

シェルスクリプトによる バッチ処理速度高速化の取組み

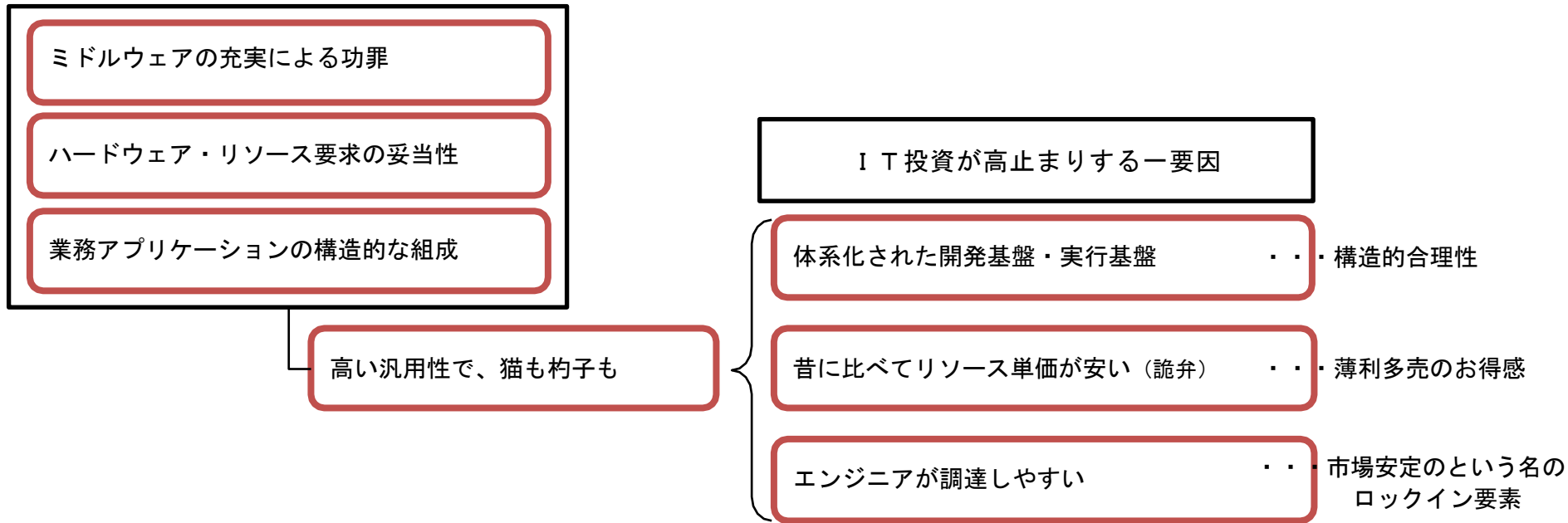
サインポスト株式会社

目次

1. システム開発を取り巻く I T 投資環境への考察
2. シェルスクリプトによる処理高速化とは
3. 実装による効果
4. 導入事例のご紹介

1. システム開発を取り巻く I T 投資環境への考察 (1 / 3)

システム投資、特にハードウェアとソフトウェア（ミドルウェア）へのイニシャルコスト／ランニングコストは、昨今の技術の進歩に比して低減されることなく「高止まり」と評価するのが妥当と考えられる。



これらの事象に対する具体的な施策は、「現行踏襲の暗黙の了解を排除」し、「RDBMSを使わずファイルシステムを使う」・「インタプリタ言語活用の領域を見いたす」など回帰的な方法を視野に入れたシステム化により、思い切ったコスト削減を行うことで効果を発揮すると思量。

※インタプリタ言語 . . . プログラミング言語で記述したソースコードを、コンピューターが実行できる機械語の形式（オブジェクトコード）へと逐次翻訳しながら、そのプログラムを実行してゆくプログラム

1. システム開発を取り巻く I T 投資環境への考察 (2 / 3)

前述の内容に照らし、投資の是正観点は（側面的に）以下のように整理される。

システム投資の 是正観点

そのミドルウェアは必要か？

アプリケーション開発の方法論に依存し、構造化されたミドルウェア群を導入して非効率な開発を助長している傾向も散見

HWリソースの用途を
理解して調達しているか？

その多くがミドルウェアを動作させるために使われ、さらに業務アプリケーションを動作させる構文解析機構（パーサー）などに使われていて、業務ロジックの実行に直接寄与するリソースは多くない。

JavaやCOBOLでないと
いけないか？

コンパイラ言語とインタプリタ言語の使い分けの有効性を考えていない

※コンパイラ言語・・・人間が書いたプログラムをコンピュータが理解できる形式に変換するプログラム言語のこと。コンパイルする前にコンピュータが読める形式に翻訳しておくため、実行速度が速い。

1. システム開発を取り巻く I T 投資環境への考察 (3 / 3)

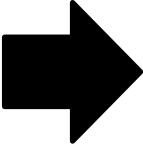
【弊社の試み】

以上の内容を受け、システム環境と業務処理の特性を評価したうえで、バッチ処理高速化の施策をお客様に提案している。

業務アプリケーション開発言語として、インタプリタであるシェルコマンドを集積した「シェルスクリプト」を採用。情報系システムのデータ処理に特化し開発する。OSに標準添付される基本コマンド群を駆使し製造する。

インタプリタ言語の弱点である低速処理は「tsukubai (つくばい)」と呼ばれる高速処理が得意な単機能シェルコマンドを組合せて処理速度を向上する。

インタプリタへの入出力処理特性に焦点を当てた開発方法論「ユニケージ」を適用し、データの組成を軸にロジック形成。ハードウェアに直接作用する記述仕様により、廉価な機器でも十分な処理性能が得られる。



弊社はバッチ処理高速化ソリューションを立ち上げ、「tsukubai」「ユニケージ」の創出元「ユニバーサル・シェル・プログラミング研究所 (USP 研究所) 様」と提携し金融業のお客様に「原点回帰」的なイノベーションをもたらす事業を推進中。

2. シェルスクリプトによる処理高速化とは

(1) シェルスクリプトとは

UNIX系オペレーティングシステム（OS）で標準実装されているコマンド（シェルコマンド）を組合せ、複数の処理をまとめて行うとき(バッチ処理)に使われる、OSが直接解釈・処理できるスクリプト。

(2) ソリューション

OS標準実装のシェルコマンドに加え、集計処理を高速かつ簡便に実行できるC言語のコマンドライブラリを提供します。独自かつ高度にチューニングしたコマンドは高速処理が得意です。

(3) 特徴

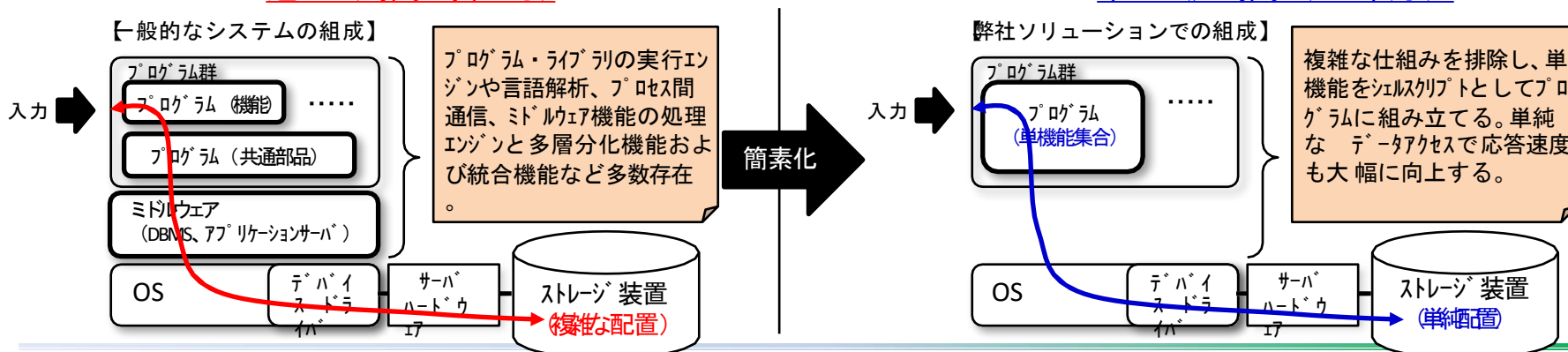
- UNIX/Linuxのシェルコマンドと同じ位置付けで利用できる。
- ミドルウェア（DBMS、アプリケーションサーバ）が不要。
- ハードウェアに直接作用し極めて高速な処理ができる（潜在力を使い切る）。

(4) 処理が速い理由

弊社のソリューションは、処理結果を応答するまでの演算量を劇的に削減しています。

遅い（時間が掛かる）

早い（短時間で処理終了）



3. 実装による効果

効果	詳細
コスト削減 (投資・経費)	<ul style="list-style-type: none">① データベースをはじめとした各種ミドルウェアが不要※1。 購入費・構築費などの投資費用、および保守経費を削減できる。② UNIXの標準的なコマンドを駆使しているため、ハード/OS更改の影響が軽微（移植性に優れる）。③ 本プログラムの稼働に高性能なハードウェアは不要。 本ソリューション適用により、CPU数や仮想化サーバ数の削減をはかることができる。
バッチ処理時間の短縮	<p>バッチ処理の所要時間が劇的に短縮するため、</p> <ul style="list-style-type: none">① 後続処理の開始時刻を早期化できる。② ユーザー様作業時間が捻出できる。③ 新しいビジネス創出のためのシステム処理時間を捻出できる。
保守性の向上	<ul style="list-style-type: none">① UNIXの基本的なコマンドを駆使しており、プログラムのステップ数も短くなるため、処理内容の把握が容易。② シェルコマンドとスクリプト記述の方法論を学び、保守の移管が行える。

※1 システム運用に必要なジョブスケジューラや監視ソフトウェア群は除く

4. 導入事例のご紹介（1 / 2）

以下、ユニバーサル・シェル・プログラミング研究所様（USP研究所）の実績。

- 東京電力株式会社
営業料金計算（稼働統計・基幹システム）
- 有限会社あきゅらいず
通販サイト構築、販売実績集計システム
- 公益財団法人流通経済研究所
データ分析サービス「マーケターズ・デスク」システム
- 生活協同組合連合会コープネット事業連合
宅配お届け明細（基幹システム）
物流統合システムの大量集品実績データのチェック及び加工処理
- 株式会社日本酒類販売
オープンCOBOLバッチ処理のマイグレーション
- 株式会社日本農業新聞
市況システム（読者向け市況情報提供サービス）
- 株式会社東急ハンズ
MD 基幹システム
- 株式会社イオンフォレスト（THEBODYSHOP）
全店舗販売状況集計・レポートシステム（BIツール）

他多数

4. 導入事例のご紹介（2 / 2）

<COBOLとの比較（某社様）>

	基幹処理（再構築対象）			ユニケー ジ 開発手法
	JCL	プログラム	計	シェル
本数	152	179	331	78
ステップ数	97,293			5,714
処理時間 （累計）	48:28:27			11:40:36

- ・ シェルスクリプトに置き換えることにより、**十分な効果**が期待できる。

ステップ数が削減 . . . 約17分の1
処理時間が短縮 . . . 約4分の1
開発工数が効率化 . . . 約20分の1



お客さまの強い味方となって、お客さまに貢献することが私たちの“喜び”です。

ありがとうございました。