

スケールアウトパフォーマンス / 可用性クラスタの活用

企業経営の主要な業務を支える基幹システムは、日々変化するマーケットのニーズに対応できる柔軟な環境や迅速にリリースできるプロセスの確立に加え、システム停止によるビジネスへの影響を回避するために、安定した稼働を強く求められます。

基幹システムの高可用性を実現する方法の1つとしてクラスタリングソフトを使用したシステムの冗長化が挙げられますが、これは物理的障害などが発生した場合に、アクティブサーバーとスタンバイサーバーを自動的に切り替え、システムの停止を回避するものです。

加えて、実行環境のスペックを増強する「スケールアップ」、実行環境を複数用意して処理の負荷を分散させる「スケールアウト」は、想定外の負荷による大幅な処理の遅延や、その結果としてのシステム停止を未然に防ぐ対策として、コンテナのオーケストレーションサービスやクラウドにおいても広く利用されています。

さらに、基幹システムに関連するすべてのソフトウェアが個々に高可用性を意識した機能を持ち合わせていれば、幾重にも対策を施すことができ、安定したシステムを提供することが可能になります。

Enterprise Developer / Enterprise Server はスケールアウトパフォーマンス/可用性クラスタ機能をデフォルトで提供しており、複数の Enterprise Server インスタンスを単一システムイメージで運用、管理することができます。この構成を取り入れることにより、JCL 処理は負荷分散され、単一障害点を最小限に抑えることができます。

本文書では、Enterprise Developer / Enterprise Server 製品を使用したスケールアウトパフォーマンス/可用性クラスタを実現する方法について解説します。

目次

1. モダナイゼーション支援製品	1
2. 開発環境製品と実行環境製品の違い	1
3. モダナイゼーション支援製品の機能分布	1
4. スケールアウトパフォーマンス/可用性クラスターの目的と効果	2
1) 目的	2
2) 効果	3
5. PAC を取り巻く構成	4
1) インメモリデータストア	4
2) RDBMS	5
3) ロードバランサ、クラスタリングソフト	5
4) PAC、SOR、PSOR	5
6. PAC のコンポーネント	6
1) Enterprise Server インスタンスの可用性	6
2) システムデータの可用性	6
3) データファイルの可用性	6
4) システムのセキュリティ	6
5) SOR の負荷分散	7
7. PAC 利用の前提条件	7
1) OS とサードパーティ製品	7
2) サポートされる構成と制限事項	7
3) PAC メンバーの条件	7
8. デモ環境	8
1) OS	8
2) プロセッサ	8
3) システムの種類	8
4) 各環境の名前解決	8
5) ユーザー権限	8
6) MFDS サービスの開始設定	8
7) 共有フォルダ	9
8) インメモリデータストア	9
9) MFDBFH	9
10) XA スイッチモジュール	9
11) ODBC	9
12) 稼働ビット数	10

1 3)	データ文字コード	10
1 4)	管理画面の表示文字コード	10
1 5)	Micro Focus 製品名とバージョン	10
1 6)	参照した Micro Focus 製品マニュアルバージョン	10
9.	PAC の構成手順	10
1)	MFDBFH 構成ファイルの作成	11
2)	インメモリデータストアの開始	11
3)	Enterprise Server インスタンスの設定	12
4)	SOR の作成	15
5)	PAC の作成	16
10.	Enterprise Server インスタンスの開始	17
1)	Enterprise Server インスタンス開始 : 1 回目	17
2)	PAC の初期化	18
3)	Enterprise Server インスタンス開始 : 2 回目以降	18
11.	JCL の実行	19
1)	JCL の内容	19
2)	JCL の実行	20
3)	実行記録の確認	20
4)	MFDBFH データストアの確認	22
12.	CICS の実行	22
1)	HACloud セッションサーバーの起動	22
2)	CICS 処理の実行	24
13.	おわりに	26

各機能の詳細に関しては、製品マニュアルページからご利用になるバージョンを選択後、内容をご確認ください。

<https://www.microfocus.co.jp/>

[サポート] > [COBOL・エンタープライズ製品のカスタマーケア : 詳細をみる] > [製品マニュアル]

1. モダナイゼーション支援製品

マイクロフォーカスのモダナイゼーション支援製品は COBOL 製品群とエンタープライズ製品群の2つに分類されています。本文書ではエンタープライズ製品群に含まれる開発環境製品の Enterprise Developer と、実行環境製品の Enterprise Server に焦点を当てて解説します。

2. 開発環境製品と実行環境製品の違い

開発環境製品の Enterprise Developer はコンパイラとランタイムを含み、開発用マシンにインストールする製品です。業界標準 IDE の Eclipse、Visual Studio、Visual Studio Code を利用した開発作業や、開発用の実行環境である Enterprise Server 機能により、開発工程における単体テストや JCL の実行、IDE を利用したデバッグなど、効率的なテストを行うことができます。

一方、実行環境製品の Enterprise Server はランタイムのみを含み、本番用マシンにインストールする製品です。実行単位となる Enterprise Server インスタンスは、例えば JCL を対象としたバッチ用、IMS, CICS オンライン用、Web サービス用など、運用用途に合わせて複数作成することができます。

【製品の配置イメージ】

開発用マシン : Enterprise Developer

COBOL, PL/I コンパイラ

COBOL, PL/I ランタイム
開発用 Enterprise Server 機能

Enterprise Server インスタンス一覧

CTGDEMO	Region	Stopped	✓	✓
IMSDemo	Region	Stopped	✓	✓
JCLDEM60	Region	Stopped	✓	✓
PLUJCL64	Region	Stopped	✓	✓
CICSDEMO	Region	Stopped	✓	✓

本番用マシン : Enterprise Server

COBOL, PL/I ランタイム
本番用 Enterprise Server 機能

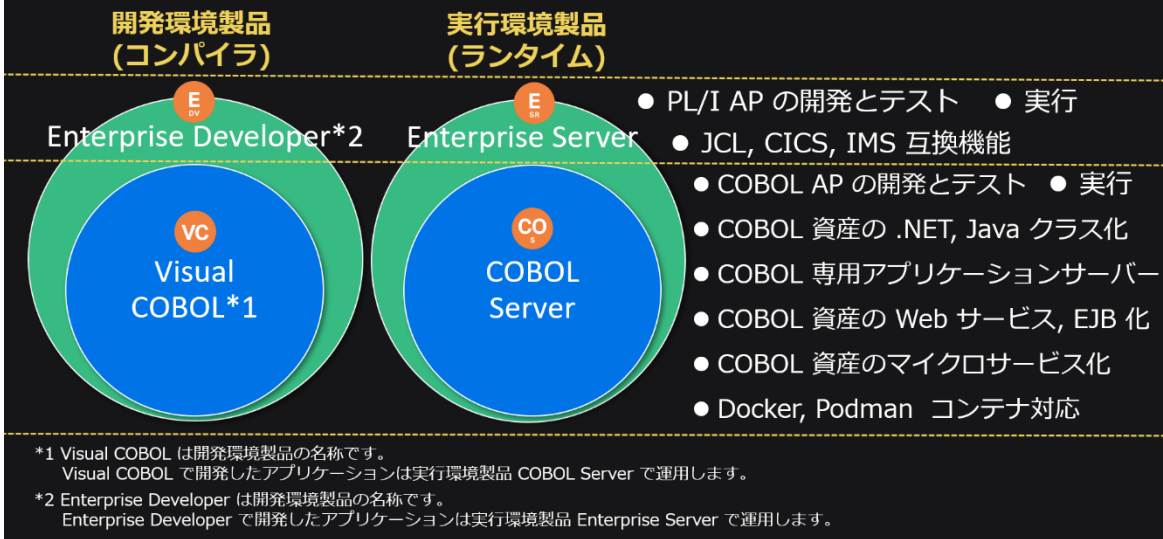
Enterprise Server インスタンス一覧

CTGDEMO	Region	Stopped	✓	✓
IMSDemo	Region	Stopped	✓	✓
JCLDEM60	Region	Stopped	✓	✓
PLUJCL64	Region	Stopped	✓	✓
CICSDEMO	Region	Stopped	✓	✓

3. モダナイゼーション支援製品の機能分布

エンタープライズ製品は COBOL, PL/I 言語のサポートや、JCL, CICS, IMS をエミュレートする機能を持ち IBM メインフレームからのリホストに最適な製品です。また、COBOL 製品の全機能を含む上位製品であることから、リホストに留まることなく最新技術と連携したモダナイゼーションを継続して実現することができます。

モダナイゼーション支援製品の機能分布



COBOL 製品は COBOL のビジネスロジックを様々な技術と連携させて有効活用できる製品です。

COBOL ソースはそのまま、ネイティブコードはもちろんのこと、マネージコードである Java クラスと .NET クラスへコンパイルすることができ、それぞれ JVM と .NET フレームワーク上で稼働させることができます。ネイティブコードは、製品に含まれている COBOL 専用のアプリケーションサーバーを使用した REST、SOAP 形式の Web サービスや EJB 連携を実現することができます。

両製品群とも、クラウド上でマイクロサービスを導入する際に欠かせないコンテナ型仮想化にも対応しています。

エンタープライズ製品の機能概要については[こちら](#)をご覧ください。

COBOL 製品の機能概要については[こちら](#)をご覧ください。

4. スケールアウトパフォーマンス/可用性クラスターの目的と効果

1) 目的

基幹システムの停止はビジネスに甚大な影響を与え、企業の信頼喪失につながることから、常に安定した稼働を強く求められています。オープン環境においてもサーバーの物理的な障害、ネットワークの切断、ソフトウェアによる障害など様々な問題に備える必要があり、複数の視点から目的に応じたツールや回避策を施すこととなります。

このようなニーズから、Enterprise Developer と Enterprise Server 製品にはスケールアウトパフォーマンス/可用性クラスター（以降 PAC と称す）機能がデフォルトで含まれており、これを利用することによりシステムの高可用性を実現します。

2) 効果

PAC 機能を利用すると、次に挙げるようなシステムの高可用性を実現できます。

① 負荷分散

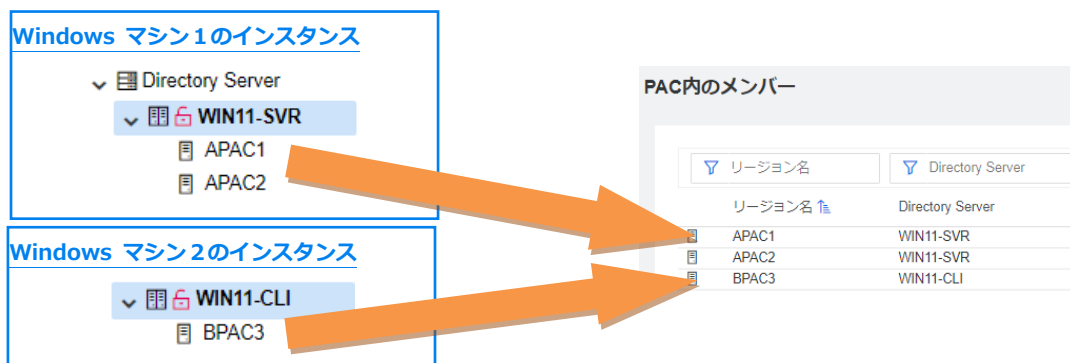
複数の Enterprise Server インスタンスを PAC 構成に参加させ、単一システムのイメージで処理の負荷分散を実現します。PAC 構成に登録された Enterprise Server インスタンスを PAC メンバーと呼びます。

例えば JCL を使用したバッチ処理では、自動的に PAC メンバーの空いているイニシエータヘジョブが割り振られ、効率的に処理を実行することができます。

② 単一障害点の回避

稼働させる OS や製品バージョン、稼働ビット数が一致すれば、IP アドレスの異なる複数のマシンに存在する Enterprise Server インスタンスを PAC メンバーに加えることができ、ハードウェアやネットワークに起因する単一障害点を最小限に抑えることができます。

2つの Windows 環境から PAC メンバーに指定した例)



③ ファイルの共有と制御

PAC メンバー間で JES のカタログファイル、スプールファイル、CICS のリソース定義ファイル、処理に使用する VSAM などのデータファイルが共有でき、排他制御が行われます。

これらのファイルを Micro Focus Database File Handler (以降 MFDBFH と称す)を利用して、RDBMS へ格納することもでき、データの可用性を高めることもできます。

④ ロードモジュールの共有

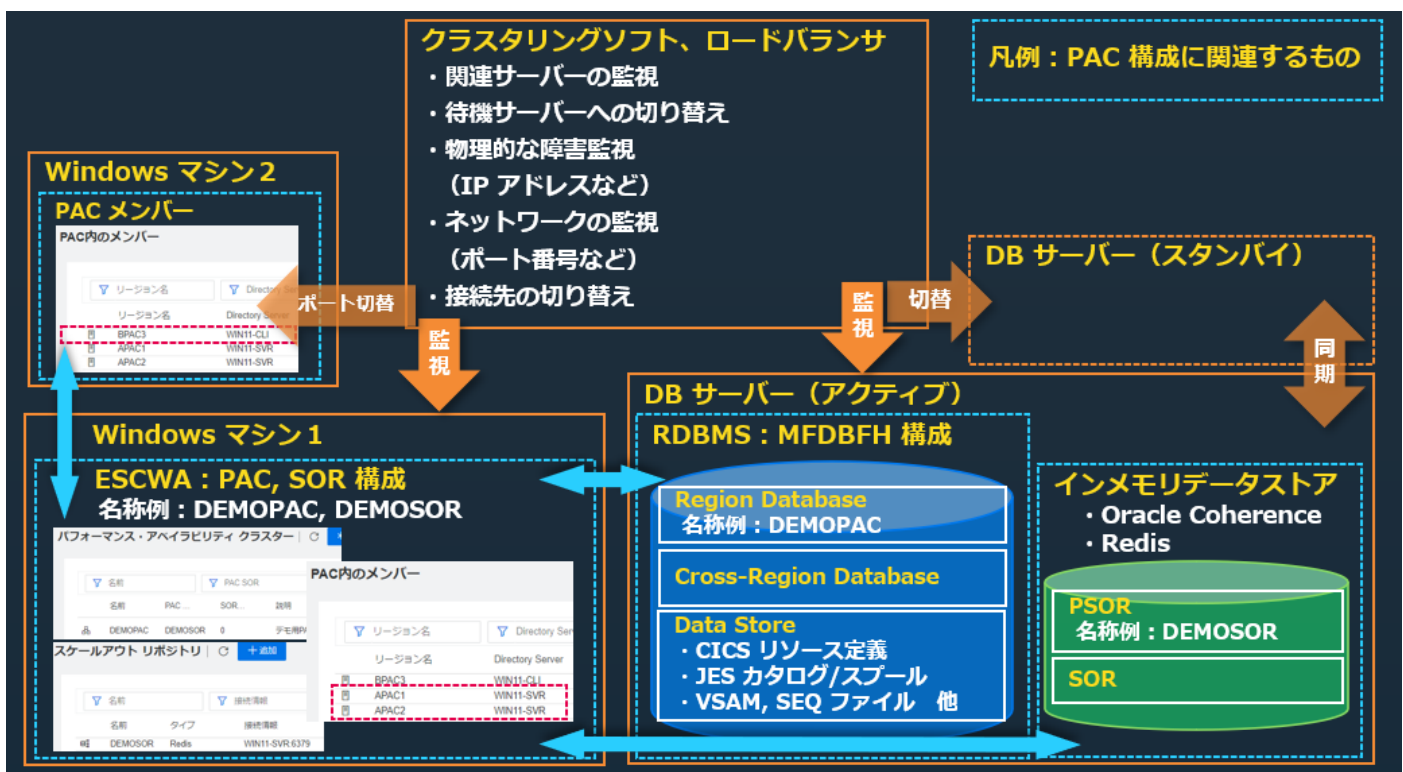
PAC メンバー間で実行可能ファイルであるロードモジュールを共有することができます。例えば1つの Enterprise Server インスタンスに対して CICS NEWCOPY を行った場合には、指定したプログラムが使用中でないことを前提に新しいモジュールがロードされ、すべての PAC メンバーへ伝播されます。

5. PAC を取り巻く構成

PAC 機能では、PAC メンバーの同期と管理のために必要となる構成要素があります。加えて、必要に応じて準備する製品以外のツールについても説明します。

PAC 構成図の例)

- ・ DB サーバーに RDBMS と インメモリデータストアを配置
- ・ Windows マシン 1 の共通管理画面である Enterprise Server Common Web Administration (以降 ESCWA と称す) で PAC と SOR を構築
- ・ Windows マシン 1 と Windows マシン 2 の Enterprise Server インスタンスを PAC メンバーに登録



1) インメモリデータストア

PAC メンバー間の同期を実現するためにインメモリデータストアを利用したスケールアウトレポジトリ (以降 SOR と称す) が必要です。PAC 構成に紐づけられた SOR を PAC スケールアウトレポジトリ (以降 PSOR と称す) と呼びます。

製品には開発、テスト目的で使用可能なインメモリデータストアの Redis が内包されていますが、本番環境や本番相当のテスト環境では Oracle Coherence の高可用性機能や Redis Cluster、Redis Sentinel の導入を計画し、PAC 構成全体の高可用性を確保することを強くお勧めします。

2) RDBMS

PAC メンバーを管理するために、RDBMS に MFDBFH を構築する必要があります。サポートされている RDBMS を製品マニュアルでご確認後、別途ご準備ください。

PAC メンバーからのアクセスは接続文字列または ODBC 経由となります。

RDBMS の機能を利用したスタンバイ環境との同期などを計画し、PAC 構成全体の高可用性を確保することを強くお勧めします。

3) ロードバランサ、クラスタリングソフト

一般的にクラスタリングソフトは物理的な障害や OS レベルの障害によりサーバー自体がダウンした場合にスタンバイ環境へ切り替えるもの、ロードバランサは処理の負荷分散とポートなどの死活監視、障害時の接続先切り替えを目的に導入するもので、これら機能の一部である業務処理の負荷分散をエンタープライズ製品の PAC 機能が提供しています。

基幹システムの停止を防ぐためには、サーバーのダウンやネットワーク障害などに備えてスタンバイ環境と同期とる、接続先を切り替えるなど複数の対策を施す必要があります、これらを実現する場合は別途ロードバランサやクラスタリングソフトをご準備ください。

PAC 構成では、物理的に異なるマシンにある Enterprise Server インスタンスを PAC メンバーとすることができ、かつメンバー間でファイルを共有することから、ロードバランサが障害を検知した場合は、稼働している PAC メンバーに接続先を切り替えて業務処理を続行することもできます。

4) PAC、SOR、PSOR

PAC を構成するために、1つのマシンの ESCWA を使用して PAC や SOR、PSOR を登録しますが、例えばマシン1の ESCWA からこれらを登録し、マシン1がダウンした場合でも、他マシンの PAC メンバーが起動中であれば PAC 機能は有効であり、処理の負荷分散は継続されます。

また、PAC、SOR を登録していないマシンであっても、他のマシンで PAC メンバーに指定されている Enterprise Server インスタンスが存在する場合は PAC メンバーとして起動されます。

PAC、SOR を登録していないマシン)



コンソールログには PAC メンバーとして起動されたことが記録)

CASMG0001I		APAC2(002)#CASSI1435I 00000 Other module(s) were synchronized from PAC DEMOPAC CasLOADS. 16:16:10#
CASMG0001I		APAC2(002)#CASSI1435I 00000 CICS module(s) were synchronized from PAC DEMOPAC CasLOADS. 16:16:10#
CASMG0001I		APAC2(002)#CASSI1435I 00000 JES module(s) were synchronized from PAC DEMOPAC CasLOADS. 16:16:10#
CASMG0001I		APAC2(002)#CASSI1432I Joining PAC DEMOPAC. 16:16:12#

PAC のソフトウェア要件は、ご利用になるバージョンの製品マニュアルを開き「ここからはじめよう > 製品情報 > Enterprise Developer のインストール > 環境別の Readme」に含まれる「パフォーマンス・アベイラビリティ クラスター (PACs) のソフトウェア要件」欄をご参照ください。

6. PAC のコンポーネント

PAC 構成には下記の項目を目的として、必須コンポーネントと任意に構成できるオプションコンポーネントがあります。

1) Enterprise Server インスタンスの可用性

PAC 構成には Enterprise Server インスタンスの指定が必須です。何かの障害により1つのインスタンスが停止した際にも可用性を保持するために、3つ以上のインスタンスを PAC メンバーにすることを推奨しています。また、メンバーの同期に必要な PSOR を PAC 構成に対して必ず1つ指定します。

2) システムデータの可用性

PAC メンバー間におけるシステムデータの共有と管理のために、MFDBFH を利用したリージョンデータベースとリージョン間データベースの構成が必須です。

3) データファイルの可用性

データの可用性を高めるために、JES のカタログファイルや CICS のリソース定義ファイル、データファイルなどを MFDBFH を利用して RDBMS へ格納し、管理することができます。

この構成は必須ではありません。

4) システムのセキュリティ

PSOR や SOR にアクセスするクライアントに対して、認証を要求することができます。

この構成は必須ではありません。

5) SOR の負荷分散

CICS TD、TS キューのデータを別の SOR に格納することができます。

これにより特定データのアクセスに対する負荷を分散することができますが、この構成は必須ではありません。

7. PAC 利用の前提条件

PAC 機能を利用するにあたり、下記の前提条件をご確認ください。

1) OS とサードパーティ製品

サポートされる OS、RDBMS、ODBC のバージョンは製品バージョンによって異なりますので、ご利用になるバージョンの製品マニュアルを開き、「デプロイ > 構成および管理 > Enterprise Server の構成および管理 > スケールアウト パフォーマンス/可用性クラスター > 前提条件」をご確認ください。

【例】

Windows

Database	ODBC Driver
Microsoft SQL Server 2008 (including Express editions) R2 and later	ODBC 13.1 for SQL Server
PostgreSQL 10.x (incl. Amazon Aurora Postgres) and later	psqlodbc 11
DB2 10.5 and later	n/a
Oracle 19c	n/a

2) サポートされる構成と制限事項

PAC 構成と制限事項は製品バージョンによって異なりますので、ご利用になるバージョンの製品マニュアルを開き、「デプロイ > 構成および管理 > Enterprise Server の構成および管理 > スケールアウト パフォーマンス/可用性クラスター > サポートされる構成」をご確認ください。

3) PAC メンバーの条件

PAC 構成に参加する PAC メンバーは以下の条件を満たす必要があります。

- Enterprise Server インスタンスが存在する OS のバージョンが同じであること。
- Enterprise Server インスタンスが存在するエンタープライズ製品のバージョンが同じであること。
- Enterprise Server インスタンスの稼働ビット数が同じであること。

8. デモ環境

これから解説するデモでは2つの Windows 環境を使用します。詳細は以下の通りです。

1) OS

Windows 11 Pro

2) プロセッサ

Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz 2.11 GHz

3) システムの種類

64 ビット オペレーティング システム、x64 ベース プロセッサ

4) 各環境の名前解決

IP アドレスは各環境の hosts ファイルにて名前解決しています。

ファイルパスの例) C:¥Windows¥System32¥drivers¥etc¥hosts
設定値)

< 管理画面を使用して運用するマシンの IP アドレス> WIN11-SVR

< Micro Focus Directory Server (以降 MFDS と称す) をリスンされるマシンの IP アドレス> WIN11-CLI

5) ユーザー権限

両環境において、アクセスや書き込み権限のあるユーザーを使用しています。

6) MFDS サービスの開始設定

WIN11-CLI の Enterprise Server インスタンスを PAC メンバーとするため、WIN11-SVR から WIN11-CLI の MFDS へアクセスを許可するよう、WIN11-CLI の MFDS 設定ファイルを変更しています。変更前に Windows サービスから MFDS サービスを停止し、変更、保存後に MFDS サービスを開始してください。念のため、オリジナルファイルはバックアップしてください。

ファイルパス) C:¥ProgramData¥Micro Focus¥Enterprise Developer¥mfdsacfg.xml

更新前の設定値) <bind_address>127.0.0.1</bind_address>

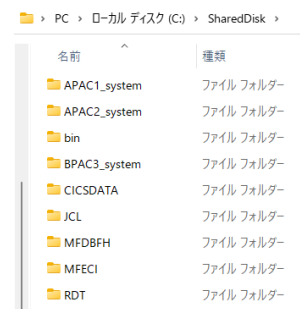
更新後の設定値) <bind_address>0.0.0.0</bind_address>

7) 共有フォルダ

すべての PAC メンバーが使用する共有フォルダを WIN11-SVR に作成しています。

8) インメモリデータストア

WIN11-SVR に存在する、製品に含まれる Redis を使用しています。



名前	種類
APAC1_system	ファイル フォルダ
APAC2_system	ファイル フォルダ
bin	ファイル フォルダ
BPAC3_system	ファイル フォルダ
CICSDATA	ファイル フォルダ
JCL	ファイル フォルダ
MFDBFH	ファイル フォルダ
MFECI	ファイル フォルダ
RDT	ファイル フォルダ

9) MFDBFH

WIN11-SVR には IBM Db2 v11.1.0.1527 がインストールされており、ここに MFDBFH を構築しています。WIN11-SVR ではリモートノードを設定し、WIN11-CLI からのアクセスを可能にしています。リモート接続方法に関しては Db2 の技術文書をご参照ください。

製品の MFDBFH 機能では、サポートしている RDBMS 別にスクリプトを提供しています。任意のデータベース名などを引数として環境構築用のスクリプトを生成し、これを実行することでデータベースやデータストアを作成することができます。

詳しくは、ご利用になるバージョンの製品マニュアルを開き、「ディプロイ > ディプロイ済みアプリケーションのメインフレーム サポート > MSS 構成および管理 > Micro Focus ネイティブ データベース ファイル処理およびエンタープライズ サーバー リージョン データベース管理 > 構成要件」をご参照ください。

10) XA スイッチモジュール

Db2 の接続に ODBC を使用する場合は、Db2 用の XA スイッチモジュールをビルドして使用します。詳しくは、ご利用になるバージョンの製品マニュアルを開き、「ディプロイ > 構成および管理 > Enterprise Server の構成および管理 > サーバー インスタンス環境の構成 > XA 準拠リソース (XAR) の使用 > RM スイッチ モジュール > RM スイッチ モジュールのビルド」をご参照ください。

11) ODBC

PAC メンバーから MFDBFH へのアクセスに ODBC を使用する場合は、Db2 の ODBC ドライバーを使用したシステムデータソースを作成します。この名前は後述の Enterprise Server インスタンスの XA リソース登録時に使用します。

デモ環境では接続文字列を使用します。

ODBC 登録例)

名前	プラットフォーム	ドライバー
DB2FH	64ビット	IBM DB2 ODBC DRIVER - DB2COPY1

1 2) 稼働ビット数

プログラム、Enterprise Server インスタンス共に 64 ビットを指定しています。

1 3) データ文字コード

製品では EBCDIC もしくは ASCII/SJIS 文字コードを指定できますが、デモ環境では ASCII/SJIS を使用しています。

1 4) 管理画面の表示文字コード

ESCWA の JES 機能で日本語が含まれるスプールやデータを表示する場合は、前もって製品インストールパスに利用する CCSID を展開し、MFACCCGI_CHARSET 環境変数を指定する必要があります。詳しい設定方法は、製品マニュアルの「Getting Started > メインフレーム チュートリアル > JCL チュートリアル」をご参照ください。

1 5) Micro Focus 製品名とバージョン

Micro Focus Enterprise Developer 8.0J for Windows

補足) Micro Focus Enterprise Server 8.0J for Windows と同等の開発用実行環境を含んでいます。

1 6) 参照した Micro Focus 製品マニュアルバージョン

Micro Focus Enterprise Developer 8.0 for Eclipse

Micro Focus Enterprise Developer 8.0 for Visual Studio 2022

9. PAC の構成手順

PAC 構成に必要な要素やコンポーネントをご理解いただいた上で、ここからは設定する手順について具体的に解説します。

1) MFDBFH 構成ファイルの作成

Db2 に構築済みの MFDBFH 環境へ接続するために MFDBFH.cfg という名前の構成ファイルを用意します。

構築済み環境)

データベース名 : MFDBFH

データストア名 : DATAS

アクセス権限を持つユーザー : tarot

ユーザーのパスワード : password

内容例)

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <datastores>
3   <server name="WIN11-SVR" type="db2" access="odbc">
4     <dsn name="DB2DATAS" type="datastore" dsname="DATAS" optio="all" dbname="MFDBFH" connect="DB=MFDBFH,uid=tarot,pwd=password"/>
5     <dsn name="DEMOPAC" type="region.cas" region="DEMOPAC" feature="all" dbname="MFDBFH" connect="DB=MFDBFH,uid=tarot,pwd=password"/>
6     <dsn name="DB2CROSSREGION" type="crossregion.cas" dbname="MFDBFH" connect="DB=MFDBFH,uid=tarot,pwd=password"/>
7   </server>
8 </datastores>
```

3行目 : RDBMS が存在する環境とその種類、アクセス方法を指定しています。

4行目 : データベース名 MFDBFH に存在するデータストア名 DATAS を指定し、tarot ユーザーでアクセスするよう指定しています。このデータストアは MFDBFH 機能を使用して事前に作成し、JES 機能で使用するファイルとデータファイルを格納しています。

5行目 : リージョンデータベースの指定となり、後述で作成する PAC 名を指定します。このリージョンデータベースは MFDBFH 機能を使用して事前に作成する必要があります。

6行目 : リージョン間データベースの指定です。このリージョン間データベースは MFDBFH 機能を使用して事前に作成する必要があります。

配置場所は任意ですが、デモ環境ではこの構成ファイルを WIN11-SVR の共有フォルダである C:¥SharedDisk¥MFDBFH に配置しています。

このファイルはすべての PAC メンバーが参照します。

2) インメモリデータストアの開始

SOR を作成する前に、WIN11-SVR の Enterprise Developer コマンドプロンプト (64 ビット) から、製品付属の Redis を開始します。WIN11-CLI からのアクセスに備えて --bind 0.0.0.0 を指定します。

コマンド)

"C:\Program Files (x86)\Micro Focus\Enterprise Developer\AdoptRedis\redis-server.exe" --bind 0.0.0.0

```
C >"C:\Program Files (x86)\Micro Focus\Enterprise Developer\AdoptRedis\redis-server.exe" --bind 0.0.0.0
Redis 3.2.100 (00000000/0) 64 bit
Running in standalone mode
Port: 6379
PID: 6944
http://redis.io

[6944] 14 Dec 11:17:19.164 # Server started, Redis version 3.2.100
[6944] 14 Dec 11:17:19.165 * The server is now ready to accept connections on port 6379
```

デフォルトの 6379 ポートで Redis が開始されます。

3) Enterprise Server インスタンスの設定

ESCWA から PAC メンバーとなる Enterprise Server インスタンスの設定を行います。デモで使用するインスタンス名は以下の3つです。

WIN11-SVR マシンに存在するインスタンス)

APAC 1

APAC 2

WIN11-CLI マシンに存在するインスタンス)

BPAC 3

① 構成情報

必要な環境変数を設定します。

APAC 1, APAC 2 設定値)

```
構成情報 ⓘ
[ES-Environment]
SHARED=C:\SharedDisk
MFACCGI_CHARSET=Shift_JIS
MFCODESET=939
ES_DB_FH=Y
ES_DB_SERVER=WIN11-SVR
MFDBFH_CONFIG=$SHARED\MFDBFH\MFDBFH.cfg
```

BPAC 3 設定値)

```
構成情報 ⓘ
[ES-Environment]
SHARED=\\\\WIN11-SVR\SharedDisk
MFACCGI_CHARSET=Shift_JIS
MFCODESET=939
ES_DB_FH=Y
ES_DB_SERVER=WIN11-SVR
MFDBFH_CONFIG=$SHARED\MFDBFH\MFDBFH.cfg
```

SHARED

共有フォルダのパスを環境変数として指定しています。

WIN11-CLI からはネットワーク経由となるため、パスの先頭に ¥¥¥¥ を指定します。

MFACCCGI_CHARSET

JES 機能において日本語を表示するため、Shift_JIS を指定します。

MFCODESET

CICS 機能において使用する文字コードの 939 を指定しています。

ES_DB_FH

MFDBFH を使用するため、Y を指定します。

ES_DB_SERVER

MFDBFH が構築されている RDBMS の存在する

IP アドレスを指定します。

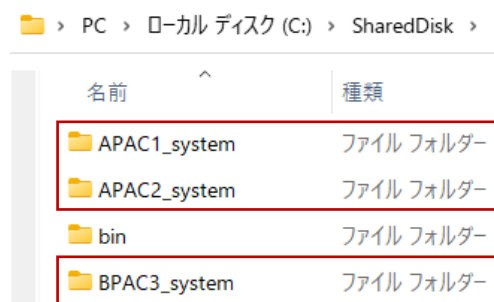
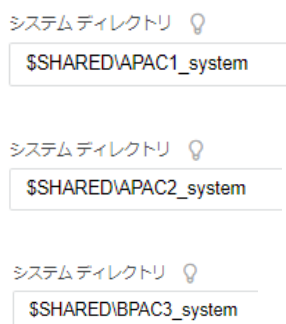
MFDBFH_CONFIG

前述で作成した MFDBFH.cfg ファイルの完全パスを指定します。



② システムディレクトリ

インスタンスごとに出力されるコンソールログなどを格納するシステムディレクトリは個別に設定する必要があります。デモ環境では共有フォルダにそれぞれのフォルダを用意しています。



③ CICS の構成

共有フォルダに格納されているリソース定義ファイル、実行モジュール、データファイル、マップファイルを共有するため、PAC メンバーに同じ値を設定しています。SIT についても同じ値を入力します。

The screenshot shows the 'CICSの構成' (CICS Configuration) window with the following settings:

- システム初期化テーブル (SIT): DBCS
- リソース定義ファイルパス: \$SHARED\RDТ
- トランザクションパス: \$SHARED\bin
- ファイルパス: \$SHARED\CICSDATA
- マップパス: \$SHARED\bin
- EZASOCKET サポート*

To the right, a file explorer shows the path 'PC > ローカル ディスク (C:) > SharedDisk >'. The following folders are listed and highlighted with red boxes:

名前	種類
APAC1_system	ファイル フォルダー
APAC2_system	ファイル フォルダー
bin	ファイル フォルダー
BPAC3_system	ファイル フォルダー
CICSDATA	ファイル フォルダー
JCL	ファイル フォルダー
MFDBFH	ファイル フォルダー
MFECI	ファイル フォルダー
RDТ	ファイル フォルダー

④ JES の構成

共有フォルダに格納されている実行モジュールを共有するため、PAC メンバーの JES プログラムパスには同じ値を設定しています。

システムカタログ、データセットの省略時ロケーションには、MFDBFH に格納するため、構築されている RDBMS が存在する IP アドレスとデータストア名を設定します。

The screenshot shows the 'JESの構成' (JES Configuration) window with the following settings:

- JES プログラムパス: \$SHARED\bin
- システム カタログ: sql://WIN11-SVR/DATAS/catalog.dat
- データセットの省略時ロケーション: sql://WIN11-SVR/DATAS?type=folder;folder=/data
- システム プロシージャ ライブラリ: SYS1.PROCLIB
- Fileshare 構成ロケーション: (empty)

Below the configuration window, a file explorer shows the path 'PC > ローカル ディスク (C:) > SharedDisk >'. The 'bin' folder is highlighted with a red box.

To the right, a terminal window shows the output of the command 'C > dbfhdeploy list sql://WIN11-SVR/DATAS':

```

Micro Focus Database File Handler - Deployment Tool Version 8.0.00
Copyright (C) 1984-2022 Micro Focus. All rights reserved.

Name      Type
----      -
/PRC      Folder
/data     Folder
ESCLRNLs.cfg  Lseq
SPLDSN.dat  Ksds
SPLJNO.dat  Seq
SPLJOB.dat  Ksds
SPLMSG.dat  Ksds
SPLOUT.dat  Ksds
SPLSUB.dat  Ksds
catalog.dat Ksds
jesinfo.dat Ksds

DEP0021I 11 files/folders
  
```

各 PAC メンバーには A と B と C クラスを実行できるイニシエータを1つ設定しています。

PAC 構成は1つのシステムイメージで稼働するため、このグループ内に同じクラスを実行するイニシエータが3つあることとなります。並列実行により負荷分散が実現されますが、逆に並列実行を防ぐ場合は PAC メンバー内で該当クラスのイニシエータが単一になるよう設定します。

名前	クラス	説明
INITABC	ABC	

PAC 構成内においても、ジョブ名が重複している並列実行は制御され、デフォルト設定では先行ジョブが終了するまで後行ジョブは待機します。

⑤ XA リソースの登録

MFDBFH 接続に ODBC を使用する場合は、PAC メンバーの XA リソースの構成に前述でビルドした XA スイッチモジュールと ODBC 名を指定して登録します。

XA リソースの構成

ID* XADB2 名前* DB2MFDB

モジュール* C:\xa\ESDB2XA.dll

有効* 再接続試行* 1

OPEN 文字列 DSN=DB2FH,LOCALTX=T

4) SOR の作成

WIN11-SVR の ESCWA から SOR を作成します。左側メニューから SOR を選択し、右側ペインの追加ボタンをクリックします。

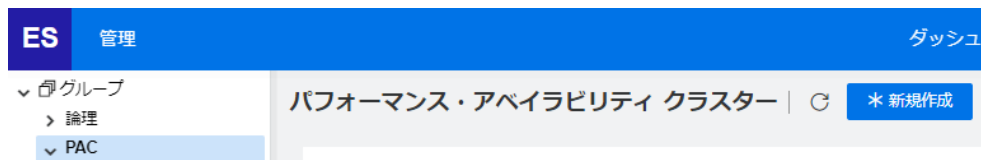
名前は任意ですが、デモ環境では DEMOSOR を、タイプには Redis を、接続パスには前述で開始した Redis が存在するマシンの IP アドレスとポート番号を指定します。

スケールアウトリポジトリの構成

名前 *	タイプ
<input type="text" value="DEMOSOR"/>	<input type="text" value="Redis"/>
説明	
<input type="text"/>	
接続パス *	
<input type="text" value="WIN11-SVR:6379"/>	

5) PAC の作成

WIN11-SVR の ESCWA から PAC を作成します。左側メニューから PAC を選択し、右側ペインの追加ボタンをクリックします。



名前は任意ですが、デモ環境では DEMOPAC を、PAC SOR には 前述で作成した DEMOSOR を選択します。

The screenshot shows the 'パフォーマンス・アベイラビリティ クラスタの新規作成' (New Performance Availability Cluster) form. It includes a '名前*' (Name) field with 'DEMOPAC', an empty '説明' (Description) field, and a 'PAC SOR' dropdown menu with 'DEMOSOR' selected. There are '保存' (Save) and '戻る' (Back) buttons at the top right.

左側メニューから、追加された DEMOPAC を選択し、右側ペインのインストールボタンをクリックします。



インストールウィンドウで、どのマシンにあるどの Enterprise Server インスタンスを PAC メンバーとするのかを指定します。

リージョンにPAC定義をインストール

Directory Server

WIN11-SVR
WIN11-CLI

Selected すべてクリア

APAC1@WIN11-SVR ×
APAC2@WIN11-SVR ×

リージョン

フィルタ

APAC1@WIN11-SVR
 APAC2@WIN11-SVR
 CICSDEMO@WIN11-SVR
 CICSIMS@WIN11-SVR
 DBDEMO@WIN11-SVR
 ECIDEMO@WIN11-SVR
 ESDEMO@WIN11-SVR
 ESDEMO64@WIN11-SVR
 IMSDEMO@WIN11-SVR

追加 戻る

3つのインスタンスをメンバーとした PAC 構成が作成されます。

PAC内のメンバー

リージョン名	Directory Server
APAC1	WIN11-SVR
APAC2	WIN11-SVR
BPAC3	WIN11-CLI

10. Enterprise Server インスタンスの開始

PAC メンバーを開始する具体的な手順について説明します。




1) Enterprise Server インスタンス開始 : 1つ目

関連する SOR の既存データをすべて削除するために、1つ目の PAC メンバーにはコールドを指定して開始します。デモ環境では APAC1 が該当します。このコールド指定は必須ではありません。

リージョンの開始オプション

 セキュアでない

スタートモード

ウォーム  コールド  アップグレード 

2) PAC の初期化

コールド指定による1つ目の PAC メンバーが正常に開始されない場合は、すべてのメンバーが停止されていることを確認後、caspac コマンドを使用して PAC の初期化を行います。これにより PSOR からすべての情報が削除されます。

コマンドでは、定義済みの PAC 名と前述で起動済みの Redis が存在するマシンの IP アドレスと接続ポート番号を指定します。

コマンド例) caspac -aInitPac=DEMOPAC -sredis,WIN11-SVR:6379

```
C>caspac -aInitPac=DEMOPAC -sredis,WIN11-SVR:6379
Connected to Redis server at WIN11-SVR port 06379
Using PACName DEMOPAC
Version=0002
NB CasLOADS Readers: 0
ES_DB_FH=Y
ES_DB_SERVER=WIN11-SVR
```

```
ES Region that started the PAC: PACUID=APAC1(P
89 (303406))
Store CasTMP deleted return code 0
Store CasPAC deleted return code 0
Store CasCSF deleted return code 0
PAC DEMOPAC initialisation return code 0
```

3) Enterprise Server インスタンス開始 : 2つ目以降

2つ目以降は関連する SOR の既存データを保持するために、ウォームを指定して開始します。デモ環境では APAC2、BPAC3 が該当します。



リージョンの開始オプション

△ 不 セキュアでない

スタートモード

ウォーム ⓘ

コールド ⓘ

アップグレード ⓘ

正常に開始されない場合は、システムディレクトリにある console.log ファイルをテキストエディターで表示し、エラーの内容を確認します。

正常に開始された場合は、ESCWA から APAC 1 のコンソールログを表示し、他の2つのインスタンスと同期していることを確認します。

CASMG0001I		APAC2(002)#CASSI1432I Joining PAC DEMOPAC. 13:42:26#
CASMG0001I		APAC2(002)#CASSI1435I 00000 Other module(s) were synchronized from PAC DEMOPAC CasLOADS. 13:42:26#
CASMG0001I		APAC2(002)#CASSI1435I 00000 CICS module(s) were synchronized from PAC DEMOPAC CasLOADS. 13:42:26#
CASMG0001I		APAC2(002)#CASSI1435I 00000 JES module(s) were synchronized from PAC DEMOPAC CasLOADS. 13:42:26#
CASMG0001I		BPAC3(003)#CASSI1432I Joining PAC DEMOPAC. 13:42:41#
CASMG0001I		BPAC3(003)#CASSI1435I 00000 Other module(s) were synchronized from PAC DEMOPAC CasLOADS. 13:42:43#
CASMG0001I		BPAC3(003)#CASSI1435I 00000 CICS module(s) were synchronized from PAC DEMOPAC CasLOADS. 13:42:43#
CASMG0001I		BPAC3(003)#CASSI1435I 00000 JES module(s) were synchronized from PAC DEMOPAC CasLOADS. 13:42:43#

11. JCL の実行

実際に3つの JCL を実行して、負荷分散とファイルの排他制御が行われているかを確認します。

1) JCL の内容

① SLEEP.LCK.JCL

ジョブ名) SLEEP.LCK

実行クラス) A

使用ファイル)

VSAM.KSDS1 ファイルを OUTPUT で OPEN

実行プログラム内容)

ファイルを OPEN 後、60 秒スリープし、

ソート後のデータを VSAM.KSDS1 ファイルへ出力

サブミットする Enterprise Server インスタンス)

APAC1 : ポート番号 61747

```
//SLEEP.LCK JOB CLASS=A,MSGCLASS=A
//*****
//DEFVSAM1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE VSAM.KSDS1 PURGE
SET LASTCC=0
DEFINE CLUSTER (NAME(VSAM.KSDS1)) -
DATA -
(RECORDS(10) FREESPACE(20 10) KEYS(5 0) -
RECORDSIZE(71 71)) -
INDEX (RECORDS(50 50))
/*
//SORTSTEP EXEC SORTD
//SORT1.SORTIN DD *
00009Shiki Masaoka 5-5,Dogo Onsen,Matsuyama-shi,Ehime-ken 1870
00001Soseki Natsume 1-1,Koishikawa,Bunkyo-ku,Tokyo-to 1886
00007Dgai Mori 3-1,Rintaro-cho,Tsuwano-shi,Shimane-ken 1886
00002Ryotaro Shiba 2-3,Sonezaki,Kita-ku,Osaka-shi,Osaka-fu 1900
00000Jirocho Shimizu 6-6,Jiro-cho,Shimizu-shi,Shizuoka-ken 1800
/*
//SORT1.SORTOUT DD DSN=&&JINJIDAT,DISP=(NEW,PASS),
// SPACE=(800,(10,10)),DCB=(RECFM=FB,LRECL=71,DSORG=PS),UNIT=SYSDA
//SORT1.SYSIN DD *
SORT FIELDS=(1,5,CH,A)
/*
//APPL1 EXEC PGM=SLEEP.LCK
//SYSOUT DD SYSOUT=*
//KSDSFILE DD DSN=VSAM.KSDS1,DISP=SHR
//INDD DD DSN=&&JINJIDAT,DISP=(OLD,DELETE)
//PRINTER DD SYSOUT=*
```

② NOWAIT.JCL

ジョブ名) NOWAIT

実行クラス) A

使用ファイル)

VSAM.KSDS2 ファイルを OUTPUT で OPEN

実行プログラム内容)

ソート後のデータを VSAM.KSDS2 ファイルへ出力

サブミットする Enterprise Server インスタンス)

APAC1 : ポート番号 61747

```
//NOWAIT JOB CLASS=A,MSGCLASS=A
//*****
//DEFVSAM1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE VSAM.KSDS2 PURGE
SET LASTCC=0
DEFINE CLUSTER (NAME(VSAM.KSDS2)) -
DATA -
(RECORDS(10) FREESPACE(20 10) KEYS(5 0) -
RECORDSIZE(71 71)) -
INDEX (RECORDS(50 50))
/*
//SORTSTEP EXEC SORTD
//SORT1.SORTIN DD *
00009Shiki Masaoka 5-5,Dogo Onsen,Matsuyama-shi,Ehime-ken 1870
00001Soseki Natsume 1-1,Koishikawa,Bunkyo-ku,Tokyo-to 1886
/*
//SORT1.SORTOUT DD DSN=&&JINJIDAT,DISP=(NEW,PASS),
// SPACE=(800,(10,10)),DCB=(RECFM=FB,LRECL=71,DSORG=PS),UNIT=SYSDA
//SORT1.SYSIN DD *
SORT FIELDS=(1,5,CH,A)
/*
//APPL1 EXEC PGM=NOWAIT
//SYSOUT DD SYSOUT=*
//KSDSFILE DD DSN=VSAM.KSDS2,DISP=SHR
//INDD DD DSN=&&JINJIDAT,DISP=(OLD,DELETE)
//PRINTER DD SYSOUT=*
```

③ WAIT.JCL

ジョブ名) WAIT

実行クラス) C

使用ファイル)

VSAM.KSDS1 ファイルを I-O で OPEN

実行プログラム内容)

ソート後のデータを SLEEP.LCK.JCL と同じ

VSAM.KSDS1 ファイルへ出力

```
//WAIT JOB CLASS=C,MSGCLASS=C
//*****
//SORTSTEP EXEC SORTD
//SORT1.SORTIN DD *
00005Eiji Yoshikawa 9-3,Miyatomomura,Mimasaka-gun,Okayama-ken 1920
00004Osamu Dazai 2-6,Tsugaru,Tsugaru-gun,Aomori-ken 1911
00008Ryoma Sakamoto 1-1,Harimayabashi,Kochi-shi,Kochi-ken 1820
00010Yukichi Fukuzawa 8-8,Keio-cho,Nakatsu-shi,Oita-ken 1835
00003Hideyo Noguchi 5-1,Inawashiro,Aizu-shi,Fukushima-ken 1911
/*
//SORT1.SORTOUT DD DSN=&&JINJIDAT,DISP=(NEW,PASS),
// SPACE=(800,(10,10)),DCB=(RECFM=FB,LRECL=71,DSORG=PS),UNIT=SYSDA
//SORT1.SYSIN DD *
SORT FIELDS=(1,5,CH,A)
/*
//APPL1 EXEC PGM=WAIT
//SYSOUT DD SYSOUT=*
//KSDSFILE DD DSN=VSAM.KSDS1,DISP=SHR
//INDD DD DSN=&&JINJIDAT,DISP=(OLD,DELETE)
//PRINTER DD SYSOUT=*
```

サブミットする Enterprise Server インスタンス)

BPAC3 : ポート番号 61749

2) JCL の実行

前述の JCL を実行します。これらの JCL は共有フォルダに配置してあり、インスタンスの Web Service J2EE ポートへサブミットします。

コマンド)

```
cassub -jC:%SharedDisk%JCL%SLEEPLCK.JCL -stcp:WIN11-SVR:61747
```

```
cassub -jC:%SharedDisk%JCL%NOWAIT.JCL -stcp:WIN11-SVR:61747
```

```
cassub -jC:%SharedDisk%JCL%WAIT.JCL -stcp:WIN11-CLI:61749
```

```
C >cassub -jC:%SharedDisk%JCL%SLEEPLCK.JCL -stcp:WIN11-SVR:61747
JCLCM0187I J0001054 SLEEPLCK JOB SUBMITTED (JOBNAME=SLEEPLCK,JOBNUM=0001054) 11:12:26
JCLCM0180I J0001054 SLEEPLCK Job ready for execution. 11:12:26
Processed "C:%SharedDisk%JCL%SLEEPLCK.JCL"

C >cassub -jC:%SharedDisk%JCL%NOWAIT.JCL -stcp:WIN11-SVR:61747
JCLCM0187I J0001055 NOWAIT JOB SUBMITTED (JOBNAME=NOWAIT,JOBNUM=0001055) 11:12:27
JCLCM0180I J0001055 NOWAIT Job ready for execution. 11:12:27
Processed "C:%SharedDisk%JCL%NOWAIT.JCL"

C >cassub -jC:%SharedDisk%JCL%WAIT.JCL -stcp:WIN11-CLI:61749
JCLCM0187I J0001056 WAIT JOB SUBMITTED (JOBNAME=WAIT,JOBNUM=0001056) 11:12:30
JCLCM0180I J0001056 WAIT Job ready for execution. 11:12:31
Processed "C:%SharedDisk%JCL%WAIT.JCL"
```

3) 実行記録の確認

各メンバーのコンソールログを表示して実行記録を確認します。

APAC1 のコンソールログ)

- ① SLEEPLCK.JCL を受け付け、60 秒のスリープ後に正常終了しています。
- ② NOWAIT.JCL を受け付けていますが、実行記録がありません。SLEEPLCK ジョブがイニシエータを占有しているため、他のインスタンスで実行されているか後述で確認します。

	JCLCM0182I	I	J0001052 SLEEPLCK JOB ENDED - COND CODE 0000
①	JCLCM0187I	I	J0001054 SLEEPLCK JOB SUBMITTED (JOBNAME=SLEEPLCK,JOBNUM=0001054)
	JCLCM0180I	I	J0001054 SLEEPLCK Job ready for execution.
	JES000004I	I	J0001054 SLEEPLCK JOB DISPATCHED
	JCLCM0187I	I	J0001055 NOWAIT JOB SUBMITTED (JOBNAME=NOWAIT,JOBNUM=0001055)
②	JCLCM0180I	I	J0001055 NOWAIT Job ready for execution.
	JCLCM0188I	I	J0001054 SLEEPLCK JOB STARTED
	CASMG0001I	I	** START SLEEP 60 sec **
①	CASMG0001I	I	** END SLEEP 60 sec **
	CASMG0001I	I	** END OF SLEEP **
	JCLCM0182I	I	J0001054 SLEEPLCK JOB ENDED - COND CODE 0000

APAC2 のコンソールログ)

APAC1 で受け付けられた NOWAIT.JCL はイニシエータが空いている APAC2 に割り振られ、正常終了しています。

JES000004I		J0001055 NOWAIT JOB DISPATCHED
JCLCM0188I		J0001055 NOWAIT JOB STARTED
CASMG0001I		** START NOWAIT **
CASMG0001I		** END OF NOWAIT **
JCLCM0182I		J0001055 NOWAIT JOB ENDED - COND CODE 0000

BPAC3 のコンソールログ)

WAIT.JCL を受け付けていますが、SLEEPLCK ジョブにより VSAM.KSDS1 ファイルがロックされているため、開放を待つて処理を実行し、正常終了しています。

JCLCM0187I		J0001056 WAIT JOB SUBMITTED (JOBNAME=WAIT,JOBNUM=0001056)
JCLCM0180I		J0001056 WAIT Job ready for execution.
JES000004I		J0001056 WAIT JOB DISPATCHED
JCLCM0188I		J0001056 WAIT JOB STARTED
MVSXR0091I		J0001056 WAIT Waiting for shared use of dataset "VSAM.KSDS1".
MVSXR0093I		J0001056 WAIT Required datasets have been acquired. Execution resumed.
CASMG0001I		** START WAIT **
CASMG0001I		** END OF WAIT **
JCLCM0182I		J0001056 WAIT JOB ENDED - COND CODE 0000

JES 機能のスプールを ESCWA から表示し、時間経過から結果を確認します。

	名前	ジョブID	クラス	ユーザー	条件コード	サブミット	開始	完了	
■	SLEEPLCK	J0001054	A	JESUSER	0000		11:12:26.37	11:12:27.00	11:13:36.03
■	NOWAIT	J0001055	A	JESUSER	0000		11:12:27.41	11:12:28.30	11:12:37.09
■	WAIT	J0001056	C	JESUSER	0000		11:12:30.82	11:12:32.30	11:13:37.97

3つの JCL はサブミット順に受け付けられています。

最初にジョブの実行が開始されたのは SLEEPLCK、次に NOWAIT、最後にファイルの開放を待っていた WAIT であることがわかります。

ジョブが最初に完了したのは NOWAIT、次に SLEEPLCK、最後に WAIT となり意図した結果となりました。

この結果から PAC メンバー内における負荷分散とファイルの排他制御が行われ、正常に終了したことがわかります。

4) MFDBFH データストアの確認

JCL の実行により作成されたデータファイル、スプールファイルが MFDBFH のデータストアに存在するか確認します。

コマンド)

```
dbfhdeploy list sql://WIN11-SVR/DATAS?folder=/data
```

```
C >dbfhdeploy list sql://WIN11-SVR/DATAS?folder=/data
Micro Focus Database File Handler - Deployment Tool Version 8.0.00
Copyright (C) 1984-2022 Micro Focus. All rights reserved.

Name                                         Type
----                                         -
JINJI.KSDS.DAT                             Ksds
VSAM.KSDS1.DAT                             Ksds
VSAM.KSDS2.DAT                             Ksds
Y2022.S1219.S111226.J0001054.D00000.DAT   Lseq
Y2022.S1219.S111226.J0001054.D00001.SYSPRINT.DAT Seq
Y2022.S1219.S111226.J0001054.D00005.SYSOUT.DAT Seq
Y2022.S1219.S111226.J0001054.D00007.PRINTER.DAT Seq
Y2022.S1219.S111226.J0001054.D00008.SYSPRINT.DAT Seq
Y2022.S1219.S111227.J0001055.D00000.DAT   Lseq
Y2022.S1219.S111227.J0001055.D00001.SYSPRINT.DAT Seq
Y2022.S1219.S111227.J0001055.D00005.SYSOUT.DAT Seq
Y2022.S1219.S111227.J0001055.D00007.PRINTER.DAT Seq
Y2022.S1219.S111227.J0001055.D00008.SYSPRINT.DAT Seq
Y2022.S1219.S111230.J0001056.D00000.DAT   Lseq
Y2022.S1219.S111230.J0001056.D00003.SYSOUT.DAT Seq
Y2022.S1219.S111230.J0001056.D00005.PRINTER.DAT Seq
Y2022.S1219.S111230.J0001056.D00006.SYSPRINT.DAT Seq

DEP0021I 17 files/folders
DEP0007I The operation completed successfully (0.296 seconds)
```

作成した VSAM ファイルとスプールファイルが格納されていることが確認できます。

12. CICS の実行

製品付属のターミナルエミュレータである Micro Focus Host Access for the Cloud (以降 HACloud と称す) を使用して、リソース定義ファイルやデータファイルが共有されているか確認します。

1) HACloud セッションサーバーの起動

HACloud は、Windows および Linux 上の 3270 ホストアプリケーションへのブラウザベースの HTML5 アクセスを提供するものです。

① 構成ファイル

製品にはデフォルト値が設定されている構成ファイルが、以下の製品インストールパスに付属されています。オリジナルファイルをバックアップ後、必要に応じて変更してください。

デモ環境では WIN11-SVR の構成ファイルをデフォルト設定のまま使用します。

パスの例) C:¥ProgramData¥Micro Focus¥Enterprise Developer

ファイル名) hacloud.properties

```
1 #logging.level.com.microfocus.zfe.enterprise.ConfigurationServiceEnterprise=INFO
2 logging.level.com.microfocus.zfe.emulation.session.transport.telnet.TelnetProvider=ERROR
3 # HTTP URL on which ESCWA should be contacted
4 hacloud.es.escwaurl=http://localhost:10086
5 # MFDS host on which to query available 3270 endpoints
6 hacloud.es.mfdshost=127.0.0.1
7 # MFDS port
8 hacloud.es.mfdsport=86
9 # Port on which HA Cloud Session Server will accept inbound HTTP requests
10 server.port=7443
11 # Session Server keystore, holds server certificate
12 server.ssl.key-store=../../etc/keystore.bcfks
13 # Session Server keystore password
14 server.ssl.key-store-password=changeit
15 # Trusted certificate store for outbound HTTP and TN3270 connections
16 server.ssl.trust-store=../../etc/trustcerts.bcfks
17 # Trusted certificate store password
18 server.ssl.trust-store-password=changeit
19 # Listen over HTTP
20 spring.profiles.active=no-tls,extensions-enabled
21 # Listen over HTTPS (if a server certificate is supplied using the keystore)
22 #spring.profiles.active=extensions-enabled
```

主な設定値)

4行目 : hacloud.es.escwaurl

ESCWA が稼働している URL を指定します。デフォルトは http://localhost:10086 です。

6行目 : hacloud.es.mfdshost

MFDS が稼働しているマシンの IP アドレスを指定します。

8行目 : hacloud.es.mfdsport

MFDS が稼働しているポート番号を指定します。

10行目 : server.port

HACloud セッションサーバーと接続する HTTP サーバーのエンドポイントを指定します。

その他の項目や詳細については製品マニュアルをご参照ください。

② HACloud セッションサーバーの開始

デモ環境では 64 ビットを使用するため、64 ビットの Enterprise Developer コマンドプロンプトを起動して開始コマンドを実行します。

開始コマンド) startsessionserver

開始状態)

```

C>startsessionserver
java -Xmx1G -XX:+CMSClassUnloadingEnabled -XX:+DisableExplicitGC -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=%logs
-Djava.net.preferIPv4Stack=true -Djava.util.prefs.PreferencesFactory=com.microfocus.emc.fileprefs.FilePreferencesFactor
y -Dcom.microfocus.prefs.rootdir=%session-data -Dsun.net.http.allowRestrictedHeaders=true -jar "sessionserver.jar"

Micro Focus
HAcloud
Session Server

VERSION: 2.7.0.3

INFO 2022-12-21T11:21:10.267+0900 [main] [c.m.z.w.SessionServerApplication]: Starting SessionServerApplication v2.7.0.3
.151136 using Java 11.0.13 on Win11VM1 with PID 17836 (C:\Program Files (x86)\Micro Focus\Enterprise Developer\bin64\hacloud
\sessionserver.jar started by tarot in C:\Program Files (x86)\Micro Focus\Enterprise Developer\bin64\hacloud)
INFO 2022-12-21T11:21:10.272+0900 [main] [c.m.z.w.SessionServerApplication]: The following profiles are active: no-tls,
extensions-enabled
INFO 2022-12-21T11:21:16.971+0900 [main] [c.m.z.w.c.c.j.JettyConfig]: HTTP connections accepted on port 7443
INFO 2022-12-21T11:21:19.185+0900 [main] [c.a.i.c.i.BouncyCastleProviderRegistrationUtil]: JCE provider added: BouncyCa
stle Security Provider (FIPS edition) v1.0.2.1
INFO 2022-12-21T11:21:19.187+0900 [main] [c.a.i.c.i.BouncyCastleProviderRegistrationUtil]: FIPS mode: false
INFO 2022-12-21T11:21:19.242+0900 [main] [c.a.i.c.i.BouncyCastleProviderRegistrationUtil]: JSSE provider added: Bouncy
Castle JSSE Provider Version 1.0.12.2

```

2) CICS 処理の実行

Web ブラウザから HAcloud セッションサーバーにアクセスして CICS 処理を実行します。

① HAcloud へのアクセス

対象となる Enterprise Server インスタンスが開始されていることを確認後、Web ブラウザを表示して <http://localhost:7443> へアクセスすると、接続可能な TN3270 リスナーポートが表示されます。構成ファイルでは 127.0.0.1 を指定しているため、WIN11-SVR のリスナーが表示されています。

Available Sessions		
3270	APAC1@	:9011
3270	APAC1@	:9012
3270	APAC1@	:9013
3270	APAC2@	:9014
3270	APAC2@	:9015
3270	APAC2@	:9016

② CICS トランザクションの実行

まずは、APAC1 の 9012 ポートへアクセスしてログオンし、ACCT トランザクションを実行します。データを一部修正します。

修正前のデータ)

```

ACCOUNT FILE: RECORD CHANGE
ACCOUNT NO: 11111 SURNAME: BALL
FIRST: JOHN MI: Y TITLE: MR
TELEPHONE: 01688 1234 ADDRESS: 23 ROSE DRIVE
EASTON
OTHERS WHO MAY CHARGE:
NO. CARDS ISSUED: 2 DATE ISSUED: 12 12 11 REASON: S
CARD CODE: 1 APPROVED BY: HIO SPECIAL CODES: A
ACCOUNT STATUS: N CHARGE LIMIT: 1000.00
HISTORY: BALANCE BILLED AMOUNT PAID AMOUNT
0.00 00/00/00 0.00 00/00/00 0.00
0.00 00/00/00 0.00 00/00/00 0.00
0.00 00/00/00 0.00 00/00/00 0.00
MAKE CHANGES AND "ENTER" OR "CLEAR" TO CANCEL

```

修正後のデータ)

```

ACCOUNT FILE: RECORD CHANGE
ACCOUNT NO: 11111 SURNAME: Janice
FIRST: JOHN MI: Y TITLE: MR
TELEPHONE: 01688 1234 ADDRESS: 23 ROSE DRIVE
EASTON
OTHERS WHO MAY CHARGE:
NO. CARDS ISSUED: 2 DATE ISSUED: 12 12 11 REASON: S
CARD CODE: 1 APPROVED BY: HIO SPECIAL CODES: A
ACCOUNT STATUS: N CHARGE LIMIT: 1000.00
HISTORY: BALANCE BILLED AMOUNT PAID AMOUNT
0.00 00/00/00 0.00 00/00/00 0.00
0.00 00/00/00 0.00 00/00/00 0.00
0.00 00/00/00 0.00 00/00/00 0.00
MAKE CHANGES AND "ENTER" OR "CLEAR" TO CANCEL

```

注意)

HACloud は、製品バージョン 8.0 時点で DBCS 文字には対応していません。

次に、APAC2 の 9015 ポートへアクセスしてログオンし、ACCT トランザクションを実行します。
修正したデータを確認します

修正データの表示)

```

ACCOUNT FILE: RECORD CHANGE

ACCOUNT NO: 11111 SURNAME: JANICE MI: Y TITLE: MR
FIRST: JOHN
TELEPHONE: 01688 1234 ADDRESS: 23 ROSE DRIVE
EASTON

OTHERS WHO MAY CHARGE:

NO. CARDS ISSUED: 2 DATE ISSUED: 12 12 11 REASON: S
CARD CODE: 1 APPROVED BY: HIO SPECIAL CODES: A

ACCOUNT STATUS: N CHARGE LIMIT: 1000.00

HISTORY: BALANCE BILLED AMOUNT PAID AMOUNT
          0.00 00/00/00 0.00 00/00/00 0.00
          0.00 00/00/00 0.00 00/00/00 0.00
          0.00 00/00/00 0.00 00/00/00 0.00

MAKE CHANGES AND "ENTER" OR "CLEAR" TO CANCEL

```

修正後のデータが確認できました。

この結果から、CICS リソース定義ファイル、FCT で指定されたデータファイルが PAC メンバーで共有されていることがわかります。

13. おわりに

企業の中核を担う基幹システムは、「物理的障害の回避」、「ネットワークの安定性確保」、「業務処理停止の回避」など、多角的な視点でメインフレームと同等の堅牢性をオープン環境で実現する必要に迫られています。

オープン環境では堅牢性を確保するための様々なツールが提供されており、目的に応じて過不足のない機能を持ったツールを自由に選択することができます。また、ツールの種類によっては、移行後の運用コストダウンも期待することができます。

マイクロフォーカスの Enterprise Developer と Enterprise Server 製品は、アプリケーションの視点から堅牢性をサポートする機能がデフォルトで搭載されており、これによる効果は前述の通りです。

基幹システムの移行に向けて、マイクロフォーカスのモダナイゼーション支援製品と共に PAC 機能の導入をご検討いただければ幸いです。