

バージョン 6.2.0



z/OS のチューニング



バージョン 6.2.0



z/OS のチューニング

お願い

本書に記載されている情報をご使用になる前に、本書末尾の特記事項セクションに記載されている情報をお読みください。

新しい版で明記されるまで、WebSphere Process Server for z/OS バージョン 6、リリース 2、モディフィケーション 0 (製品番号 5655-N53) 以降のすべてのリリースとモディフィケーションが本書の対象となります。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： WebSphere® WebSphere Process Server for z/OS  
Version 6.2.0  
Tuning for z/OS

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2009.5

© Copyright International Business Machines Corporation 2005, 2009.

---

## PDF ブックおよびインフォメーション・センター

PDF ブックは、印刷およびオフラインでの参照用に提供されています。最新情報は、オンラインのインフォメーション・センターを参照してください。

セットとして、PDF ブックには、インフォメーション・センターと同一の内容が含まれます。

PDF 資料は、バージョン 6.0 またはバージョン 6.1 など、インフォメーション・センターのメジャー・リリースの後の四半期以内にご利用いただけます。

PDF 資料の更新頻度は、インフォメーション・センターより低いですが、Redbooks® よりも頻繁に更新されます。通常、PDF ブックはブックに十分な変更が累積されたときに更新されます。

PDF ブックの外部にあるトピックへのリンクを選択すると、Web 上のインフォメーション・センターに移動します。PDF ブックの外部にあるターゲットへのリンクには、そのターゲットが PDF ブックと Web ページのどちらなのかを示すアイコンによるマークが付いています。

表 1. 本書の外部にあるトピックへのリンクのプレフィックスとなるアイコン

アイコン	説明
	<p data-bbox="540 254 1325 281">インフォメーション・センターのページを含む、Web ページへのリンク。</p> <p data-bbox="540 310 1427 407">インフォメーション・センターへのリンクは、ターゲット・トピックが新しい場所に移動した場合でもその機能を保つように、間接参照ルーティング・サービスを経由します。</p> <p data-bbox="540 436 1427 604">ローカルのインフォメーション・センターでリンク先ページを見つけたい場合は、リンクのタイトルを検索することができます。あるいは、トピック ID を検索することもできます。検索の結果、タイプが異なる製品についてのトピックがいくつか見つかった場合は、検索結果の「グループ別 (Group by)」コントロールを使用して、表示するトピック・インスタンスを識別できます。以下に例を示します。</p> <ol data-bbox="540 619 1427 903" style="list-style-type: none"> <li>1. リンク URL をコピーします。例えば、リンクを右クリックして「リンク先をコピーする (Copy link location)」を選択します。例: <code>http://www14.software.ibm.com/webapp/wsbroker/redirect?version=wbpm620&amp;product=wesb-dist&amp;topic=tins_apply_service</code></li> <li>2. <code>&amp;topic=</code> の後のトピック ID をコピーします。例: <code>tins_apply_service</code></li> <li>3. ローカル・インフォメーション・センターの検索フィールドに、トピック ID を貼り付けます。文書機能がローカルにインストールされている場合は、検索結果にそのトピックが表示されます。以下に例を示します。</li> </ol> <div data-bbox="581 913 1427 1108" style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p data-bbox="589 926 816 947">1 result(s) found for</p> <p data-bbox="589 972 1102 1018">Group by: None   Platform   Version   Product Show Summary</p> <p data-bbox="589 1043 1346 1089">Update Installer を使用したフィックスパックおよびリフレッシュ・パックのインストール</p> </div> <ol data-bbox="540 1142 1183 1169" style="list-style-type: none"> <li>4. 検索結果のリンクをクリックしてトピックを表示します。</li> </ol>
	<p data-bbox="540 1184 797 1211">PDF ブックへのリンク。</p>

# 目次

PDF ブックおよびインフォメーション・センター . . . . .	iii
図 . . . . .	vii
表 . . . . .	ix
<b>第 1 章 概要 . . . . .</b>	<b>1</b>
<b>第 2 章 パフォーマンス調整の方法 . . . . .</b>	<b>3</b>
<b>第 3 章 調整チェックリスト . . . . .</b>	<b>5</b>
<b>第 4 章 調整パラメーター . . . . .</b>	<b>7</b>
トレースおよびロギングのフラグ . . . . .	7
64 ビット・モード . . . . .	7
Java 調整パラメーター . . . . .	8
Java GC ポリシー . . . . .	8
Java ヒープ・サイズ . . . . .	8
ワークロード・プロファイル . . . . .	9
ワークロード・マネージャー・サービス・クラス . . . . .	9
MDB ActivationSpec . . . . .	9
MQ リスナー・ポート . . . . .	10
MDB スロットル . . . . .	11
スレッド・プール・サイズ . . . . .	11
JMS 接続プール・サイズ . . . . .	12
データ・ソースの接続プール・サイズ . . . . .	13
データ・ソース準備済みステートメント・キャッシュのサイズ . . . . .	13
メッセージング・エンジン・プロパティ . . . . .	13
セキュリティの最小限化 . . . . .	14
Network Deployment での自動同期の無効化 . . . . .	14
<b>第 5 章 ビジネス・プロセスの調整 . . . . .</b>	<b>15</b>
長期間にわたって実行するプロセスのチューニング . . . . .	16
メッセージング・エンジンの設定計画 . . . . .	17
メッセージング・プロバイダーの細密チューニング . . . . .	17
ビジネス・プロセス・ナビゲーションのパフォーマンスの向上 . . . . .	17
microflow のチューニング . . . . .	19
<b>第 6 章 ヒューマン・タスクを含むビジネス・プロセスのチューニング . . . . .</b>	<b>21</b>
ヒューマン・タスクへの同時アクセス数の削減 . . . . .	21
照会の応答時間の短縮 . . . . .	21
全テーブルのスキャンニングの回避 . . . . .	22
タスクおよびプロセス照会の最適化 . . . . .	23

<b>第 7 章 Business Process Choreographer Explorer の調整 . . . . .</b>	<b>25</b>
Business Process Choreographer Explorer レポート作成機能のチューニング . . . . .	26
<b>第 8 章 拡張調整 . . . . .</b>	<b>29</b>
トレースおよびモニターに関する考慮事項 . . . . .	29
作業域サービスの最大送受信サイズの構成 . . . . .	29
CEI エミッターの validateEvent プロパティの無効化 . . . . .	29
<b>第 9 章 大規模オブジェクトの調整 . . . . .</b>	<b>31</b>
大規模オブジェクトの処理中の他の処理の削減または排除 . . . . .	31
<b>第 10 章 最大並行性の調整 . . . . .</b>	<b>33</b>
並行性確保のためのエッジ・コンポーネントの調整 . . . . .	33
MDB ActivationSpec . . . . .	33
リスナー・ポートの構成 . . . . .	34
スレッド・プール・サイズの構成 . . . . .	34
MDB の専用スレッド・プールの構成 . . . . .	35
並行性確保のための中間コンポーネントの調整 . . . . .	35
JMS および JMS キュー接続ファクトリーの構成 . . . . .	36
DataSource オプションの構成 . . . . .	36
データ・ソース準備済みステートメント・キャッシュのサイズ . . . . .	36
<b>第 11 章 メッセージング調整 . . . . .</b>	<b>39</b>
メッセージング・エンジンに使用するデータ・ストアまたはファイル・ストアの選択 . . . . .	39
データ・バッファ・サイズの設定 (廃棄可能またはキャッシュ) . . . . .	39
メッセージング・エンジン・データ・ストアの高性能 DBMS への移動 . . . . .	40
メッセージング・エンジンに対するデータ・ストアの設定 . . . . .	40
DB2 データベースの作成およびデータ・ストア・スキーマのロード . . . . .	40
メッセージング・エンジンに対するデータ・ソースの作成 . . . . .	41
メッセージング・エンジンのデータ・ストアの変更 . . . . .	42
<b>第 12 章 WebSphere MQ の調整 . . . . .</b>	<b>45</b>
トレースの適切な使用 . . . . .	45
メッセージ管理の使用 . . . . .	45
CCSID の設定 . . . . .	45
適切なロギングの構成 . . . . .	45
WebSphere Application Server での MQ 接続プールの使用可能化 . . . . .	45

<b>第 13 章 データベース: 一般的な調整</b>	<b>47</b>
データベース・ログ・ファイルの高速ディスク・サブシステムへの配置	47
表スペース・コンテナとは異なる装置へのログの配置	47
<b>第 14 章 データベース: DB2 固有の調整</b>	<b>49</b>
データベース統計の更新	49
バッファ・プール・サイズの正しい設定	49
適正な表索引付けの維持	50
ログ・ファイルの適切なサイズ変更	50
<b>第 15 章 拡張 Java ヒープ調整</b>	<b>53</b>
ガーベッジ・コレクションのモニター	53

大半の構成に対するヒープ・サイズの設定	54
1 つのシステム上で複数の JVM を実行する場合のヒープ・サイズの設定	55
メモリー不足エラーが発生した場合のヒープ・サイズの削減または増加	55
<b>第 16 章 共通の参照資料</b>	<b>57</b>
<b>第 17 章 z/OS 固有の参照情報</b>	<b>59</b>
<b>特記事項</b>	<b>61</b>
<b>索引</b>	<b>65</b>





---

## 表

1. 本書の外部にあるトピックへのリンクのプレフィックスとなるアイコン . . . . .	iv	3. メッセージング・エンジン・データ・ソース名	41
2. メッセージング・エンジン・スキーマ名	40	4. 新規データ・ソース名 . . . . .	42



---

## 第 1 章 概要

パフォーマンスを最適化するには、デフォルトのシステム設定を変更する必要があります。このセクションでは、システム調整時に検討するいくつかの領域を示します。その内容は、WebSphere Business Process Management 製品やその他の製品 (DB2 など) の調整です。これらの製品のそれぞれの資料には、パフォーマンス、キャパシティー・プランニング、および構成に関する情報が記載されています。

この資料では、さまざまな動作環境でのパフォーマンスの考慮事項について説明します。これらの問題が実際の製品の観点からすべて対応済みであるとする、付加的なレベルでのパフォーマンスへの影響は、これらの製品とこのセクションで説明する製品とのインターフェース部分に発生します。

システム管理者が使用できる構成パラメーターが多数存在します。このセクションでは、パフォーマンスに影響すると認められた特定のパラメーターをいくつか示しますが、使用可能なすべてのパラメーターを扱っているわけではありません。構成パラメーターおよび可能な設定の完全なリストについては、関連の製品資料を参照してください。

次のセクションでは、デプロイ済みのシステムを調整する場合に使用する方法について説明します。その後、主なコンポーネントとその関連調整概念を列挙している基本的な調整チェックリストを示します。

後続のサブセクションでは、調整について詳しく説明します。最初に、いくつかの調整パラメーターとその推奨設定値 (該当する場合) について説明し、最後にシステムの重要な領域を詳細に説明する拡張調整ガイドラインを示します。この章の説明内容に従うことで、満足できるパフォーマンスがすぐに得られることが保証されているわけではありませんが、これらのパラメーターの設定に誤りがあると、多くの場合、パフォーマンスの低下が予想されます。

このセクションの最後の数トピックでは、特定の構成を調整する場合に役立つ関連資料への参照先を示します。



---

## 第 2 章 パフォーマンス調整の方法

WebSphere® Business Process Management 環境のパフォーマンス調整を行う場合は、システム全体にわたる手法を採用することを推奨します。システムのパフォーマンス調整には、訓練と経験が必要であり、すべてを完全に説明することは困難です。調整の主要な側面のうち特に重要ないくつかの事項について説明します。

重要な点として、調整にはデプロイメント・トポロジーのすべての要素が含まれることに注意してください。

- 物理的なハードウェア接続形態の選択項目
- オペレーティング・システムのパラメーター調整
- WebSphere Process Server、WebSphere Application Server、およびメッセージング・エンジンの調整

調整の方法は、以下に示すように反復ループとして非常に簡単に開始できます。

- 一連の適切な初期パラメーター設定値を選択する
- システムを稼働させる
- システムをモニターして、パフォーマンスが制限されるかどうかを示す測定基準を取り込む
- モニター・データを使用して、より詳細な調整変更点を導き出す
- 完了するまで繰り返す

今度は、以下の各項目を順に確認します。

- 一連の適切な初期パラメーター設定値を選択する。
- 系統立った方法でパラメーターを設定するため、調整チェックリストを使用する。
- システムをモニターして、システムの正常性を調べ、さらに調整が必要かどうかを確認する。以下の項目をモニターする必要があります。
  - Web サーバーやデータベース・サーバーなどのフロントエンド・サーバーおよびバックエンド・サーバーを含むトポロジーでの各物理システム
  - CPU 使用率、メモリー使用率、ディスク使用率、ネットワーク使用率。SMF や RMF™ など、関連の z/OS® ツールを使用
  - 物理的システム (例えば、WebSphere Process Server コントローラー、サーバント、付属品) 上で開始された各 JVM プロセス。
  - verbosegc 統計データ
- 各 WebSphere Process Server サーバーに対して、TPV (Tivoli® Performance Viewer) を使用して以下の項目をモニターします。
  - 各データ・ソースについて、データ接続プールの使用率
  - 各 JCA リソース・コネクタについて、接続プールおよびセッション・プールの統計データ
  - 各スレッド・プール (Web コンテナ、デフォルト、作業マネージャー) について、スレッド・プールの使用率

- モニター・データを使用して、より詳細な調整変更点を導き出します。

このトピックは、スキルや経験を要する広範なものです。この段階の調整では、収集したモニター・データの分析、パフォーマンスのボトルネックの検出、および追加調整の実行を行う分析者が必要です。この段階の調整についての重要な特性は、前の段階で収集したモニター・データを主体として形成されます。

パフォーマンスのボトルネックの例は以下のとおりですが、これ以外にもあります。

- CPU、ディスク、メモリーなどの物理リソースの過剰な使用。こうした問題は、物理リソースを追加するか、使用可能なリソース全体にわたってより均一に負荷を再均衡化することで解決できます。
- 仮想リソースの過剰な使用。仮想リソースの例としては、ヒープ・メモリー、接続プール、およびスレッド・プールがあります。これらについては、調整パラメーターを使用してボトルネックを除去する必要があります。

---

## 第 3 章 調整チェックリスト

システムを調整する場合は、以下のチェックリストを使用します。

### 共通

- トレースおよびモニターは可能な限りオフにする
- データベースをデフォルトの Derby から DB2 for z/OS などの高性能 DBMS に移動する
- 実用的な場面では、セキュリティーを有効化しない
- パフォーマンス測定のために適切なハードウェア構成を使用する。反復可能なパフォーマンス・テストを実行できるようにするため、環境は可能な限り専用リソースで構成する必要がある。共用リソースや、可用性が他のシステムの影響を受ける可能性があるリソースは使用しない。
- 実動サーバーを開発モードで使用しない
- 単体テスト環境 (UTE) はパフォーマンス測定には使用しない
- MDB アクティベーション・スペックを構成する
- クラスタリングに合わせて構成する (可能な場合)
- スレッド・プール・サイズを構成する
- データ・ソース (接続プール・サイズ、準備済みステートメントのキャッシュ・サイズ) を構成する
- CEI エミッターでの妥当性検査を無効にする

### メッセージングおよびメッセージ・バインディング

- アクティベーション・スペックを最適化する (JMS)
- キュー接続ファクトリーを最適化する (JMS)
- 接続プール・サイズを構成する (JMS)
- リスナー・ポート構成を最適化する (MQJMS、MQ)
- SIBus データ・バッファ・サイズを構成する

### データベース

- データベース表スペースおよびログを高速のディスク・サブシステムに配置する
- 表スペース・コンテナとは異なる装置にログを配置する
- 表の現行索引を保守する
- データベース統計の更新
- ログ・ファイル・サイズを正しく設定する
- バッファ・プール・サイズの最適化

### Java

- (新世代領域の) ヒープ・サイズを設定して、メモリーを効率的に管理する
- 適切なガーベッジ・コレクション・ポリシーを選択する



---

## 第 4 章 調整パラメーター

このセクションでは、このレポートに記載されている製品の調整時によく使用するパフォーマンス調整パラメーターを示します。一部のフラグまたはチェック・ボックスは製品のすべてまたは一部に共通ですが、その他のフラグまたはチェック・ボックスは特定の製品に固有のものです。

---

### トレースおよびロギングのフラグ

ほとんどのトレース機能およびモニター機能は、管理コンソールを使用して制御されます。PMI モニター、ロギング、およびトレースに対して適切なレベルのトレース/モニターが設定されていることを管理コンソールを使用して確認してください。

#### このタスクについて

トレース機能をコンソールから調整するには、以下の手順を実行します。

##### 手順

1. 管理コンソールを開き、「トラブルシューティング」>「ロギングおよびトレース」>「(サーバー名)」>「ログ詳細レベルの変更」を選択します。
2. 構成およびランタイムの両方を **\*=all=disabled** に設定します。
3. トレース機能を z/OS コンソールから使用不可にするには、サーバーに対して `modify` コマンドを実行します。例えば、z/OS コンソールからすべての Java トレースを使用不可にするには、コマンド `f (server name) ,tracejava=*=all=disabled` を実行します。
4. モニター機能は管理コンソールから制御できます。モニター機能をコンソールから使用不可にするには、「モニターおよびチューニング」>「Performance Monitoring Infrastructure (PMI)」>「(サーバー名)」を選択し、「なし」を選択します。

---

## 64 ビット・モード

WebSphere Process Server 6.2.0 は WebSphere Application Server for z/OS 6.2.0 上で動作しますが、64 ビット・モードで動作するように構成できます。64 ビット・モードをアクティブ化すると、パフォーマンスのオーバーヘッドが発生します。記述されている計測結果の一部では、64 ビット・モードで実行した場合に大幅な性能低下が認められ、しかも明らかなメリットはありませんでした。性能低下時のガーベッジ・コレクション統計データによると、Java ヒープは影響を受けていないことが判明したため、31 ビット・モードで利用可能だった値よりヒープ・サイズを増やしたことにメリットはありませんでした。

ヒープが影響を受けている場合 (例えば、並行性の高い処理を行う場合や大量のオブジェクト/メッセージを処理する場合) は、64 ビット・モードにすると、一定の負荷軽減を実感できるポイントまでヒープ・サイズを増加することができます。ただし、この点について、インフォメーション・センターのこの調整セクションでは評価されていません。

64 ビット・モードをアクティブ化するには、サーバーの一般プロパティーで「**64 ビット JVM モードでの実行**」オプションを設定することが必要です。さらに、サーバーの z/OS start コマンドに **AMODE=64** パラメーターを指定する必要があります。詳しくは、『MVS コンソールからのサーバーの開始』を参照してください。

---

## Java 調整パラメーター

このセクションでは、よく使用されるいくつかの Java 仮想マシン (JVM) 調整パラメーターを示します。

完全なリストについては、「WAS for z/OS V6.1 Tuning Guide」の JVM 調整のセクションを参照してください。

JVM 管理パネルにアクセスするには、管理コンソールを開き、「サーバー」>「アプリケーション・サーバー」>「(ご使用のサーバー名)」>「サーバー・インフラストラクチャー」>「Java およびプロセス管理」>「プロセス定義」>「(制御または付属またはサーバント)」>「追加プロパティー」>「Java 仮想マシン」を選択します。

## Java GC ポリシー

デフォルトのガーベッジ・コレクション (GC) アルゴリズムは、Mark-Sweep-Compact です。場合によっては、次のセクションで説明するように、世代別並行 (gencon) アルゴリズムが、新世代領域の調整済みヒープ・サイズで良好なパフォーマンスを発揮します。

GC ポリシーを gencon に変更するには、「Java 仮想マシン」管理パネルで、**-Xgcpolicy:gencon** を汎用 JVM 引数に追加します。

詳しくは、「WAS for z/OS V6.1 Tuning Guide」で、ストレージおよび JVM の調整に関する情報を参照してください。

## Java ヒープ・サイズ

ガーベッジ・コレクション (GC) とは、JVM ヒープの部分を再利用できるように未使用のオブジェクトを解放する処理のことです。Java 言語仕様では、明示的な delete() または free() バイトコードが提供されないため、アクティブな参照データがなくなったオブジェクトを検出して削除し、そのスペースを再利用するために解放することが非常に重要です。

ガーベッジ・コレクションは、メモリーの要求があると自動的に起動します。例えば、オブジェクトが作成され、ヒープ内で利用できる空きメモリーでは要求をすぐに満たすことができない場合 (割り振り失敗) が該当します。ガーベッジ・コレクションは、Java クラス・ライブラリーである System.gc() 呼び出しを介してプログラムで起動することもできます。この場合には、ガーベッジ・コレクションが同期した状態ですぐに実行されます。

実稼働環境では、適切な Java ヒープ・サイズを使用してください。デフォルトの Java ヒープ・サイズを変更するには、「Java 仮想マシン」管理パネルで、「初期ヒープ・サイズ」および「最大ヒープ・サイズ」を明示的に設定します。

世代別並行ガーベッジ・コレクターを使用すると、Java ヒープは新規オブジェクトが割り振られる新領域 (新世代領域) と、存続時間の長いオブジェクトが存在する旧領域 (旧世代スペース) に分割されます。総ヒープ・サイズは、新領域と旧世代スペースの合計になります。新領域のサイズは、総ヒープ・サイズとは独立して設定できます。通常、新領域のサイズは総ヒープ・サイズの約  $\frac{1}{2}$  に設定する必要があります。関連パラメーターは以下のとおりです。

- -Xmns<サイズ> : 初期の新領域サイズ
- -Xmnx<サイズ> : 最大の新領域サイズ
- -Xmn<サイズ> : 固定の新領域サイズ

---

## ワークロード・プロファイル

サーバント・スレッドの数を構成するには、サーバーのワークロード・プロファイルを設定します。各プロファイル設定のスレッド数は、使用可能なプロセッサの数を含む数式を使用して求めます。

詳しくは、WebSphere for z/OS インフォメーション・センターを参照してください。WebSphere Process Server for z/OS サーバーにスレッドを割り振るときは注意が必要です。z/OS の場合は、ワークロードに基づいて調べる必要がある、ISOLATE、IOBOUND、CPUBOUND、または LONGWAIT の標準スレッド設定があります。

ワークロード・プロファイルを設定するには、管理コンソールを開き、「サーバー」>「アプリケーション・サーバー」>「(サーバー名)」>「コンテナ設定」-「コンテナ・サービス」-「ORB サービス」>「z/OS 追加設定」を選択して、ドロップダウン・リストからワークロード・プロファイルを選択します。

---

## ワークロード・マネージャー・サービス・クラス

z/OS オペレーティング・システムでは、事前に割り当てられているワークロード・マネージャー・サービス・クラスに基づいて要求を処理するために、マシン・リソースが割り当てられます。要求を分類して、該当するレベルのサービスを受け取るため、ワークロード・マネージャーを必ず事前に構成しておくようにしてください。

ワークロードは、CB というワークロード・マネージャーのサブシステム・タイプを使用して分類できます。

ワークロードをきめ細かく分類するには、ワークロード分類ファイルを使用します。詳しくは、WebSphere Application Server インフォメーション・センターを参照してください。

---

## MDB ActivationSpec

JMS アクティベーション・スペックは、MDB onMessage メソッドを駆動するメッセージ・キューにアプリケーション MDB をバインドするときに使用します。特に注目に値する 2 つのアクティベーション・スペック・パラメーターは、最大バッチ・サイズおよび最大並行エンドポイント数です。

各 JMS エクスポート・コンポーネントには、MDB とそれに対応する ActivationSpec (JNDI 名: モジュール名/エクスポート・コンポーネント名\_AS) があります。最大並行エンドポイント数では、特定の MDB が並行して処理できるメッセージの数が指定されます。ただし、アクティブな MDB インスタンスは、それぞれコンテナ・スレッド・プールから実行されることに注意してください。JMS エクスポート MDB のデフォルト値は 10 です。これは、JMS キューから MDB スレッドへ同時に配信できる BO が最大で 10 であることを意味します。

アクティベーション・スペックの最大バッチ・サイズもパフォーマンスに影響します。デフォルト値は 1 です。最大バッチ・サイズの値は、1 回のステップで、メッセージング層から取り出されてアプリケーション層に配信されるメッセージの数を指定しています (これは、この処理が 1 回のトランザクションで実行されるという意味ではないため、この設定値がトランザクション対象範囲に影響することはありません)。長時間実行するビジネス・プロセスに関連付けられているアクティベーション・スペックでこの値を大きくすると、MDB メッセージ処理の効率を高めることができます。

アクティベーション・スペック・パラメーターを構成するには、管理コンソールで「リソース」>「JMS」>「アクティベーション・スペック」>「(ActivationSpec 名)」を選択します。

---

## MQ リスナー・ポート

MQ または MQJMS バインディングの場合は、WebSphere Process Server または WebSphere Enterprise Service Bus へのインバウンド・メッセージの配信を構成するためにリスナー・ポートを使用します。リスナー・ポートは、アプリケーション MDB をメッセージ・キューにバインドする場合の JMS アクティベーション・スペックに相当します。リスナー・ポートには、WAS 管理コンソールの「メッセージ・リスナー・サービス」部分で構成できるパラメーターと同様のパラメーターがあります。

「最大セッション数」パラメーターは、JMS アクティベーション・スペックの「最大並行エンドポイント数」に相当し、特定のリスナー・ポートでの MDB 処理の並行性のレベルを指定します。

「最大メッセージ数」パラメーターは、MDB の onMessage メソッドを同じコンテキスト内で何回呼び出すことができるかを指定します。

「最大セッション数」の値は、メッセージ・リスナーのスレッド・プール値よりも小さくする必要があります。このパラメーターを設定する場合は、MDB スロットルの効果も考慮に入れる必要があります。

「リスナー・ポート」パラメーターを設定するには、管理コンソールを開き、「サーバー」>「アプリケーション・サーバー」>「(サーバー名)」>「メッセージ・リスナー・サービス」>「リスナー・ポート」>「(リスナー・ポート名)」を選択します。

---

## MDB スロットル

WebSphere Application Server for z/OS では、MDB 処理のコントロールの 1 つとして、MDB スロットルがあります。これを使用する目的は、WebSphere Application Server サーバントに存在する作業スレッドによる処理待ちのメッセージで z/OS WLM 作業キューが満杯にならないようにすることです。MDB スロットルは、WLM 要求キューでの処理対象メッセージのバックログが大きくなりすぎないように、メッセージ・リスナー・ポートによる JMS キュー (またはトピック) の先読み範囲を制限する役割を果たします。

MDB スロットルの上限しきい値および下限しきい値は、リスナー・ポートの「最大セッション数」パラメーターの値で決まります。上限しきい値は、リスナー・ポートの「最大セッション数」の値に設定されます。この数のメッセージが WLM 作業要求キューに入ると、並行要求の数が下限しきい値 (デフォルトでは「最大セッション数」の値の 50%) に低下するまで、追加の要求がキューに入ることはありません。

WebSphere Application Server サーバントに活動停止中のスレッドがある一方で、追加の要求がキューに入るのを MDB スロットルがブロックしている状態を回避するには、「最大セッション数」の値を、サーバントにある作業スレッドの数の 2 倍に設定することを推奨します。

「最大セッション数」の値を設定するには、管理コンソールを開き、「サーバー」>「アプリケーション・サーバー」>「(サーバー名)」>「メッセージ・リスナー・サービス」>「リスナー・ポート」>「(リスナー・ポート名)」を選択します。

---

## スレッド・プール・サイズ

WebSphere では、スレッド・プールを使用して並行タスクを管理します。スレッド・プールの「最大サイズ」プロパティを設定するには、管理コンソールを開き、「サーバー」>「アプリケーション・サーバー」>「(サーバー名)」>「追加プロパティ」>「スレッド・プール」>「(スレッド・プール名)」を選択します。

通常は、以下のスレッド・プールを調整する必要があります。

- デフォルト
- ORB.thread.pool
- WebContaineran の対象箇所

作業マネージャーを使用して、「スレッドの最大数」の値を、スレッド・プールからスレッドが枯渇することのないように十分大きい値に設定します。この値を設定するには、管理コンソールを開き、「非同期 Bean」>「作業マネージャー」を選択します。

WebSphere MQ メッセージは、メッセージ・リスナーを使用して MDB に対するアクションを呼び出します。MDB インスタンスとリスナー・ポートを並行して実行できるようにするには、メッセージ・リスナーのスレッド・プールを構成する必要があります。このスレッド・プールは、リスナー・ポートの最大セッション数の値をサポートできるように大きくする必要があります。スレッド・プールの設定はサ

ーバー全体が対象であり、設定するには、管理コンソールで「アプリケーション・サーバー」>「(サーバー名)」>「メッセージ・リスナー・サービス」>「スレッド・プール」を選択します。

並行性が不十分であることを示す症状の 1 つは、プロセッサの不使用状態です。並行性を変更して、プロセッサの使用率およびスループットが最大になるようにしてください。十分な数の作業スレッドを確保する方法については、「WebSphere Application Server for z/OS Tuning Guide」の『Tuning for Subsystems』を参照してください。

---

## JMS 接続プール・サイズ

最大接続プール・サイズを大きくすると、並行性を高めることができます。ただし、1 つのデプロイ済みアプリケーションだけが特定の接続ファクトリーから接続を獲得する場合は、アクティベーション・スペックで構成した最大並行性よりキュー接続プール・サイズの最大値を大きくしてもメリットはありません。

最大接続プール・サイズでは、このプールで作成できる物理接続の最大数が指定されます。これらはバックエンド・リソースへの物理接続です。この数値に到達すると、新規の物理接続は作成されなくなるため、現在使用中の物理接続がプールに返されるまで、または ConnectionWaitTimeout 例外が発生するまで、要求側は待機します。

例えば、「最大接続数」の値を 5 に設定していて、5 つの物理接続が使用中の場合、プール・マネージャーは、「接続タイムアウト」に指定されている時間だけ、物理接続が解放されるのを待ちます。下位層のリソースへの接続を待機しているスレッドは、その接続が解放され、プール・マネージャーによってスレッドに割り振られるまでブロックされます。指定のインターバル内に接続が解放されない場合は、ConnectionWaitTimeout 例外が発生します。

「最大接続数」を 0 に設定すると、接続プールのサイズを無限に大きくすることができます。この設定にすると、「接続タイムアウト」の値は無視されます。

リスナー・ポートが関係している場合は、接続ファクトリーのプール・サイズを決めるときにそのことを考慮する必要があります。接続ファクトリーの接続プールはリスナー・ポートの数より大きくする必要があり、接続ファクトリーのセッション・プールは、この接続を使用するすべてのリスナー・ポートの最大セッション数の値をカバーするのに十分な大きさにする必要があります。

最大接続プール・サイズは、管理コンソールの「リソース」で、以下のようにして設定できます。

- 「リソース」>「JMS」>「接続ファクトリー」>「(ファクトリー名)」
- 「リソース」>「JMS」>「キュー接続ファクトリー」>「(ファクトリー名)」

「接続ファクトリー」管理パネルで、「追加プロパティ」>「接続プール・プロパティ」を開きます。接続プールの最大サイズについて「最大接続数」プロパティを設定します。

---

## データ・ソースの接続プール・サイズ

DB2® for z/OS への最小接続数および最大接続数、および接続待ちのタイムアウト値は調整できます。 :

管理コンソールを開き、「リソース」>「JDBC」>「データ・ソース」>「(データ・ソース名)」を選択します。

データ・ソース接続プールの最大サイズは、「最大接続数」プロパティの値で制限されています。この値を構成するには、管理コンソールを開き、「データ・ソース」パネルで「追加プロパティ」->「接続プール・プロパティ」を選択します。

通常は以下のデータ・ソースを調整する必要があります。

- BPE DB 向けのデータ・ソース
- SCA アプリケーション・バス ME データ・ソース
- SCA システム・バス ME データ・ソース
- CEI バス ME データ・ソース
- ESB ロガー・メディエーション・データ・ソース
- WBI データ・ソース
- イベント・データ・ソースおよびイベント・カタログ・データ・ソース

---

## データ・ソース準備済みステートメント・キャッシュのサイズ

準備済みステートメント・キャッシュは、準備済み SQL ステートメントおよび呼び出し可能ステートメントの処理を最適化する役割を果たします。

キャッシュのサイズは、キャッシュ項目が廃棄されないように調整できます。キャッシュのサイズを調整するには、管理コンソールを開き、「リソース」>「JDBC」>「データ・ソース」>「(データ・ソース名)」>「WebSphere Application Server データ・ソース・プロパティ」を選択します。

BPEL macroflow (長時間動作するプロセス) では、プロセスに関連するデータを持続させるため、BPEDB データ・ソースが大量に使用されます。さまざまなデータを持続させると、さまざまなステートメントを多数使用することになるため、データ・ソースのキャッシュのデフォルトの容量をはるかに超えてしまいます。この結果、過剰な数のキャッシュ不足が発生して、キャッシュの効果が失われてしまいます。この問題は、キャッシュのサイズを増やすことで解決できます。

WebSphere Process Server の BPC コンテナでは、準備済みステートメントが大規模に使用されます。このデータ・ソースのステートメント・キャッシュ・サイズは、128 以上にする必要があります。データ・ソースのキャッシュ廃棄数は、管理コンソールの Tivoli Performance Viewer を使用するとモニターできます。

---

## メッセージング・エンジン・プロパティ

2 つのメッセージング・エンジン・カスタム・プロパティは、メッセージング・エンジンのパフォーマンスに影響します。

`DiscardableDataBufferSize` は、ベスト・エフォート型の非持続メッセージを処理するときに使用するデータ・バッファのサイズをバイト単位で表します。廃棄可能なデータ・バッファの目的は、メッセージ・データをメモリー内に保持することです。これは、このサービス品質の場合には、メッセージ・データがデータ・ストアに書き込まれることがないためです。サイズが大きすぎてこのバッファには収まらないメッセージは廃棄されます。

`CachedDataBufferSize` は、ベスト・エフォート型の非持続メッセージ以外のすべてのメッセージを処理するときに使用するデータ・バッファのサイズをバイト単位で表します。このキャッシュ・データ・バッファの目的は、メモリー内のデータをキャッシュに格納することにより、パフォーマンスを最適化することです。これらのデータは、キャッシュに格納されない場合にはデータ・ストアから読み取る必要があります。

`DiscardableDataBufferSize` および `CachedDataBufferSize` を設定するには、管理コンソールを開き、「サービス統合」>「バス」>「(バス名)」>「メッセージング・エンジン」>「(メッセージング・エンジン名)」>「カスタム・プロパティ」を選択します。

- `sib.msgstore.discardableDataBufferSize`
- `sib.msgstore.cachedDataBufferSize`

---

## セキュリティの最小限化

WebSphere Application Server のセキュリティ機能は、リソースの無許可での使用に対する重要な保護機能を提供します。セキュリティを完全に無効にすることは推奨されないため、さまざまなセキュリティ・メカニズムが使用可能になっており、それぞれが処理オーバーヘッドを増加させています。

WebSphere Application Server セキュリティ機能のどれをアクティブにするかは、慎重に検討する必要があります。これらは管理コンソールのセキュリティのセクションで構成されますが、セキュリティのいくつかの側面については、アプリケーション・アセンブリー時やデプロイメント時に構成されます。

---

## Network Deployment での自動同期の無効化

Network Deployment 構成では、ノード・エージェントがデプロイメント・マネージャー構成データをノードの構成データと同期化します。この同期の発生頻度が高すぎると、全体的なシステム・パフォーマンスに影響することがあります。

Network Deployment の同期頻度を調整するか、自動化された同期を無効にするには、管理コンソールを開き、「システム管理」>「ノード・エージェント」>「`nodeagent`」>「ファイル同期化サービス」を選択します。

---

## 第 5 章 ビジネス・プロセスの調整

このタスクを使用して、ビジネス・プロセスのパフォーマンスを改善します。

### 始める前に

ビジネス・プロセスが正常に実行されたら、このタスクを実行してパフォーマンスを改善することができます。

### 手順

1. 基本的なパフォーマンスの測定方法、および最適化する測定値を定義します。

例えば、ビジネス・アプリケーションによっては、負荷のピーク時にエンド・ユーザーのために応答時間を短縮することが望ましいことがあります。その他のアプリケーションの場合は、システムで可能なトランザクションの処理速度のほうが、各トランザクションの実際の期間よりも重要になります。

2. 基本的な測定を行います。

アプリケーションのチューニングに適した負荷、時刻、および曜日の条件下で、基本的な測定を行います。通常、最も重要な基本的な測定値は、スループットおよび応答時間です。例えば CPU 負荷が 100% に、ディスク I/O が最大値に、またはネットワーク I/O が 100% に達するなど、特定のボトルネック容量に達した後で、スループット値が測定されます。信頼性の高い応答時間の値は、単一のプロセス・インスタンスで、サーバー使用率が低いときに、最も適切に測定されます。

3. アプリケーションを調整します。

アプリケーションには複数のプロセスを含めることができます。Microflow は長期実行プロセスよりもパフォーマンスが優れているため、パーシスタンスが不要な状況で、機能を 1 つのトランザクションの 1 つのスレッドで処理できる場合は、長期実行プロセスではなく Microflow をモデル化する方式を選択してください。また、長期実行プロセスの分岐を Microflow に分離することを検討してください。さらに、同期サービス呼び出しの速度は通常、非同期サービス呼び出しより速くなります。そのためパフォーマンス上の理由で、同期サービス呼び出しが長期実行プロセスでのデフォルトの動作でないとしても、同期サービス呼び出しを使用してください。

長期実行プロセスでは、トランザクション境界を変更できます。ほとんどの場合、トランザクション境界の数を減らすことによって、パフォーマンスを向上できます。ただし、トランザクション境界の最適の数はパフォーマンス・テストによってのみ検出できます。データの直列化と非直列化にもコストがかかるため、アクティビティを直列化する代わりにプロセス内で並列実行パスを使用し、プロセスの一部であるデータのサイズや複雑さを最小化することも検討してください。さらに、発行されるイベントの数を最小にします。

4. プロセスを調整します。

アプリケーションが長期間にわたって実行するプロセスと microflow のどちらを使用するかに応じて、以下のステップを実行します。

- 長期間にわたって実行するプロセスを調整するには、『長期間にわたって実行するプロセスのチューニング』で説明されているステップを実行します。これらのプロセスは、長時間実行が継続する傾向がありますが、イベントまたは人との対話で中断される場合もあります。したがってそのパフォーマンスは、Business Process Choreographer データベースとメッセージング・サービスのパフォーマンスによって異なります。
  - microflow を調整するには、19 ページの『microflow のチューニング』で説明されているステップを実行します。このプロセスは、どちらかといえば短期間のみ実行するプロセスです。このプロセスは、監査ロギング用のみにデータベースを使用して (監査ロギングが使用可能な場合)、テンプレート情報を取得します。このプロセスは、人との対話を必要としません。
5. 除去可能なパフォーマンスのボトルネックがないかどうか、現在の構成を検討します。

考慮すべき可能性のある項目は以下のとおりです。

- より多くのプロセッサ、メモリー、およびより高速なディスクをインストールします。
  - データベースのログをデータとは異なる物理ディスクに保管して、データをいくつかのディスクに分散させます。
  - 最適なパフォーマンスを得るために、Cloudscape® ではなく、DB2を使用します。
6. 基本的な測定項目に関して、同様の負荷条件でベンチマーク測定を繰り返します。

アプリケーションのパフォーマンス測定の永続レコードを保持して、パフォーマンスにおける将来の変更点を客観的に測定します。

## タスクの結果

ビジネス・プロセスは、はっきりと分かるほど高速に実行されるように構成されます。

---

## 長期間にわたって実行するプロセスのチューニング

このタスクを使用して、長期間にわたって実行するビジネス・プロセスのパフォーマンスを改善します。

### このタスクについて

長期間にわたって実行するプロセスには、ユーザーとの対話、非同期呼び出し、複数の受信、ピック、イベント・ハンドラーなどがあります。これらのプロセスでは、永続状態を保管するためにデータベースとメッセージング・サブシステムを使用します。以下のトピックでは、長期間にわたって実行するプロセスのパフォーマンスを向上させる方法を説明します。

## メッセージング・エンジンの設定計画

メッセージング・エンジンの初期設定の計画を立てるには、このタスクを使用します。

### このタスクについて

長期実行プロセスの最善のパフォーマンスを引き出すには、永続メッセージのパフォーマンスが最大化するようにメッセージング・システムを調整します。データ・ストア・バックエンド・タイプの場合、ファイル・ストアは適切に実行しているため、そのファイル・ストアが優先されます。ご使用の環境がクラスターで稼働しており、ファイル・ストアを使用できない場合は、データベース・データ・ストアを使用します。

WebSphere Application Server のサービス統合機能を使用する場合は、WebSphere Application Server for z/OS インフォメーション・センターのサービス統合の調整のプロパティの設定にある説明に従って、メッセージング・エンジン用のデータ・ストアをセットアップおよび調整してください。

### タスクの結果

ご使用のメッセージング・エンジンは最適な状態で作動しています。

## メッセージング・プロバイダーの細密チューニング

メッセージング・プロバイダーのパフォーマンスを向上させるには、このタスクを使用します。

### 手順

WebSphere Application Server のサービス統合機能を使用する場合は、WebSphere Application Server インフォメーション・センターにあるを参照してください。

### タスクの結果

メッセージング・プロバイダーのパフォーマンスが向上します。

## ビジネス・プロセス・ナビゲーションのパフォーマンスの向上

パフォーマンスの最適化を使用可能にし、各種構成パラメーターを調整することによって、長期実行プロセスのパフォーマンスを調整することができます。

### このタスクについて

長期実行プロセスは、複数のトランザクションにわたっています。デフォルトでは、トランザクションは Java™ Messaging Service (JMS) メッセージによってトリガーされます。プロセス・ナビゲーションのパフォーマンスを向上させるには、Business Flow Manager を構成して、JMS メッセージの代わりにトランザクションをトリガーするために作業マネージャー・ベースの実装を使用することができます。使用するナビゲーション・モードが JMS か作業マネージャーかに関係なく、トランザクション間キャッシュのサイズを調整できます。

以下に、2 つのプロセス・ナビゲーション・モードの特性を要約します。

## JMS メッセージ・ベースのナビゲーション

プロセス・ナビゲーションのメッセージ駆動型 Bean (MDB) によって制御される JMS メッセージで処理されるプロセス・ナビゲーション。

- メッセージング・エンジンがアプリケーションに対してローカルになるようにトポロジーがセットアップされている場合、プロセスはサーバー・アフィニティを維持してナビゲートします。ただし、非同期メッセージやヒューマン・タスクなどの外部イベントによってプロセスが起動される場合は除きます。
- 単一アプリケーション・クラスター内の複数のサーバーが単一のリモート・メッセージング・エンジンを使用するようにトポロジーがセットアップされている場合、プロセス内のナビゲーションはクラスター内の複数のサーバーに分散されます。

## 作業マネージャー・ベースのナビゲーション

プロセス・ナビゲーションは、作業マネージャーが制御するスレッド・プールによって処理されます。プロセス・インスタンスの正常なナビゲーションは、サーバー・アフィニティを維持して完了します。

トランザクションの整合性を確保するために、ナビゲーション・ステップをトリガーするメッセージは Business Process Choreographer データベースに保管されます。バックグラウンド・リカバリー・スレッドは、定期的にこれらのメッセージをスキャンし、指定された最大経過時間よりも古いメッセージが存在する場合、プロセス・ナビゲーション MDB によって選出させるためにこれらのメッセージを JMS キューに送信します。Business Process Choreographer により、各メッセージが 1 回だけ実行されることが保証されます。

エラーが発生し、ナビゲーション・ステップがロールバックされた場合は、プロセスのナビゲーションは JMS 制御ナビゲーションに戻ります。

サーバー・アフィニティは、プロセス・インスタンス内のナビゲーションが単一の WebSphere Application Server で行われることを意味します (非同期サービスが起動された場合、待機状態またはタイムアウト状態が発生した場合、receive アクティビティまたは pick アクティビティがアクティブ化された場合、ヒューマン・タスクが実行された場合を除く)。これらのイベントの結果、あるプロセス内のナビゲーションが別の WebSphere Application Server で続行される場合があります。

## 手順

1. Business Flow Manager を構成して、作業マネージャー・ベースのプロセス・ナビゲーションを使用します。

管理コンソールで、以下のステップを実行します。

- a. 「サーバー」 → 「クラスター」 → 「*cluster\_name*」または「サーバー」 → 「アプリケーション・サーバー」 → 「*server\_name*」にナビゲートし、「ビジネス・インテグレーション」の下で「Business Process Choreographer」を展開し、「Business Flow Manager の構成」 → 「構成」をクリックします。
- b. 「高度なパフォーマンスの最適化を使用可能にする」オプションを選択します。以下の構成パラメーターの値を変更できます。
  - メッセージ・プール・サイズ

- 停止したメッセージの最大存続期間
  - 停止したメッセージのリカバリー間隔
  - スレッドでの最大処理時間
  - トランザクション間キャッシュ・サイズ
2. オプション: ワークフロー・マネージャーが使用可能なスレッドの最大数を増やします。

Business Flow Manager では、内部処理のために 2 つのスレッドが必要です。残りのスレッドは、プロセス・ナビゲーションで使用可能です。最初に各プロセス・ナビゲーターにスレッドを 1 つ追加します。スレッド・プール・サイズを増加させる場合は、Business Process Choreographer データベース (BPEDB) および接続ファクトリー (BPECFC) の接続プール・サイズも増加させる必要があります。

スレッドの最大数を変更するには、管理コンソールを使用して以下を実行します。

- a. 「リソース」 → 「非同期 Bean」 → 「作業マネージャー」 → 「BPENavigationWorkManager」にナビゲートします。
  - b. 「スレッド・プールのプロパティ」の下、「スレッドの最大数」の値を変更します。
  - c. 「作業要求キュー・サイズ」の値を「スレッドの最大数」と同じ値に設定します。
3. 変更を保管します。
  4. 変更を有効にするために、サーバーを再始動します。

## タスクの結果

これで作業マネージャーがプロセス・ナビゲーションを制御します。

---

## microflow のチューニング

microflow のパフォーマンスを向上させるには、このタスクを使用します。

### このタスクについて

microflow は、ユーザーとの対話や永続メッセージングのサポートなしに、メモリー内で実行されます。監査ロギングまたは Common Event Infrastructure (CEI) が microflow で使用可能になっている場合のみ、データベース・アクセスが必要です。microflow の処理が、単一スレッド内で行われ、また通常は単一トランザクション内で行われます。microflows のパフォーマンスは、主として呼び出されるサービスによって決まります。ただし、サーバーが使用可能なメモリーが小さすぎる場合、microflow のパフォーマンスは低下します。

### 手順

1. Java 仮想マシン (JVM) ヒープ・サイズを調整します。

Java ヒープ・サイズを増やすことにより、microflow のスループットを向上させることができます。これは、ヒープ・サイズを大きくするほど、必要なガーベ

ジ・コレクション・サイクル数を減らすことができます。値は、ディスクへのヒープ・スワッピングを避けるため、十分に低い値に維持します。

2. JVM のガーベッジ・コレクションを調整します。世代別ガーベッジ・コレクター・ポリシーによって最適なスループットを実現できます。そのポリシーは、JVM 設定の汎用 JVM 引数としてアクティブ化されます。収集の初期値は、合計ヒープ・サイズの半分に設定してください。例えば、`-Xgcpolicy:gencon -Xmn512M` の場合は、ヒープ・サイズ 1024 MB に対応するポリシーがアクティブ化されます。
3. オブジェクト・リクエスト・ブローカー (ORB) のスレッド・プール・サイズを調整します。リモート・クライアントがサーバー・サイド ORB に接続する場合は、使用できるスレッドが十分あることを確認してください。「アプリケーション・サーバー」 → 「`server_name`」 → 「ORB サービス」 → 「z/OS 追加設定」 → 「ワークロード・プロファイル」にナビゲートします。
4. デフォルトのスレッド・プール・サイズを調整します。同時に実行できる `microflow` の数を増やすには、デフォルトのスレッド・プール・サイズを大きくする必要があります。値を変更するには、管理コンソールで「サーバー」 → 「アプリケーション・サーバー」 → 「`server_name`」 → 「プロパティの追加」 → 「スレッド・プール」 → 「デフォルト」にナビゲートします。

## タスクの結果

`microflow` は、現在の環境とロード条件の下で可能な限り高速に実行されます。

---

## 第 6 章 ヒューマン・タスクを含むビジネス・プロセスのチューニング

ヒューマン・タスクを含むビジネス・プロセスのパフォーマンスを向上させるには、さまざまな方法があります。

以下のトピックでは、ヒューマン・タスクを含むビジネス・プロセスを調整する方法を説明します。

---

### ヒューマン・タスクへの同時アクセス数の削減

複数のユーザーが同じヒューマン・タスクを要求しようとした場合、1 人のユーザーだけが要求に成功します。他のユーザーのアクセスは拒否されます。

1 つのヒューマン・タスクを要求できるのは、1 人のユーザーのみです。何人かのユーザーが同時に同じヒューマン・タスクで作業を行おうとすると、衝突が起きる可能性が増します。衝突が起きると、データベースやロールバックでのロック待機のため、遅延が発生します。衝突の発生を回避または削減する方法として、以下のものがあります。

- 同時アクセス数が多い場合、特定のヒューマン・タスクにアクセスできるユーザー数を制限します。
- インテリジェントな要求機構を使用して、クライアントからの不必要なヒューマン・タスクの照会をなくします。例えば、以下の手順のいずれかを実行します。
  - 最初の要求に失敗した場合、リストから別の項目を要求します。
  - ヒューマン・タスクを常に無作為に要求します。
  - メンバー数の少ないグループにタスクを割り当てるなどして、タスクの潜在的な所有者の数を減らします。
  - リストの検索に使用する照会にしきい値を指定して、タスク・リストのサイズを制限します。ヒット数を制限するフィルター処理の使用も検討します。タスクのプロパティをフィルターに掛けることによって、例えば、優先順位が 1 番のタスクまたは 24 時間以内に期限の切れるタスクのみを表示できます。インライン・タスクの場合、カスタム・プロパティまたは照会プロパティを使用するタスクに関連するビジネス・データをフィルターに掛けることもできます。このようなフィルター処理を行うには、タスク・リストを検索する照会に適切な WHERE 文節を指定する必要があります。
  - 動的担当者照会 (置換変数を使用する照会) を最小限に抑えるか、または回避します。
  - ヒューマン・タスクの照会にクライアント・キャッシュ機構を使用して、一度に複数の照会が実行されないようにします。

---

### 照会の応答時間の短縮

データベースが照会への応答に必要とする時間を短縮します。

カスタム・クライアントを使用する場合は、必ず照会でしきい値を設定してください。使用可能度の観点から、一般的に、何百、何千もの項目の検索は望ましいことではありません。データベースの操作数が増えるほど、タスクの完了までにかかる時間が長くなること、および人は一度に少数の結果しか管理できないことがその理由です。しきい値を指定すると、データベースの負荷やネットワーク・トラフィックを最小限に抑え、クライアントが確実にデータをすぐに提供できるようになります。

多数の項目を戻す照会の処理方法としてよりよい方法は、照会を作成し直して、戻される結果セットの項目数を減らすことです。これを行うには、特定のプロセス・インスタンスのみについて作業項目を照会するか、一定の日付を持つ作業項目のみを照会します。

フィルター基準を使用して、照会結果を削減することもできます。

---

## 全テーブルのスキャンニングの回避

照会アプリケーション・プログラミング・インターフェース (API) を使用して、データベース内のオブジェクトをリストする場合、フィルター操作を指定して、取得する結果の範囲を絞ることができます。これらのフィルターに、オブジェクト属性の値と範囲を指定することができます。

データベース照会の処理時に、フィルター情報が変換され、Structured Query Language (SQL) ステートメントの WHERE 文節に格納されます。この WHERE 文節は、影響を受けるデータベース表内の列名にオブジェクト属性をマップします。

照会に索引付きテーブル列への変換を行わないフィルターが指定されている場合、SQL ステートメントによってテーブルがスキャンされることとなります。このスキャンニングによって、パフォーマンスに悪影響があり、デッドロックが発生するリスクが高くなります。このパフォーマンスの悪影響は、1日に数回のみという程度であれば容認されますが、1分間に数回という頻度になれば、効率に悪影響を及ぼします。

このような状況では、カスタム索引でこの影響を劇的に抑えることができます。実際のお客様の状況で、カスタム索引により、API の応答時間を 25 秒から 300 ミリ秒に減らすことができました。データベース表の 724 000 行を読み取る代わりに、6 行を読み取るだけでよいのです。Business Process Choreographer データベースのインスタンス・テーブルに対して volatile フラグを定義すると、DB2 オプティマイザーが適切なデータ・アクセス・プランを決定する上で役立ちます。このフラグは、空のテーブルやほぼ空のテーブルであっても、常にテーブル・スキャンではなく索引が使用されることを指定します。

指定するフィルター基準により、一部の列が索引に含まれないように設定することができます。該当する場合、テーブル・スキャンを使用し、その結果照会のパフォーマンスが低下する場合は、例えば DB2 Explain を使用して、ステートメントのアクセス・パスを確認してください。必要であれば、新しい索引を定義してください。

## タスクおよびプロセス照会の最適化

タスク・リストおよびプロセス・リストを取得するための query および queryAll API 呼び出しを実行すると、複数のデータベース表の組み合わせを含む複雑な SQL 照会を生成できます。データ表示を最適化することは、パフォーマンス要件、特に複数のユーザーがタスク・リストに同時にアクセスするヒューマン・ワークフロー・アプリケーションのパフォーマンス要件に取り組むのに役立ちます。

### このタスクについて

Business Process Choreographer が照会用に調整される場合、応答時間は通常、高負荷であっても、適合サイズのシステム上のサブ秒の領域にあります。照会の応答時間を計算するには、標準のデータベース計算を適用できます。

大容量のヒューマン・ワークフローのシナリオは、照会テーブルでの使用に合わせて最適に調整されています。照会テーブルは、特定の照会に関係のある事前計算データのセットを提供します。例えば、照会プロパティは照会の実行時にデータベースによってタスクまたはプロセス・インスタンスと結合される必要があります。照会テーブルが使用される場合、こうした SQL 結合は照会実行時にさらに計算される必要はありません。

照会テーブルの実装および保守の取り組みは、標準のデータベース・チューニング技法と比べて大きくなります。照会テーブルを使用する前に、索引、ログ・ファイル配布、およびメモリーなどの、標準のデータベース最適化手法を注意深く検討してください。

テーブルを照会するための 2 つの方法がサポートされています。それは、実体化ビューとカスタム・テーブルです。実体化ビューとカスタム・テーブルのどちらを使用するかは、保守コスト、開発コスト、さらにタスクおよびプロセス・リストの照会によって戻されるデータの現行性に関する要件に基づいて決定してください。

- 実体化ビューは、非同期更新メカニズムを利用する場合に使用します。これは、最適の照会とプロセス・ナビゲーション・パフォーマンスを提供します。
  - 更新は、実体化ビューが使用される場合にのみ行われます。
  - セットアップ、使用、および保守が比較的簡単です。
  - アプリケーション・ソース・コードを変更しないで実装できます。
- カスタム・テーブルは、query または queryAll インターフェースを使用して、他のアプリケーションからのデータを標準照会に組み込む場合に使用します。さらに、カスタム・テーブルは、タスクおよびプロセス照会に必要なデータの表示を最適化するためにも使用できます。
  - データベース・トリガーまたはその他の技法を使用して、タスクおよびプロセス・リストの照会用に最適化されたカスタム・テーブルを同期して更新できます。
  - カスタム・テーブルで提供されたデータを照会するために、照会を変更する必要があります。



---

## 第 7 章 Business Process Choreographer Explorer の調整

以下の提案は、Business Process Choreographer Explorer のパフォーマンスを向上させるにはさまざまな方法があることを示しています。

### 手順

1. サーバーの最大ヒープ・サイズを増やすことを検討します。

Web クライアントは、システム上の負荷を自然に増やします。サーバーに接続されるクライアントが増えると、メモリー内に保持しなければならないオブジェクトが増えます。そのため、サーバーの最大ヒープ・サイズを増やすことを検討してください。これにより、アプリケーションの応答時間が改善され、アプリケーションと並行して作業できるユーザーの最大数が増えます。

2. Web コンテナ・スレッド・プールを調整します。

スレッド・プールのサイズおよびスレッド非アクティビティー・タイムアウトが Web コンテナのパフォーマンスに影響を与える可能性があります。これらの設定を変更するには、管理コンソールで「サーバー」 → 「アプリケーション・サーバー」 → 「*server\_name*」 → 「スレッド・プール」 → 「Web コンテナ」の領域にナビゲートします。

- a. 最大プール・サイズおよび最小プール・サイズを調整します。

Web クライアント・アプリケーションの HTTP 要求はすべて、Web コンテナ・スレッド・プールからのスレッドを使用して処理されます。Web クライアントのパフォーマンスを反映するように、最小および最大プール・サイズを調整できます。

プール内の最大スレッドの数は、アプリケーション・サーバーが同時に処理できる要求の数を表すわけではありません。プール内のすべてのスレッドが使用中の場合、追加要求は、スレッドに割り当てることができるまでキューに入れられます。クライアント要求がスレッドの割り当てを待機する場合、クライアントの応答時間は大きくなります。しかし、最大数の設定が高すぎる場合、システムは過負荷になり、その結果クライアントの応答時間はさらに悪くなります。また、他のアプリケーションが著しくスローダウンする可能性もあります。

コンテナのサイズ変更がパフォーマンス向上につながるかどうかを判別するには、Tivoli Performance Viewer を使用して、スレッド上の負荷 (PercentMaxed カウンター) および Web コンテナ・モジュールのアクティブ・スレッドの数 (ActiveThreads カウンター) をモニターします。

PercentMaxed カウンターの値が常に 2 桁の数字になっている場合、Web コンテナがボトルネックである可能性があります。この場合、スレッドの数を増やしてください。アクティブ・スレッドの数がプール内のスレッドの数より小さい場合、スレッド・プール・サイズを小さくするとパフォーマンス向上につながる可能性があります。

- b. スレッド非アクティビティー・タイムアウトを調整します。

スレッド非アクティビティ・タイムアウトは、スレッドが再利用される前に経過しなければならない非アクティビティの期間をミリ秒で定義します。このタイムアウトに小さい値 (1 など) を設定すれば、多数のユーザーがスレッド・プールで空きスレッドを待たずに同時に処理を実行できるようになります。値 0 は待機時間がないことを示します。

3. 大規模なリストのために検索限界の値を小さくします。

大規模なタスク・リストまたはプロセス・リストで作業している場合、リストの検索限界を小さくして、ユーザーがアクセスしないデータを収集しないようにすることもできます。この設定を変更するには、管理コンソールで「サーバー」 → 「クラスター」 → 「*cluster\_name*」または「サーバー」 → 「アプリケーション・サーバー」 → 「*server\_name*」の領域にナビゲートし、「ビジネス・インテグレーション」の下で「**Business Process Choreographer**」を展開し、「**Explorer の構成 (Explorer Configuration)**」をクリックします。

---

## Business Process Choreographer Explorer レポート作成機能のチューニング

レポート生成に要する時間は、さまざまな要因によって変動します。レポート生成のパフォーマンスを向上させるためのさまざまな方法を以下に示します。

### データベース統計を更新する

DB2 データベースでは、実動データベースにデータを取り込んだ時点でデータベース統計を更新すると、パフォーマンスが大幅に向上します。

- DB2 データベースの統計を更新するには、以下のコマンドを入力します。

```
RUNSTATS ON TABLE schema_prefix.EVENT_ACT_T WITH DISTRIBUTION AND DETAILED INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.EVENT_PRC_T WITH DISTRIBUTION AND DETAILED INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.INST_ACT_T WITH DISTRIBUTION AND DETAILED INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.INST_PRC_T WITH DISTRIBUTION AND DETAILED INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.OPEN_EVENTS_T WITH DISTRIBUTION AND DETAILED INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.QUERY_T WITH DISTRIBUTION AND DETAILED INDEXES ALL;  
RUNSTATS ON TABLE schema_prefix.SLICES_T WITH DISTRIBUTION AND DETAILED INDEXES ALL;
```

データベースが大規模な (例えばプロセス・インスタンスが 500 000 件を超える) 場合は、RUNSTATS ユーティリティを実行するときに WITH DISTRIBUTION AND SAMPLED DETAILED INDEXES ALL ステートメントを使用してください。

ここで *schema\_prefix* は、Business Process Choreographer Explorer レポート作成機能用データベースの作成時に使用されたデータベース・スキーマの名前です。データベース統計の更新についての詳細は、ご使用のデータベースの資料を参照してください。

### 発行イベントの数を削減する

WebSphere Integration Developer では、アクティビティまたはプロセスのログ記録を非常に詳細なレベルで定義できます。アクティビティ監査イベントがレポート作成の対象になるのは、アクティビティを含むプロセスに対してもイベントが生成される場合に限られます。プロセスに関連付けることができないアクティビティ・イベントは、イベント・コレクター・アプリケーションでは無視され、データベースに格納されません。発行イベント数を減らすには、以下の手順を実行します。

1. 監査するプロセス・テンプレートを選択し、関心のないプロセスに関するイベントの発行を無効にします。
2. 監査するプロセス・テンプレートのアクティビティを選択します。レポート結果に影響を与えずに一部のイベントを省略できるかどうかを確認します。

アクティビティまたはプロセスの正確な情報を取得するには、すべてのイベント・タイプを監査するか、またはイベント・タイプを一切監査しないかのどちらかが必要です。

## SQL ユーザー定義関数実装を使用する

レポートを作成するには、Business Process Choreographer Explorer レポート・データベースに特定のユーザー定義関数 (UDF) をいくつかインストールする必要があります。UDF は SQL ベースの実装と Java ベースの実装として提供されます。SQL 実装は、Java 実装よりも処理速度が高速ですが、欠点もあります。Java 実装を使用している場合は、SQL 実装への切り替えを検討してください。

SQL 実装と Java 実装の長所と短所について詳しくは、ユーザー定義関数の選択に関するトピックを参照してください。

## タイムアウト値を増加させる

レポート生成には長時間かかることがあります。生成にかかる時間が長すぎると、JDBC ドライバーのトランザクション・タイムアウトまたは接続タイムアウトが発生することがあります。この状況が発生した場合は、次のようにタイムアウト値を増加させます。

1. 管理コンソールで「サーバー」 → 「アプリケーション・サーバー」 → 「*server\_name*」 → 「トランザクション・サービス」にナビゲートします。
2. 「合計トランザクション存続時間タイムアウト」の値が「最大トランザクション・タイムアウト」の値よりも小さい場合は、同じ値にします。
3. それでもまだパフォーマンスの問題が発生する場合は、「合計トランザクション存続時間タイムアウト」の値を 0 に設定し、「最大トランザクション・タイムアウト」の値を増やします。
4. それでもまだパフォーマンスの問題が発生する場合は、「合計トランザクション存続時間タイムアウト」と「最大トランザクション・タイムアウト」の両方の値を 0 に設定し、JDBC ドライバーの接続タイムアウトの値を増加させます。これを行うには、「JDBC」 → 「JDBC プロバイダー」 > *JDBC provider* → 「データ・ソース」 → *data\_source\_name* → 「接続プール・プロパティ」の下にあるデータ・ソースの接続プール・プロパティにナビゲートし、「接続タイムアウト」の値を増加させます。

サーバー・クラスターでは、すべてのクラスター・メンバーのトランザクション・タイムアウト値を調整する必要があります。

## 不要なデータを削除する

レポートのパフォーマンスは、レポート・データベース内のインスタンスとイベント・データの容量に応じて変化します。レポート生成のために大量のデータを照会すると、パフォーマンスは低下します。レポート・データベース内のプロセス・インスタンスやアクティビティ・インスタンスの数を減らすと、レポートのパフォ

パフォーマンスを向上させることができます。不要な情報や古い情報を定期的に削除すると、パフォーマンスの向上に役立ちます。

---

## 第 8 章 拡張調整

拡張調整を構成するには、以下のトピックを参照してください。

---

### トレースおよびモニターに関する考慮事項

さまざまなシステム・コンポーネントを対象に、トレースおよびモニターを異なるレベルで構成する機能は、システム分析またはデバッグを行う期間中、非常に有用であることが証明されています。WebSphere BPM 製品群は、以下の 2 つの点で豊富なモニター機能を備えています。1 つは Common Event Interface (CEI) および監査ログを介したビジネス・モニターであり、もう 1 つは PMI および ARM を介したシステム・パフォーマンス・モニターです。これらの機能を使用すると、実行中のソリューションのパフォーマンスに対する分析結果が得られますが、これらの機能によって全体的なシステム・パフォーマンスやスループットが低下することがあります。したがって、トレースおよびモニターの使用には適切な判断が必要であり、最適なパフォーマンスを確保するため、可能な場合はすべてオフにしてください。

ほとんどのトレース機能およびモニター機能は、WebSphere Application Server 管理コンソールを介して制御されます。PMI モニター、ロギング、およびトレースに対して適切なレベルのトレース/モニターが設定されていることを管理コンソールを介して確認してください。

管理コンソールを使用して、ビジネス・プロセス・コンテナにある監査ロギングおよび Common Event Infrastructure ロギングのチェック・ボックスのマークが外されていることを確認します。ただし、ビジネス上の理由でこれらの機能が必要な場合は除きます。効率を求められる作業の場合は、記録されるデータが有用な場合を除き、モニター機能を全面的にオフにすることをお勧めします。

イベント・モニターを制御する場合には、WebSphere Integration Developer (WID) も使用されます。コンポーネントおよびビジネス・プロセスのイベント・モニター・タブを確認して、イベント・モニターを慎重に適用するようにしてください。

### 作業域サービスの最大送受信サイズの構成

同期 SCA 呼び出しを JVM を介して行う場合は、下位層のインフラストラクチャーが要求の作業域を直列化し、その後応答から作業域を非直列化します。デフォルトの作業域サービスや作業域区画サービスでは、作業域で送信または受信できるサイズの上限として、ユーザー指定のサイズ限度が使用されます。可能な値は、0 (限度なし)、-1 (デフォルト) またはゼロ以外の正数です。サイズを「0」(限度なし) に設定すると、負荷の高いアウトバウンド作業域サイズまたは着信作業域サイズの検査を回避できるため、効率の面で有利になります。

### CEI エミッターの validateEvent プロパティの無効化

CEI イベント・エミッターのデフォルトの動作は、イベントを検証して、すべてのフィールドが正しく入力されていることを確認することです。この処理は、

WebSphere Process Server for z/OS システムによって生成されるイベントについては必要ありません。イベントの検証を無効にするには、エミッター・ファクトリー・プロファイルにカスタム・プロパティを追加する必要があります。

イベント検証を無効にするには、管理コンソールを開き、「サービス統合」->「**Common Event Infrastructure**」->「イベント・エミッター・ファクトリー」->「デフォルトの **Common Event Infrastructure** エミッター (Default **Common Event Infrastructure emitter**)」の順に選択します。 **validateEvent** という名前の新規プロパティを追加して、その値を **false** に設定します。

---

## 第 9 章 大規模オブジェクトの調整

大規模オブジェクトの調整を構成するには、以下のトピックを参照してください。

---

### 大規模オブジェクトの処理中の他の処理の削減または排除

比較的大きいオブジェクト・サイズを許容するための方法の 1 つとして、JVM 内部での並行処理を制限する方法があります。考えられる最大のオブジェクトを、他の WebSphere Process Server for z/OS や WebSphere アダプター・アクティビティと並行して、安定した流れで処理できると期待してはなりません。大規模オブジェクトについて検討する場合に必要な運用上の前提は、必ずしもすべてのオブジェクトが「大規模」または「非常に大規模」ではないということと、大規模オブジェクトが到着する頻度はさほど頻繁ではなく、1 日に 1 回または 2 回である可能性が高いということです。非常に大規模の複数のオブジェクトを並行して処理する場合には、障害の発生する確率が劇的に増加します。

通常処理される、比較的小規模のオブジェクトのサイズおよび数は、システムでの Java ヒープのメモリー使用量に影響します。通常、システム上の負荷が高くなると、大規模オブジェクトを処理するときにメモリーの問題が発生する可能性が高くなります。



---

## 第 10 章 最大並行性の調整

サーバー・クラスのハードウェアでの大量のデプロイメントの大半では、同時に実行される操作が多数存在します。最大の並行性を確保するための調整を行うと、サーバーの CPU を飽和させるのに十分な量の負荷をサーバーが確実に受け入れることができるようになります。調整が不十分な構成を示す兆候の 1 つは、CPU が全面的に使用されていないにもかかわらず、負荷が増加しても CPU 使用率が増加しない場合です。並行性の最大化を目標にこれらの動作を最適化するための一般的なガイドラインは、実行フローに従って、一度に 1 つずつボトルネックを取り除くことです。

注: 並行処理の割合が高いと、サーバーのリソース要件 (メモリーおよびスレッド数) が高くなります。大規模オブジェクトの処理、大量のユーザーの処理、良好な応答時間の確保など、その他の調整目的とのバランスを取ることが必要です。

---

### 並行性確保のためのエッジ・コンポーネントの調整

第一段階は、ビジネス・オブジェクトが WebSphere Process Server のエッジ・コンポーネントで並行して処理されるようにすることです。入力 BO の出力元がアダプターである場合は、入力メッセージの並行配信に合わせてアダプターが調整されていることを確認します。

入力 BO の出力元が Web サービス・エクスポート・バインディングである場合や、JSP またはサーブレットから直接呼び出された場合には、Web コンテナのスレッド・プール・サイズが適正であることを確認してください。入力 BO の出力元がメッセージングである場合は、アクティベーション・スペック (MDB バインディング) およびリスナー・ポート (MQ または MQJMS バインディング) を調整する必要があります。

---

### MDB ActivationSpec

JMS アクティベーション・スペックは、MDB `onMessage` メソッドを駆動するメッセージ・キューにアプリケーション MDB をバインドするときに使用します。特に注目する 2 つのアクティベーション・スペック・パラメーターは、最大バッチ・サイズおよび最大並行エンドポイント数です。

各 JMS エクスポート・コンポーネントには、MDB とそれに対応する `ActivationSpec` (JNDI 名: モジュール名/エクスポート・コンポーネント名\_AS) があります。最大並行エンドポイント数では、特定の MDB が並行して処理できるメッセージの数が指定されます。ただし、アクティブな MDB インスタンスは、それぞれコンテナ・スレッド・プールから実行されることに注意してください。JMS エクスポート MDB のデフォルト値は 10 です。これは、JMS キューから MDB スレッドへ同時に配信できる BO が最大で 10 であることを意味します。

アクティベーション・スペックの最大バッチ・サイズもパフォーマンスに影響します。デフォルト値は 1 です。最大バッチ・サイズの値は、1 回のステップで、メッ

セージング層から取り出されてアプリケーション層に配信されるメッセージの数を指定しています (これは、この処理が 1 回のトランザクションで実行されるという意味ではないため、この設定値がトランザクション対象範囲に影響することはありません)。長時間実行するビジネス・プロセスに関連付けられているアクティベーション・スペックでこの値を大きくすると、MDB メッセージ処理の効率を高めることができます。

アクティベーション・スペック・パラメーターを構成するには、管理コンソールで「リソース」>「JMS」>「アクティベーション・スペック」>「(ActivationSpec 名)」を選択します。

---

## リスナー・ポートの構成

MQ または MQ/JMS バインディングを使用する場合は、WebSphere Process Server for z/OS へのインバウンド・メッセージの配信を構成するためにリスナー・ポートを使用します。この場合は、「最大セッション数」プロパティが、JMS バインディングの ActivationSpec での maxConcurrency パラメーターと同じ役割を果たします。この値を適切な値まで増やしてください。

注: リスナー・ポートは、MQ または MQ/JMS バインディングを含む SCA モジュールをサーバーにデプロイすると作成されます。

---

## スレッド・プール・サイズの構成

スレッド・プールのサイズは、アプリケーションを並行して実行するためのサーバーの処理能力に直接影響します。最大限の並行性を確保するには、スレッド・プール・サイズを最適値に設定する必要があります。maxConcurrency パラメーターや「最大セッション数」パラメーターの値を大きくしても、JMS キューまたは MQ キューからの BO の同時配信が有効化されるだけです。WebSphere Process Server for z/OS サーバーが複数の要求を同時に処理するには、対応するスレッド・プール・サイズも大きくして、これらのメッセージ駆動型 Bean (MDB) スレッドの同時実行能力を高める必要があります。

MDB 処理は、デフォルトのスレッド・プールから割り振られたスレッドにディスパッチされます。別のスレッド・プールが指定されていない限り、アプリケーション・サーバー内のすべての MDB がこのスレッド・プールを共有します。つまり、デフォルトのスレッド・プール・サイズは、個々の MDB の maxConcurrency よりも大幅に大きくする必要があるという意味です。

Web コンテナ・スレッド・プール内にあるスレッドは、着信 HTTP 要求および Web サービス要求を処理する目的で使用されます。このスレッド・プールは、サーバー上にデプロイされたすべてのアプリケーションによって共有されるため、デフォルト値よりも大きい値に調整する必要があります。

ORB スレッド・プールのスレッドは、リモート EJB 呼び出しなどの ORB 要求を実行するために使用されます。スレッド・プール・サイズは、EJB インターフェース (特定のヒューマン・タスク・マネージャー API など) を介して着信する要求を処理するのに十分な大きさにする必要があります。

---

## MDB の専用スレッド・プールの構成

デフォルトのスレッド・プールは、多くの WebSphere Application Server タスクによって共有されます。JMS MDB の実行を専用スレッド・プールに分離することが必要になる場合もあります。JMS MDB スレッドに使用されているスレッド・プールを変更するには、以下の手順に従ってください。

1. サーバー上に新規のスレッド・プール (例えば MDBThreadPool) を作成します。
2. 「リソース」>「リソース・アダプター」>「リソース・アダプター」を選択して、サービス統合バス (SIB) JMS リソース・アダプター管理パネル (サーバー・スコープ付き) を開きます。アダプターが表示されない場合は、「設定」へ移動して、「組み込みリソースの表示」チェック・ボックスを選択します。
3. SIB JMS リソース・アダプターとノードおよびセルの有効範囲について、手順 1 および 2 を繰り返します。
4. 変更内容を有効にするため、サーバーを再始動します。

非同期 SCA 呼び出しに対応する SCA モジュール MDB では、独立したリソース・アダプターである Platform Messaging Component SPI Resource Adapter が使用されます。スレッド・プールを別のプールに変更するには、前述の手順に従ってください。

**注:** 専用スレッド・プールを使用した場合でも、リソース・アダプターに関連付けられているすべての MDB は、引き続き同じスレッド・プールを共有します。ただし、デフォルトのスレッド・プールを使用する他の WAS タスクとこれらの MDB が必ずしも競合するわけではありません。

---

## 並行性確保のための中間コンポーネントの調整

入力 BO を端から端まで単一スレッドで処理する場合、通常はエッジ・コンポーネントの調整が適しています。ただし、多くの場合、実行バスの両端間には複数のスレッドの切り替わりが存在します。実行バスの非同期セグメントごとに適切な並行性を確保するようにシステムを調整することが重要です。

SCA コンポーネントの非同期呼び出しでは、関連の入力キューに到着する着信イベントを、MDB を使用して listen します。各 SCA モジュールでは、MDB とそれに対応するアクティベーション・スペック (JNDI 名: sca/モジュール名/ActivationSpec) が定義されます。

**注:** SCA モジュール MDB は、モジュール内にあるすべての非同期 SCA コンポーネント (SCA エクスポート・コンポーネントを含む) により共有されます。

ActivationSpec の maxConcurrency プロパティ値を構成する場合には、このことを考慮に入れてください。SCA モジュール MDB では、JMS エクスポートのデフォルトのスレッド・プールと同じスレッド・プールを使用します。

長時間実行されるビジネス・プロセスでの非同期は、トランザクション境界で発生します。BPE では、内部の MDB とその ActivationSpec (BPEInternalActivationSpec) が定義されます。maxConcurrency パラメーターは、SCA モジュールおよび JMS エクスポート MDB の場合と同じガイドライン (前述) に従って調整する必要があります。WebSphere Process Server サーバーに存在する BPEInternalActivationSpec は 1 つです。

---

## JMS および JMS キュー接続ファクトリーの構成

複数の同時実行スレッドでは、JMS 接続プールやデータベース接続プールなどのリソースが適切に調整されていない場合、これらのリソースがボトルネックになることがあります。最大接続プール・サイズは、このプールで作成できる物理接続の最大数を指定します。これらは、DB2 データベースなどのバックエンド・リソースへの物理接続です。スレッド・プール限度に到達すると、新規の物理接続は作成できなくなるため、現在使用中の物理接続がプールに返されるまで、要求側は待機します。接続が返されなかった場合は、ConnectionWaitTimeout 例外が発生します。

例えば、「最大接続数」の値を 5 に設定していて、5 つの物理接続が使用中の場合、プール・マネージャーは、「接続タイムアウト」に指定されている時間だけ、物理接続が解放されるのを待ちます。下位層のリソースへの接続を待機しているスレッドは、その接続が解放され、プール・マネージャーによってスレッドに割り振られるまでブロックされます。指定のインターバル内に接続が解放されない場合は、ConnectionWaitTimeout 例外が発生します。

「最大接続数」を 0 に設定すると、接続プールのサイズを無限に大きくすることができます。この設定にも、接続タイムアウト値が無視されるという副作用があります。接続ファクトリーを調整するための一般的なガイドラインとしては、接続ファクトリーの最大接続プール・サイズが、並行スレッド数にスレッドごとの同時接続数を掛けた値と一致する必要があります。

各 JMS、MQ、または MQJMS インポートでは、アプリケーションのデプロイメント時に作成された接続ファクトリー (例えば、ContactManager の SAPSAPAdpt.JMSImport\_CF など) があります。JMS 接続ファクトリーの接続プールの最大接続数プロパティは、インポート・コンポーネントで同時に実行するすべてのスレッドに対して接続を確保するのに十分な大きさにする必要があります。デフォルト値は 10 です。

---

## DataSource オプションの構成

DataSource の最大接続プロパティは、すべてのスレッドからデータベースへの同時アクセスが十分可能な大きさにする必要があります。通常、WebSphere Process Server for z/OS サーバーで構成されるデータソースは、BPEDB データ・ソース、WPSDB データ・ソース、および Message Engine DB データ・ソースなど、多数あります。各データ・ソースの最大接続プロパティは、その他のシステム・リソースの最大並行性に一致するように設定します。

---

## データ・ソース準備済みステートメント・キャッシュのサイズ

準備済みステートメント・キャッシュは、準備済み SQL ステートメントおよび呼び出し可能ステートメントの処理を最適化する役割を果たします。

キャッシュのサイズは、キャッシュ項目が廃棄されないように調整できます。キャッシュのサイズを調整するには、管理コンソールを開き、「リソース」>「JDBC」>「データ・ソース」>「(データ・ソース名)」>「WebSphere Application Server データ・ソース・プロパティ」を選択します。

BPEL macroflow (長時間動作するプロセス) では、プロセスに関連するデータを持続させるため、BPEDB データ・ソースが大量に使用されます。さまざまなデータを持続させると、さまざまなステートメントを多数使用することになるため、データ・ソースのキャッシュのデフォルトの容量をはるかに超えてしまいます。この結果、過剰な数のキャッシュ不足が発生して、キャッシュの効果が失われてしまいます。この問題は、キャッシュのサイズを増やすことで解決できます。

WebSphere Process Server の BPC コンテナでは、準備済みステートメントが大規模に使用されます。このデータ・ソースのステートメント・キャッシュ・サイズは、128 以上にする必要があります。データ・ソースのキャッシュ廃棄数は、管理コンソールの Tivoli Performance Viewer を使用するとモニターできます。



---

## 第 11 章 メッセージング調整

メッセージング調整を構成するには、以下のトピックを参照してください。

---

### メッセージング・エンジンに使用するデータ・ストアまたはファイル・ストアの選択

メッセージング・エンジンのパーシスタンス (永続性) は、データベースでバックアップします。WebSphere Process Server for z/OS バージョン 6.2 のスタンドアロン構成の場合、BPE バスおよび SCA バスのパーシスタンス・ストレージは、ファイル・システム (ファイル・ストア) でバックアップできます。ファイル・ストアはプロファイルの作成時に選択しておく必要があります。新規のスタンドアロン・エンタープライズ・サービス・バス・プロファイルまたはスタンドアロン・プロセス・サーバー・プロファイルを作成するには、プロファイル管理ツールを使用します。「プロファイル作成オプション」>「拡張プロファイル作成」>「データベース構成」を選択し、「メッセージング・エンジン (ME) に対してファイル・ストアを使用する (Use a file store for Messaging Engine (MEs))」チェック・ボックスを選択します。このプロファイルを使用すると、BPE および SCA サービス統合バスではファイル・ストアが使用されます。

---

### データ・バッファ・サイズの設定 (廃棄可能またはキャッシュ)

`DiscardableDataBufferSize` は、ベスト・エフォート型の非持続メッセージを処理するときに使用するデータ・バッファのサイズをバイト単位で表します。廃棄可能なデータ・バッファの目的は、メッセージ・データをメモリー内に保持することです。これは、このサービス品質の場合には、メッセージ・データがデータ・ストアに書き込まれることがないためです。サイズが大きすぎてこのバッファには収まらないメッセージは廃棄されます。

`CachedDataBufferSize` は、ベスト・エフォート型の非持続メッセージ以外のすべてのメッセージを処理するときに使用するデータ・バッファのサイズをバイト単位で表します。このキャッシュ・データ・バッファの目的は、メモリー内のデータをキャッシュに格納することにより、パフォーマンスを最適化することです。これらのデータは、キャッシュに格納されない場合にはデータ・ストアから読み取る必要があります。

`DiscardableDataBufferSize` および `CachedDataBufferSize` を設定するには、管理コンソールを開き、「サービス統合」>「バス」>「(バス名)」>「メッセージング・エンジン」>「(メッセージング・エンジン名)」>「追加プロパティ」>「カスタム・プロパティ」を選択します。

---

## メッセージング・エンジン・データ・ストアの高性能 DBMS への移動

パフォーマンス向上のため、メッセージング・エンジンのデータ・ストアには、デフォルトの Derby ではなく、DB2 などの実稼働品質データベースを使用する必要があります。各データ・ストアは、それ固有のデータベース内に置かれています。DB2 の場合、この配置は管理上の視点からは最適ではありません。システムには既に多くのデータベースがあり、データベースをさらに 4 つ追加すると、保守や調整の手間が大幅に増加します。ここで推奨する解決方法では、4 つのデータ・ストアすべてに対して単一の DB2 データベースを使用します。個々のデータ・ストア/表は完全に分離されており、各メッセージング・エンジンは、その一連の表の排他ロックを始動時に取得します。各メッセージング・エンジンは、固有のスキーマ名を使用してその一連の表を識別します。

---

## メッセージング・エンジンに対するデータ・ストアの設定

メッセージング・エンジンは、デフォルトではそれぞれ Derby のデータ・ストアを使用するように構成されています。各データ・ストアは、それ固有のデータベース内に置かれています。DB2 for z/OS に関しては、この配置は管理上の視点からは最適ではありません。システムには既に多くのデータベースがあり、データベースをさらに 4 つ追加すると、保守や調整の手間が大幅に増加します。

ここで推奨する解決方法では、4 つのデータ・ストアすべてに対して単一の DB2 for z/OS データベースを使用します。個々のデータ・ソース/表は、各メッセージング・エンジンに関連付けられている一連の表に対して固有のスキーマ名を割り当てることで分離されます。これにより、各メッセージング・エンジンは、その一連の表の排他ロックを始動時に取得できます。

---

## DB2 データベースの作成およびデータ・ストア・スキーマのロード

メッセージング・エンジンごとに DB2 データベースを 1 つ用意する代わりに、以下に示すように、すべてのメッセージング・エンジンを同じデータベースに配置し、異なるスキーマを使用して各エンジンを分離します。

注: テーブル・スペース名は固有でなければなりません。

表 2. メッセージング・エンジン・スキーマ名

スキーマ	メッセージング・エンジン
SCASYS	LPAR01-server1.SCA.SYSTEM.box01Node01Cell.Bus
SCAAPP	LPAR01-server1.SCA.APPLICATION. box01Node01Cell.Bus
CEIMSG	LPAR01-server1.CommonEventInfrastructure_Bus
BPCMSG	LPAR01-server1.BPC.box01Node01Cell.Bus

Windows® 上で以下のコマンドを使用して、メッセージング・エンジンごとに 1 つのスキーマ定義を作成します。 <WAS Install> は WebSphere Process Server のインストール・ディレクトリを表し、<user> はユーザー名を表します。

```
<WAS Install>%bin%sibDDLGenerator.sh -system db2 -version 8.1 -platform zOS  
-statementend ; -schema BPCMSG -user <user>createSIBSchema_BPCMSG.ddl
```

各スキーマ/メッセージング・エンジンごとにこのコマンドを繰り返します。

データベースを複数のディスクにまたがって分散させるには、作成したスキーマ定義を編集し、表スペースのそれぞれの表に名前を付けて、使用したスキーマの後に置きます。例えば、SCAAPP は SCANODE\_TS になり、CEIMSG は CEIMSG\_TS になります。編集後、スキーマ定義は以下のような必要があります。

```
CREATE SCHEMA CEIMSG;
CREATE TABLE CEIMSG.SIBOWNER (
  ME_UUID VARCHAR(16),
  INC_UUID VARCHAR(16),
  VERSION INTEGER,
  MIGRATION_VERSION INTEGER
) IN CEIMSG_TB;

CREATE TABLE CEIMSG.SIBCLASSMAP (
  CLASSID INTEGER NOT NULL,
  URI VARCHAR(2048) NOT NULL,
  PRIMARY KEY(CLASSID)
) IN CEIMSG_TB;
```

ここにあるさまざまな表に別個の表スペースを割り当てることが可能です。最適な配分は、アプリケーション構造と負荷特性により異なります。この例では、データ・ストアごとに 1 つの表スペースが使用されました。

すべてのスキーマ定義、および表の定義済み表スペースを作成したら、「SIB」というデータベースを作成します。

システム管理表スペースに対して以下のコマンドを実行することにより、表スペースを作成し、使用可能な複数のディスクにまたがってコンテナを分散させます。

z/OS の場合は、データベース管理者が調整する目的で、製品に DDL が付属しています。

可能な場合は、データベース・ログを別個のディスクに配置します。

4 つのスキーマ定義をデータベースに読み込むことにより、データベースのスキーマを作成します。

---

## メッセージング・エンジンに対するデータ・ソースの作成

データ・ソースをメッセージング・エンジンごとに作成し、管理コンソールを使用して新規のデータ・ストアを使用するように各メッセージング・エンジンを構成します。

以下の表にデフォルトの状態を示します。

表 3. メッセージング・エンジン・データ・ソース名

メッセージング・エンジン	JDBC プロバイダー
box01-server1.SCA.SYSTEM.box01Node01Cell.Bus	Derby JDBC プロバイダー (XA)
box01-server1.SCA.APPLICATION.box01Node01Cell.Bus	Derby JDBC プロバイダー
box01-server1.CommonEventInfrastructure_Bus	Derby JDBC プロバイダー
box01-server1.BPC.box01Node01Cell.Bus	Derby JDBC プロバイダー

JDBC ドライバーが欠落している場合は、新規の JDBC ドライバー「DB2 Universal JDBC ドライバー・プロバイダー」を XA 以外のデータ・ソースに対して最初に作成します。BPC が DB2 に対して正しく構成されていた場合は、XA DB2 JDBC ドライバー・プロバイダーが必ず存在します。

新規の JDBC データ・ソースを 4 つ作成します。1 つは XA データ・ソースとしての CEI が対象であり、残りの 3 つは、シングル・フェーズ・コミット (XA 以外) データ・ソースとして作成します。

以下の表に新規名を示します。

表 4. 新規データ・ソース名

データ・ソースの名前	JNDI 名	JDBC プロバイダーのタイプ
CEIMSG_sib	jdbc/sib/CEIMSG	DB2 Universal (XA)
SCAAPP_sib	jdbc/sib/SCAAPPLICATION	DB2 Universal
SCASYSYSTEM_sib	jdbc/sib/SCASYSYSTEM	DB2 Universal
BPCMSG_sib	jdbc/sib/BPCMSG	DB2 Universal

データ・ソースを作成する場合は、以下の操作を行います。

- 「コンテナ管理パーシスタンス (CMP) 内でこのデータ・ソースを使用する」というチェック・ボックスのチェック・マークを外します。
- コンポーネント管理認証別名を設定します。
- データベース名を、メッセージングのために以前に作成したデータベースに使用した名前に設定します (例: SIB)。
- ドライバーのタイプとして「2」または「4」を選択します。DB2 の推奨事項に従って、ローカル・データベースにアクセスする場合は JDBC ユニバーサル・ドライバー・タイプ 2 接続を使用し、リモート・データベースに接続する場合はタイプ 4 接続を使用します。タイプ 4 のドライバーでは、データベースに対してホスト名および有効なポートを構成することが必要になります。

## メッセージング・エンジンのデータ・ストアの変更

管理コンソールを使用して、メッセージング・エンジンのデータ・ストアを変更します。

ナビゲーション・パネルで、「サービス統合」->「バス」を選択し、表示されているバス/メッセージング・エンジンごとにデータ・ストアを変更します。

データ・ストアごとに新規の JNDI 名およびスキーマ名を書き込みます。テーブルは既に作成されているため、「テーブルの作成」チェック・ボックスのチェック・マークを外します。

サーバーは即座にメッセージング・エンジンを再始動します。SystemOut.log には変更の結果が示され、さらにメッセージング・エンジンが正常に始動したかどうかも示されます。

サーバーを再始動して、すべてのシステムが更新済みの構成を使用して起動することを確認します。



---

## 第 12 章 WebSphere MQ の調整

以下のチェックリストでは、WebSphere MQ の調整に関する主な考慮事項について詳しく説明します。

- トレースの適切な使用
- メッセージ管理の使用
- CCSID の設定
- 適切なロギングの構成
- WebSphere Application Server での MQ 接続プールの使用可能化

---

### トレースの適切な使用

TRACE(G) OFF を設定すると、プロセッサの使用率を大幅に節減できます。また、キュー・マネージャーを開始してからトレースをオフにするのではなく、トレースをオフに設定してキュー・マネージャーを開始することにより、さらに効果を上げることができます。

---

### メッセージ管理の使用

持続時間の短いメッセージと長いメッセージを別のページ・セットに分けると、便利な場合があります。複数の小サイズのメッセージを 1 つの作業単位にまとめて処理できますし、可能な場合は多数の小サイズ・メッセージを 1 つの大サイズ・メッセージにまとめて送信する方法もあります。

非持続メッセージを使用している場合、すべてのメッセージが非持続である場合は NPMSPEED(FAST) チャンネルによるロギングは実行されません。

---

### CCSID の設定

キュー・マネージャー定義パラメーター・データ・セット (xxxxZPRM) に QMCCSID パラメーターが設定されていない場合は、パフォーマンスが低下する可能性があります。

---

### 適切なロギングの構成

できる限り多くのリカバリー・ログ・データを DASD のアクティブ・ログに保持します。理想的には、使用頻度の低いボリュームにアクティブ・ログ・データ・セットを割り振る必要があります。ログのオフロードが 30 分に 1 回以上の割合で実行できるように、ログのサイズは十分大きくする必要があります。

---

### WebSphere Application Server での MQ 接続プールの使用可能化

WebSphere Application Server の WebSphere MQ 接続ファクトリー定義では、「MQ 接続プールを使用可能にする」チェック・ボックスを選択する必要があります。

このためには、管理コンソールを開き、「リソース」>「JMS プロバイダー」>「WebSphere MQ」>「WebSphere MQ 接続ファクトリー」>「(接続ファクトリー名)」を選択します。

---

## 第 13 章 データベース: 一般的な調整

一般的な調整を構成するには、以下のトピックを参照してください。

---

### データベース・ログ・ファイルの高速ディスク・サブシステムへの配置

データベースは、可用性、トランザクション処理、および回復可能性の向上を考慮して設計されます。パフォーマンス上の理由により、表データをディスクにすぐに書き込むことはできません。これらの変更内容は、データベース・ログに書き込まれることで回復可能になります。データベース・ログ・ファイルが更新されるのは、ログ・バッファがいっぱいになった場合と、トランザクションのコミット時です。このため、データベース・ログ・ファイルの使用頻度は非常に高くなります。さらに重要なこととして、ログの書き込みが行われるとコミット操作が保留状態になることが多くなります。つまりアプリケーションが、書き込みに同期するために書き込みの完了を待機します。このため、データベース・ログ・ファイルへの書き込みアクセスのパフォーマンスは、システム全体のパフォーマンスにとって非常に重要です。データベース・ログ・ファイルは、ライトバック・キャッシュを備えた高速ディスク・サブシステムに格納することを推奨します。

---

### 表スペース・コンテナとは異なる装置へのログの配置

すべてのデータベース・ストレージ構成の基本的な方針は、データベース・ログを表スペース・コンテナとは別の物理ディスク装置に配置することです。こうすると、表スペース・コンテナに対する入出力とデータベース・ログに対する入出力との間のディスク・アクセスの競合が低減され、ログ・ストリームに対する順次アクセス・パターンがほぼ維持されます。このような分離を行うことにより、ログのアーカイブ処理を使用した場合の回復可能性も向上します。



---

## 第 14 章 データベース: DB2 固有の調整

総合的な DB2 調整ガイドについては、インフォメーション・センターのこのセクションでは取り扱いません。ただし、DB2 環境のパフォーマンスを向上する上で役立つ一般的な原則がいくつかあります。以下のセクションでは、これらの原則について説明し、より詳細な情報への指針を示します。

DB2 パフォーマンス調整のクイック・リファレンスについては、[developerWorks®](#) の記事『DB2 Tuning Tips for OLTP Applications』を参照してください。

最新の DB2 調整ガイドライン一式については、DB2 インフォメーション・センターを参照してください。

DB2 インフォメーション・センターにはパフォーマンス関連情報も豊富に記載されており、容易に検索できます。

---

### データベース統計の更新

DB2 でキー・テーブルへのアクセスを最適化できるように、統計データを更新することが重要です。統計データは、照会を評価するためのアクセス・プランを決定するために DB2 照会最適化プログラムによって使用されます。統計データは、表および索引で保守されます。統計データの例としては、表の行数や表の特定の列にある特異な値の数などがあります。DB2 8.2 には、DB2 自動表保守機能と呼ばれる新機能が組み込まれています。この機能では、正しい統計データを確実に収集および維持するために、必要に応じて RUNSTATS コマンドがバックグラウンドで実行されます。デフォルトでは、この機能は使用可能になっていません。この機能は DB2 コントロール・センターでオンにすることができます。統計データを更新すると、良好に機能するアクセス・プランを DB2 照会最適化プログラムで作成して、照会を評価することができます。

---

### バッファーク・プール・サイズの正しい設定

バッファーク・プールは、データベース・ページの処理中にその読み取り先、変更先、および保持先となるメモリーの領域です。バッファーク・プールは、データベースのパフォーマンスを向上する役割を果たします。必要なデータのページが既にバッファーク・プール内にある場合、そのページへのアクセスは、ディスクから直接読み取る必要があるページへのアクセスよりも高速になります。従って、DB2 バッファーク・プールのサイズは、パフォーマンスにとって非常に重要です。

バッファーク・プールが使用するメモリーの容量は、バッファーク・プール・ページのサイズと割り当てるページ数の 2 つの要因により異なります。バッファーク・プールのページ・サイズは作成時に固定され、4、8、16、または 32 KB に設定できます。最もよく使用されるバッファーク・プールは、ページ・サイズが 4 KB の IBMDEFAULTBP です。DB2 では、バージョン 8 以降、バッファーク・プールが使用するページ数は明示的に設定することを推奨しています。このためには、CREATE BUFFERPOOL コマンドまたは ALTER BUFFERPOOL コマンドを使用し

ます。デフォルトのページ数は 250 であるため、デフォルトのバッファー・プール・サイズの合計は非常に小さく、4 KB \* 250 = 1000 KB になります。

**注:** すべてのバッファー・プールは、データベース・システム上に割り振られているデータベース・グローバル・メモリー内にあります。バッファー・プールは、使用可能なメモリーを使い果たすことなく、その他のデータ構造体やアプリケーションと共存する必要があります。通常、バッファー・プールのサイズを大きくすると、入出力動作を削減することで到達できるポイントまでパフォーマンスを向上できます。そのポイントを超えると、メモリーを追加で割り振ってもパフォーマンスは向上しなくなります。バッファー・プール・サイズの適切な設定値を選択するには、システム・ツールを使用するか DB2 バッファー・プールのスナップショットを使用して、データベース・コンテナの入出力動作をモニターします。システムでのページング動作につながるため、バッファー・プール・サイズの設定値は大きい値に構成しないようにしてください。

---

## 適正な表索引付けの維持

WebSphere BPM 製品では、多くのインストール済み環境に適合する一連のデータベース索引が作成されますが、一部の環境では索引を追加することが必要になります。索引の追加が必要なデータベース環境では、時間が経過するにつれてパフォーマンスが低下することがよくあります。場合によっては、パフォーマンスが大幅に低下することもあります。索引の追加が必要な環境では、表スペース・コンテナを保有している装置で、多数の読み取り入出力操作が発生することがよくあります。パフォーマンスの向上のために追加する索引の決定を支援するため、DB2 には設計アドバイザーが用意されています。設計アドバイザーは DB2 コントロール・センターで使用できますが、コマンド行プロセッサから開始することもできます。設計アドバイザーは、特定の作業負荷に適した索引を定義および設計するのに役立つ機能を備えています。

---

## ログ・ファイルの適切なサイズ変更

循環ロギングを使用する場合は、使用可能なログ・スペースにより、バッファー・プール内のダーティー・ページを適度な低速度で一掃できることが重要です。データベースに対する変更内容はログに即座に書き込まれますが、調整の行き届いているデータベースでは、変更されたページが最終的にディスクに書き戻される前に、複数の変更内容が 1 つのページにまとめられます。ログ内にのみ記録されている変更内容は、循環ロギングでは上書きできません。DB2 はこの状態を検出して、新しいログ・ファイルへ切り替えるために必要なダーティー・ページの即時消去を強制的に実行します。この仕組みでは、ログに記録された変更内容は保護されますが、すべてのアプリケーション・ロギングは、必要なページが消去されるまで中断する必要があります。

DB2 では、データベース・レベルの **softmax** パラメーターで制御している状態で、ページの消去処理を事前に起動することにより、ログ・ファイルを切り替えるときの一時停止を回避します。softmax のデフォルト値 100 の設定では、ログの現在の先頭項目と、ダーティー・ページへの変更を記録しているログ項目のうち経過時間の最も長い項目との差が 1 つのログ・ファイルのサイズの 100% を超えると、バ

ックグラウンドでの消去動作が開始されます。 極端な場合には、この非同期ページ消去ではログ動作に追従できないため、ログの切り替えによる一時停止が発生して、パフォーマンスの低下を招きます。

使用可能なログ・スペースを増やすと、非同期のページ消去によるダーティ・バッファ・プール・ページの書き込み時間が増えて、ログの切り替えによる一時停止の発生が防止されます。 消去処理と次の消去処理との間隔を長くすると、複数の変更内容を、書き込む前に 1 つのページにまとめることができるため、ページ消去をより効率的にすることにより、必要な書き込みスループットの値を減らすことができます。

使用可能なログ・スペースは、ログ・ファイル・サイズと 1 次ログ・ファイルの数の積で制御されます。これらはデータベース・レベルで構成されます。 **logfilsiz** は、各ログ・ファイルの 4K ページの数を表します。 **logprimary** は、1 次ログ・ファイルの数を制御します。通常動作では 1 分以上折り返す必要がないほどサイズの大きい 1 次ログ・ファイルを 10 ファイル使用します。

1 次ログ・ファイルのサイズを大きくすると、データベースのリカバリーに対して明らかな影響が出ます。 **softmax** の値が一定の値であるとする、ログ・ファイルのサイズが大きいことは、リカバリーの所要時間が長くなることを意味します。これに対応するために **softmax** パラメーターの値を小さくすることはできますが、ページ消去をより積極的に行うと、効率の低下につながることを念頭においてください。 **softmax** の値を大きくすると、書き込みの一括化という自由度が増える反面、リカバリー時間は増大します。



---

## 第 15 章 拡張 Java ヒープ調整

WebSphere BPM 製品群は Java で記述されているため、Java 仮想マシン (JVM) のパフォーマンスは、これらの製品によるパフォーマンスに大きく影響します。JVM では、ツール処理および実行時のパフォーマンスの両方を向上するために使用できる複数の調整パラメーターが外部化されています。最も重要なパラメーターは、ガーベッジ・コレクションと、Java ヒープ・サイズの設定に関連しています。このセクションでは、これらのトピックについて詳しく説明します。

注: このレポートで取り上げる製品では、ほとんどのプラットフォーム上での IBM® JVM を使用しますが、ここでは Java 5 の調整についてのみ説明します。

以下は、「IBM Java 5 Diagnostics Guide」へのリンクです。この参照先のガイドでは、このレポートで説明する調整パラメーターよりも多くのパラメーターが説明されていますが、その大半は特定の状態を対象にしており、汎用ではありません。IBM Java 5 のガーベッジ・コレクション・アルゴリズムについては、『Understanding the IBM JDK for Java』の章の『Garbage Collectors』というセクションを参照してください。

---

### ガーベッジ・コレクションのモニター

ヒープを正しく設定するには、ヒープの使用方法を決定する必要があります。これは、verbosegc トレースを収集することにより簡単に実行できます。verbosegc トレースでは、ガーベッジ・コレクションのアクションおよび統計データが標準エラー出力に出力されます。verbosegc トレースは、Java ランタイム・オプションの verbosegc を使用することによってアクティブになります。出力例を以下に示します。

#### verbosegc トレースの出力例

```
<af type="tenured" id="12" timestamp="Fri Jan 18 15:46:15 2008" intervalms="86.539">
  <minimum requested_bytes="3498704" />
  <time exclusiveaccessms="0.103" />
  <tenured freebytes="80200400" totalbytes="268435456" percent="29" >
    <soa freebytes="76787560" totalbytes="255013888" percent="30" />
    <loa freebytes="3412840" totalbytes="13421568" percent="25" />
  </tenured>
  <gc type="global" id="12" totalid="12" intervalms="87.124">
    <refs_cleared soft="2" threshold="32" weak="0" phantom="0" />
    <finalization objectsqueued="0" />
    <timesms mark="242.029" sweep="14.348" compact="0.000" total="256.598" />
    <tenured freebytes="95436688" totalbytes="268435456" percent="35" >
      <soa freebytes="87135192" totalbytes="252329472" percent="34" />
      <loa freebytes="8301496" totalbytes="16105984" percent="51" />
    </tenured>
  </gc>
  <tenured freebytes="91937984" totalbytes="268435456" percent="34" >
    <soa freebytes="87135192" totalbytes="252329472" percent="34" />
    <loa freebytes="4802792" totalbytes="16105984" percent="29" />
  </tenured>
  <time totalms="263.195" />
</af>
```

テキスト・エディターを使用して verbosegc 出力を解析するには、単調で手間のかかる作業が必要です。より効果的な Java ヒープ分析に使用できる優れた視覚化ツールが Web 上にあります。IBM Pattern Modeling and Analysis Tool (PMAT) for Java Garbage Collector は、そうしたツールの 1 つです。このツールは、IBM alphaWorks から無償でダウンロードできます。

---

## 大半の構成に対するヒープ・サイズの設定

このセクションでは、大半の構成に対して適切な Java ヒープ・サイズを設定するための基準について説明します。大半の実稼働アプリケーションでは、IBM JVM の Java ヒープ・サイズのデフォルト値は小さすぎるため、大きくする必要があります。さらに、`-Xgcpolicy:gencon` オプションを使用した場合は、新世代領域のヒープ・サイズを大きくする必要があります。

最適なヒープ・サイズを設定する方法は、いくつかあります。有用な方法の概要を以下に示します。初期ヒープ・サイズを (`-Xms` オプションで) 適度な値 (例: 256 MB) に設定し、最大ヒープ・サイズを (`-Xmx` オプションで) 適度ではあるが大きい値 (例: 1024 MB) に設定します。ヒープが強制的にページングされるような最大ヒープ・サイズには絶対にしないでください。ヒープが常に物理メモリ内にとどまることが必須条件です。この結果、JVM は、ヒープ・サイズを増減することによって GC 時間を適度な限度内に維持しようとします。また、verbosegc からの出力を使用して GC の動作をモニターする必要があります。

世代別並行 GC (`-Xgcpolicy:gencon`) を使用する場合は、新世代領域のサイズも特定の値に設定できます。デフォルトでは、新しいサイズはヒープ・サイズ合計の 4 分の 1 か 64 MB のいずれか小さいほうになります。パフォーマンスを向上するには、新世代領域のヒープ・サイズをヒープ・サイズ全体の 4 分の 1 以上にして、その上限を 64MB には設定しないようにする必要があります。新世代領域のサイズは、JVM オプション (`-Xmn <size>`、`-Xmns<initialSize>`、および `-Xmnx<maxSize>`) で設定します。

初期ヒープ・サイズを設定したら、verbosegc トレースを使用して GC の動作をモニターする必要があります。出力を分析したら、その結果に応じてヒープの設定値を変更します。例えば、GC の所要時間の割合が高く、ヒープが最大サイズまで大きくなった場合は、最大ヒープ・サイズを大きくすることでスループットを高められます。経験法則として、合計時間の 10% を超える時間を GC に費やした場合は、GC 所要時間の割合が高いとみなされます。

**注:** このタイプの問題はメモリ過剰使用の問題である可能性があるため、Java ヒープの最大サイズを大きくしても、必ずしも問題を解決できないことがあります。したがって、GC 一時停止時間のために応答時間が過剰に長くなっている場合は、ヒープ・サイズを縮小してください。両方の問題が確認された場合は、アプリケーションによるヒープ使用量の分析が必要です。

---

## 1 つのシステム上で複数の JVM を実行する場合のヒープ・サイズの設定

実行中の Java プログラムには、それぞれに関連付けられているヒープがあります。このため、複数の Java プログラムが単一の物理システム上で実行されている構成では、ヒープ・サイズを適切に設定することが特に重要です。仮想メモリの全使用量の合計 (Java ヒープだけでなく、その他すべての仮想メモリの割り振り) が物理メモリのサイズを超えた場合、Java ヒープはページング処理の対象になります。前述したように、ページングが実行されるとシステム全体のパフォーマンスは大幅に低下します。この状況が発生する可能性を最小限にとどめるため、以下の基準に従ってください。

- 実行中の JVM ごとに `verbosegc` トレースを収集する
- `verbosegc` トレース出力に基づいて、初期ヒープ・サイズを比較的小さい値に設定する。例えば、`verbosegc` トレース出力の示す結果によると、ヒープ・サイズは 256 MB までは急激に増加し、その後 400 MB までは比較的緩やかに増加するとします。この結果に基づいて、初期ヒープ・サイズを 256 MB (`ms256m`) に設定します。
- `verbosegc` トレース出力に基づいて、最大ヒープ・サイズを適切な値に設定する。最大ヒープ・サイズの設定値は小さすぎることのないようにしてください。小さすぎると、メモリ不足エラーが発生します。最大ヒープ・サイズはスループットのピーク値をクリアできるのに十分な大きさにする必要があります。前述の例を使用した場合、最大ヒープ・サイズは 768 MB に設定することが適切と思われます(`mx768m`)。
- ヒープ・サイズの設定値は小さすぎることのないようにしてください。この値が小さすぎるとガーベッジ・コレクションが頻繁に実行され、スループットが低下する可能性があります。 `verbosegc` トレースは、この設定値を決定するのに役立ちます。ガーベッジ・コレクションが頻繁に実行されることのないように、ヒープ・サイズは十分大きくする必要がありますが、その一方でヒープ・サイズが次第に増加してヒープがページングされる原因にならないように、バランスを取る必要があります。

---

## メモリ不足エラーが発生した場合のヒープ・サイズの削減または増加

`java.lang.OutOfMemory` 例外は、JVM によってさまざまな環境で使用されるため、この例外の発生原因を追跡するのが困難な場合があります。これらのエラー発生原因候補の差異を見分けるための決定的な仕組みはありませんが、まず、`verbosegc` を使用してトレースを収集することをお勧めします。問題がヒープ内のメモリ不足である場合、これは `verbosegc` 出力で容易に読み取ることができます。

ただし、`java.lang.OutOfMemory` 例外が発生したとき空きメモリが十分ある場合、次に検査する項目は `verbosegc` からのファイナライザー・カウントです。ファイナライザー・カウントが大きい場合は、ヒープ外部のリソースがヒープ内部のリソースに保持され、ファイナライザーによって削除されることが微妙な影響を及ぼす可能性があります。ファイナライザーの実行頻度を増やすことによってヒープ・サイズを縮小すると、この状態を軽減できます。さらに、アプリケーションを調べて、ファイナライザーの実行を回避するか、または最小限に抑えることができるかどうかを確認してください。

メモリー不足のエラーは、特定のシステム・リソースの枯渇など、JVM ヒープの使用量に関連しない問題でも発生することに注意してください。この例としては、ファイル・ハンドル数の不足や、スレッドのスタック・サイズが小さすぎるなどがあります。

構成を調整して、ネイティブ・ヒープを枯渇しないようにすることができる場合があります。スレッドのスタック・サイズを小さくしてみてください (-Xss パラメーター)。ただし、メソッドのネストが深いと、スタック・サイズが不足している場合に、スレッドのスタック・オーバーフローが発生する恐れがあります。

ミドルウェア製品では、処理動作中の JDBC ドライバーを使用していると、通常は固有のメモリー所要量に多大な影響がある処理動作対象外のドライバーを検出できません。例えば、タイプ 4 JDBC ドライバーを使用できることや、MQSeries をバイインディング・モードからクライアント・モードに切り替えることができるなどです。詳しくは、対象となっている製品の資料を参照してください。WAS 6.1 z/OS の Java ヒープの調整について詳しくは、「WAS for z/OS Version 6.1 Tuning guide」を参照してください。

---

## 第 16 章 共通の参照資料

以下のリストでは、WebSphere Process Server を使用した調整作業を実行するとき  
に役立つ参照資料をいくつか示します。

1. メッセージングの最良実施例
2. DB2 ベスト・プラクティス (英語)
3. DB2 チューニング (英語)
4. DB2 管理ガイド: パフォーマンス
  - <http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp>
  - [http://www-306.ibm.com/software/data/db2/support/db2\\_9](http://www-306.ibm.com/software/data/db2/support/db2_9)
5. Red Paper for Websphere Business Process Management 6.1 Performance Tuning
6. Business Process Choreographer (BPC) Tuning Paper for WebSphere Process Server 6.1
7. WebSphere Application Server パフォーマンスの URL (英語)
8. WebSphere Application Server V6.1 インフォメーション・センターの『パフォーマンスのチューニング』セクション (メッセージングのチューニングを含む)
9. WebSphere Process Server V6 – Business Process Choreographer: Performance Tuning of Human Workflows Using Materialized Views
10. Modeling Efficient BPEL Processes
11. Using J2CA Adapters with WebSphere Enterprise Service Bus
12. Guided Tour of WID Part 1
13. WebSphere Enterprise Service Bus サポート
14. Web Services Architectures and Best Practices
15. Best Practices for Web Services. Part1-Basics (文書 9 および 10 はパフォーマンスについて説明しています)。
16. Best Practices for Web Services. Part9- Performance considerations
17. Best Practices for Web Services. Part10- Other considerations which influence performance
18. ダイナミック・キャッシング・サービスに関する WebSphere Journal の記事 (英語)
19. IBM Java 5.0 Diagnostic Guide
20. Redbook: Production Topologies for WPS and WebSphere ESB V6



---

## 第 17 章 z/OS 固有の参照情報

以下のセクションでは、WebSphere Process Server のパフォーマンス調整作業を実行するときに役立つ参照資料を示します。

### WebSphere パフォーマンスのベスト・プラクティスおよびリソース

1. Performance Harness for Java Messaging Service

### WebSphere Application Server の参照資料

1. WebSphere Application Server Performance Best Practices and Resources
2. WebSphere Process Integration Version 6.1.0 information center
3. メッセージングの最良実施例
4. WebSphere Application Server パフォーマンスの URL (英語)
5. WebSphere Application Server for z/OS バージョン 6.1.0 製品ライブラリーのページ (英語)
6. WebSphere Application Server for z/OS Tuning guide Version 6.1.0
7. WebSphere Application Server for z/OS V6.1 インフォメーション・センターの『パフォーマンスのチューニング』セクション (メッセージングのチューニングを含む)

### DB2 および DB2 for z/OS の参照資料

1. DB2 for z/OS および DB2 for OS/390 の製品ページ (英語)
2. DB2 Universal Database for z/OS の製品概要ページ (英語)
3. Best practices for tuning DB2 UDB V8.1 and its databases
4. A quick reference on DB2 tuning
5. DB2 管理ガイド: パフォーマンス
6. DB2 for z/OS and WebSphere for z/OS Redbook

### WebSphere MQ の参照資料

1. WebSphere MQ for z/OS ライブラリー (英語)
2. Performance Tuning for Java Messaging Service on WebSphere Application Server on z/OS
3. How the maximum sessions property on the listener port affects WebSphere Application Server performance
4. WebSphere MQ for z/OS の製品ページ (英語)

### WebSphere Process Server および WebSphere Enterprise Service Bus の参照資料

1. z/OS Technical Overview: WebSphere Process Server and WebSphere Enterprise Service Bus
2. WebSphere エンタープライズ・サービス・バスのインフォメーション・センター
3. Getting Started with WebSphere Enterprise Service Bus V6

4. WebSphere Enterprise Service Bus and WebSphere Enterprise Service Bus version 6.0.x for z/OS Configuration Tools and Samples

### **WebSphere Integration Developer の参照資料**

1. WebSphere Integration Developer インフォメーション・センター
2. Modeling Efficient BPEL Processes (英語の資料)

### **Web サービスの参照資料**

1. Web Services Architectures and Best Practices
2. Best Practices for Web Services. Part9- Performance considerations
3. Best Practices for Web Services. Part10- Other considerations which influence performance
4. ダイナミック・キャッシング・サービスに関する WebSphere Journal の記事 (英語)

### **z/OS および System z の参照資料**

1. z/OS Internet Library

---

## 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒106-8711  
東京都港区六本木 3-2-12  
日本アイ・ビー・エム株式会社  
法務・知的財産  
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation  
577 Airport Blvd., Suite 800  
Burlingame, CA 94010  
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性があります。その測定値が、一般に利用可能なシステムのもと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者をお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。(c) (お客様の会社名) (西暦年).このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。(c) Copyright IBM Corp. \_年を入れる\_. All rights reserved.

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

## プログラミング・インターフェース情報

プログラミング・インターフェース情報は、このプログラムを使用してアプリケーション・ソフトウェアを作成する際に役立ちます。

一般使用プログラミング・インターフェースにより、お客様はこのプログラム・ツール・サービスを含むアプリケーション・ソフトウェアを書くことができます。

ただし、この情報には、診断、修正、および調整情報が含まれている場合があります。診断、修正、調整情報は、お客様のアプリケーション・ソフトウェアのデバッグ支援のために提供されています。

**警告:** 診断、修正、調整情報は、変更される場合がありますので、プログラミング・インターフェースとしては使用しないでください。

## 商標

IBM、IBM ロゴ、ibm.com は、International Business Machines Corporation の米国およびその他の国における商標または登録商標です。これらおよび他の IBM 商標に、この情報の最初に現れる個所で商標表示 (® または ™) が付されている場合、これらの表示は、この情報が公開された時点で、米国において、IBM が所有する登録商標またはコモン・ロー上の商標であることを示しています。このような商標は、その他の国においても登録商標またはコモン・ロー上の商標である可能性があります。現時点での IBM の商標リストについては、[www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml) の「Copyright and trademark information」をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

Java およびすべての Java 関連の商標およびロゴは Sun Microsystems, Inc. の米国およびその他の国における商標です。

Microsoft および Windows は、Microsoft Corporation の米国およびその他の国における商標です。

Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Intel Centrino、Intel Centrino ロゴ、Celeron、Intel Xeon、Intel SpeedStep、Itanium、Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

この製品には、Eclipse Project (<http://www.eclipse.org>) により開発されたソフトウェアが含まれています。



IBM WebSphere Process Server for Multiplatforms バージョン 6.1.2

---

## 索引

日本語、数字、英字、特殊文字の順に配列されています。なお、濁音と半濁音は清音と同等に扱われています。







Printed in Japan