



世界の COBOL 最新動向

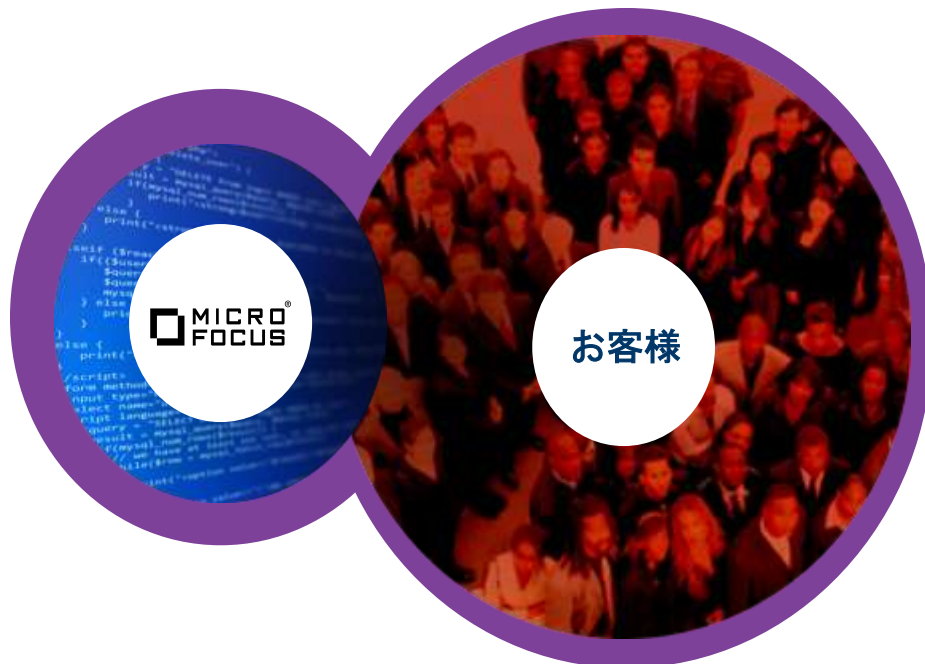
～ワールドワイドで実績豊富なコンパイラを使用して
クラウド、ビッグデータへ柔軟にキャッチアップ～

マイクロフォーカス株式会社

世界のビジネスアプリケーションをモダナイズする Micro Focus

- 1976年英国で設立（日本法人は1984年設立）
- オープンシステムターゲットのCOBOL製品でトップシェア
- ロンドン証券市場に上場

Annual revenues - 2012: \$434.8 million



- 全世界で約1,200名の従業員を持つグローバル企業
- 英国、米国、ドイツ、フランス、日本をはじめ世界中に34事業所

Micro Focus の歴史

COBOL



1976

世界初のマイクロプロセッサ
向けCOBOL

Micro Focus の歴史

1990年代

エンタープライズアプリケーション
をオープンな世界に開放する
ユニークなテクノロジー

モダナイゼーションと
マイグレーション

COBOL



Micro Focus の歴史

2009

アプリケーション開発プロセスを
モダナイズする品質向上ツール群

COBOL

モダナイゼーション
とマイグレーション



テスト・開発支援製品

製品の領域

COBOL

COBOLの開発と
運用における名高い実績



メインフレーム

600社以上での
成功事例

テスト・開発支援製品

ソフトウェア品質分野で
Gartnerの認めるリーダー

世界中のお客様に製品を供給

COBOL

分散コンピューティング企業
ITと独立系ソフトウェアベン
ダー



メインフレーム

フォーチュン 1,000
メインフレームユーザ

テスト・開発支援製品

全IT業界に及ぶ顧客層

最新の Micro Focus COBOL製品



Micro Focus Visual COBOL 2.1



開発環境製品

Micro Focus Visual COBOL 2.1

Windows版

Visual StudioとEclipse の両方を使用可

ネイティブ・.NET・JVMターゲットの開発

AIX, z/Linux, x64/x86 Linux, Solaris, HP-UX版

Windows上のEclipseによる分散開発

ネイティブ・JVMターゲットの開発

実行環境製品

Micro Focus COBOL Server 2.1

COBOL言語ランタイム＋基本運用ユーティリティ

Micro Focus COBOL Server for SOA 2.1

Java EE接続性、Webサービス展開の実行エンジンを含む

Enterprise Serverの 64-Bitモードをサポート

Micro Focus エンタープライズ・ソリューション



IBMメインフレームアプリケーション向けの開発・管理支援

お客様の目的

要求事項

Micro Focus製品

分析

モダナイゼーションの正しい優先度付け

アプリケーション構造の正確な理解複雑度を軽減

Enterprise Analyzer

開発

開発チームの効率化と品質向上

古い開発プロセスを排し、バックログを解消

Enterprise Developer

品質

タイムリーなアプリケーションサービス提供

サービス提供サイクルの改善

Enterprise Test Server

運用

最新のアーキテクチャの採用

低コストの基幹システム運用

Enterprise Server

アプリケーションサービス提供のモダナイゼーション

多くの企業が抱くゴール



標準開発ツールへの統合



+

Visual Studio

+

Eclipse

.NET

JVM

がスキルギャップをつなぐ！



- ✓ .NET や Java 開発に慣れ親しんだ開発者の取り込み
- ✓ 効率的な COBOL 言語の習得
- ✓ 開發生産性の向上
- ✓ 開発プラクティスの統合
- :
- ✓ 従来のMicro Focus COBOL 開発技術の利用
 - メインフレーム COBOL 方言、ACUCOBOL 方言等との互換
 - 強力なデータベースプログラミング機能
 - COBOL ファイル専用のデータツール

:

強力なポータビリティ



- 多様化するプラットフォームに対応
- ビジネスロジックの書き直し
不要
- 要件に応じた移行計画
が可能



Write Once, Run everywhere



スケーラブルで且つ費用対効果の高い環境でCOBOLを運用

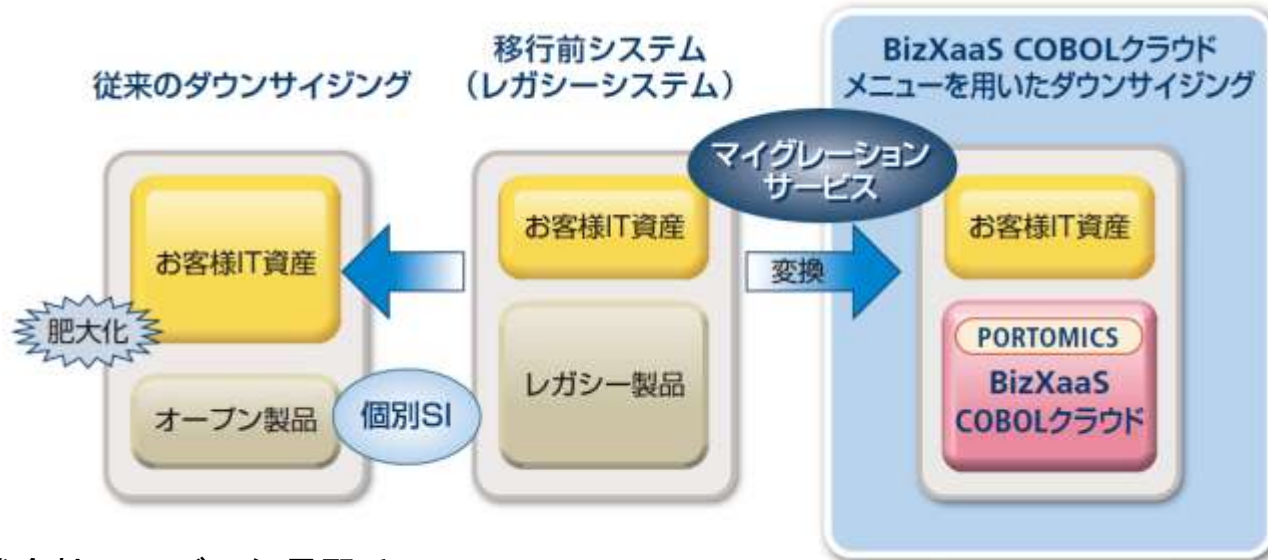


- ✓ Visual COBOL の豊富なサポート対象 OS 群より要件に応じて柔軟に計画
- ✓ Visual COBOL for Azure を使えば Windows Azure 向けアプリケーション開発が更に簡便に

Micro Focus 製品のクラウド利用事例

株式会社 NTT データ様『BizXaaS COBOL クラウドサービス』

主にメインフレーム上に構築されている IT 資産のマイグレーション先として
COBOL アプリケーションの実行環境をプライベートクラウドにて提供するサービス



株式会社 NTTデータ 星野氏:
選定理由について

「理由は単純明快で、グローバルベースで最もスタンダードなCOBOLといえば、
Micro Focus COBOLだから。選択するまでもなかった」。

IT Pro Special - COBOL 最新動向「クラウドでの COBOL 活用ソリューション」より引用
<http://special.nikkeibp.co.jp/ts/article/aa0e/109170/>

Visual COBOL for Azure

Windows Azure
が PDC 2008 で発
表されるのと同じ
に公開

- Azure クラウドサービスにおける開発
及び実行機能をサポート提供
 - ランタイム機能
 - デモプログラム
 - ヘルプ
 - Visual Studio IDE への統合
 - COBOL で開発するクラウドプロジェクトテンプレートを搭載
 - ASP.NET Web Role
 - WCF Service Web Role
 - Worker Role
 - VHD(Virtual Hard Drive) アップロードツール
- インストール要件
 - Windows Azure Tools for Visual Studio 1.6
 - Azure SDK を含む
 - IIS 7.0
- プラットフォーム
 - Windows 7, Windows Vista SP1 以上, Windows Server 2008

シナリオ

レガシーな自動車保険料試算システムを Azure にマイグレーション

メインルーチン

runv.exe - COBOL テキストウィンドウ

☆☆☆ 自家用普通乗用車年間保険料試算 ☆☆☆

主な使用目的: ☐ (K - 主に家庭用, G - 主に業務用)

契約距離区分: ☐ (今後一年間のお車の最大走行距離)
(1 - 3000km以下, 2 - 5000km以下, 3 - 7000km以下,
4 - 9000km以下, 5 - 11000km以下, 6 - 無制限)

記名保険者の生年月日: 年 月 日

記名被保険者の免許証の色: ☐ (1 - ゴールド, 2 - ブルー, 3 - グリーン)

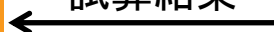
運転者の範囲: ☐ (1 - 本人のみ, 2 - 本人と配偶者, 3 - 家族のみ, 4 - 限定なし)

年間保険料試算額: 662,400 円

入力パラメータ



保険料の
試算結果

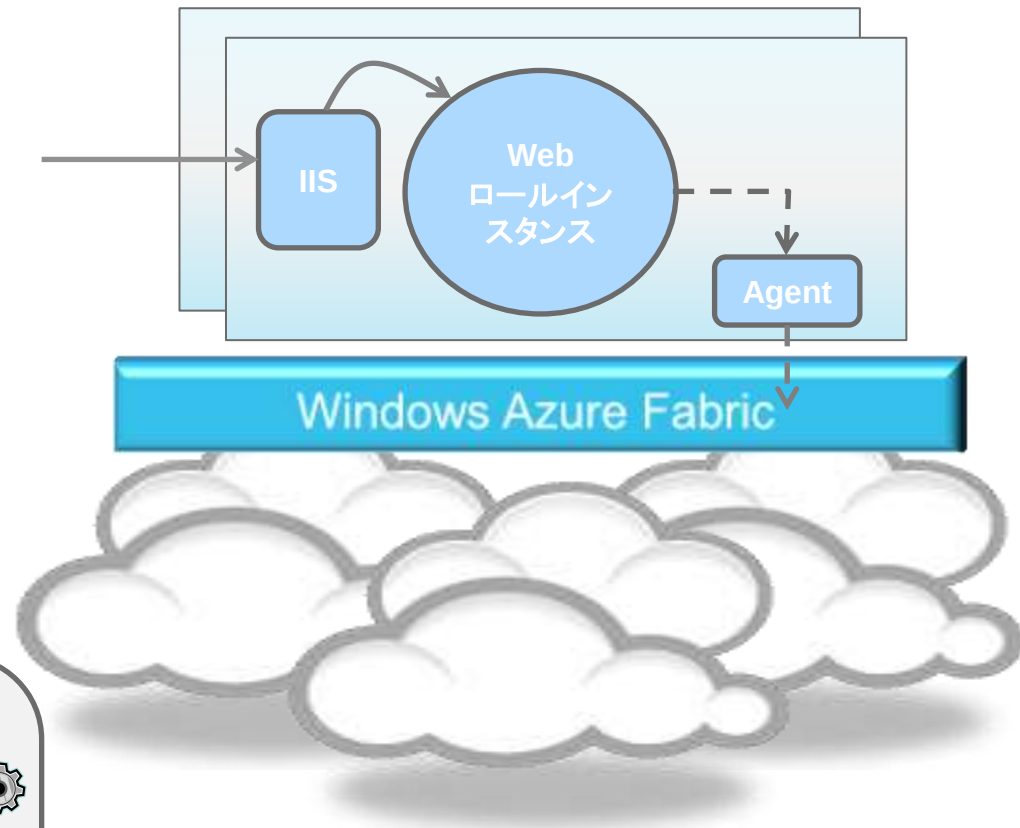
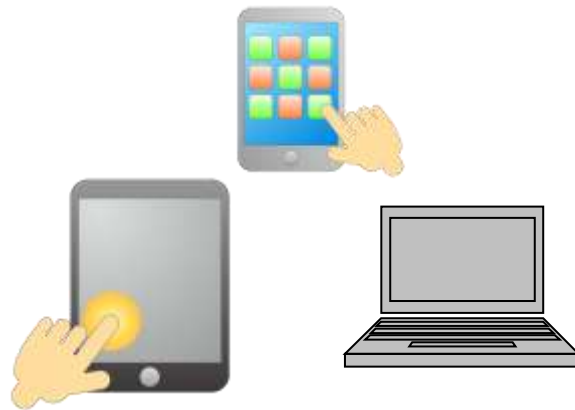


サブルーチン



```
1 000100*****
2 000200* システム I D : UXT038T
3 000300* システム名 : マイクロフォーカス構架保険
4 000400* 自動車保険料試算システム
5 000500* プログラム I D : UOT0026P
6 000600* プログラム名 : 自家用普通乗用車年間保険料試算
7 000700*****
8 000800* VER: 0 REV: 0 1998/04/27 初期リリース
9 000900* VER: 2 REV: 1 2005/02/14 T.YAMADA 免許証の色対応
10 001000*****
11 001100 IDENTIFICATION DIVISION.
12 001200 CLASS-ID UOT0026P public.
13 001300 ENVIRONMENT DIVISION.
14 001400 CONFIGURATION SECTION.
15 001500 SOURCE-COMPUTER IBM-390.
16 001600 OBJECT-COMPUTER IBM-390.
17 001700*
18 001800 INPUT-OUTPUT SECTION.
19 001900 FILE-CONTROL.
20 002000/
21 002100 DATA DIVISION.
22 002200/
23 002300 WORKING-STORAGE SECTION.
24 002400*
25 002500* << WORK-AREA >>
26 002600*
27 002700 01 WORK-AREA.
28 002800*
29 002900 03 HOKEN-RYOKUIN PTC 9(08) COMP-3.
30 003000* << その他作業項目 >>
31 003100 03 TODAY-X PTC 9(08).
32 003200 03 REDEFINES TODAY-X.
33 003300 05 TODAY-YYYY PTC 9(04).
34 003400 05 TODAY-MM PTC 9(02).
35 003500 05 TODAY-DD PTC 9(02).
36 003600 03 HK-NAME PTC 9(03).
37 003700 03 HK-PURPOSE PTC 9(02).
38 003800 03 HK-DISTANCE PTC 9(02).
39 003900 03 HK-YYYY PTC 9(04).
40 004000 03 HK-MM PTC 9(02).
41 004100 03 HK-DD PTC 9(02).
```

[デモ] Visual COBOL for Azure の利用



レガシーアプリケーション

メインルーチン

Visual COBOL for Azure Test Window

☆☆☆ 日本電気株式会社 年次報告書 ☆☆☆

主な使用目的: (1 - 主に業務用, 0 - 主に開発用)

契約締結区分: (1 - 100000以下, 2 - 100000以上, 3 - 100000以上, 4 - 100000以上, 5 - 100000以上, 6 - 無制限)

記名開始日の生年月日: 年 月 日

記名開始日の免許証の色: (1 - グリーン, 2 - ブルー, 3 - グリーン)

運転者の性別: (1 - 本人のみ, 2 - 本人と配偶者, 3 - 家族のみ, 4 - 無制限)

年間報告書提出額: 円

サブルーチン

```
1 000000 *****  
2 000000 システム ID : 1000000  
3 000000 プログラム名 : 日本電気株式会社 年次報告書  
4 000000 プログラム ID : 1000000  
5 000000 プログラムバージョン : 1.0  
6 000000 開発者 : 日本電気株式会社 年次報告書  
7 000000 開発者 ID : 1000000  
8 000000 開発者バージョン : 1.0  
9 000000 開発者コメント : 1000000  
10 000000 開発者コメントバージョン : 1.0  
11 000000 IDENTIFICATION DIVISION.  
12 000000 PROGRAM IDENTIFICATION.  
13 000000 PROGRAM NAME.  
14 000000 PROGRAM VERSION.  
15 000000 PROGRAM COMMENTS.  
16 000000 PROGRAM COMMENTS VERSION.  
17 000000 PROGRAM COMMENTS.  
18 000000 PROGRAM COMMENTS VERSION.  
19 000000 PROGRAM COMMENTS.  
20 000000 PROGRAM COMMENTS VERSION.  
21 000000 PROGRAM COMMENTS.  
22 000000 PROGRAM COMMENTS VERSION.  
23 000000 WORKING-STORAGE SECTION.  
24 000000  
25 000000  
26 000000  
27 000000  
28 000000  
29 000000  
30 000000  
31 000000  
32 000000  
33 000000  
34 000000  
35 000000  
36 000000  
37 000000  
38 000000  
39 000000  
40 000000  
41 000000
```

Visual COBOL を使ったモダナイゼーションのモデル

プレゼンテーション

- COBOL, Java, C#, VB...

ビジネスアクセス

- COBOL ラッパー
- Run unit

ビジネスロジック

- レガシーで稼働中の手続き型 COBOL

データアクセス

- 分離データ IO モジュール

データ

- リレーショナルデータベース
- COBOL データファイル

[デモ] Visual COBOL for Azure の利用



ビッグデータの活用



ソーシャルメディア

- ・意見
- ・好み
- ・ニーズ、要求 等

行動データ

- ・走行データ(GPS)
- ・センサーデータ
- ・支払履歴 等

Interaction データ

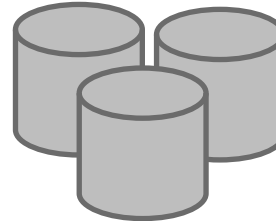
- ・Web クリックストリーム
- ・検索
- ・E-mail/チャット履歴 等

構造化 データ

日々の業務実績等を RDB
やCOBOL データファイル
に格納されたデータ

ビッグデータ処理

データの
収集・蓄積



統計学手法を
使って分析



活用例

需要管理

決済管理

運行管理

セキュリティ
管理

健康・医療
管理

ビッグデータを収集し分析可能な状態で蓄積させるまでの技術

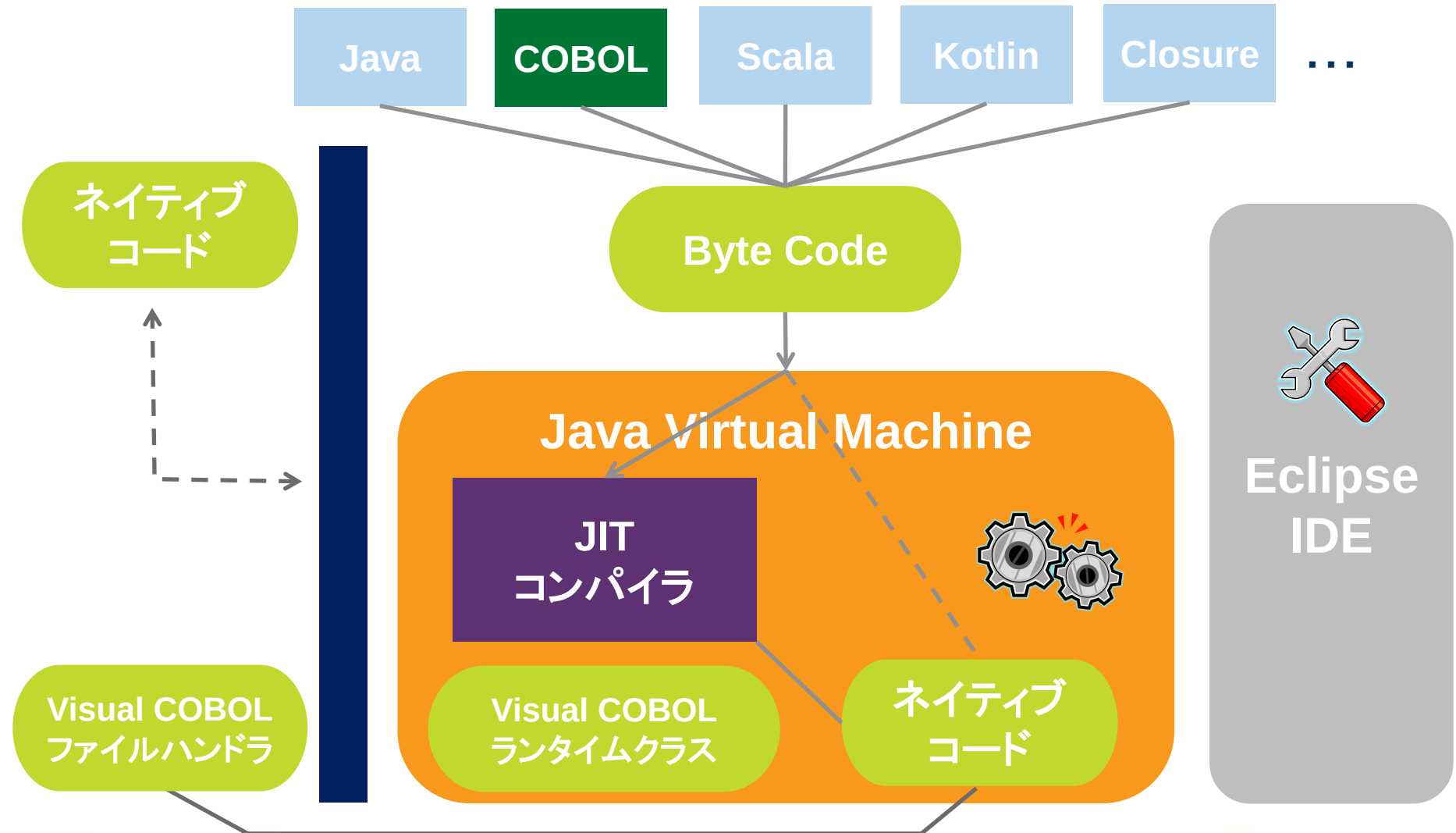
アクセスする API	操作に使われる主なテクノロジー
Twitter Streaming API	Web サービス、Java, Python, Ruby
Google アナリティックス API	Java, Python, PHP
Facebook Graph API	JavaScript, PHP

処理するフレームワーク	操作に使われる主なテクノロジー
Apache Hadoop	Java, C++, Python, PHP, Ruby
Greenplum MapReduce	Perl, Python
Aster Data SQL-MapReduce	Java, C#, Python, C++, R

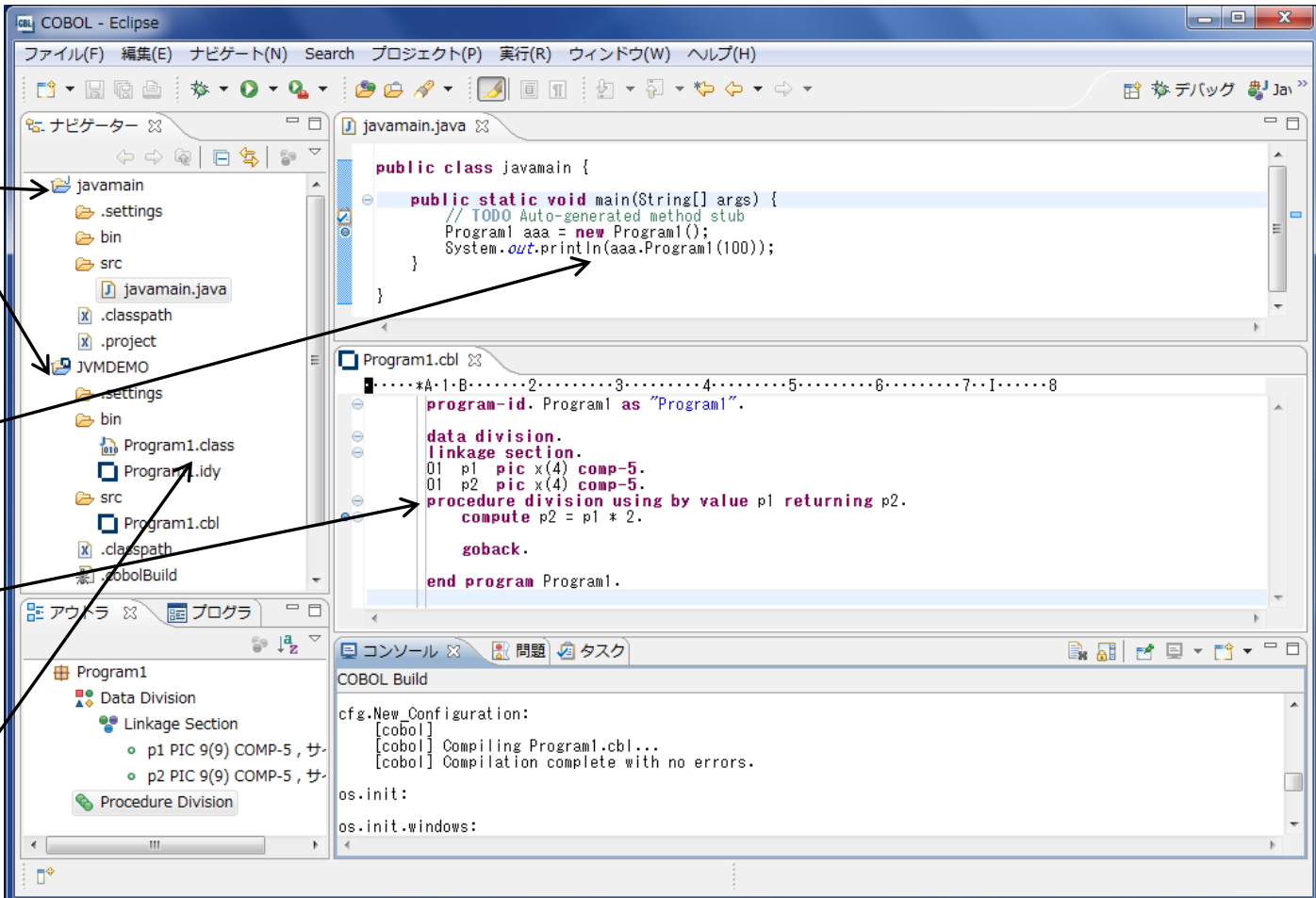


多くの技術は、COBOL for JVM で利用可能

COBOL for JVM 運用環境



COBOL for JVM 開発環境



The screenshot shows the Eclipse IDE interface for COBOL. The left sidebar displays the project structure with folders like .settings, bin, src, and files like javamain.java, Program1.class, Program1.idy, Program1.cbl, .classpath, and .project. The main editor shows the source code for javamain.java and Program1.cbl. The bottom panel shows the COBOL Build output window.

Java/COBOL 混在ワークスペース (Mixed Java/COBOL Workspace): Points to the project structure in the left sidebar.

手続き型COBOLの自動クラスメソッド化 (Automatic class methodization of procedural COBOL): Points to the Program1.cbl file in the left sidebar.

Java/COBOL 間データ授受 (Data exchange between Java and COBOL): Points to the Program1.class file in the left sidebar.

.class/.jar へのコンパイル (Compilation to .class/.jar): Points to the Program1.class file in the left sidebar.

javamain.java

```
public class javamain {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Program1 aaa = new Program1();
        System.out.println(aaa.Program1(100));
    }
}
```

Program1.cbl

```
.....*A..1..B.....2.....3.....4.....5.....6.....7..I.....8
program-id. Program1 as "Program1".

data division.
linkage section.
01 p1 pic x(4) comp-5.
01 p2 pic x(4) comp-5.
procedure division using by value p1 returning p2.
    compute p2 = p1 * 2.
    goback.
end program Program1.
```

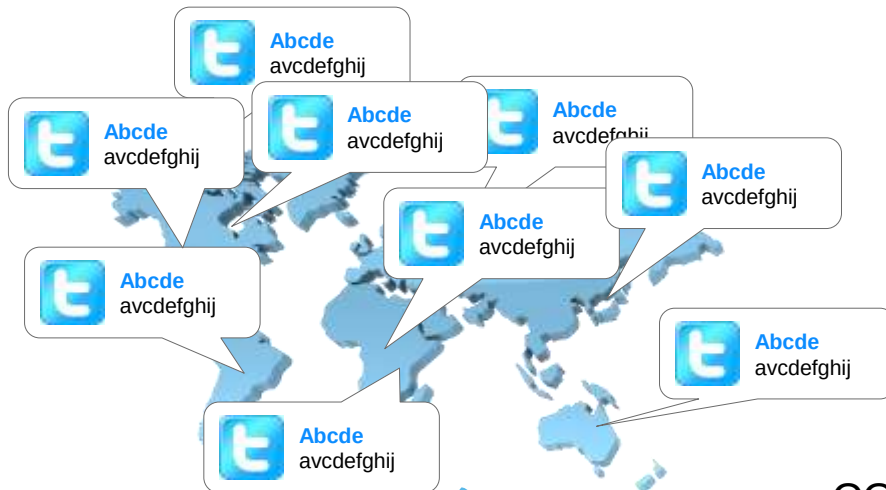
COBOL Build

```
cfg.New Configuration:
[cobol]
[cobol] Compiling Program1.cbl...
[cobol] Compilation complete with no errors.

os.init:
os.init.windows:
```


[デモ] Visual COBOL によるビッグデータの収集

twitter からデータを収集



Twitter4J
(Twitter API の Java ラッパ)

COBOL プログラム

```

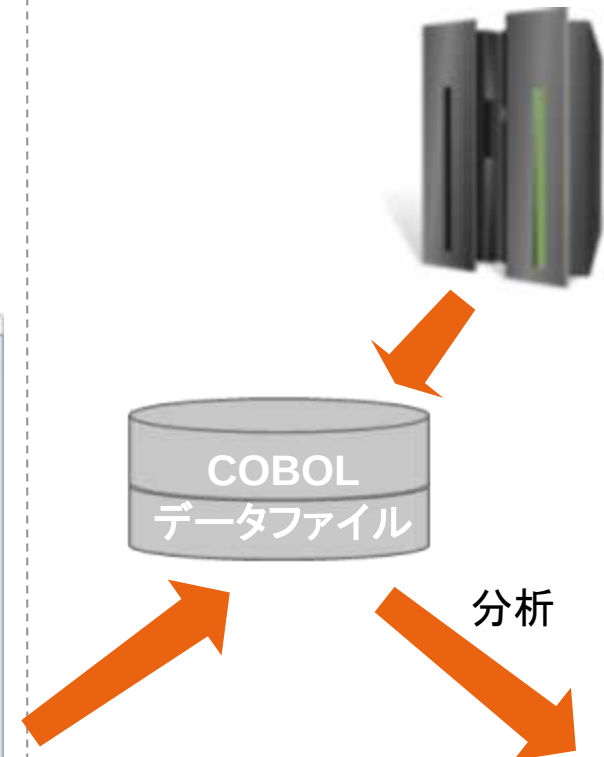
1. TweetSearchMain.cbl
2. SearchTweets.cbl

DO-QUERY.
  INITIALIZE RET-VAL-THL.
  PERFORM VARYING P-QUERY-IDX FROM 1 BY 1 UNTIL P-QUERY-IDX > 2
  PERFORM VARYING P-LANG-IDX FROM 1 BY 1 UNTIL P-LANG-IDX > 1
  SET RET-VAL(P-QUERY-IDX, P-LANG-IDX)
  TO SEARCHMENTS-REF:performQuery(P-QUERY-STR(P-
  RET-VAL(P-QUERY-IDX, P-LANG-IDX) TO RES-VAL-DISP(P-Q
  END-PERFORM.
  EXIT.

SET-QUERY-STR.
  DISPLAY "検索キーワードを入力してください(1単語): " WITH NO ADVANC
  ACCEPT QUERY-STR(1).
  DISPLAY "検索対象の言語を入力してください(2単語): " WITH NO ADVANC
  ACCEPT QUERY-STR(2).
  EXIT.

DISP-RES.
  PERFORM VARYING QUERY-IDX FROM 1 BY 1 UNTIL QUERY-IDX > 2
  DISPLAY "***** 検索結果による tweet *****"
  DISPLAY "***** 検索対象の言語 *****"
  PERFORM VARYING RES-IDX FROM 1 BY 1 UNTIL RES-IDX > 10
  DISPLAY " " RES-VAL-DISP(QUERY-IDX, RES-IDX) " "
  FULL-NAME(RES-IDX)
  END-PERFORM
  DISPLAY " "
  
```

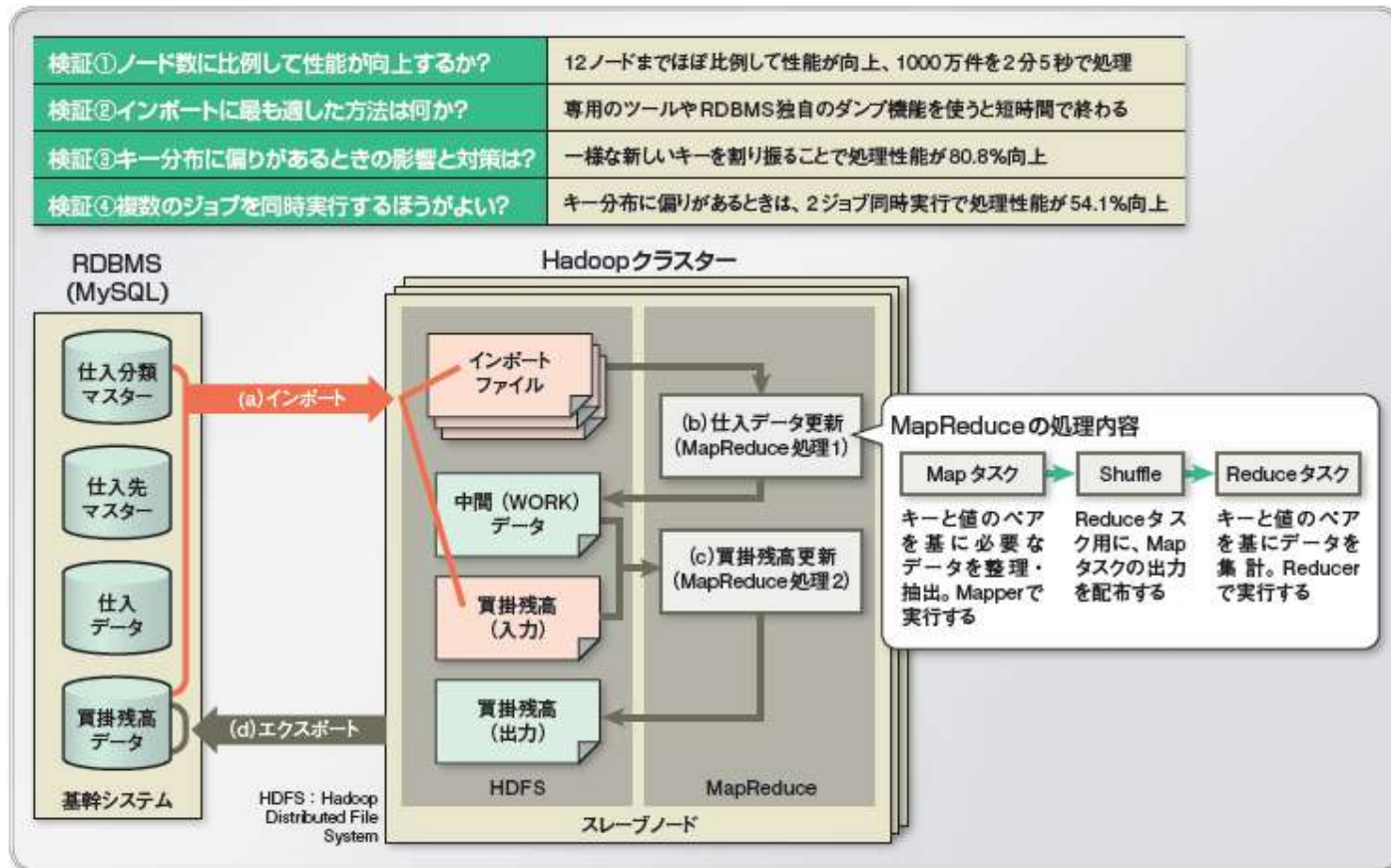
[実運用を想定した応用イメージ]



[デモ] Visual COBOL によるビッグデータの収集



データの蓄積 - Hadoop の活用例



出典: 株式会社日経 BP「IT Pro」

検証ラボ - 連載 Hadoop—企業システムでの実力は？

「1000万件のバッチを2分で実行 ノード数に比例して性能が向上」

<http://itpro.nikkeibp.co.jp/article/COLUMN/20110707/362172/>

[デモ] COBOL + Hadoop で高速にバッチ処理

Hadoop を使って処理可能な状態で蓄積

COBOL ファイル – 行順編成
受注データ

注文番号1	明細2
注文番号3	明細1
注文番号2	明細2
注文番号3	明細2
注文番号2	明細1
注文番号1	明細1
注文番号3	明細3

Mapper

(注文番号1, 明細2)
(注文番号3, 明細1)
(注文番号2, 明細2)
(注文番号3, 明細2)
(注文番号2, 明細1)
(注文番号1, 明細1)
(注文番号3, 明細3)

Shuffle

(注文番号1, [明細1、明細2])
(注文番号2, [明細1、明細2])
(注文番号3, [明細1、明細2、明細3])

Reducer

注文番号1	詳細
注文番号2	詳細
注文番号3	詳細

顧客名(姓)	顧客名(名)	注文毎の合計金額
--------	--------	----------

顧客名(姓)	顧客名(名)	製品名	製品数量	製品単価
--------	--------	-----	------	------

[デモ] COBOL + Hadoop で高速にバッチ処理

Hadoop を使って処理可能な状態で蓄積

Mapper

固定長のレコード毎に、KEY の部分と VALUE の部分に分割

Shuffle

KEY 値毎に VALUE を集めたり
ストを作成

KEY でソート

10進演算、固定長
データ操作は
COBOL が得意と
するところであり、
COBOL
で処理させたい。

Reducer

VALUE から固定長の各エレメントを抽出

KEY 値毎に合計金額を算出

$$\sum_{i=1}^n (\text{数量}_i \times \text{製品単価}_i)$$

[デモ] COBOL + Hadoop で高速にバッチ処理



Visual COBOL で ∞ の可能性へ



ご清聴ありがとうございました。

◆ 技術セミナー開催情報

「COBOL言語超入門」：4月26日（金）／ 7月12日（金）

他の言語での開発経験はあるがCOBOLは初めて、という方に、
COBOL言語を使用するシステム開発のための基礎知識を提供します。

「COBOL- .NETプログラミング言語の新たな選択肢」：6月7日（金）

.NET Frameworkをターゲットとしたシステム開発においてCOBOLを活用したいとお考えの方に、
プログラミングレベルのデモを含めて.NET 環境でのCOBOL活用手法を解説します。

「COBOLコンポジットアプリケーション開発」：6月21日（金）

コンポジットアプリケーション開発において、COBOL既存資産をサービス化して活用したい
とお考えの方に、Micro Focus製品による、Java EEコネクタ、Webサービスの有効活用法を、
プログラミングレベルのデモを含めて解説します。

セミナー詳細とお申込みはこちら；

<http://www.microfocus.co.jp/resources/events/COBOL/>

◆ 製品・サービスに関するお問い合わせ先

マイクロフォーカス株式会社 営業部

Tel: 0120-20-9612

e-mail: sales@microfocus.co.jp

[http: //www.microfocus.co.jp/](http://www.microfocus.co.jp/)