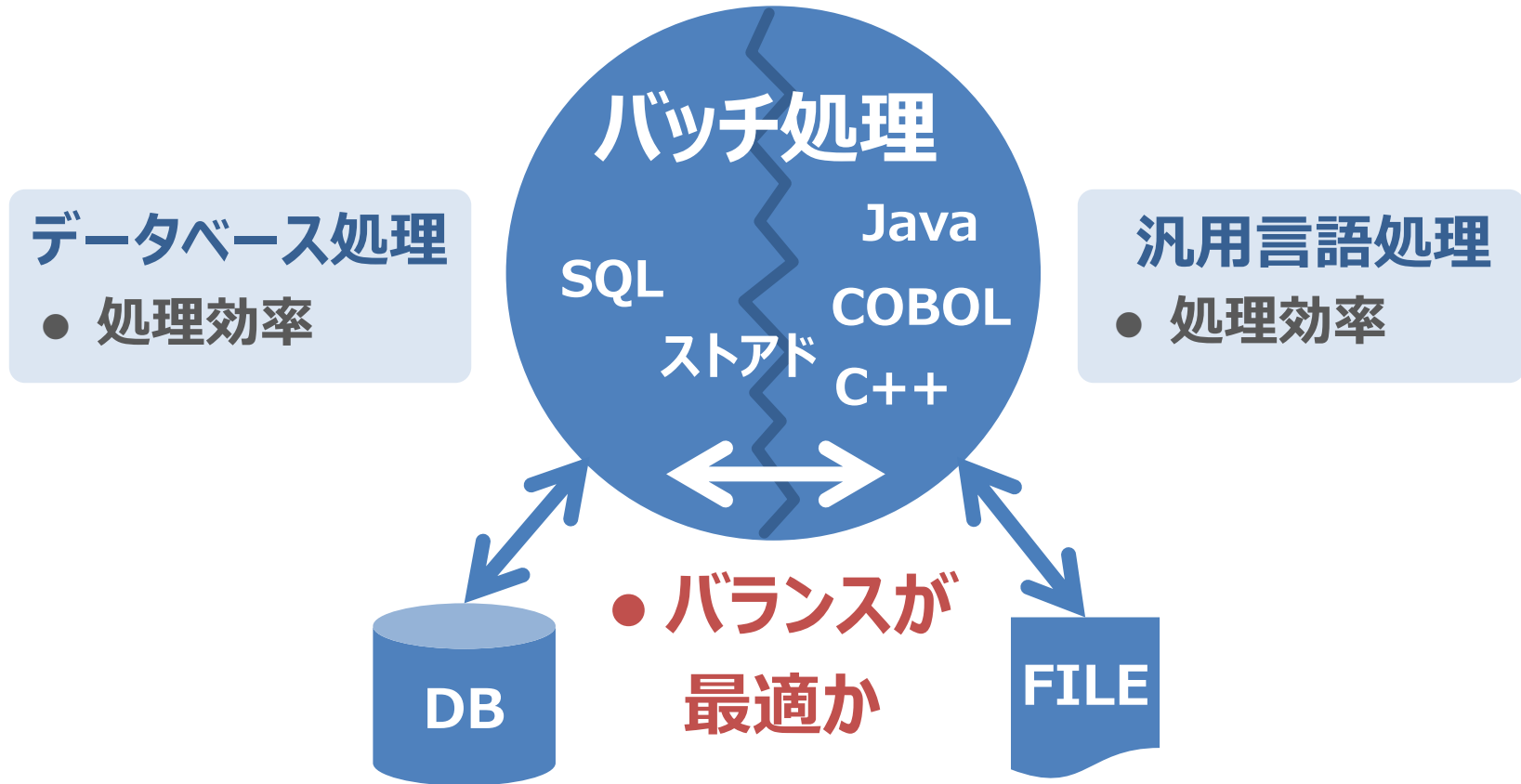
ファイル処理



データベース処理

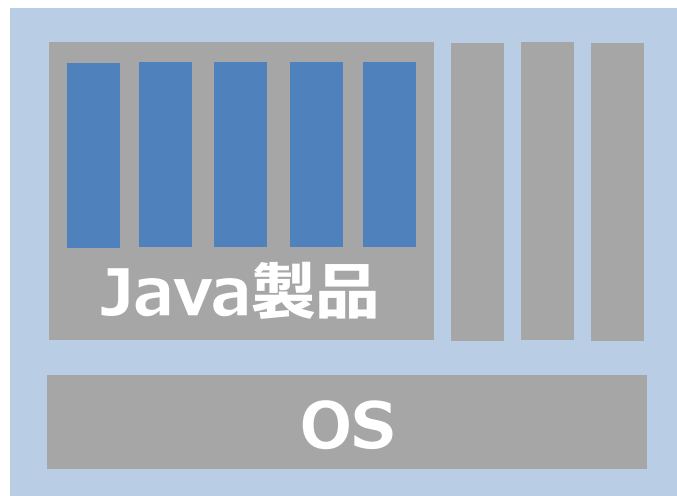


バッチ処理内部の役割構造

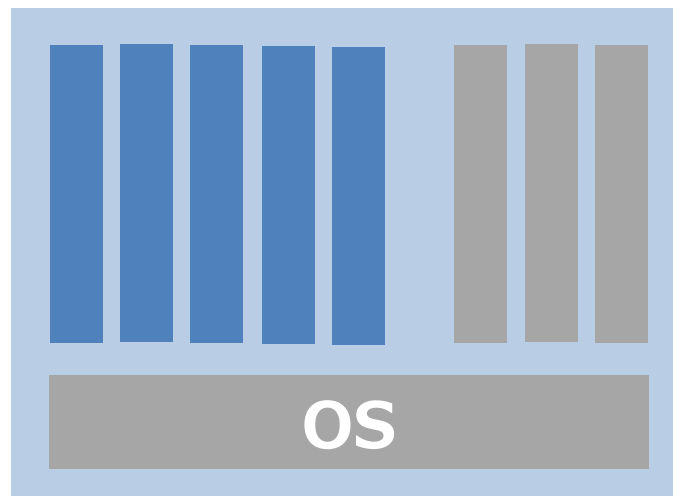


ミドルウェア構造

APサーバ指向

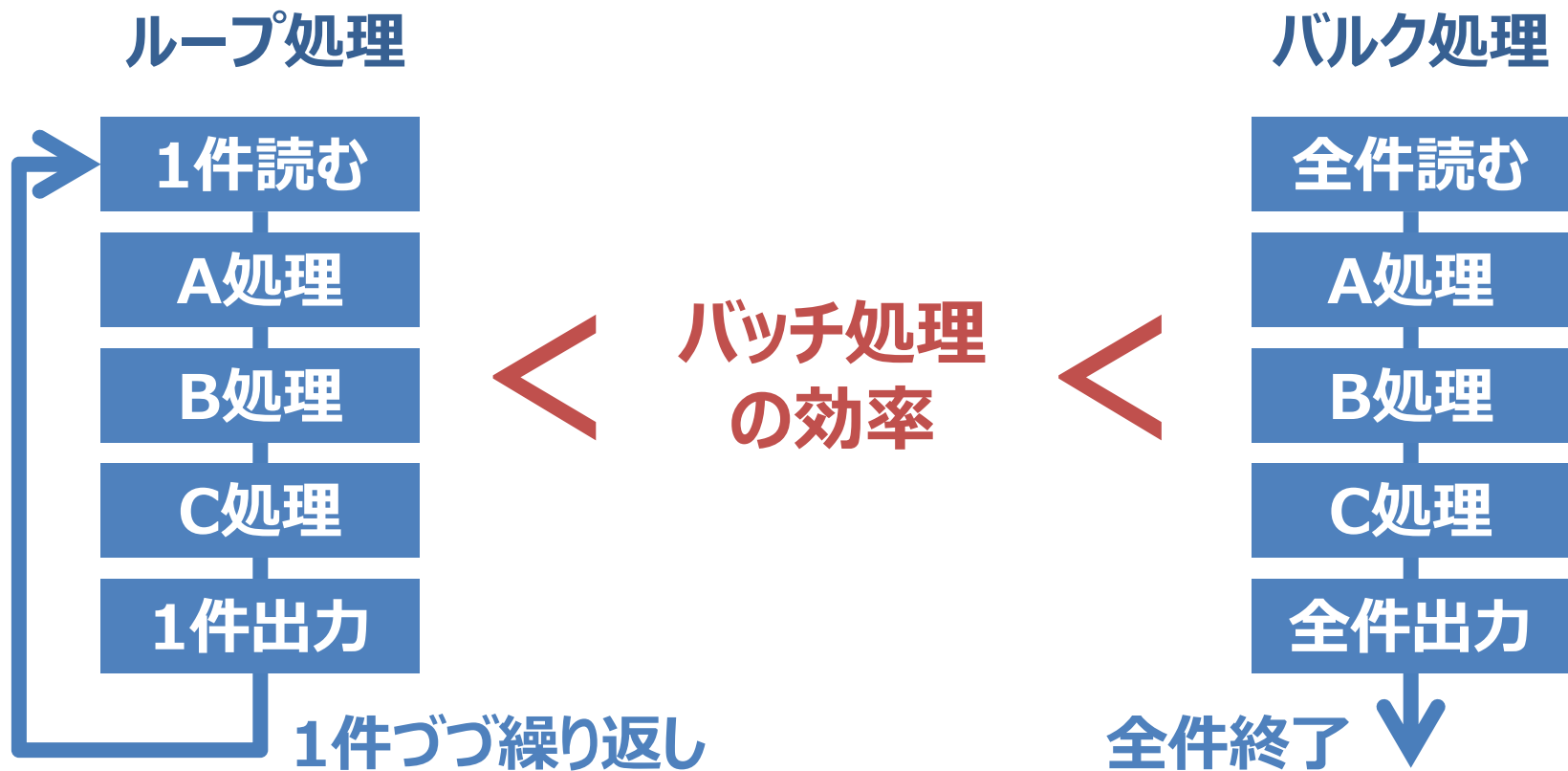


OSアプリ指向

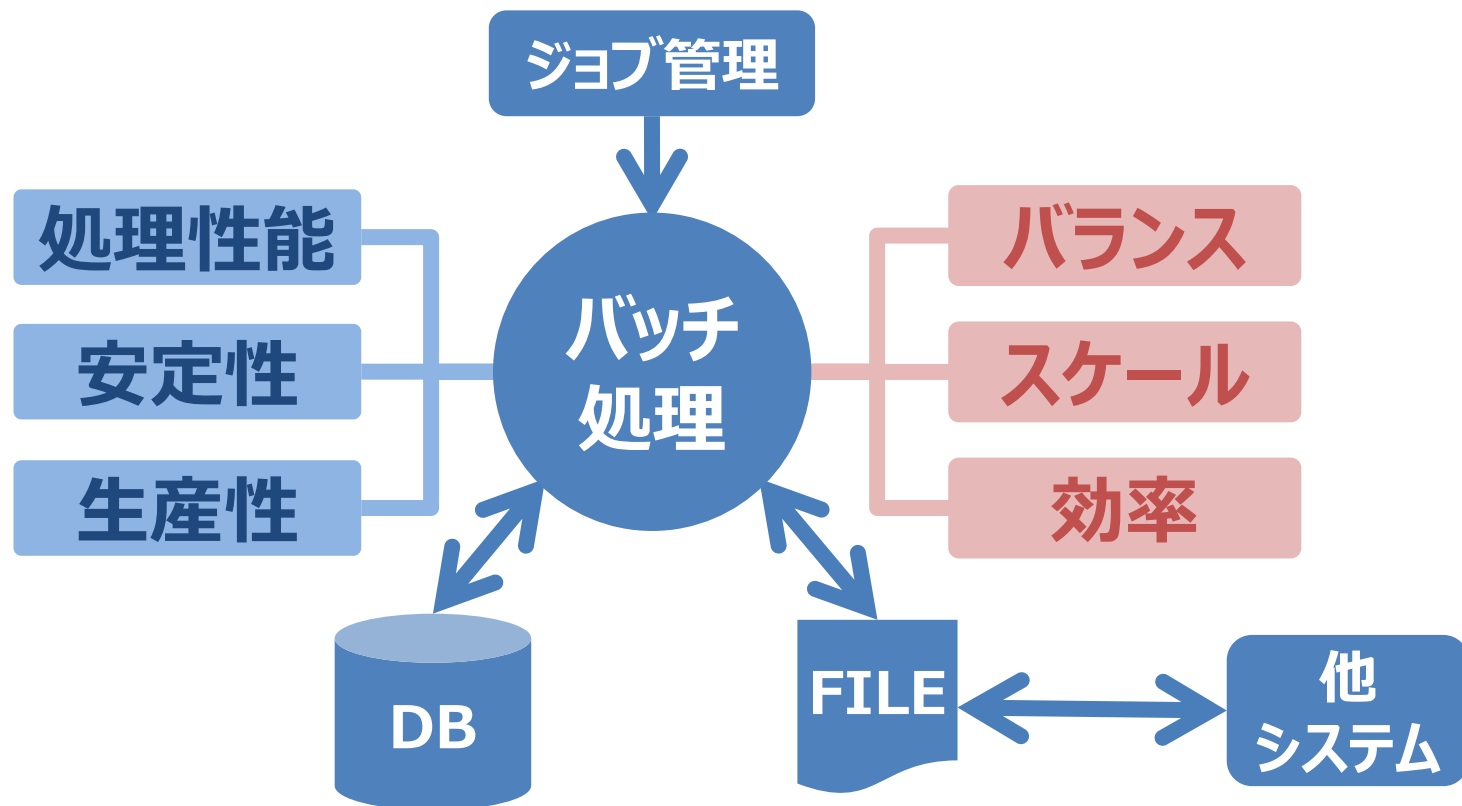


- バッチ処理の規模に対して性能がスケールするか

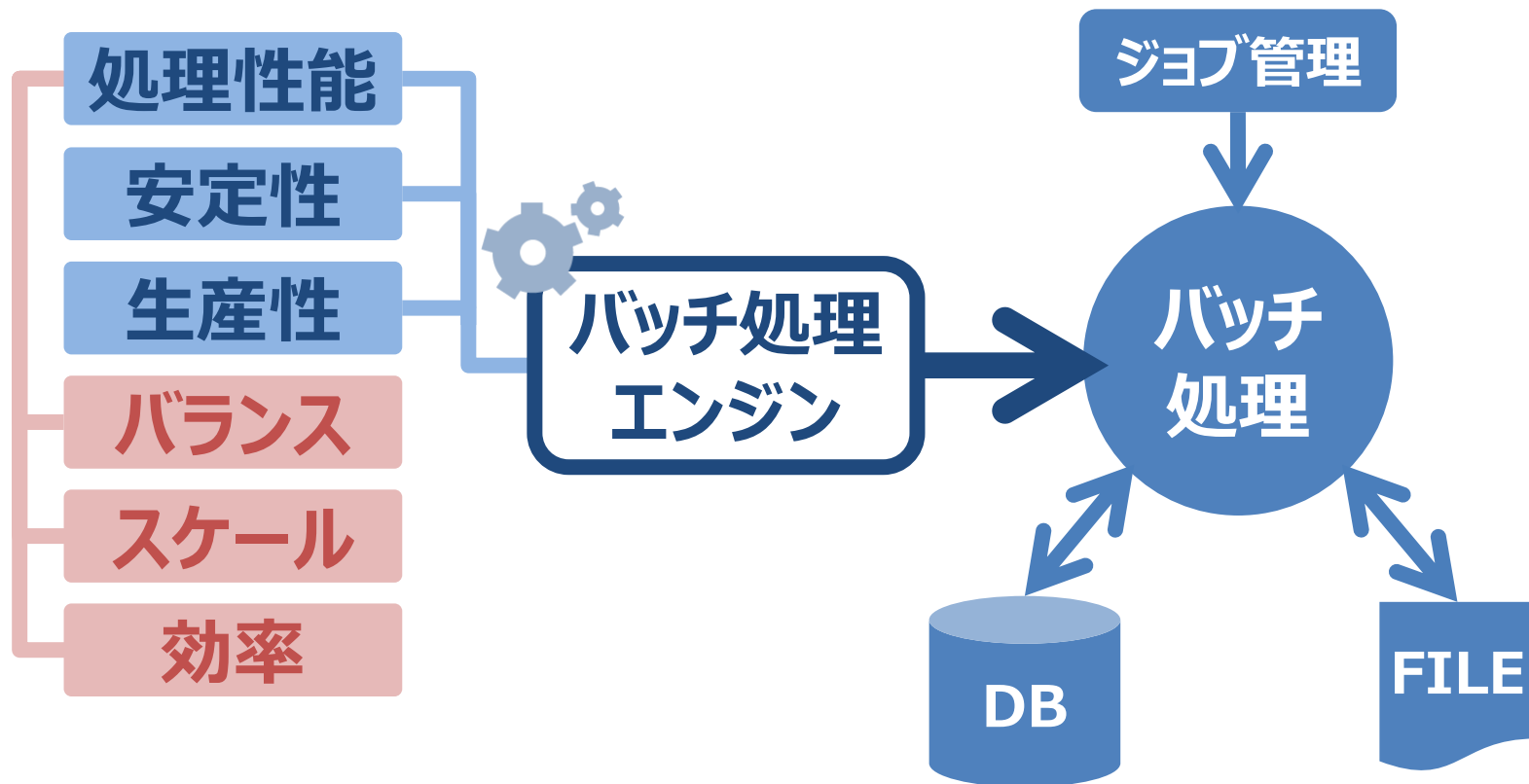
バッチ処理内部のロジック構造



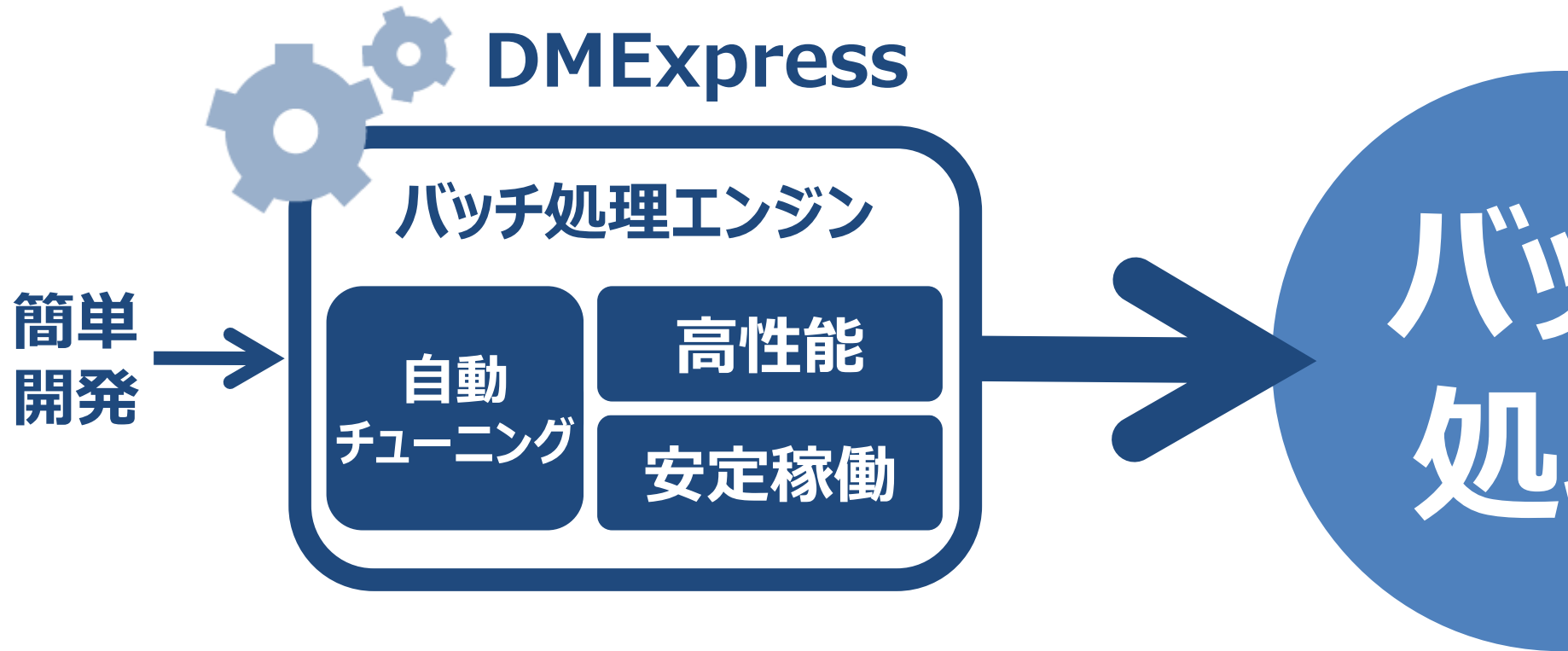
課題解決の要素



バッチ処理エンジンを組み込む方式



バッチ処理エンジン DMExpress



3. モダナイゼーション事例をまとめる

小島プレス工業様

メインフレームの販売・物流管理を含むバッチ処理基盤をオープン化するプロジェクトにDMExpressを採用

方式設計でPL/SQLやJavaを検討したが、処理性能とスキル習得に不安があり

処理性能

主要なバッチ処理の一つ、仕入れ処理。PL/SQLで65分かかったものが、DMExpressでは10分で完了。

6倍

安定性

外側で処理することで、Oracleのワークロードを軽減。現在、1000本以上のバッチ処理を安定稼働。

生産性

PL/SQLでは28時間かかった実装が5時間に。開発担当8名全員がバッチ処理開発できるチームに。

5倍

メディカルホールディングス様

20年以上使用したホストの基幹システムをオープン化するプロジェクトにDMExpressを採用

当初、バッチ処理適性を考慮してPL/SQLを選択したが、DMExpressへ

処理性能

一番重いPL/SQL処理は6時間以上たっても終わらず。DMExpressを採用したところ、1時間30分で完了。

4倍

安定性

約2兆円相当の薬の安定供給に関わる基幹処理。総数4000本超の処理がピーク時でも規定時間内に完了。

生産性

PL/SQLと比較すると、GUIで容易に開発可能。別の開発に人員を回すことでプロジェクトをスピードアップ。

中電シーティーアイ様

オープン化検討対象のバッチ処理に対するDMExpressの有効性を技術検証。共通ETL基盤としてシステム展開

技術検証における比較対象の開発手法：Java

処理性能

Javaを6多重で実行した処理時間は、10分25秒。
DMExpressは3分19秒で処理完了。

3倍

安定性

共通ETL基盤として運用を開始してから3年間、
障害発生は0件。安定稼働を実現。

生産性

JavaのPG/単体テストに要した128時間が33時間に。
JavaよりDMExpressの方が圧倒的に楽！

4倍

モダナイゼーション事例のサマリ

企業名	方式	利用製品・言語
小島プレス工業様	リビルド	DMExpress、Java
メディカルホールディングス様	リビルド	DMExpress、PL/SQL
某通信企業様	リビルド	DMExpress、C、Java、SQL
某製造企業様	リビルド	DMExpress、Java、SQL
中電シーティーアイ様	リライト	DMExpress、COBOL、Java
AGS様	リライト	DMExpress、C
某金融企業様	リホスト	DMExpress、COBOL

7事例の共通点

- 処理時間制約をクリアしている
- 障害発生を抑えて安定稼働を実現している
- 新規案件に迅速に対応できている
- スキルに依存しない開発推進体制を獲得している
- 対象システムのメインフレームを撤廃している
- ファイル処理を活用している
- データベース処理とのバランスを取っている

7事例

7事例

5事例

6事例

6事例

7事例

5事例

解決につながっているか

課題

処理量が多い

制約時間が厳しい

ミッションクリティカル

変化への対応力

人財の活用

運用コストの削減

● 処理時間制約をクリア

● 障害発生を抑えた運用

● 新規案件への迅速対応

● スキルに依存しない体制

● メインフレーム撤廃

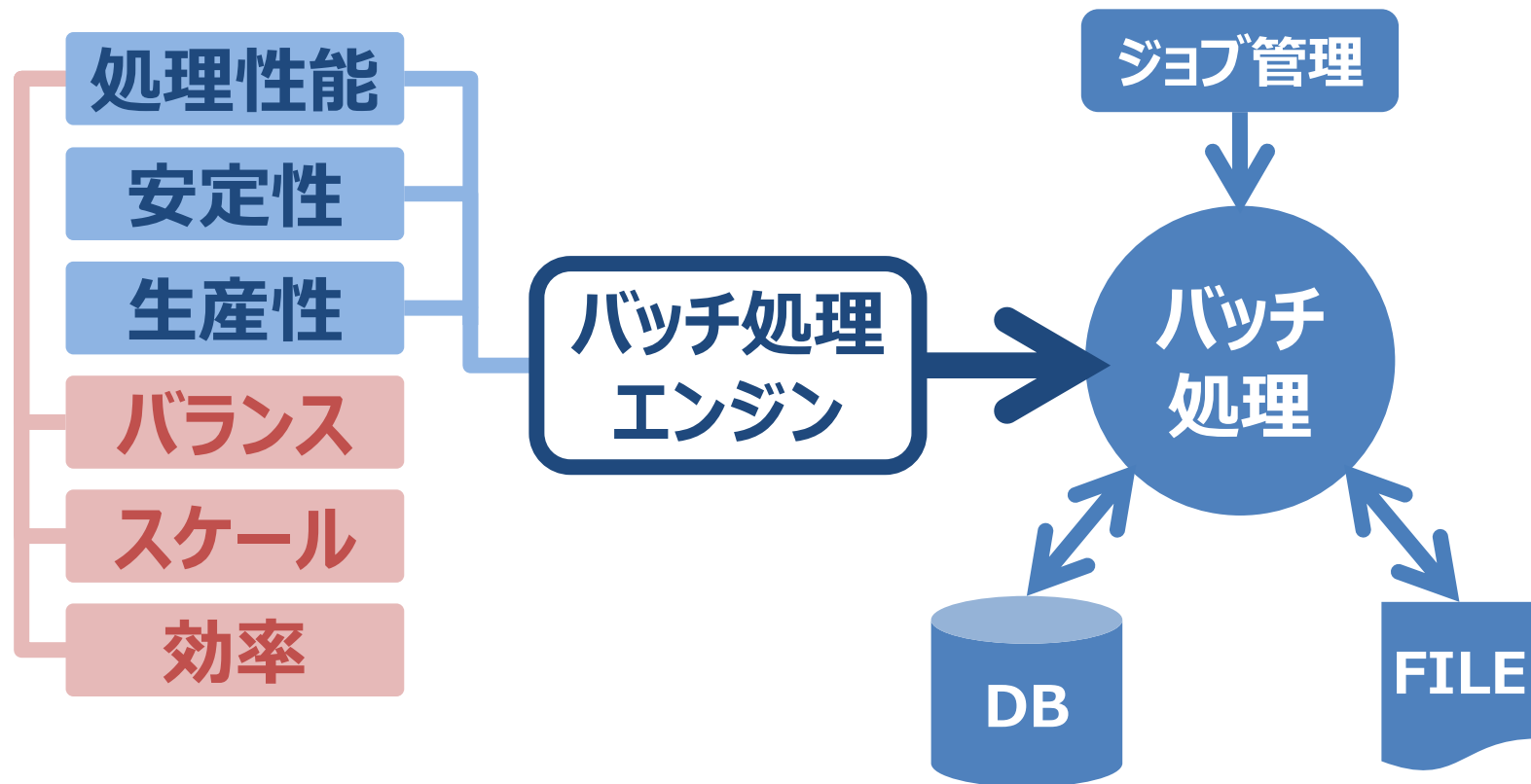
● ファイル処理を活用

● データベースとのバランス

期待

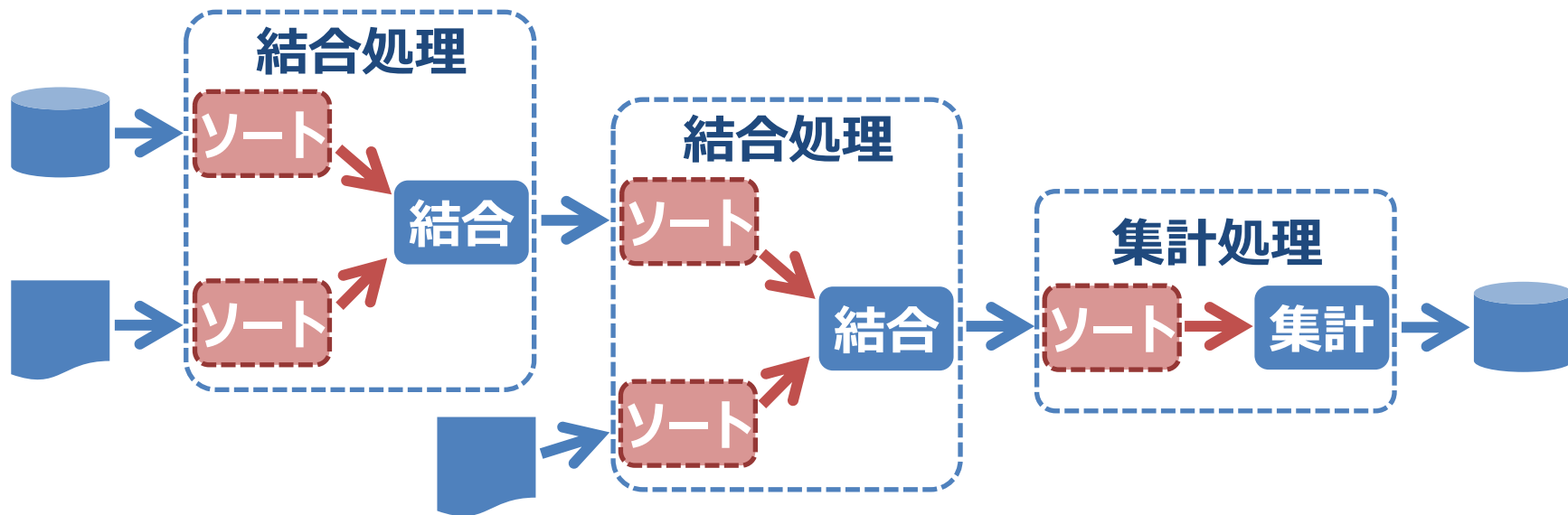
4. エンジンの特性

バッチ処理エンジンを組み込む方式（再掲）



ソートがボトルネックになりやすい

バッチ処理（バルク処理）の構造



- ソートで並べ替えが終わるまで、データは後続に流れない

処理性能 ソートを高速にさばくDMExpress

- データ量が増えたときにスケールするか

スケール

No	Data Size (GB)	スクラッチ開発 (C++)				ツール
		バブルソート	マージソート	クイックソート	クイック挿入ソート	DMExpress
1	0.5	N/A	13	10	10	4
2	1	N/A	27	24	24	9
3	5	N/A	136	124	116	41
4	10	N/A	992	962	925	84

単位 (秒)

2倍

8倍

2倍

処理性能 バッチ処理を速くする5つの仕組み

バランス



DMExpress

- ①ソートを速くする仕組み
- ②自動チューニングの仕組み
- ③ハイブリットな並列処理の仕組み
- ④結合を速くする仕組み
- ⑤DB入出力を速くする仕組み

処理性能 要点は処理効率

- 処理性能 \neq スピード

効率

	同じ 入力	同じ 機能性	同じ 結果	処理時間	CPU時間
他ETL製品				28分10秒	66分29秒
				2倍	5倍
DMExpress				13分26秒	13分29秒

[処理概要]

標準ベンチマークTPC-Hに準拠したデータを入力にしたテキストファイル処理。

8600万件（合計20GB）の8ファイルを入力として結合処理を行い、6000万件（43GB）のファイルを出力。

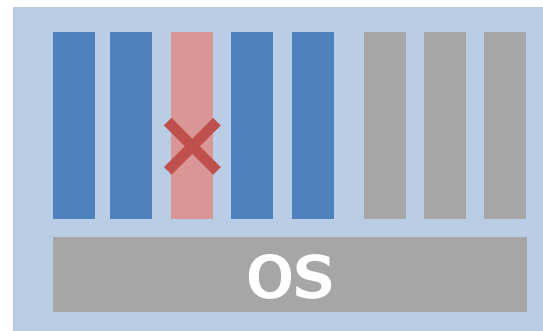
安定性 障害が発生しにくい

● 障害点が少ない製品構造

- ・ リポジトリ用のデータベース
- ・ Javaなどのコンパイラ
- ・ コンソール用のAPサーバ
- ・ 実行時のランタイムエンジン

全て不要

● 非APサーバ指向で 障害の影響を極小化



● 優れたソフトウェア品質が、ユーザから高く評価される

安定性 長期利用可能なサポートポリシー

弊社アフターフォロー訪問におけるご案内事項

- 全てのバージョンにおいて、同一のサポートレベルを提供します。
 - ただし、不具合が生じた際の個別のパッチ提供はなく、新バージョンによって提供されます。
-
- 安定稼働しているシステムに対して、次回のシステム更改まで全く手を入れない運用も可能

生産性 わかりやすいGUI

5種類のパーツから
処理を選ぶ



集計



コピー



結合

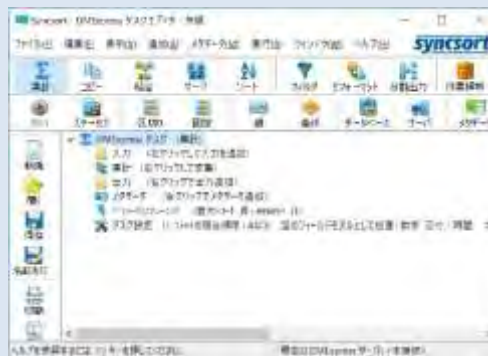


マージ



ソート

テンプレートに
パラメータを入力

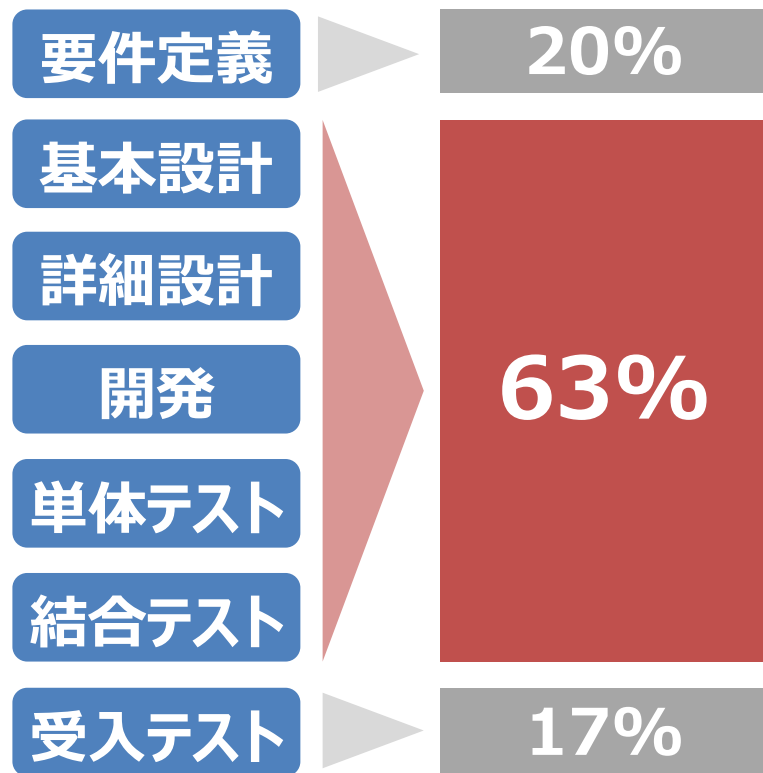


フローでつないで
ジョブが完成



- 一日の製品トレーニングを受講すれば、開発者として立ち上がる

生産性 開発工程フェーズの比率



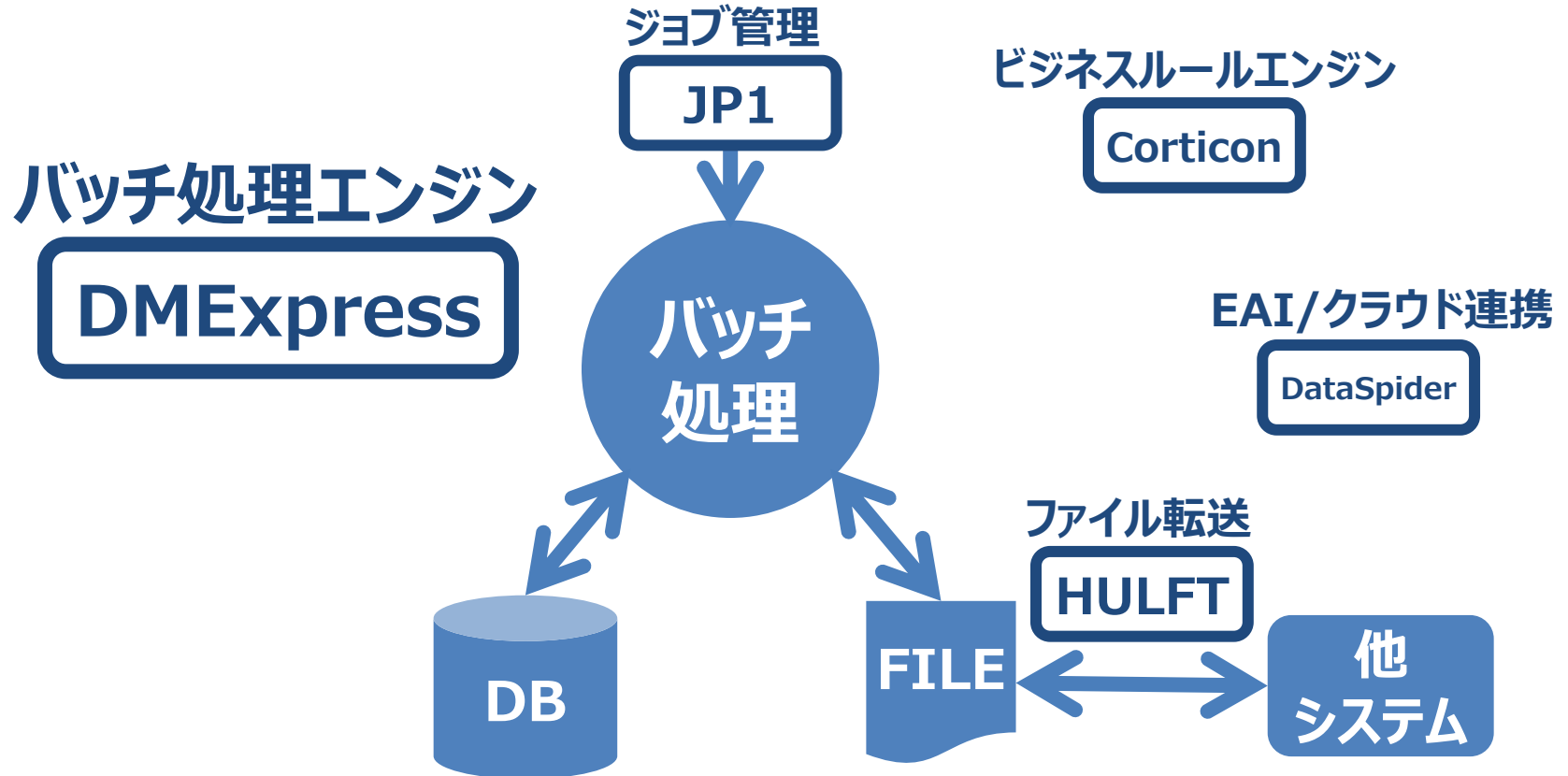
性能の作り込み、チューニングの
工数削減にも寄与するため、
バッチ処理システム開発における
全工数の 63% に対して
改善アプローチが可能

生産性 改善実績

#	お客様名	処理内容	比較対象	生産性
1	A G S 様	公共系システムのバッチ処理	C#	5倍
2	小島プレス工業様	生産系システムのバッチ処理	Java PL/SQL	4倍
3	データホライゾン様	レセプトデータの集計、 加工処理	スクラッチ	2倍
4	某製造メーカー様	生産管理システム周辺の システム間連携	Java	5倍
5	中電シーティーアイ様	データ抽出、ソート、分割の 多重処理	Java	3倍

5. 最後に

アシストがお役に立てるところ



処理方式をご検討の際に

バッチ処理システム構築における処理方式のご検討にDMExpressを加えていただき、お役立ちの機会をいただくと幸いです。



超|サ|ポ
愉|快|カ|ン|パ|ニ|ー
アシスト

※本資料に記載している情報は、2019年10月17日現在のものです。

※本資料の内容は、今後予告なく変更されることがあります。

※OracleとJavaは、Oracle Corporation 及びその子会社、関連会社の米国及びその他の国における登録商標です。

※文中の社名、商品名等は各社の商標または登録商標である場合があります。