



WebSphere Application Server for z/OS V7.0

**参考: Large Page Support - Compressed
References の使用**

WebSphere software



© 2009 IBM Corporation

免責事項

当資料は、2008年9月に発表されたWebSphere Application Server for z/OS Version 7.0 を前提として作成したものです。

当資料に含まれている情報は正式なIBMのテストを受けていません。また明記にしろ、暗黙的にしろ、何らの保証もなしに配布されるものです。

この情報の使用またはこれらの技術の実施は、いずれも使用先の責任において行われるべきものであり、それらを評価し実際に使用する環境に統合する使用先の判断に依存しています。

それぞれの項目は、ある特定の状態において正確であることがIBMによって調べられていますが、他のところで同じ、または同様の結果が得られる保証はありません。これらの技術を自身の環境に適用することを試みる使用先は、自己の責任において行う必要があります。

登録商標

1. AIX, CICS, Cloudscape, DB2, IBM, IMS, Language Environment, Lotus, MQSeries, MVS, OS/390, RACF, Redbooks, RMF, Tivoli, WebSphere, z/OS, zSeriesは IBM Corporation の米国およびその他の国における商標です。
2. Microsoft, Windows は Microsoft Corporation の米国およびその他の国における商標です。
3. Java, J2EE, JMX, JSP, EJB は Sun Microsystems, Inc. の米国およびその他の国における商標です。
4. 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。
5. 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。

目次

- 概要及び設定手順
 - 概要
 - 前提H/W、S/W
 - WASでの設定方法
- Large Page Support
- Compressed References
- 使用時のパフォーマンス参考情報

概要及び設定手順

概要

▶ Large Page Support

- IBM System z10で新しく追加されたハードウェア・フィーチャー
- 今までの4KB単位に加え、Large Pageでは、1MBチャンクでメモリーをアロケーション・管理する。
- オンにすると、ページングしなくなる。
- 効果: パフォーマンスの向上

▶ Compressed references

- JVMのオブジェクトのポインターを内部的に管理する新しい方法
- 64bitのオブジェクトのポインターのサイズを4バイトに減らすことができる。
 - z/OSのアドレス空間にある隙間を活用
- 効果: 使用メモリーの削減、パフォーマンスの向上

▶ デフォルトは共にオフなので、明示的に指定する必要あり

前提H/W、SW

➤ Large Page Support

- System z10以降
- z/OS V1R9以降
 - APAR OA20902 (z/OS V1R9の場合のみ)
 - APAR OA25485
- IBM Developer Kit for Java™ 6, 64bit版 SR2 以降
- WebSphere Application Server を使用する場合、V7.0.0.1以降

➤ Compressed References

- System z10以降
- z/OS V1R7以降
 - APAR OA26294
- IBM Developer Kit for Java™ 6, 64bit版 SR4 以降
- WebSphere Application Server を使用する場合、V7.0.0.3以降

z/OSの設定手順

➤ Large Page Support

- RACFの設定
 - FACILITY クラスの IARRSM.LRGPAGES プロファイルに対して、SRのユーザー (WSSRU1など) をREAD PERMIT する
 - CR/CRAについても同様。
- z/OS PARMLIBの設定
 - IEASYSxxに、次を追加する
 - LFArea=(xx% | xxxxxxM | xxxxxxG | xxxxxxT)
 - xx%を指定する場合、IPL時のオンライン・ストレージ全体に占めるLarge Pageの割合
 - xxxxxxM, xxxxxxG, xxxxxxTを指定する場合、Large Pageの量

➤ Compressed References

- 特に不要

WAS for z/OS V7.0の設定手順

➤ Large Page Support

- 管理コンソールにて、以下を開く
 - Servers > Server Types > WebSphere application servers > サーバー名 > Configuration > Server Infrastructure > Java and process management > Process definition > servant > Java virtual machine
- Generic JVM argumentsに、**-Xlp**を追加。
- 必要に応じて、上記**servant**を、**control**、**adjunct**に置き換えて、それぞれ追加する。
- **-Xmx**のサイズが起動時にREALストレージとしてアロケーションされることに注意。(=使用メモリーとしては、最大ヒープサイズが取られたときと同じになる)

➤ Compressed References

- 管理コンソールにて、以下を開く
 - Servers > Server Types > WebSphere application servers > サーバー名 > Configuration > Server Infrastructure > Java and process management > Process definition > servant > Environment entries
- 次を新規追加(存在する場合は、値を追加)。
 - Name: **IBM_JAVA_OPTIONS**
 - Value: **-Xcompressedrefs**
- 必要に応じて、上記**servant**を、**control**、**adjunct**に置き換えて、それぞれ追加する。

確認方法

▶ Large Page Support

- RMF Monitor IIIの7Aにて、

```

RMF V1R10 Storage Memory Objects Line 1 of 6
.
. Samples: 500 System: WSZ2 Date: 05/22/09 Time: 19.25.00 Range: 100 Sec
.
. ----- System Summary -----
. -- Memory Objects -- Frames ----- Area Used % ---
. Common Shared Large Common Fixed Shared 1 MB Common Shared 1 MB
. 1 4 4 256 0 32230 524 0.0 0.0 6.4
.
. -----
. Service --- Memory Objects --- Frames --- Bytes ---
. Jobname C Class ASID Total Comm Shr Large 1 MB Total Comm Shr
.
. HF70S S WASSTART 0078 214 0 3 0 0 9.80G 0 90.0M
. HF70SS S WASSTART 0079 165 0 2 4 524 7.60G 0 82.0M
. HF70D S WASSTART 0077 93 0 2 0 0 4.00G 0 40.0M
. SMSPDSE S SYSTEM 0008 23 0 0 0 0 87.0M 0 0
. GRS S SYSTEM 0007 4 0 0 0 0 132G 0 0
. TRACE S SYSTEM 0004 3 0 0 0 0 3.00M 0 0
.
.
.

```

▶ Compressed References

- WASのJoblogにて、

```

<?xml version="1.0" ?>
<verbosegc version="20090213_AA_CMPRSS">
<initialized>
  <attribute name="gcPolicy" value="-Xgcpolicy:optthruput" />
  <attribute name="maxHeapSize" value="0x140000000" />
  <attribute name="initialHeapSize" value="0xa0000000" />
  <attribute name="compressedRefs" value="true" />
  <attribute name="compressedRefsDisplacement" value="0x0" />
  <attribute name="compressedRefsShift" value="0x1" />
  <attribute name="pageSize" value="0x100000" />
  <attribute name="requestedPageSize" value="0x100000" />
</initialized>

```

Large Page Support

Large Page Supportとは

➤ 背景

- MVSの仮想アドレスを実メモリにマッピングするとき、キャッシュされる(Translation Lookaside Buffer, TLB)
- 昨今の使用メモリ増加においてもTLBのサイズは変わらなかったため、相対的にキャッシュヒット率が下がってしまい、パフォーマンスに影響していた。

➤ Large Pageを使用すると、

- z/Architecture DAT構造に則した、1MBのページサイズ
- 一つのTLBに多くのアドレス変換のマッピングが入る
- TLBのカバレッジが拡大できる
- キャッシュヒット率が上がるため、パフォーマンスが向上する
- 起動時に始めから最大ヒープサイズ(-Xmx)のLEヒープが確保されることに注意。十分なメモリが搭載されていることが条件。

Compressed References

Javaのオブジェクト・リファレンスの圧縮

- ▶ Javaのオブジェクトを指すポインターをオブジェクト・リファレンスと呼ぶ
- ▶ 31bit Javaの場合は、これは24バイト。64bit Javaの場合は、48バイト取られる。
- ▶ 64bitでCompressed Referenceモードの場合は、クラス・データやオブジェクト・フィールド(インスタンス)のアドレスが32bit(2^{32})に収まるため、それぞれ短くなり、ムダ(Padding)があった64bitのオブジェクト・リファレンスの半分の24バイトで済む。

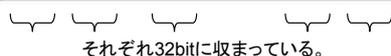
- **31-bit Object (24 bytes)**



- **64-bit Object (48 bytes)**



- **64-bit Compressed References (24 bytes)**



それぞれ32bitに収まっている。

クラス・データやオブジェクト・フィールドのアドレスの圧縮

- ▶ クラス・データやオブジェクト・フィールドが置かれる、JVMヒープ内のアドレスは、ダブルワード境界アライメント(アドレスが8の倍数)で取られる
 - つまり、アドレスの最後の3ビットは常にゼロ
 - この3ビットを有効に使うために、Compressed Referenceモードでは、「右シフト」して圧縮する。
 - (補足)Compressed Referenceオフの場合(デフォルト)、JVMヒープのアドレスは、`x'80000000'`以降に取られる(=2GBまでは 2^{32} に収まる。)

例: 6GBのアドレスの場合



2bit右シフトすると、 2^{32} の範囲に入る。

`x'00000002 80000000'`
 $\begin{array}{c} \diagup \quad \diagdown \\ 0010 \quad 2^{32} \\ \diagdown \quad \diagup \\ \underbrace{\hspace{10em}}_{2^{34}} \end{array}$

`x'00000000 70000000'`

Compressed Referencesによるシフト

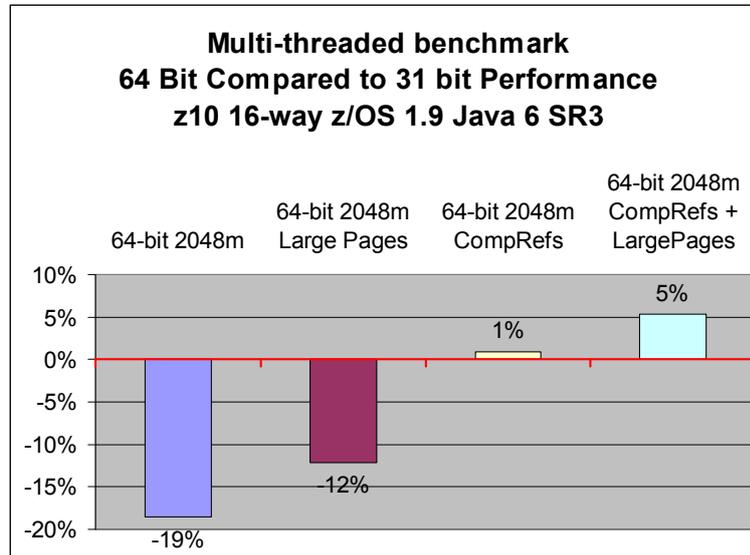
- ▶ Compressed Referencesモードではアドレスを右シフトする。
- ▶ シフトされるビット数は、アドレスのレンジによって表のとおり。
 - シフト量1ビットの場合 (-Xmxが2GB～6GB) は、z/Architectureのbase/index register memory referenceを使うことができるため、それ以上の場合よりパフォーマンスが良いとされる。
- ▶ 注意: -Xmxを30GB以上に指定した場合、Compressed Referenceモードで起動しようとするとう失敗します。

-Xmxで指定される、JVM Heapサイズ	JVM Heapの上限アドレス	Compressed referenceのためにシフトされるビット数
0～2GB未満	2 ³²	0
2GB～6GB未満	2 ³³	1
6GB～14GB未満	2 ³⁴	2
14GB～30GB未満	2 ³⁵	3
30GB以上	2 ³⁵ 以上	Compressed Referenceは使用できない

使用時のパフォーマンス
参考情報

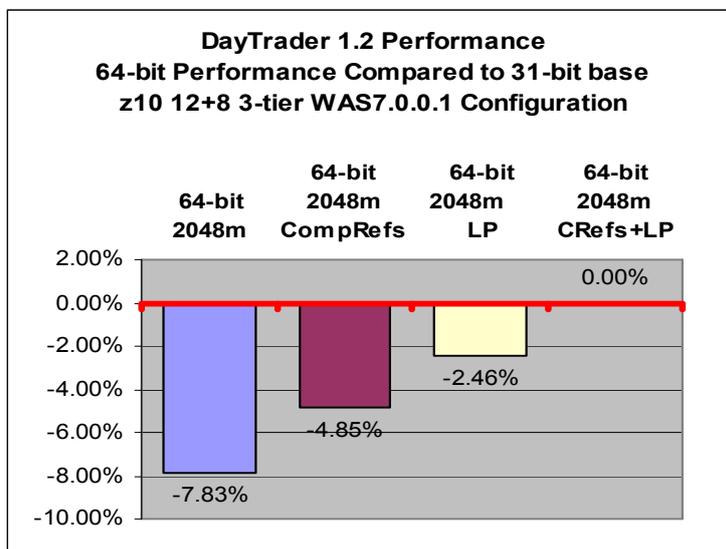
単体Javaのパフォーマンス

- 64bit Javaでも、Large PageとCompressed Referencesを併用すると、31bit Java (0%ライン)のスループットを上回る。



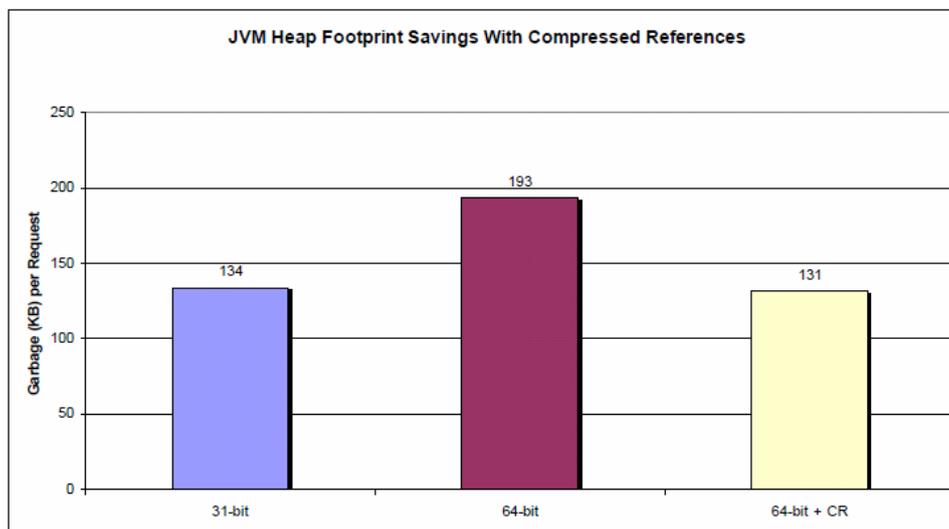
WAS V7.0のパフォーマンス

- ▶ 64bitモードでも、Large PageとCompressed Referenceを併用すると、31bitモードと同等のパフォーマンスが得られる。



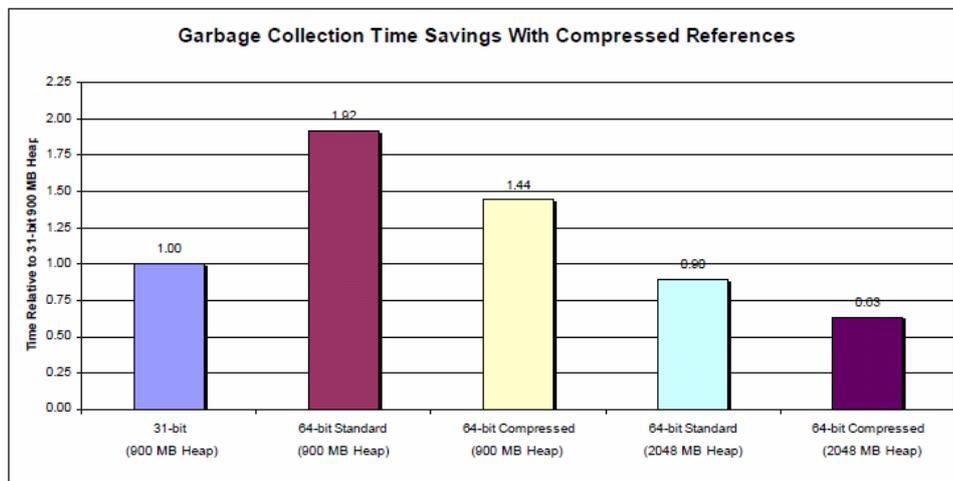
WAS V7.0のパフォーマンス

- ▶ 64bitモードでも、Compressed References (図中CR)をオンにすれば、1リクエストあたりに消費されるGC対象のHeapが31bitモードと同等まで削減される。



WAS V7.0のパフォーマンス

- ▶ 64bitモードでも、Compressed References (図中Compressed)をオンにすると、GCの所要時間が短縮される。
 - Heapサイズが900MBの場合よりも2GBの場合の方が短縮される。



參考資料

- ▶ Match 31-bit WebSphere Application Server performance with new features in 64-bit Java on System z
 - https://www-304.ibm.com/jct09002c/partnerworld/wps/servlet/ContentHandler/whitepaper/systemz/java_websphere/performance?S_TACT=105AGX10&S_CMP=LP