

# Oracle BPEL Process Manager 10.1.3.4のメッセージ・スレッド・アーキテクチャ

このドキュメントでは、リリース10.1.3.4以降のOracle BPEL Process Managerでのメッセージ・スレッド・アーキテクチャについて説明します。

## メッセージ・スレッド・アーキテクチャ

プロセスをインスタンス化するためにクライアントに受信されたメッセージや、非同期プロセスの処理中に生成された内部メッセージは、ディスパッチャ・モジュールによって管理されます。このモジュールには次の3種類のディスパッチ・セットが含まれています。

- システム
- 呼出し
- エンジン

内部メッセージ（ディスパッチ・メッセージとも呼ばれる）は、3つのディスパッチ・セットのいずれかに属します。

ディスパッチ・メッセージの実行をスケジューリングする必要がある場合、ディスパッチャはスレッド・プールにメッセージをエンキューします。それぞれのディスパッチ・セットに、1つのスレッド・プール（`java.util.concurrent.ThreadPoolExecutor`）が含まれます。スレッド・プール実装は、メッセージがエンキューされるとスレッドに通知を行うとともに、プール内で適切な数のスレッドがインスタンス化されるようにします。図1-1に、その詳細を示します。

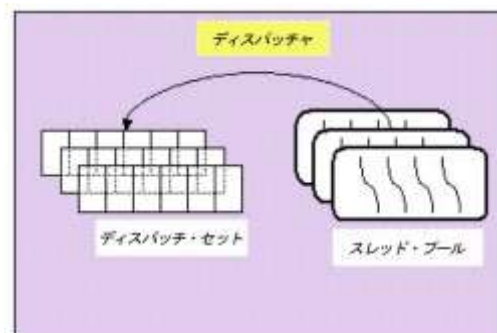


図1-1 スレッド・プール実装

ディスパッチ・セットのスレッドをチューニングするため、次の3つの構成プロパティが提供されています。表1-1に、その詳細を示します。

表1-1 プロパティ

プロパティ	説明	値
<b>dspSystemThreads</b>	<p>システム・ディスパッチャ・メッセージを処理するために割り当てられるスレッドの合計数。システム・ディスパッチャ・メッセージは汎用のクリーンアップ・タスクであり、通常はサーバーによってすぐに処理されます（ステートフルなメッセージBeanのプールへの解放など）。</p> <p>一般に、実行中に生成されるシステム・ディスパッチャ・メッセージ数の処理に必要なスレッド数はわずかです。</p>	<p>デフォルト値は、2スレッドです。1スレッド未満の値が指定されると、デフォルト値に変更されます。</p>
<b>dspInvokeThreads</b>	<p>呼出しディスパッチャ・メッセージを処理するために割り当てられるスレッドの合計数。呼出しディスパッチャ・メッセージは、Oracle BPELサーバーが受信したペイロードごとに生成され、新規インスタンスのインスタンス化に使用されます。</p> <p>エンジンによって処理されるリクエストの大半がインスタンス呼出しである（インスタンス・コールバックではなく）場合、呼出しスレッドの数を増やすことで、パフォーマンスが向上する可能性があります。スレッド数を増やすと、コンテキストのスイッチング・コストが高くなるため、CPU利用率が高くなる場合があります。</p>	<p>デフォルト値は、40スレッドです。1スレッド未満の値が指定されると、デフォルト値に変更されます。</p>
<b>dspEngineThreads</b>	<p>エンジン・ディスパッチャ・メッセージを処理するために割り当てられるスレッドの合計数。エンジン・ディスパッチャ・メッセージは、Oracle BPELサーバーによってアクティビティが非同期で処理される必要のある場合は常に生成されます。</p> <p>Oracle BPELサーバー上に配置されたプロセスの大半が多数のデハイドレーション・ポイント（midprocess receive、onMessage、onAlarm、waitアクティビティ）を持つ永続プロセスである場合、エンジン・スレッドの数を増やすことでパフォーマンスが向上する可能性があります。スレッド数を増やすと、コンテキストのスイッチング・コストが高くなるため、CPU利用率が高くなる場合があります。</p>	<p>デフォルト値は、60スレッドです。1スレッド未満の値が指定されると、デフォルト値に変更されます。</p>

これらのプロパティの説明は、Oracle BPEL Process Manager 10.1.3.4のドキュメントにも記載されています。

[http://download.oracle.com/docs/cd/E12524\\_01/relnotes.1013/e12523/bpelrn.htm#sthref16](http://download.oracle.com/docs/cd/E12524_01/relnotes.1013/e12523/bpelrn.htm#sthref16)

各スレッド・プールの最小スレッド数は1です。上記プロパティの値に0または負数を指定することはできません。

それぞれの個別ディスパッチャ・セットに対するメッセージ統計は、Oracle BPEL ControlのThreadsページで提供されています。図1-2に、その詳細を示します。

Pending Requests		Scheduled	Active
BPEL Domain Management Requests		0	0
BPEL Process Management Requests		0	0
Callback Requests		22	4
Activity Execution Requests		12	16
	<b>Process</b>	<b>Count</b>	<b>Process</b> <b>Count</b>
	Flow (v. 1.0)	22	Flow (v. 1.0) 5
	Queue size: <input type="text" value="5"/> <input type="button" value="Go"/>		
	Flow (v. 1.0)		
	Flow (v. 1.0)		
	Flow (v. 1.0)		
	Flow (v. 1.0)		
	Flow (v. 1.0)		
	Flow (v. 1.0)		
New Instance Requests		0	6
	<b>Process</b>	<b>Count</b>	<b>Process</b> <b>Count</b>
	Queue size: <input type="text" value="20"/> <input type="button" value="Go"/>		AmericanLoan (v. 1.0) 3
	UnitedLoan (v. 1.0)		3
Low Level System Requests		0	0

Request Statistics	Total Requests	Errored	Avg Pending Time (ms)	Avg Execution Time (ms)	Throughput (msg/sec - past 60s)
System Thread Statistics	244	0	7	0	1.05
Invoke Thread Statistics	144	16	24623	48316	1.05
Engine Thread Statistics	161	9	2727	15322	0.73

Thread Allocation Activity	
Active system threads	2
Active invocation threads	40
Active engine threads	20

図1-2 Oracle BPEL ControlのThreadsページ

Request Statistics表に、スレッド統計が表示されます。この表は、次のフィールドで構成されています。

**注：**統計情報の計算方法は、すべてのスレッド・タイプにおいて同じです。

■ Request Statistics :

3種類のディスパッチ・セット・スレッドを表示します。

- System Thread Statisticsのチューニングには、**dspSystemThreads**プロパティを使用します。
- Invoke Thread Statisticsのチューニングには、**dspInvokeThreads**プロパティを使用します。
- Engine Thread Statisticsのチューニングには、**dspEngineThreads**プロパティを使用します。

■ Total Requests :

スレッド・タイプごとの合計リクエスト数を表示します。この数値は、最後にサーバーが起動されて以来、またはOracle BPEL Controlで最後に統計情報が消去されて以来の合計数を表します。

■ Errored :

スレッド・タイプごとの合計エラー処理件数を表示します。

このエラー件数は、EJB内部から例外がスローされ、ディスパッチャ・レイヤーで取得された場合にのみ増加します。エンジン内部で取得されて処理（非同期監査中のデハイドレート）されたエラーは、エラー処理としてディスパッチャに対して表面化することはありません。ログ内に、"Dispatch error, failed to handle message ..."のような例外が表示されます。

次に、発生する可能性のあるエラーの例を挙げます。

- システム・メッセージ：現在処理されている唯一のシステム・ディスパッチ・セット・エラーは、ステートフルEJBをプールに解放する際のエラーです。
- 呼出しメッセージ：呼出しディスパッチ・セットのエラーには、あらゆるエラーが当てはまります。呼出しメッセージの結果として、インスタンスが作成され、保存されます。したがって、トランザクション・タイムアウトやデータベース・アクセス・エラー、メモリ不足エラー、EJBの問題など、処理中に発生する可能性のあるすべてのシステム・エラーが、呼出しメッセージ・エラーの原因となります。
- エンジン・メッセージ：エンジン・ディスパッチ・セットは、コールバックの解決、アクティビティ期限切れのコールバック、アクティビティの実行（bpelx:execによってチェックポイントが発行された場合）、ノンブロッキング呼出しのコールバック、非同期監査書込みなど、多数のメッセージ・タイプを処理します。ここで発生したエラーはすべて、エンジン・ディスパッチ・セット・エラー件数の増加につながります。

---

**注：**呼出しディスパッチ・セットとエンジン・ディスパッチ・セットのエラー条件は同じです（トランザクション・タイムアウト、メモリ不足エラー、データベース・アクセス・エラーなど）。重要なのは、どちらのスレッドの処理中にエラーが発生したかという点です。これら2つのディスパッチ・セットにおいて、エラー・タイプとエラー件数の相関関係はありません。

---

■ **Avg Pending Time :**

メッセージがキューに入ってから、スレッドに取り出されて実行されるまでの平均時間をミリ秒で表示します（実行時間は含まれません）。この数値は、保留時間の合計をメッセージ数で割ったものになります。

■ **Avg Execution Time :**

スレッド・タイプごとに、メッセージ処理にかかった平均時間をミリ秒で表示します。実行時間の測定は、メッセージがデキューされた時点から開始され、メッセージに関連付けられたアクションが完了した時点で停止されます。この数値は、実行時間の合計をメッセージ数で割ったものになります。

■ **Throughput (msg/sec Past 60s) :**

スレッドごとに、1秒あたりに処理されたメッセージ数を表示します。この数値は、実行されたメッセージ数とそれぞれの実行時間を循環配列リストに入れ、最後の1分間の時間をチェックする方法で計算されます。