



WebSphere Application Server for z/OS V7.0

WAS for z/OSのカスタマイズ 基礎編

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。

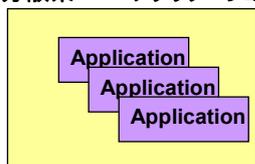
目次

- スレッド数、SR数の変更
- JVMの設定のカスタマイズ
- アプリケーション稼働環境の整備
- WAS for z/OSのログ出力のカスタマイズ
- 導入時セキュリティ

スレッド数、SR数の変更

アプリケーションの稼働数

分散系WASアプリケーション・サーバー



WAS for z/OSアプリケーション・サーバー

HTTP, MDB,
RMI/IIOPリクエ
ストを受け取る

コントローラー
(CR)

WLM
キュー

サーバント(SR)

Application
Application
Application

サーバント(SR)

Application
Application
Application

JVMが動作し、サー
ブレットやEJBが稼
働する

は、一つのアドレススペースを表す

WAS for z/OSは2種類のリージョンでサーバー・イメージを構成します(CRとSR)。

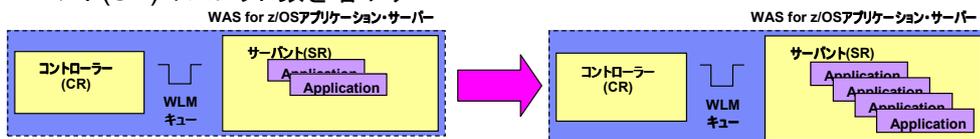
CRは外部との通信を行います。

SRはクローン・イメージのものを複数毎構成することが可能です。

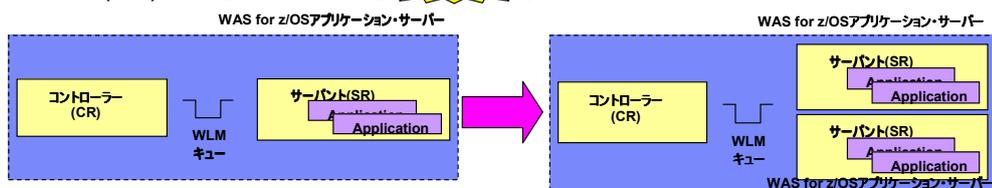
アプリケーションはSRにDeployされます。

アプリケーションの稼働数を増やすには？

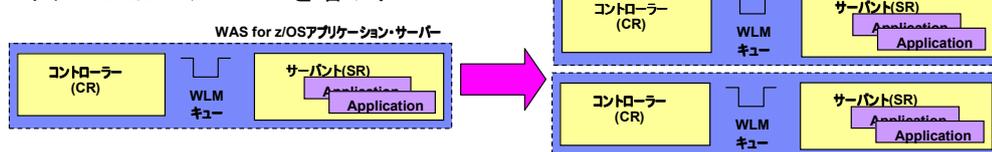
▶ サーバント(SR)のスレッド数を増やす



▶ サーバント(SR)の数を増やす



▶ アプリケーション・サーバーを増やす



アプリケーションの同時稼働数を増やす方法の説明です。

一番上はスレッドの数を増やしています。SRの中のスレッド数を変えています。

真中はWAS for z/OSユニークな方法で、SRの稼働枚数を増やしています。

一番下は所謂垂直クラスター構成になります。

スレッド数の変更

▶ ワークロード・プロファイル

- ワークロード・プロファイルとスレッド数の関係
- Webコンテナ、EJBコンテナ 共通のアプリケーション用のスレッド数
- 設定箇所 :
 - サーバー> サーバー・タイプ>WebSphere Application Server> server_name > コンテナ・サービス>ORB サービス>z/OS の追加設定 (z/OS additional settings)



ワークロード・プロファイル	スレッド数	CPUが1つの場合	CPU (zAAP) が2つの場合	CPU (zAAP) が3つの場合	CPU (zAAP) が12個の場合
ISOLATE	1	1	1	1	1
CPUBOUND	MAX((CPU数-1),3)	3	3	3	11
IOBOUND *1	MIN(30, MAX(5,(CPU数*3)))	5	6	9	30
LONGWAIT	40	40	40	40	40
New V7 CUSTOM	任意の設定数 (最小:1, 最大:100)*2				

*1 : デフォルト値

*2 : servant_region_custom_thread_count サーバー・カスタム・プロパティに指定される値 (設定場所: 環境 > WebSphere 変数)

スレッド数の変更方法です。

従来SRで稼働するスレッド数はCPUの数とワークロードのタイプ毎に4種類のワークロード・プロファイルを選択して決めていました (ISOLATE/CPUBOUND/IOBOUND/LONGWAIT)。

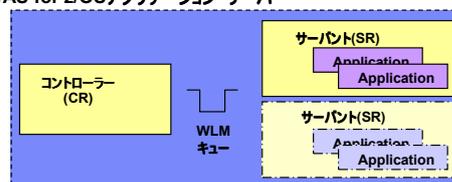
V7からは新たに任意の数値を設定できるようになりました (CUSTOM)。

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.zseries.doc/info/zseries/ae/uorb_rorb_service.html

SRを2枚にする(1/2)

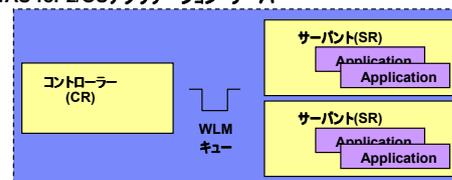
- 最小数=1, 最大数=2
 - 起動時 SR 1枚
 - SR 1枚で処理できなくなってきたらSR 2枚
 - SR 2枚で余裕ができたならまた1枚に戻る

WAS for z/OSアプリケーション・サーバー



- 最小数=2, 最大数=2
 - 常時 SR 2枚
 - サーバントの起動は順次
 - 最小数起動完了でサーバー起動完了

WAS for z/OSアプリケーション・サーバー



WAS for z/OSではSR数の最小値と最大値を設定することができます。

管理コンソールで設定します。

V7からはSRの最大・最小値をコマンドでも変更できるようになりました。それまでは、WASの再起動が必要でした。

SRを2枚にする(2/2)

管理コンソールでの操作

- 設定箇所:
 - サーバー > サーバータイプ > WebSphere Application Server > server_name > 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 > サーバー・インスタンス
- 複数インスタンス使用可能 (チェック・ボックス)
- 最小インスタンス数
- インスタンスの最大数



複数SRを設定するには管理コンソールの該当ページ(上記)で、「複数インスタンス使用可能」がチェックされていること前提です。

➤ MODIFYコマンド(F)によりSR数の最小・最大値を動的に変更できる

- 前提として複数SRの設定をONIにしておく必要がある。

- 設定箇所

- サーバー>サーバータイプ>WebSphere Application Server> server_name > 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 > サーバー・インスタンス > “複数インスタンス使用可能” にチェック
- 内部的には、wlm_dynapplenv_single_server という変数の設定となる

- コマンド実行例

- MINの設定値が現在のSR数より大きい場合はSRが直ちに起動

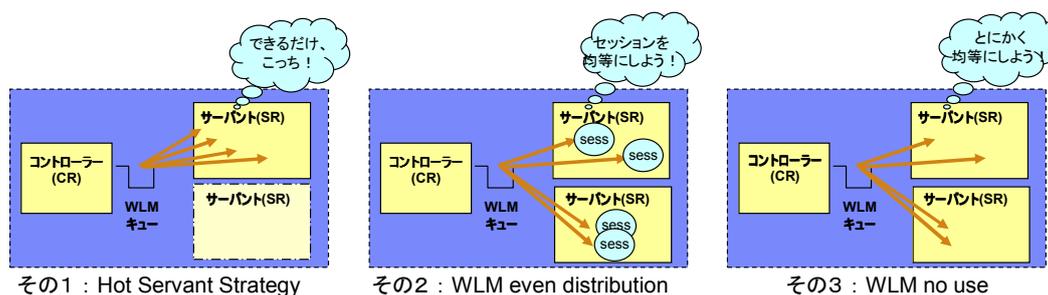
```
F W7S02,WLM_MIN_MAX=(2,3)
BB000211I MODIFY COMMAND WLM_MIN_MAX=(2,3) COMPLETED SUCCESSFULLY
IWM034I PROCEDURE W7S01S STARTED FOR SUBSYSTEM W7S02 574
APPLICATION ENVIRONMENT W7S02
PARAMETERS AMODE=64,JOBNAME=W7S02S,ENV=W7ND1.W7N01.W7S02
```

- 現在の設定状況の確認

```
F W7S02,DISPLAY,WLM
BB000343I WLM MIN/MAX SERVANTS: 2/3
BB000188I END OF OUTPUT FOR COMMAND DISPLAY,WLM
```

SRが複数枚の場合の振分け(1/2)

- ▶ 複数SR時のリクエスト振分けは、パラメーターの設定により3タイプ（どのタイプもセッション・アフィニティの処理が可能）
- その1 : Hot Servant Strategy
 - リクエストを可能な限り一つのSRに振分ける方式
 - デフォルト時の挙動
 - その2 : WLM even distribution
 - セッション・オブジェクト数を均等にするよう振分ける方式（セッションの無い処理には向かない）
 - リクエスト数が均一になるとは限らない
 - その3 : WLM no use
 - WLMを使わずに、リクエストをラウンド・ロビンで処理させる方式
 - セッション・オブジェクト数が均一になるとは限らない
 - 複数サービスクラス使用不可、SR枚数増減不可、の制約がある



複数SRの使われ方には3種類の方法があります。その1 : Hot Servant Strategy はデフォルトの振舞です。

SRが複数枚の場合の振分け(2/2)

➤ 設定箇所

- 環境>WebSphere 変数

	概要	server_use_wlm_to_queue_work	WLMStatefulSession (wlm_stateful_session_placement_on)	server_work_distribution_algorithm
その1 Hot Servant Strategy <省略値>	リクエストを可能な限り一つのSRに振分ける方式。	1 (WLM使用) <省略値>	false (アプリケーション・サーバーの省略値)	- (無効)
その2 WLM even distribution	セッション・オブジェクト数を均等にしよう振分ける方式。 リクエスト数が均一になるとは限らない。	1 (WLM使用) <省略値>	true	- (無効)
その3 WLM no use (Round Robin)	WLMを使わずに、リクエストをラウンド・ロビンで処理させる方式。 セッション・オブジェクト数が均一になるとは限らない。	0 (zero) (WLM未使用)	- (無効)	1 (ラウンド・ロビン・アルゴリズム)

※組合せとしては、server_use_wlm_to_queue_work=0 server_work_distribution_algorithm=0 があるが、これは非推奨設定

複数SRの振分け方法毎と設定方法を表にまとめています。

- ▶ 複数SR時、従来は1枚ずつSRは順番にスタートしていた。
 - 平行・スタートをセットすると、初めのSRが単独で起動した後のSRは平行に起動する。
 - 設定箇所：
 - 環境 > WebSphere 変数 > Cell/Node/Serverを選択
 - WebSphere変数 : wlm_servant_start_parallel
 - 設定値 : 1 で平行・スタート、0でシリアル・スタート

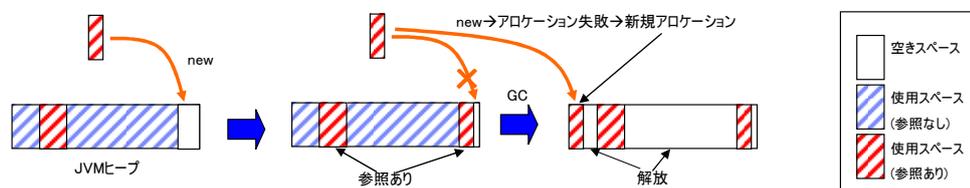
参照URL : http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/zseries/ae/urun_rproperty_custproperties.html

従来は複数SR設定時、SRは必ず一アドレス空間毎にシリアルに起動されていました。多数のSR設定をすると、CPUの数に余裕がある場合でも全SRの起動を完了するのに余計な時間がかかってしまっていたのを解消するのが平行・スタートです。平行・スタートの設定をしても、最初の一枚は単独でスタートしますので、この設定が有効なのはSR数(最小のSR数設定)が3以上の時です。

JVMの設定のカスタマイズ

JVMヒープとは？ ガーベッジ・コレクション(GC)とは？

- JVMヒープとは？
 - オブジェクトのスペースを割り当てる領域
- Javaでのメモリー管理
 - オブジェクト生成でメモリーが割りあてられる
 - オブジェクトが不要になり、オブジェクトへの参照がなくなる
 - プログラマーが解放するメモリーを覚えている必要はない
- ガーベッジ・コレクション(GC)とは？
 - 参照がないオブジェクトに割り当てられたメモリーを探し解放する行為
 - 新規オブジェクト生成で割りあててるスペースが無い場合に行われる



© 2009 IBM Corporation

15

JavaはそのランタイムであるJVM(Java Virtual Machine)によるMemory管理に任せることで、ユーザーによるメモリーの獲得・解放をアプリケーションから意識しない。言語稼働環境を提供します。

Javaのプログラム・インスタンス(オブジェクト)の稼働するメモリー・エリアをJVMヒープと呼びます。

JVMによる自動メモリー開放のための仕組みをGCと呼んでいます。

JVMヒープに関するカスタマイズのポイント(1/2)

- ▶ JVMが稼働するアドレス・スペース
 - デモン以外のすべての全WASアドレス・スペース
 - デプロイメント・マネージャーのコントローラー(CR)、サーバント(SR)
 - ノード・エージェント
 - アプリケーション・サーバーのコントローラー(CR)、サーバント(SR)、アジャクト(CRA)
 - 基本的に調整が必要なのはアプリケーション・サーバーのサーバント(SR)のJVMヒープ・サイズ
- ▶ JVMヒープ・サイズの調整
 - 初期ヒープ・サイズ(SRの初期値:256MB)
 - 最大ヒープ・サイズ(SRの初期値:512MB)
- ▶ JVMヒープ・サイズ調整のための情報取得
 - JVMヒープの使用状況は、GCログにて把握する
 - サーバント(SR) では、`servant.jvm.options`に以下の設定が追加される
 - verbose:gc
 - GCログはJOBLOG(SYSOUT)へ出力される
- ▶ GCポリシーの変更(オプション)
 - 省略時値 : `-Xgcpolicy:optthruput` (JDK 1.4.2までのGCロジックと似ている)
 - 設定例 : `-Xgcpolicy:gencon` (世代別GC)
 - 参考情報 : http://publib.boulder.ibm.com/infocenter/javasdk/v6r0/topic/com.ibm.java.doc.user.zos.60/user/garbage_options.html

WASではJVMヒープを管理コンソールで設定します。

GCの方式には何種類かありますが、WAS V7 (Java SDK6)ではデフォルトのoptthruput、もしくはgenconのどちらかを選択するのが一般的です。

JVMヒープに関するカスタマイズのポイント(2/2)

➤ 設定箇所

- サーバー > サーバータイプ > WebSphere Application Server > server_name > サーバー・インフラストラクチャー > プロセス定義 > Servant > Java仮想マシン

Integrated Solutions Console ようこそ wsadmin ヘルプ | ログアウト

表示: すべてのタスク

- ようこそ
- ガイド付きアクティビティ
- サーバー
 - 新規サーバー
 - サーバータイプ
 - WebSphere Application Server
 - WebSphere プロキシサーバー
 - 汎用サーバー
 - バージョン 5 JMS サーバー
 - WebSphere MQ サーバー
 - Web サーバー
- クラスター
 - DataPower
 - コアグループ
- アプリケーション
- サービス
- リソース
- セキュリティ
- 環境
- システム管理
- ユーザーおよびグループ
- モニターおよびチューニング

冗長クラスロード

冗長ガーベージコレクション

冗長 JNI

初期ヒープサイズ
256 MB

最大ヒープサイズ
512 MB

HProf の実行

HProf 引数

デバッガーモード

デバッガー引数
-agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=7777

汎用 JVM 引数
-Xgcpolicy:gccon

実行可能 JAR ファイル名

JIT を使用不可にする

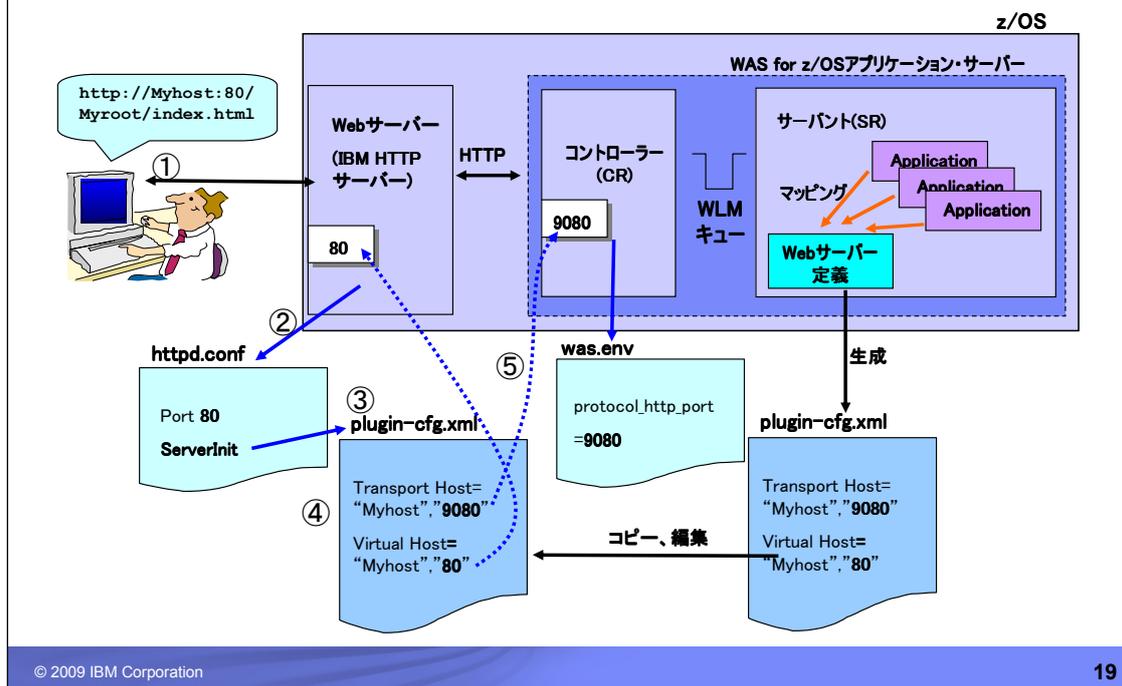
オペレーティングシステム名
E/OS

適用 OK リセット 取り消し

JVMヒープに関する管理コンソールでの設定イメージです。

アプリケーション稼働環境の整備

Webアプリケーション稼働環境



© 2009 IBM Corporation

19

Webアプリケーション (HTTP) の一般的な処理順序です。

リクエストの流れ

- ① クライアントからWebサーバーへリクエスト送付 <ホスト名(Myhost)、ポート番号(80) >
- ② Webサーバーでリクエスト受信 <ホスト名(Myhost)、ポート番号(80) >
- ③ httpd.confに指定されたURIにマッチし、Webサーバー・プラグインにリクエスト転送
- ④ Webサーバー・プラグインがplugin-cfg.xmlの情報を元に転送先サーバーを決定
- ⑤ WebサーバーからWAS(HTTPトランスポート・ハンドラー)へリクエスト転送 <ホスト名(Myhost)、ポート番号(9080) >

2種類のz/OS版IHS

➤ IBM HTTP Server v5.3 for z/OS

- 略称 : IHS for z/OS
- Lotus Domino GO Webserver ベース
- 参照情報 :

http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/IMWZIU18/CCONTENTS?SHELF=EZ2ZBK0G&DN=SC34-4826-09&DT=20070518142743

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/zseries/ae/trun_plugin_390.html

➤ IBM HTTP Server for WebSphere Application Server on z/OS

- 略称 : IHS for WAS on z/OS
- Apacheベース
- 参照情報 :

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.ihs.doc/info/welcome_ihs.html

どちらもプラグイン・モジュールを組み込むことでWAS for z/OSのフロントエンドWebサーバーとして使用できます。

※ 分散環境のWebサーバーもWAS for z/OSのフロントエンドWebサーバーになります。この場合分散環境のWASで提供されるプラグイン・モジュール(Cプログラム)が必要になります。

z/OS環境でWAS for z/OSのフロント・エンドとして使用できるIHSには2種類あります。

IBM HTTP Server v5.3 for z/OSはz/OSの基本コンポーネントとしてパッケージされたIHSで、OS/390の頃から提供されてきました。

従来からz/OS上でWebサーバーを使われている場合はこちらのIHSを使われるケースが多く、今後も継続使用していただけます。

IBM HTTP Server for WebSphere Application Server on z/OSはWASのオプション・コンポーネントとして提供されるIHSで、WAS for z/OS v6.1 から提供されるようになりました。

Apacheは分散環境を含め多くのユーザーに親しまれているWebサーバーのため、そのスキルの調達の上易さがメリットとして挙げられ、今後の利用拡大が期待されます。

IHSでのWASプラグイン・モジュールの組み込み

➤ IHS for z/OS

1. プラグイン・モジュール・ファイルに許可 (755 +p) を設定
 - ✓ パスの例 : /usr/lpp/zWebSphere/V7R0/bin/ihs390WAS70Plugin_http.so
 - ✓ chmod 755 ihs390WAS70Plugin_http.so
 - ✓ extattr +p ihs390WAS70Plugin_http.so
2. 構成ファイル (httpd.conf) の編集
 - ✓ ServerInit、ServerTerm、Service ディレクティブを httpd.conf 構成ファイルに追加
 - ✓ ServerInitディレクティブでplugin-cfg.xmlのパス名を指定
 - ✓ ServiceディレクティブでWASで実行するアプリケーションのURIを指定
3. Webサーバーの再始動

httpd.confのサンプル

```
Port      80
#
ServerInit /usr/lpp/zWebSphere/V7R0/bin/ihs390WAS70Plugin_http.so:init_exit /etc/plugin-cfg.xml
Service /Myroot1/* /usr/lpp/zWebSphere/V7R0/bin/ihs390WAS70Plugin_http.so:service_exit
Service /Myroot2/* /usr/lpp/zWebSphere/V7R0/bin/ihs390WAS70Plugin_http.so:service_exit
Service /Myroot3/* /usr/lpp/zWebSphere/V7R0/bin/ihs390WAS70Plugin_http.so:service_exit
:
ServerTerm /usr/lpp/WebSphere/V7R0/bin/ihs390WAS70Plugin_http.so:term_exit
```

➤ IHS for WAS on z/OS

1. install_plugin.sh でランタイムイメージを作成
2. ConfigureIHSPlugin.shで構成ファイルを作成
 - ✓ IHS for z/OSのServiceディレクティブのようにURLとプラグイン・モジュールのマッピングを定義する必要が無い
3. Webサーバーの再始動

各々のIHSへのWASプラグイン・モジュールの組み込み方法の概要です。

詳細は以下のサイトをご参照下さい。

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.zseries.doc/info/zseries/ae/trun_plugin_390.html?resultof=%22%70%6c%75%67%69%6e%22%20%22%73%65%72%76%69%63%65%22%20%22%73%65%72%76%69%63%22%20%22%73%65%72%76%65%72%69%6e%69%74%22%20

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.zseries.doc/info/zseries/ae/trun_plugin_ihsz.html?resultof=%22%43%6f%6e%66%69%67%75%72%65%49%48%53%50%6c%75%67%69%6e%2e%73%68%22%20

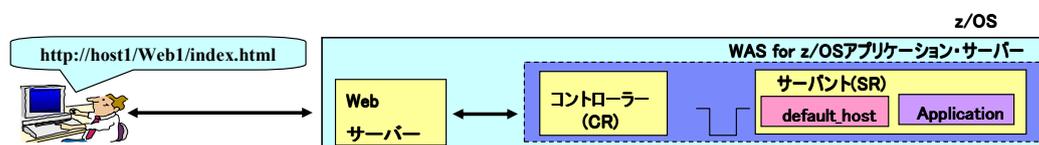
Webアプリケーション稼働環境設定 (1/5)

- ① 仮想ホストとホスト別名の定義(オプション)
- ② Webサーバーの定義
- ③ アプリケーションをWebサーバーに関連付けてデプロイ
- ④ Webサーバー・プラグイン構成ファイル(plugin-cfg.xml)の生成と伝搬
 - ✓ plugin-cfg.xmlはASCIIで生成されます
- ⑤ Webサーバー・プラグイン構成ファイル(plugin-cfg.xml)の編集(オプション)

Webアプリケーション稼働環境設定 (2/5)

▶ (①の補足) 仮想ホストとは

- 単一のホスト・マシンを複数のホスト・マシンに見せかけて使用できるようにするための構成
- 仮想ホストを 1 つ以上の Web モジュールに関連付けることができます
- 各 Web モジュールは、1 つの仮想ホストにのみ関連付けることができます
- 個々の仮想ホストには1 つ以上の ホスト別名があります
- 仮想ホストはこのエイリアスによって認識されます
- ホスト別名は、TCP/IP のホスト名およびポート番号です
- デフォルト仮想ホスト
 - アプリケーション・サーバー構築時、デフォルト仮想ホストが構成されています(名前は default_host)
 - デフォルト仮想ホストのホスト別名は、 *:80 および *:9080 に構成されています
 - ポート 80 は HTTP サーバーのポート
 - ポート 9080 はデフォルトのサーバーの HTTP トラフィック用のポートです
 - default_host が用意されているため、多くのユーザーにとっては仮想ホストを作成する必要はありません



http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.zseries.doc/info/zseries/ae/ccws_vhost.html?resultof=%22%76%69%72%74%75%61%6c%22%20%22%68%6f%73%74%22%20

Webアプリケーション稼働環境設定 (3/5)

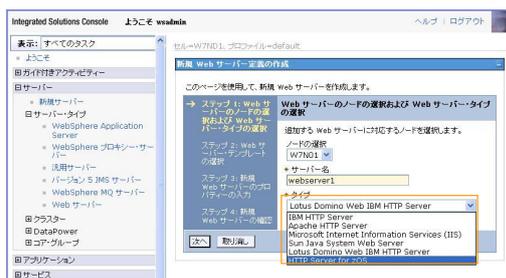
➤ ① 仮想ホストとホスト別名の定義

- 環境 > 仮想ホスト > default_host > ホスト別名
- IHSプラグインからHTTP/HTTPSリクエストを受け入れるには Web Serverポート(この例では80/443)の設定が必要(サーバー構築時に自動設定)
- WASのHTTP/HTTPSリクエストの受け口はHTTP/HTTPSトランスポート(この例では9080/9443)



➤ ② Webサーバーの定義

- サーバー > Webサーバー > 新規作成
- (ポイント)ステップ1
 - IHS for z/OS : HTTP Server for z/OS
 - IHS for WAS on z/OS : IBM HTTP Server
- (ポイント)ステップ3
 - プラグインのインストール・ロケーションにIHSが読む plugin-cfg.xmlのパスをセット



Webアプリケーション稼働環境設定 (4/5)

➤ ④ Webサーバー・プラグイン構成ファイル(plugin-cfg.xml)の生成と伝搬

■ その1. セル全体

- 設定箇所 :
 - 環境 > グローバルWebサーバー・プラグイン構成の更新 > OK
- 生成ファイル・パス /<config root>/DeploymentManager/config/cells/plugin-cfg.xml

■ その2. Webサーバー単位

- 設定箇所 :
 - サーバー > Webサーバー > (Webサーバーの選択) > プラグインの生成
- 生成のパス : <WASROOT>/profiles/<profilename>/config/cells/././nodes/./servers/<webservname>/
- 伝搬のパス : <PluginInstallRoot>/config/<webservname>/
- プラグイン構成ファイルの自動生成、自動伝搬も可能



セル内のすべてのアプリ・サーバーが同じWebサーバーを経由してリクエストを受ける場合には、その1.の方法をします。

個々のWebサーバー毎に個別の plugin-cfg.xml を使用する場合はその2.の方法をします。

Webアプリケーション稼働環境設定 (5/5)

⑤ plugin-cfg.xmlの編集



- `<Log>`セクション: ログ・ファイルとログ・レベルを定義します
- `<ServerCluster>`セクション: サーバー・クラスターとサーバー・メンバーを定義します。
CloneIDでセッション・アフィニティのリクエストを該当サーバーへ転送
- `<VirtualHostGroup>`セクション: 仮想ホスト名を定義します
- `<UriGroup>`セクション: 処理するURIを定義します
- `<Route>`セクション: UriGroupをServerClusterとVirtualHostGroupに関連付けます

`plugin-cfg.xml` はasciiエンコードですので、個別編集をする場合はviasciiを使用します。

`plugin-cfg.xml` は定期的に再読み込みされます(`RefreshInterval`)ので、IHSの再起動は不要です。

問題判別の際に**LogLevel="Trace"**に変更して資料採りを行うことがありますが、資料収集後は必ず元に戻しましょう。

補足 : plugin-cfg.xmlの例

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- HTTP server plugin config file for the cell HY6CBP generated on 2006.06.02 at 03:49:48 AM GMT+00:00 -->
<Config ASDisableNagle="false" AcceptAllContent="false" AppServerPortPreference="HostHeader" ChunkedResponse="false" FIPSEnable="false" IISDisableNagle="false" IISPluginPriority="High" IgnoreDNSFailures="false"
RefreshInterval="60" ResponseChunkSize="64" WildMatchingCompat="false">
  <Log LogLevel="Error" Names="/opt/IBM/WebSphere/plugins/logs/http_plugin.log" />
  <Property Name="ESIEnable" Value="true" />
  <Property Name="ESIMaxCacheSize" Value="1024" />
  <Property Name="ESIInvalidationMonitor" Value="false" />
  <VirtualHostGroup Name="default_host">
    <VirtualHost Names="*:9080" />
    <VirtualHost Names="*:80" />
    ...(- 節省略) ...
  </VirtualHostGroup>
  <ServerCluster CloneSeparatorChange="false" IgnoreAffinityRequests="true" LoadBalance="Round Robin" Name="WSCLA" PostBufferSize="64" PostSizeLimit="-1" RemoveSpecialHeaders="true" RetryInterval="60">
    <Server ConnID="BEE450CB9A9024C7000001FC00000007C0A82B85" ConnectTimeout="0" ExtendedHandshake="false" LoadBalanceWeight="1" MaxConnections="-1"
      Name="WSS11A" ServerIOTimeout="0" WaitForContinue="false">
      <Transport Hostname="wsz1.makuhari.ibm.com" Port="9080" Protocol="http" />
      <Transport Hostname="wsz1.makuhari.ibm.com" Port="9443" Protocol="https" />
      <Property Name="keyring" Value="/WebSphere/WSZ1/DeploymentManager/profiles/default/etc/plugin-key.kdb" />
      <Property Name="stashfile" Value="/WebSphere/WSZ1/DeploymentManager/profiles/default/etc/plugin-key.sth" />
    </Transport>
    <Server ConnID="BEE450CCE3D62E47000001FC00000007C0A82B85" ConnectTimeout="0" ExtendedHandshake="false" LoadBalanceWeight="2" MaxConnections="-1"
      Name="WSS12A" ServerIOTimeout="0" WaitForContinue="false">
      <Transport Hostname="wsz2.makuhari.ibm.com" Port="9080" Protocol="http" />
      <Transport Hostname="wsz2.makuhari.ibm.com" Port="9443" Protocol="https" />
      <Property Name="keyring" Value="/WebSphere/WSZ1/DeploymentManager/profiles/default/etc/plugin-key.kdb" />
      <Property Name="stashfile" Value="/WebSphere/WSZ1/DeploymentManager/profiles/default/etc/plugin-key.sth" />
    </Transport>
  </Server>
  <PrimaryServers>
    <Server Name="WSS11A" />
    <Server Name="WAA12A" />
  </PrimaryServers>
  </ServerCluster>
  <UriGroup Name="default_host_WSCLA_URIs">
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jessionid" Name="/HelloWeb/**" />
  </UriGroup>
  <Route ServerCluster="WSCLA" UriGroup="default_host_WSCLA_URIs" VirtualHostGroup="default_host" />
</Config>
```

Log

VirtualHostGroup

ServerCluster

UriGroup

Route

WAS for z/OSのログ出力の
カスタマイズ

WAS for z/OSのログ

- WAS for z/OSのログ
 - JOBLOG – SYSPRINT
 - 標準出力など
 - System.out.println, System.err.println, Tr.Sysout, stdout, client output
 - JOBLOG –SYSOUT
 - JVM GCトレース情報など
 - Tr.Syserr, STDERR
- ログのタイムスタンプ
 - ログのタイムスタンプをローカル・タイムにする
 - ※マシンのTOD(Time Of Day)をローカル・タイムにしているシステムではこの設定は必要ありません
- 出力先のカスタマイズ
 - ログファイルをHFSに書き出すこともできる
- SYSOUT, SYSPRINTの分割

WAS for z/OSは分散系WASでのSystem.outやSystem.err に相当するログがJOBログに書き出されます。

ログのタイムスタンプのカスタマイズ

- JVMから出力しているログ
 - 対象ログ
 - SYSOUTのGCログ など
 - TZ変数にJST-9を設定
 - 管理コンソール設定
 - ・ 環境>WebSphere 変数 >新規作成
- JVM以外から出力しているログ
 - 対象ログ
 - エラー・ログ
 - トレース
(SYSPRINT または TRCFILE DD)
 - ras_time_local変数に 1 を設定
 - 0はGMT, 1はローカル時間
 - 管理コンソール設定
 - ・ 環境>WebSphere 変数 >新規作成

※マシンのTOD(Time Of Day)をローカル・タイムにしているシステムではこの設定は必要ありません

© 2009 IBM Corporation

WebSphere 変数

WebSphere 変数 > 新規作成

このページを使用して、置換変数を定義します。この変数は、ファイルリネームなどの、システム定義値の置換指示のレベルを指定します。変数またはセルの有効範囲レベルがあります。1つの有効範囲レベルになる値にできます。変数が有効範囲値と矛盾する場合は、より細かい範囲レベルの値をオーバーライドします。すなわちサーバー変数はノード変数をオーバーライドし、クラスター変数はセル変数をオーバーライドします。

構成

一般プロパティ

* 名前
TZ

値
JST-9

説明

WebSphere 変数

WebSphere 変数 > ras_time_local

このページを使用して、置換変数を定義します。この変数は、ファイルリネームなどの、システム定義値の置換指示のレベルを指定します。変数またはセルの有効範囲レベルがあります。1つの有効範囲レベルになる値にできます。変数が有効範囲値と矛盾する場合は、より細かい範囲レベルの値をオーバーライドします。すなわちサーバー変数はノード変数をオーバーライドし、クラスター変数はセル変数をオーバーライドします。

構成

一般プロパティ

* 名前
ras_time_local

値
1

30

ログのタイム・スタンプの調整が必要な場合、書き出し元の違いによって2箇所のカスタマイズが必要です。

HFSへのログ出力リダイレクト

▶ サーバー起動JCLの書き換え例

```

/* * SYSOUT DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSOUT DD PATHMODE=(SIRWXU,SIRWXG,SIROTH),
// PATHOPTS=(OWRONLY,OCREAT),
// PATH='/var/WebSphere/logs/sysout.log.d&LYMMDD..t&LHHMMSS.'
/* * SYSPRINT DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSPRINT DD PATHMODE=(SIRWXU,SIRWXG,SIROTH),
// PATHOPTS=(OWRONLY,OCREAT),
// PATH='/var/WebSphere/logs/sysprt.log.d&LYMMDD..t&LHHMMSS.'

```

▶ HFSへの出カイメージ

```

WSZ1:E20128:/WSZ1/var/WebSphere/logs: >ls -l
total 784
-rwxrwxr-- 1 ASSR1 WSCFG1 13443 Apr 15 16:32 sysout.log.d090415.t163148
-rwxrwxr-- 1 ASSR1 WSCFG1 114775 Apr 15 16:32 sysprt.log.d090415.t163148
WSZ1:E20128:/WSZ1/var/WebSphere/logs: >tail -f sysout.log.d090415.t163148
BossLog: { 0091} 2009/04/15 07:32:17.492 02 SYSTEM=WSZ1 CELL=W7ND1 NODE=W7N01 CLUSTER=W7S02
SERVER=W7S02 PID=0X00010191 TID=0X00000022 0X371FC22C c=./bbgrjT
BossLog: { 0092} 2009/04/15 07:32:18.214 02 SYSTEM=WSZ1 CELL=W7ND1 NODE=W7N01 CLUSTER=W7S02
SERVER=W7S02 PID=0X00010191 TID=0X00000022 0X371FC22C c=./bbgrj.
BossLog: { 0093} 2009/04/15 07:32:18.229 02 SYSTEM=WSZ1 CELL=W7ND1 NODE=W7N01 CLUSTER=W7S02
SERVER=W7S02 PID=0X00010191 TID=0X00000022 0X371FC22C c=./bbgrj.
BossLog: { 0094} 2009/04/15 07:32:18.704 02 SYSTEM=WSZ1 CELL=W7ND1 NODE=W7N01 CLUSTER=W7S02
SERVER=W7S02 PID=0X00010191 TID=0X00000020 0X269FC22C c=./bbgrj.
:
:

```

JOBログに書き出される情報をHFS(zFS)にリダイレクトするには上記のようにWASの起動JCLに修正を加えます。

JOBログとHFS両方に出力することはできません。

SYSOUT, SYSPRINTの分割

- 長期稼働対策で、SYSOUT, SYSPRINTを分割して運用できます
 - WASサーバーを停止しなくても、SYSOUT, SYSPRINTをJESプール内で分割できます
- 設定方法
 - 改ページ文字を出力する間隔を定義
 - 管理コンソール:
 - 環境 > WebSphere 変数 > 新規作成
 - ras_stderr_ff_interval (単位:分)
 -  – ras_stderr_ff_line_interval (フォーム・フィード・キャラクター間の出力行数)
 - ras_stdout_ff_interval (単位:分)
 -  – ras_stdout_ff_line_interval (フォーム・フィード・キャラクター間の出力行数)
 - スタート・プロシージャのDDのSEGMENT定義 – SEGMENTするページ数を指定
 - SYSPRINT DDとSYSOUT DDステートメントにSEGMENTパラメーターを追加
 - (例)


```
//CEEDUMP DD SYSOUT=*, SPIN=UNALLOC, FREE=CLOSE
//SYSOUT DD SYSOUT=*, SPIN=UNALLOC, FREE=CLOSE, SEGMENT=2
//SYSPRINT DD SYSOUT=*, SPIN=UNALLOC, FREE=CLOSE, SEGMENT=2
```

JOBログを時間単位または量(行数)で分割(ログを切る)ことができます。
HFSにリダイレクトした場合、この機能は使えません、

補足: WAS v7.0 SRスタート・プロシージャ修正例

【修正前】 SRスタート・プロシージャ:W7S01S

```
//W7S01S PROC ENV=,AMODE=00
// SET ROOT='/WebSphere/W7ND1/W7N01'
// SET WSDIR='AppServer'
//BBOPASR EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,MEMLIMIT=NOLIMIT,
// PARM='PGM &ROOT./&WSDIR./lib/bboosrmr &AMODE.'
//STDENV DD PATH='&ROOT/&ENV/was.env'
//*
//DEFAULTDD DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//HRDCPYDD DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSOUT DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//CEEDUMP DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSPRINT DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
/*
```

【修正後】 SRスタート・プロシージャ:W7S01S

```
//W7S01S PROC ENV=,AMODE=00
// SET ROOT='/WebSphere/W7ND1/W7N01'
// SET WSDIR='AppServer'
//BBOPASR EXEC PGM=BPXBATSL,REGION=0M,TIME=NOLIMIT,MEMLIMIT=NOLIMIT,
// PARM='PGM &ROOT./&WSDIR./lib/bboosrmr &AMODE.'
//STDENV DD PATH='&ROOT/&ENV/was.env'
//*
//DEFAULTDD DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//HRDCPYDD DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSOUT DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE,SEGMENT=2
//CEEDUMP DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE
//SYSPRINT DD SYSOUT=*,SPIN=UNALLOC,FREE=CLOSE,SEGMENT=2
/*
```

SYSOUT, SYSPRINTの分割の出力例

➤ SYSOUT, SYSPRINTを分割の設定例

- サーバーにras_stderr_ff_interval=3、ras_stdout_ff_interval=3 を設定
- SYSPRINT, SYSOUTに、SEGMENT=2 を設定
 - サーバー起動完了後からほぼ6分ごとに分割されます (1ページが3分、2ページで分割)

```

SDSF JOB DATA SET DISPLAY - JOB WSS11AS (STC05185)
NP  DDNAME  StepName ProcStep DSID Owner  Rec-Cnt  Byte-Cnt  CrTime
JESMSG LG JES2      2 ASSR1   368    26,595   18:07:51
JESJCL  JES2      3 ASSR1   24     1,151   18:07:51
JESYSMSG JES2     4 ASSR1    3     210    18:07:51
SYSPRINT WSS11AS 103 ASSR1 1,654  84,496  18:07:51
SYSOUT   WSS11AS 104 ASSR1 35    2,119   18:07:51
-----
SYSPRINT WSS11AS 105 ASSR1 0      8    18:17:56
SYSOUT   WSS11AS 106 ASSR1 0      8    18:17:56
-----
SYSPRINT WSS11AS 107 ASSR1 0      8    18:23:55
SYSOUT   WSS11AS 108 ASSR1 0      8    18:23:55
-----
SYSPRINT WSS11AS 109 ASSR1 0      8    18:29:55
SYSOUT   WSS11AS 110 ASSR1 0      8    18:29:55

```



※ ras_stderr_ff_line_interval, ras_stdout_ff_line_intervalはLOGにラインフィード文字が書かれるタイミングに依存するため、出カライン数は設定値よりも多少ずれます。

3 x 2 で6分です。

管理セキュリティー

管理セキュリティ

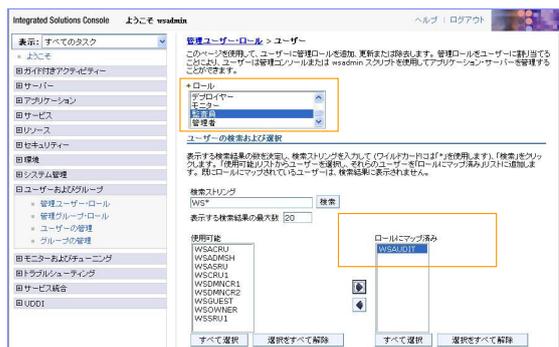
- 管理コンソールの使用前に認証を実施するには管理セキュリティをセットします。
 - 設定箇所：
 - セキュリティー > グローバル・セキュリティ
 - ここでは、管理ユーザーはローカル・オペレーティング・システム(RACF)を前提を想定しています。



管理者ユーザーの追加

➤ RACFユーザーの追加

- 前提: グループWSCFG1が管理者に必要なEJBROLEのアクセス権を持っていて、新たなユーザーWSAUDITを監査用のユーザーとして追加したい ⇒ WSAUDIT を WSCFG1 にコネクトする
CONNECT WSAUDIT GROUP (WSCFG1)
- 設定箇所 :
 - ユーザー及びグループ > 管理ユーザー・ロール
- ロールの選択で監査員を選択
- RACFユーザー (WSAUDIT) をロールにマップ済みに登録



管理ユーザー (WSADMIN) から他のユーザー (WSAUDIT) が管理できるようになる。



詳細情報 : http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.zseries.doc/info/welcome_nd.html