HITACHI
Inspire the Next

開発・運用時のガイド

JDK17 への移行に伴う留意点

2024.9
September

はじめに

uCosminexus Application Server を使用し，システムを設計・構築・運用する方が留意すべき点について説明します。本書は，開発・運用フェーズで使用するドキュメントとして，Java™ Development Kit 11 から Java™ Development Kit 17 への移行に伴う留意点について記述しています。

１．対象とする読者

本書は，Java™ Development Kit 17 を使用し，システムを設計・構築・運用する立場にある方を対象としています。

■商標類

・HITACHI は，（株）日立製作所の商標または登録商標です。

・Oracle と Java は、Oracle Corporation 及びその子会社，関連会社の米国及びその他の国における登録商標です。文中の社名，商品名等は各社の商標または登録商標である場合があります。

・その他記載の会社名，製品名は，それぞれの会社の商標もしくは登録商標です。

■英略語の表記

本書では，英略語を次のように表記しています。

表記	製品名
Java	Java™
Java EE	Java™ Platform, Enterprise Edition
Java SE	Java™ Platform, Standard Edition
JDK	Java™ Development Kit
	JDK™

■発行元

株式会社日立製作所 クラウドサービスプラットフォームビジネスユニット　マネージド&プラットフォームサービス事業部

目　次

1 JDK17 への移行に伴う留意事項

JDK17 への移行に伴う留意事項を説明します。

本章の構成

1.1 JDK17 への移行に伴う GC についての留意事項

本節では，JDK11 から JDK17 に移行した際の GC についての留意事項を記載します。

(1) ZGC の追加

JDK17 以降では GC（ガベージコレクタ）として，ZGC が選択できます。
ZGC はチューニングが簡単で，かつ停止時間が非常に短いスケーラブルな GC です。そのため，低レイテンシが要求されるシステム，および大規模なメモリ環境のシステムに適しています。

従来からサポートされている SerialGC・G1GC と ZGC の比較を以下の表に示します。

#	GC 方式	特徴	推奨システム
1	SerialGC	GC 処理の全てをアプリケーションを停止させて行うため，アプリケーション実行中は GC 処理による CPU リソースの消費が起こりません。このため，アプリケーションのスループットが他 GC よりも高いです。一方で，GC によるアプリケーションの停止時間は Java ヒープサイズに比例して長くなります。	最悪レスポンス時間の要件がなく，スループットの方が重要なシステム。
2	G1GC	GC 処理の一部をアプリケーションの実行と並行で行います。アプリケーションの実行中に GC 処理による CPU リソースの消費が起こります。このため，アプリケーションのスループットは低くなります。一方，上記の並行処理で取得した情報を利用することで，アプリケーションの停止時間をソフトリアルタイム制御できます。	GC による最悪レスポンス時間の要件があり，スループットよりソフトリアルタイム性が重要なシステム。
3	ZGC	一部を除き，GC 処理をアプリケーションの実行と並行で行うためアプリケーションの停止時間が非常に短いです。これは数テラバイトの非常に大きなヒープを使用するようなシステムにおいても同様ですが，その反面スループットは低下します。	低レイテンシが要求されるシステム。

(2) デフォルト GC の変更

JDK17 以降ではデフォルトで選択される GC が SerialGC から G1GC に変更されました。ただし，論理プロセッサが 2 未満または物理メモリが 1792MB 未満の場合は自動で SerialGC が選択されます。

(3) 明示管理ヒープ機能の非サポート化

JDK17 以降では明示管理ヒープ機能は非サポートです。明示管理ヒープ機能は SerialGC における GC 停止時間の長期化を防ぐために使用されました。JDK17 以降ではアプリケーションの要件に合わせて G1GC または ZGC を使用することで GC 停止時間の長期化を防ぎます。

JDK11 以前に明示管理ヒープ機能を使用していて JDK17 へ移行する場合，以下に注意してください。

・明示管理ヒープ機能の JavaVM オプションを指定している場合，プロセスの起動に失敗します。そのため，明示管理ヒープ機能の JavaVM オプションはすべて削除してください。

・明示管理ヒープ機能の API を使用している場合，アプリケーションの改修は必要ありません。明示管理ヒープ機能が無効の場合と同じ挙動となります。

・明示管理ヒープ機能は，Java ヒープとは別の Explicit ヒープという独自の領域を使用しています。これまで明示管理ヒープ機能を使用していた場合，Explicit ヒープに配置されていたオブジェクトは，すべて Java ヒープに配置されることになります。そのため，これまで Explicit ヒープに設定していたヒープサイズ（-XX:HitachiExplicitHeapMaxSize に指定した値）を Java ヒープに加算してください。G1GC，ZGC を使用する場合は，さらにそれぞれの GC で必要な Java ヒープサイズを加算して使用してください。

1.2 JDK17 への移行に伴うその他の留意事項

本節では，JDK11 から JDK17 に移行した際の GC 以外の留意事項を記載します。

(1) クラスファイルのバージョンについて

Java SE 17 からクラスファイルのバージョンが 61 になりました。
これにより，クラスファイルフォーマット仕様に対し，Record/ PermittedSubclasses の Attribute が追加されています。

クラスファイル変換等でクラスファイルを出力する場合，出力するクラスファイルのバージョンとクラスファイルのフォーマットが一致している必要があります。一致していない場合はクラスロード時に java.lang.VerifyError が発生するため，クラスファイルの読み込みや書き換え等を行っている，もしくは，そのようなツールを使用している場合は新仕様対応を行ってください。
なお，クラスファイルのバージョンはバイナリエディタや javap コマンドを使用してクラスファイル内の”major version”から確認してください。

また, javac コマンドのクロスコンパイルオプション(source/target もしくは release)を用いることで指定したバージョンのクラスファイルを作成することもできます。詳細は(2) javac コマンドの source/target/release オプションに指定できる値 を確認してください。

(2) javac コマンドの source/target/release オプションに指定できる値

javac コマンドのクロスコンパイルオプション(source/target もしくは release)を用いることで指定したバージョンのクラスファイルを作成できます。

ただし，その場合はクラスバージョンに対応する Java SE の機能までしか使用することはできません。
JDK17 以降ではクロスコンパイルオプション(source/target もしくは release)に”6”を指定できなくなったため，指定可能なバージョンは”7”以上”17”以下です。

(3) sun.nio.cs.map システムプロパティの削除

JDK17 以降では sun.nio.cs.map システムプロパティは削除されました。そのため，JDK11 以前は以下の指定によりエンコーディング名 shift_jis, csshiftjis,ms_kanji,x-sjis を Windows-31J（MS932）の別名として使用できましたが，JDK17 以降では使用できません。

```
sun.nio.cs.map=Windows-31J/Shift_JIS
```

この動作変更により，Shift_JIS に含まれていない Windows-31J には含まれる日本語の特殊文字について文字化けが発生する可能性があります。その場合には，アプリケーションの改修が必要です。

(4) 内部 API へのアクセス制限厳格化

JDK17 以降では内部 API でのアクセス制限がより厳格化されたため，内部 API を使用する際には--add-opens オプションを指定する必要があります。これまでは–add-opens オプションの代替として--illegal-access オプションを指定することで警告やエラーを回避することが可能でしたが，JDK17 以降では--illegal-access オプションは指定できなくなります。
以下に--add-opens オプションの使用方法を記載します。

--add-opens [使用する内部 API が含まれるモジュール]/[使用する内部 API が含まれるパッケージ] =[アクセス元のモジュール] (,[アクセス元のモジュール])*

(5) その他

その他の移行に関する留意事項については，以下の Web ページを参照してください。
https://docs.oracle.com/en/java/javase/17/migrate/significant-changes-jdk-release.html#GUID-561005C1-12BB-455C-AD41-00455CAD23A6

－以上－