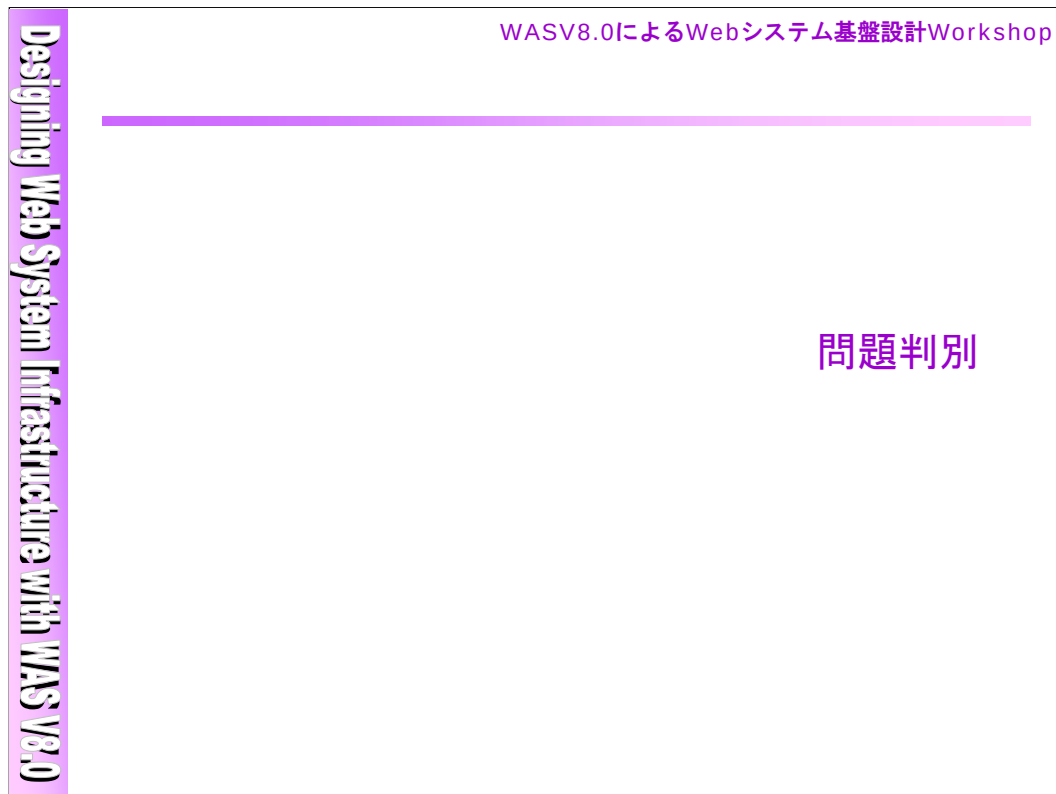




Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## 当セッションの目的

- 問題判別を行うに当たっての心得について確認する
  - ◆ 基本手順、確認すべきサイト、情報収集方法
- 問題判別のポイントについて理解する
  - ◆ 各コンポーネントのログ・トレース
    - Edge Component、IHS、プラグイン、WASのそれぞれのログと、トレース取得方法
  - ◆ IBM Support Assistant (ISA) 5.0 Technical Preview
    - ISA で使用可能な各種ツールの使用方法
  - ◆ 症状別問題判別方法
    - ブラウザへ応答が返ってこない時の問題判別の一例、ハング/ループ、スローダウン、OutOfMemory、クラッシュなど、代表的な症状における問題判別方法
  - ◆ JVM 問題判別はセッション 08 JVM を参照

2

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## Agenda

1. 問題判別の心得
  - 1-1 問題判別の基本
  - 1-2 情報収集・問題判別ツール
2. 問題判別
  - 2-1 ログ、トレース
  - 2-2 症状別問題判別

まとめ・参考文献

3

Designing Web System Infrastructure with WAS V8.0

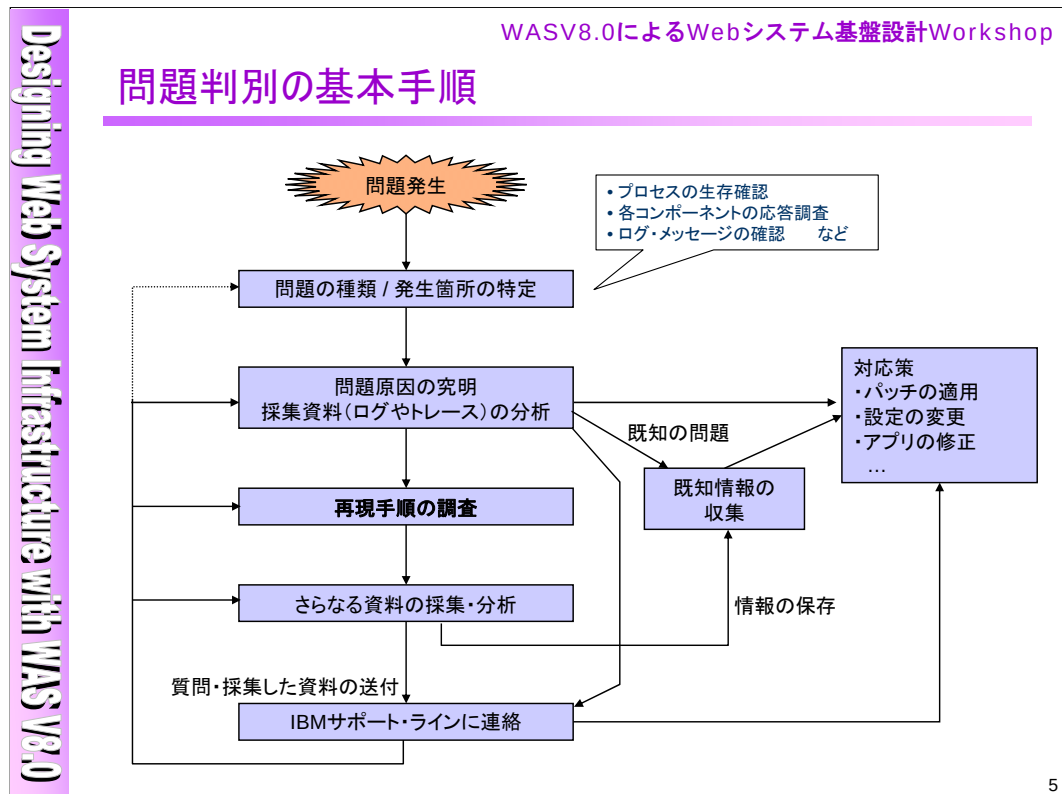
## 1.問題判別の心得

---

### 1-1 問題判別の基本

### 1-2 情報収集・問題判別ツール

4



上記は、WebSphere製品に関わらず、一般的なOS/ソフトウェア製品の問題判別の基本手順になります。

問題判別とは、発生した問題の原因を特定し、その問題を解決するための対応策を実施して、問題が解決したことをテストで確認するための作業です。

問題が発生したら、その問題の種類や発生箇所の分析を行います。そして、取得したログやトレースを使用して、その問題が起こっている原因を究明します。その原因に応じて、パッチの適用や設定の変更、アプリケーションの修正などといった対応をします。原因が既知の問題である場合には既知情報を収集し、その問題に対して適切な対応をします。

次に、その問題の再現性について調査します。再現性がある場合はその再現手順について調査します。そして、さらに公開されている資料や既知情報をもとに問題の分析を行います。ここで調査した情報を、今後同じような問題が起こったときのために残しておくことも大切になります。

再現性の有無は、問題解決の可能性に大きく影響します。再現性のない問題の解決は非常に困難です。

一度は再現性がないように思えても、問題発生ごとに状況を詳しく記録・分析することで、初めはわからなかった再現手順が明らかになることがあります。

最終的に問題への対応方法が分からず、解決されない場合は、サポート・センターに連絡し解析を依頼します。その際、質問内容と一緒にログやトレースなどの採集した資料も送付します。

Designing Web System Infrastructure with WAS V8.0

WAS V8.0によるWebシステム基盤設計Workshop

問題が起こる前に

- 障害情報や修正プログラム情報の収集
  - ◆ WASサポート・サイトから最新リリースをチェック
    - 最新リリースの適用を推奨
  - ◆ 障害情報の確認
- 問題の発生を想定した設計・構成
  - ◆ WASクラッシュ時にヒープ・ダンプを出力させる
  - ◆ ログやダンプなど、ファイル出力先のサイズに余裕を持たせる
  - ◆ 計画的なFixpackの適用
- 機能テスト、パフォーマンス・テストの実施
- プロジェクト体制
  - ◆ 障害発生時の役割分担、体制などの明確化

6

問題が発生する前に備えておくべきこととしては、まず障害情報や修正プログラム情報の収集が挙げられます。これらの情報は、WASのサポート・サイトから得ることが出来ます。WASサポート・サイトは、最新リリースの情報や障害に関する情報が随時更新されており、これらを事前に調べておくことが望ましいです。

また、WASの導入にあたっては最新リリースの適用が推奨されます。更に本番システムを構築するにあたり、機能テスト、パフォーマンス・テストを事前に行っておくことや障害発生時の役割分担などプロジェクト体制を明確化することも重要です。

お客様要件や運用方針などにもよりますが、システムがサービス・インしてからも極力最新リリースにすることが望ましく、適用を見送る場合でも、その検討は随時行うことをおすすめします。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## はじめに確認すべきサイト

- Information Center – (製品マニュアル)
  - ◆ <http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp>
- WAS サポート・ホーム
  - ◆ WAS サポートに関する情報を一元的に集約
  - ◆ [http://www.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere\\_Application\\_Server](http://www.ibm.com/support/entry/portal/Overview/Software/WebSphere/WebSphere_Application_Server)
- Fix List for WAS 8.0
  - ◆ FixPackによる機能追加・修正内容の一覧表示
    - LLB, ULB, Caching Proxy も含む
    - ダウンロード・ページへのリンクもある
  - ◆ <http://www.ibm.com/support/docview.wss?uid=swg27022958>
- developerWorks
  - ◆ 日本語の技術情報や連載記事、Workshop資料を入手可能
  - ◆ <http://www.ibm.com/developerworks/jp/websphere/index.html>
- テクニカル・サポート検索
  - ◆ 過去のQ&A情報などが検索できる
  - ◆ <http://www.ibm.com/support/jp/search/index.html>

7

上記は、問題判別の際に、“はじめに確認すべきサイト”をまとめたものです。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## WAS サポート・ホーム

---

- デザイン一新!
- 欲しい情報にすばやくたどり着けるようになった。



メニューを選択して、  
欲しい情報に  
迅速アクセス!

8

WAS サポート・ホームページはデザインを一新し、欲しい情報にすばやくたどり着くことができるようになりました。

ぜひ、お使いのブラウザにブックマークしておくことをお勧めします。

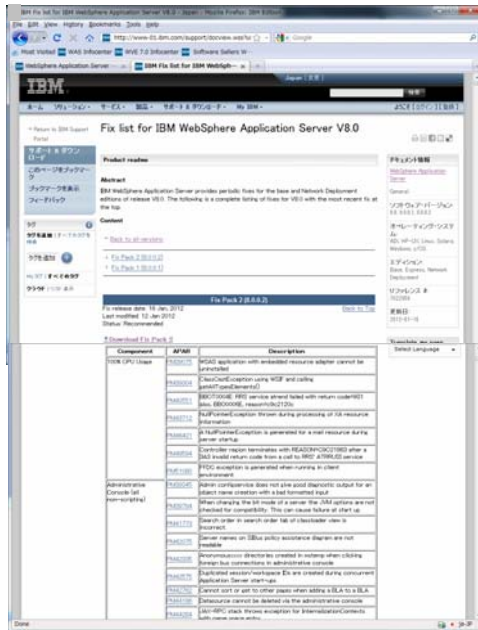


Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## Fix List for WAS 8.0

- Fix Pack に含まれる修正・機能拡張の一覧
- ダウンロード・ページへのリンクも掲載
  - ◆ ダウンロード・ページでは Edge Components の Fix も提供



9

Fix List for WAS 8.0 ページでは、Fix Pack に含まれる修正や機能拡張の一覧を見ることができます。各 Fix Pack のダウンロード・ページへのリンクも掲載されており、ダウンロード・ページでは Edge Components の Fix も入手することが可能です。

Designing Web System Infrastructure with WAS V8.0

## 1.問題判別の心得

---

- 1－1 問題判別の基本
- 1－2 情報収集・問題判別ツール

10

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## Edge Components (ULB) の情報収集 (1/2)

- 現行構成ファイル
  - ◆ dscontrol file save <ファイル名>
    - [AIX/Linux] <ULB\_root>/servers/configurations/dispatcher
    - [Windows] <ULB\_root>%servers%configurations%dispatcher
- ULB のバージョン
  - ◆ ULB が稼働している場合
    - dscontrol executor report
- ◆ 稼働していない場合
  - [AIX/Linux] <ULB\_root>/servers/logs/dispatcher 以下のファイル
  - [Windows] <ULB\_root>%servers%logs%dispatcher 以下のファイル
- Dispatcherと割り振り先サーバーのネットワーク情報
  - ◆ 設定 [Unix] netstat -in [Windows] ipconfig -all
  - ◆ ルーティング [Unix] netstat -rn [Windows] route print
  - ◆ 接続テーブル [Unix] netstat -an [Windows] netstat -an
  - ◆ インターフェース [Unixのみ] ifconfig -a

```

[/]# dscontrol executor report
Executor 報告書:

バージョン・レベル ..... 8.0.0-1 - 20110719-131239
[wsbld342] Linux/x86_64/gcc32
受信された総バケット数 ..... 14,337
転送されたバケット数 ..... 0
生成されたバケット数 ..... 0
廃棄されたバケット数 ..... 0
ローカルに処理されたバケット数 ..... 14,337
エラー・バケット数 ..... 0
接続数/秒 ..... 0
          
```

11

Edge Components で問題判別の際に収集すべき情報には、現行構成ファイル、バージョン、ネットワーク情報、ログ・ファイル、ネットワーク・トレースがあります。現行構成ファイル、バージョン、ネットワーク情報は、上記のコマンドを実行することで取得が可能です。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## Edge Components (ULB) の情報収集 (2/2)

---

- ログ・ファイル
  - ◆ <ULB\_root>/servers/logs/dispatcher 以下のログ・ファイル
    - ハイ・アベイラビリティの問題
      - ログ・レベル5のhamon.log
        - ✓ <ULB\_root>/bin/dscontrol set loglevel 5
      - ログ・レベル5のreach.log
        - ✓ <ULB\_root>/bin/dscontrol manager reach set loglevel 5
    - アドバイザーの問題
      - ログ・レベル5のアドバイザー・ログ
        - ✓ <ULB\_root>/dscontrol advisor loglevel http 80 5
        - ✓ http はプロトコル名、80 はポート番号
- ネットワーク・トレース
  - ◆ [AIX] ユーザー・トラフィックに問題がある場合
    - iptrace -a -s <クライアントIP> -d <クラスター・アドレス> -b <出力ファイル名>
  - ◆ [AIX] Heartbeatに問題がある場合
    - iptrace -abp <HeartBeat使用ポート> <出力ファイル名>
  - ◆ [Linux] tcpdump -w <出力ファイル名>
  - ◆ [Windows] WireShark など

12

ログ・ファイルは必要に応じてレベルを変更することが可能です。

Windows のネットワーク・トレースについては OS に付属するツールがありませんので、別途、チャートで照会するようなツールを入手し、利用して下さい。

- ◆ `<IHS_root>/conf/httpd.conf`

13

## WAS V8.0 によるWebシステム基盤設計ワークショップ

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## WASの情報収集

---

- collector ツール <WAS\_PROFILE\_ROOT>/bin/collector.sh(.bat)
  - ◆ <Hostname>-<Cellname>-<Nodename>-<Profilename>-WASenv.jarファイル  
に出力
    - コレクター・ツールのログ
    - <WAS\_ROOT>と<WAS\_PROFILE\_ROOT>以下のbin、config、logs、profiles、propertiesディレ  
クトリー以下のファイルのコピー
    - OS情報
    - Java 情報
    - WAS情報 (<WAS\_ROOT>以下のディレクトリー構造とファイルが記載)
    - コレクター・ツールが実行したシェル・スクリプト (バッチファイル) の実行情報
    - MQ情報
    - JAR ファイルのマニフェスト
  - ◆ Collectorコマンドはプロファイル単位で実行
  - ◆ WAS 7.0 から非推奨 (後述) 
- バージョン
  - ◆ versionInfoコマンド
- ログファイル、設定ファイル (後述)

14

WASで問題発生時に情報を収集するツールとしてcollectorがあります。

collectorコマンドを実行すると上記の情報を一つのJARファイルにアーカイブします。実際のcollectorコマンドは、<WAS\_ROOT>/binにあり、取得対象とするプロファイル名や、出力ファイル名などが指定できますが、<WAS\_PROFILE\_ROOT>/bin配下にあるcollectorコマンドを実行することでこれらが省略可能です。

collectorコマンドは、WASのインストール・ディレクトリーの外で実行する必要があり、出力先を指定しない場合はカレント・ディレクトリーにJARファイルが出力されることを念頭に、適切なディレクトリーから<WAS\_PROFILE\_ROOT>/bin/collector.sh(.bat)を実行します。

collector ツールは V7.0 から非推奨となっています。このことにつきましては後ほど詳しく説明します。

バージョンの確認は、ログ・ファイルやcollectorのサマリー情報以外に、<WAS\_PROFILE\_ROOT>/bin/versionInfo.sh(.bat) コマンドで確認することが出来ます。

その他、WASで問題判別の際に収集すべき情報にはログや設定ファイルがあります。

次ページ以降でご説明します。

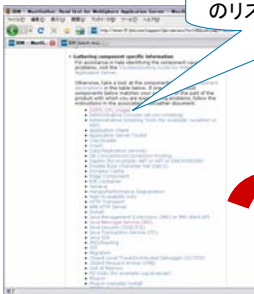
Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

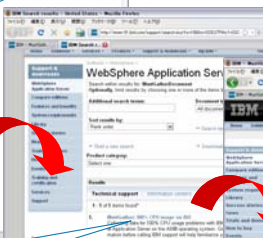
## MustGather

---

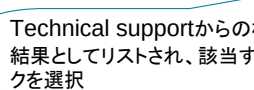
- WASの問題判別のサポート・サイト
  - ◆ 発生した問題に対する情報収集の作業方法などを掲載
- MustGather: Read first for WebSphere Application Server
  - ◆ <http://www.ibm.com/support/docview.wss?rs=180&uid=swg21145599>



問題の種類、発生コンポーネントのリストから選択



問題の種類ごとに取得すべき情報と作業方法を記載



Technical supportからの検索結果としてリストされ、該当するリンクを選択



15

その他、問題が発生した際に参考にするサポート・サイトとして、MustGather:Read first for all WebSphere Application Serverがあります。このページでは、問題の種類や問題が発生したコンポーネントがリストになっており、問題の種類に応じて取得すべき情報収集の作業方法などが示されています。問題判別を行う際には参考にして下さい。

(例) 問題の種類、発生コンポーネント

- 100% CPU Usage
- Classloader
- DB Connections/Connection Pooling
- EJB

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## IBM Support Assistant (ISA)

---

- IBM 製品に関する課題を解決するためのツール
  - ◆ WebSphere 製品だけではない
- 各種のログやトレースを基に、問題解析を手助けしてくれる
- レポート機能が豊富
  - ◆ ログに対して grep や sed や awk や表計算ソフトを駆使していた方は活用を検討してみてもよいかもしれません。
- 現行バージョンは V4.1.2 だが…
  - ◆ Agent のサポート・プラットフォームが古い
    - Windows XP, Windows Server 2003
    - Red Hat Enterprise Linux AS 5.0
    - AIX 6.1, …
- そこで今回は、V5.0 Tech Preview の情報をお伝えします。
  - ◆ ドキュメンテーションもまだほとんどされていないため、検証ベースの情報となります。
  - ◆ 正式リリースの際には、動作が変更されている場合があります。



16

IBM Support Assistant (ISA) とは、WebSphere製品だけにとどまらず、多くのIBM製品に関する課題を解決するためのサポート・ツールです。WAS製品導入CDもしくは、次のリンク先 (<http://www.ibm.com/software/support/isa/>)より導入することができます。

WAS が出力する各種のログやトレースを基に、問題判別を手助けしてくれます。

特徴の一つとして、レポート機能が豊富であることが挙げられます。

これまで運用状況の報告のために、ログに対して grep や sed や awk や 表計算ソフトなどを駆使してレポートを作成されていた方は、これを機に ISA の活用を検討してみるのもよいと考えます。

現行バージョンは V4.1.2 ですが、チャートに示しましたように Agent のサポート・プラットフォームが古く、これからの導入は困難と考えます。

そこで今回は、Tech Preview ではありますが、V5.0 の情報をお伝えしたいと思います。

V5.0 Tech Preview (以下、ISA 5.0) はドキュメンテーションもほとんどされていないため、これからご紹介するのは基本的に検証ベースの情報となります。

従いまして、正式リリースの際には動作が変わっている可能性があることをご了承下さい。



Designing Web System Infrastructure with WAS V8.0

WAS V8.0によるWebシステム基盤設計Workshop

## ISA 5.0 Tech Preview - 構成

■ 構成

- ◆ 「ISA サーバー」と Web ブラウザーで構成される
  - ISA サーバーの実体は WAS Community Edition (WAS CE)
    - WAS CE: Build 2.1.1.5-20110224 / Apache Geronimo 2.1 ベース
- ◆ Web ブラウザーには Java プラグインが必要
  - Java 6 Update 10
- ◆ V4 のアーキテクチャーとは大きく異なる
  - Agent, Agent Manager, Workbench の組み合わせではない
  - 必ずしも WAS を実行するマシンにインストールする必要はない
    - ログやダンプなどの資料はブラウザからアップロードする

ISA V4 アーキテクチャー

ISA V5 アーキテクチャー

17

ISA 5.0 は、構成が 4.0 と大きく異なります。

ISA 4.0 は WAS が稼働する各サーバーに ISA エージェントと、構成内に一つの ISA エージェント・マネージャーを配置し、作業者が使用する ISA ワークベンチはエージェント・マネージャーの管理の下で各エージェントに接続し、作業を行っていました。

ISA 5.0 は、正式な名称ではありませんが「ISA サーバー」とも呼ぶべきコンポーネントに、Web ブラウザーから接続して使用する構成になっています。

ISA サーバーの実体は WAS Community Edition であり、Apache Geronimo をベースとしたアプリケーション・サーバーです。

Web ブラウザーには Java 6 Update 10 (以上のレベル) のプラグインが必要です。

問題判別に用いるログやダンプなどの資料はブラウザからアップロードする仕組みになっているため、ISA 4.0 とは異なり、WAS が稼働する各ノードに ISA サーバーをインストールする必要はありません。

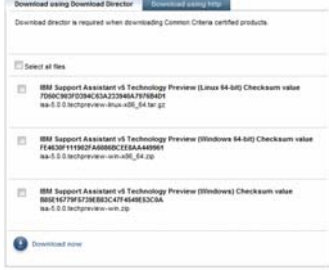
Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## ISA 5.0 Tech Preview - 稼働要件

■ 稼働要件

- ◆ ISA サーバー
  - (まだ?) 明示的に書かれていない
  - ダウンロード・ページは Linux 64bit, Windows 64bit, Windows となっている



- ◆ Web ブラウザー
  - Windows (XP とそれ以降)
    - Internet Explorer 8, Firefox 3.6 (とそれ以降)
    - Internet Explorer 9, Safari と Google Chrome は稼働するはずだがテストされていない
  - Linux
    - Firefox 3.6 とそれ以降

18

ISA サーバーの稼働要件は明示的に書かれてはおりませんが、先のページで示しましたように WAS CE / Apache Geronimo 2.1 ベースですので、こちらの稼働要件に準ずるものと考えられます。

Web ブラウザーにつきましては明示的に記載がありますので、これらのブラウザーに Java 6 Update 10 プラグインを適用して使用して下さい。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## ISA 5.0 Tech Preview - 起動・停止

---

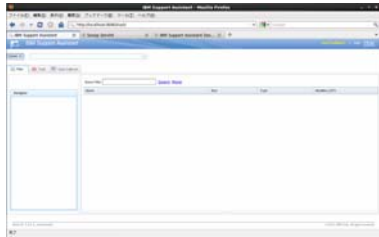
- ISA サーバー
  - ◆ 起動: start\_isa.[sh | bat]
 

```
[root@rhel12 ISA5]# ./start_isa.sh
Using GERONIMO_HOME: /root/ISA/isa-5.0.0.techpreview-linux-x86_64/ISA5
Using GERONIMO_TMPDIR: var/tmp
Using JRE_HOME: /root/ISA/isa-5.0.0.techpreview-linux-x86_64/ISA5/_jvm/jre
Using GERONIMO_OUT: /root/ISA/isa-5.0.0.techpreview-linux-x86_64/ISA5/var/log/server.out

Geronimo started in background. PID: 24671

Once the server has started, IBM Support Assistant can be launched
in a browser at:
http://localhost:8080/isa5 OR http://<hostname>:8080/isa5
```
  - ◆ 停止: stop\_isa.[sh | bat]
- Web ブラウザー
  - ◆ http://<hostname>:8080/isa5 へアクセス

初期画面



19

起動・停止は至ってシンプルで、それぞれ start\_isa.[sh | bat], stop\_isa.[sh | bat] を発行するだけです。

起動後、上のメッセージが出力されてしばらくしてから、http://<hostname>:8080/isa5 へアクセスすると、初期画面が表示されます。

ISA サーバーの起動には時間がかかることがありますので、もし初期画面が正常に表示されない場合は、しばらく待ってから Web ブラウザーでリロードを行ってみてください。

現在の ISA 5.0 Tech Preview はまだ日本語環境に完全に対応しておらず、日本語環境で start\_isa を発行すると問題が発生することがあります。

Linux 環境では start\_isa.sh の前に export LANG=C を発行しておくことで、問題を回避することが可能です。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## ISA 5.0 Tech Preview - ケース管理

■ ケースとは？

- ◆ ある一つの問題に対する、以下の資料などの「入れ物」
  - 各種のログやトレース
  - 分析結果
- ◆ 単にファイルの入れ物なので、ファイルであれば何でも格納できる
- ◆ ケースに対してファイルをアップロードし、ファイルを格納する
- ◆ 問題判別ツールは、ケース内のファイルを対象として起動する

20

ケースとは、ある一つの問題に対する資料の入れ物です。

主に、各種のログやトレースと、それらを ISA で利用できるツールを用いて分析した結果を格納することになります。

ケース自体は単にファイルの入れ物ですので、ファイルであれば何でも格納することが可能です。

ケースにアップロードされたファイルを右クリックすると、そのファイルに対して適用可能な作業がコンテキスト・メニューに現れます。

コンテキスト・メニューから Problem Analysis を選択すると、そのファイルに対して実行可能な問題判別ツールを選択することが可能です。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

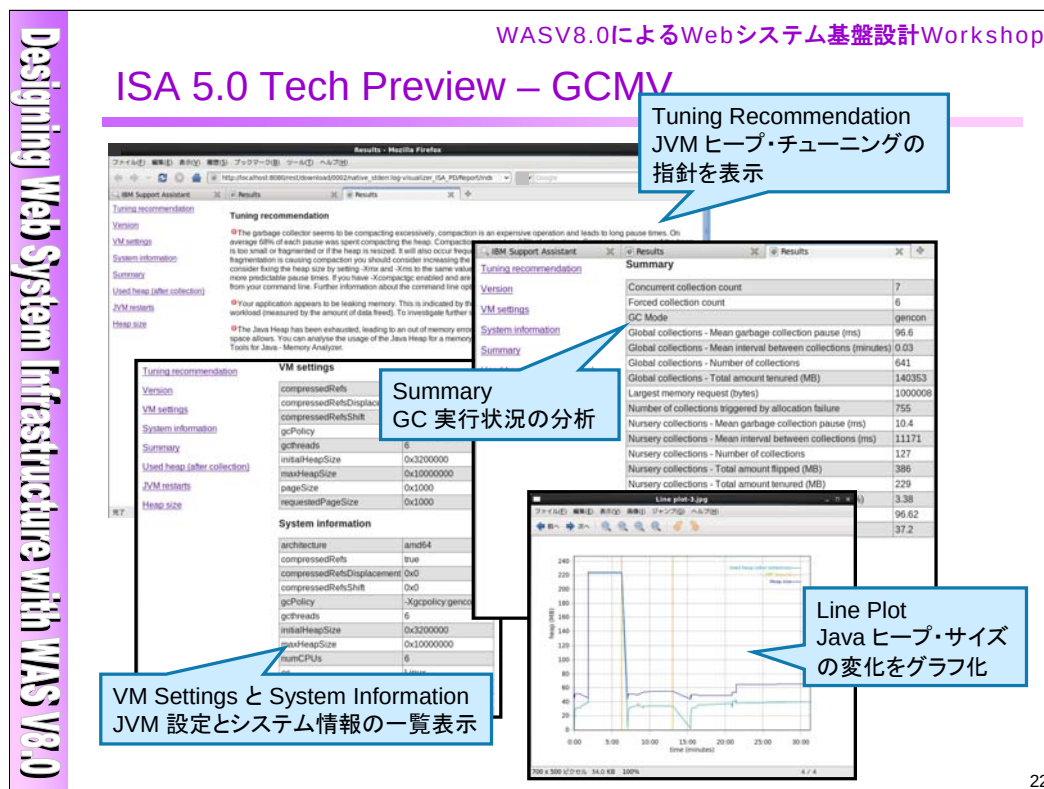
## ISA 5.0 Tech Preview - 利用可能な問題判別ツール

| ツール                                                                         | 説明                                                                    |
|-----------------------------------------------------------------------------|-----------------------------------------------------------------------|
| <b>Garbage Collection and Memory Visualizer (GCMV)</b>                      | 冗長ガーベージ・コレクション (verboseGC) の結果をビジュアルに表示します。                           |
| <b>Memory Analyzer</b>                                                      | OutOfMemoryError をはじめとするメモリーに関するさまざまな問題の原因を探すため、JVM ダンプを分析します。        |
| <b>Pattern Modeling and Analysis Tool for Java Garbage Collector (PMAT)</b> | verboseGC ログの分析と、Java ヒープの使用状況の分析を行います。Java ヒープのチューニングのためのレポートを生成します。 |
| <b>Profile Port Checker</b>                                                 | WAS 構成におけるポートの衝突をチェックします。                                             |
| <b>Thread and Monitor Dump Analyzer (TMDA)</b>                              | 同一 Java プロセスの異なる時刻のスレッド・ダンプを比較することで、JVM ハングの疑いを検出します。                 |
| <b>WebSphere Application Server Configuration Visualizer</b>                | WAS の構成をビジュアル化し、HTML 形式のレポートを生成します。                                   |
| <b>IBM Portal Log Analyzer</b>                                              | ConfigTrace.log を分析し、WebSphere Portal 環境の問題を分析します。(今回はご説明しません。)       |

21

ISA 5.0 で利用可能な問題判別ツールの一覧です。

次ページ以降で、各ツールの使い方と、何ができるのかをご説明します。



22

GCMV の入力 は GC ログを含む native\_stderr.log です。  
 チャートに示した情報を始め、GC 実行状況の分析をレポート化します。  
 また、分析結果に基づくチューニングの指針を示してくれます。

WASV8.0によるWebシステム基盤設計Workshop

## ISA 5.0 Tech Preview – PMAT

**Designing Web System Infrastructure with WAS V8.0**

**GC 状況 全体の分析**

**Java ヒープ異常発生と 推奨される設定**

**Java ヒープ 状況推移**

23

こちらは PMAT のレポートになります。

入力は GCMV と同様、GC ログを含む native\_stderr.log です。

機能的には前のページでご紹介しました GCMV と似ていますが、状況推移グラフの種類が非常に豊富です。

GCMV のグラフは Java ヒープ全体の様子のみでしたが、PMAT では GenCon ポリシーにおける Nursery / Tenured 領域それぞれについて GC 実施前後の使用量をグラフ化することができ、より詳細な分析が可能になっています。

生成可能なグラフは以下の 20種類です。

Used Tenured Java Heap Size After Garbage Collection  
 Free Temured Java Heap Size After Garbage Collection  
 Total Tenured Java Heap Size After Garbage Collection  
 Used Nursery Java Heap Size After Garbage Collection  
 Free Nursery Java Heap Size After Garbage Collection  
 Total Nursery Java Heap Size After Garbage Collection  
 Used Tenured Java Heap Size Before Garbage Collection  
 Free Tenured Java Heap Size Before Garbage Collection  
 Total Tenured Java Heap Size Before Garbage Collection  
 Used Nursery Java Heap Size Before Garbage Collection  
 Free Nursery Java Heap Size Before Garbage Collection  
 Total Nursery Java Heap Size Before Garbage Collection  
 Requested Java Heap Size After Garbage Collection  
 Interval between Garbage Collections  
 Allocation Failure Duration  
 Mark Phase Duration  
 Sweep Phase Duration  
 Compact Phase Duration  
 Garbage Collection Duration  
 Garbage Collection Overhead (%)

WAS V8.0によるWebシステム基盤設計Workshop

## ISA 5.0 Tech Preview – Memory Analyzer

Designing Web System Infrastructure with WAS V8.0

The screenshot displays the 'Leak Suspects' window in Mozilla Firefox. It shows a pie chart representing memory usage with two segments: (a) 194.6 MB and (b) 30.1 MB. The total is 224.6 MB. Below the chart, 'Problem Suspect 1' is listed. To the right, a detailed view of 'Problem Suspect 1' is shown, including a description of a memory leak in 'pd.pro.MyVector' and a table of 'Shortest Paths To the Accumulation Point'.

| Class Name                                             | Shallow Heap | Retained Heap |
|--------------------------------------------------------|--------------|---------------|
| java.lang.Object[220] @ 0x10000000                     | 1,296        | 204,004,560   |
| com.ibm.ws.classloader.CompoundClassLoader @ 0x1862208 | 24           | 204,004,584   |
| java.util.Hashtable\$Entry @ 0x10000000                | 24           | 24            |
| java.util.Hashtable\$Entry @ 0x10000000                | 64           | 88            |
| java.util.Hashtable\$Entry @ 0x10000000                | 40           | 128           |
| java.util.Hashtable\$HashEnumerator @ 0x10000000       | 56           | 72            |
| java.util.Hashtable\$Entry @ 0x10000000                | 16           | 16            |
| Total: 2 entries                                       |              |               |
| java.util.Hashtable\$Entry @ 0x10000000                | 24           | 24            |
| Total: 2 entries                                       |              |               |

24

Memory Analyzer のレポートです。

Heap Dump を入力として、Java ヒープ内のインスタンスの合計サイズの大きい、メモリー・リークの疑いのあるクラスを検出してくれます。



Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## 【参考】管理コンソールでの javacore, ヒープ・ダンプ生成

- WAS 8.0 では管理コンソール上から javacore, ヒープ・ダンプの生成が可能になった。 New v8



25

ISA そのものから少し話題がそれますが、WAS 8.0 では管理コンソール上のボタンを押すだけで、Java コアやヒープ・ダンプの生成が可能になりました。

WASV8.0によるWebシステム基盤設計Workshop

## ISA 5.0 Tech Preview – Profile Port Checker

Designing Web System Infrastructure with WAS V8.0

26

Profile Port Checker のレポートです。

Collector ツールの生成する JAR ファイルを入力とし、プロファイル内・プロファイル間で競合する可能性のあるポート設定を検出してくれます。

ただし、物理ノードが異なるために実際には競合しないケースも報告されることがありますので、最終的には人手での判断が必要なケースが存在します。

WASV8.0によるWebシステム基盤設計Workshop

## ISA 5.0 Tech Preview – TMDA

Designing Web System Infrastructure with WAS V8.0

レポート目次

各スレッドの状況

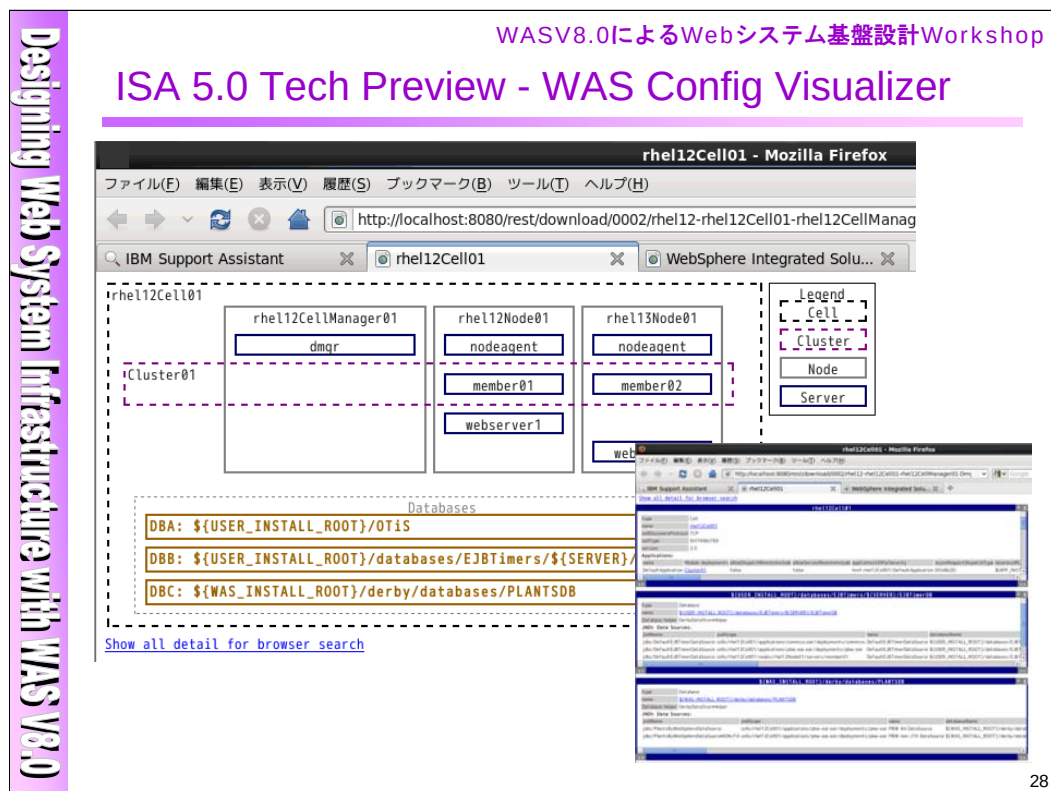
ハングの疑いのあるスレッドと実行中のクラス、メソッド

27

TMDA のレポートです。

同一 Java プロセスの複数の時刻の Java コアを与えることで、ハングしている可能性のあるスレッドを検出してくれます。

単一の Java コアを与えた場合でも、各スレッドの状況を報告してくれます。



WAS Configuration Visualizer のレポートです。

入力は Collector ツールの生成する JAR ファイルで、WAS と Web サーバー、データベースの構成図を作成してくれます。

各コンポーネントの詳細な情報のレポート表示も可能です。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## AutoPD = IBM Support Assistant Lite ?

---

- WAS 7.0 から collector ツールは非推奨となった。
  - ◆ WAS 8.0 ND Infocenter: 非推奨のフィーチャー
    - [http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/rmig\\_depfeat.html](http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/rmig_depfeat.html)
  - ◆ ただし、IBMサービス・ラインから collector の実行結果を送付するよう指示があった場合などは、それに従って下さい。
- 推奨されるマイグレーション・アクションは AutoPD を使用することとなっている。
- AutoPD = IBM Support Assistant Lite?
  - ◆ Infocenter 上で AutoPD に関する詳細情報のリンクをたどると、ISA Lite の説明へ飛ぶ。
- ここでは、ISA Lite についてご説明します。

29

WAS 8.0 Infocenter には、Collector ツールが非推奨になった旨の記載があります。

(ただし、IBMサービスラインから Collector の実行結果を送付するよう指示があった場合などは、それに従って下さい。)

同じく Infocenter には、推奨されるマイグレーション・アクションとして AutoPD への移行が記載されています。

詳細情報へのリンクが示されていますが、このリンクをたどりますと IBM Support Assistant Lite の記事へ飛ぶようになっています。

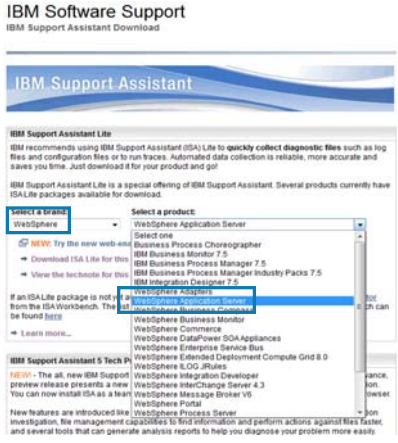
そこで今回は、ISA Lite についてご説明します。

Designing Web System Infrastructure with WAS V8.0

WAS V8.0によるWebシステム基盤設計Workshop

## ISA Lite の入手

- ISA Lite は製品ごとに存在する。
  - ◆ WAS を選んでダウンロード下さい。
    - <http://www.ibm.com/software/support/isa/download.html>
  - ◆ まず、WebSphere ブランドを選択。
  - ◆ その後、WAS を選択する。



30

ISA Lite は製品ごとに存在します。

上記の URL へアクセスし、初めにソフトウェア・ブランドから WebSphere を選択します。

その後、WebSphere Application Server を選択し、ダウンロードします。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## ISA Lite の利用方法

■ 問題を選択し、ガイドに従って操作して情報収集を行う。




31

起動直後の画面では、情報収集の対象となる問題のカテゴリーが表示されます。

問題カテゴリーを展開すると個々の問題が現れますので、問題を選択してガイドに従い、情報収集を完了します。

結果は単一の zip ファイルとしてアーカイブされます。

Designing Web System Infrastructure with WAS V8.0

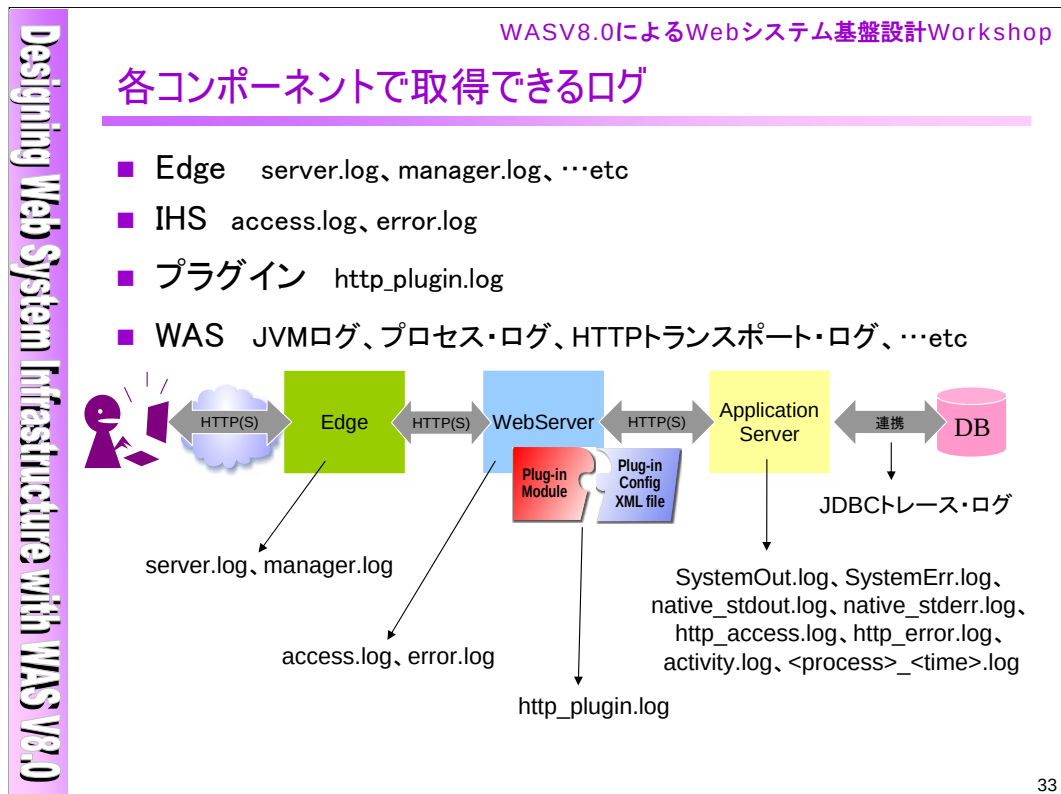
## 2.問題判別

---

2-1 ログ、トレース  
2-2 症状別問題判別

32





ログは問題判別の基本となるものであり、出力する製品や階層に応じてさまざまなログが出力されます。この章では、EdgeからWASに至るまでのログについて紹介します。

ブラウザから実行されるリクエストはまずEdgeからIHSに割り振られ、そのリクエストがWASに処理を委ねるものとプラグインが判断した場合、プラグインはそのリクエストをWASに渡します。WAS内で処理されたリクエストはプラグイン、IHSを通過してブラウザに戻されます。実際に記録されるタイミングは、各コンポーネントによって違うため、問題判別を行う際には注意が必要です。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## Edgeのログ

---

- Load Balancerのログは循環ログ
  - ◆ サイズがオーバーしたら古いログから上書き
  - ◆ ログ・サイズは変更可能
  
- ログの種類
  - ◆ server.log
    - dsserverの活動の記録、発行したコマンド等の情報
  - ◆ manager.log
    - managerの活動記録等
  - ◆ 使用advisor名\_ポート.log
    - 使用するadvisorに関する情報
  - ◆ hamon.log
    - high availabilityに関する情報 (Primary, Backupの両方)
  - ◆ reach.log
    - reach advisorがとばしたping情報等

34

上記のログは、コマンドdscontrol loglevel <0～5> にて書き込む情報量を変更することが可能です。

ログレベルを0～5まで変更することが出来ます。ログ・レベルが3以上の場合は、最大サイズを設定することが推奨されます。指定のサイズに達すると、ファイルの先頭から以前のログ項目を上書きします。ログにはタイム・スタンプが記録されるため、書き込まれた順番は判断可能です。

(※) Load Balancerのログは問題判別用ですので、常時ログ・レベルを5にして運用する事は一般的ではありません。ログ・レベルを高くするとパフォーマンスに影響します。

Load Balancerを再起動したときに、前回使用していたログ・ファイルに .bak 拡張子を付加して保存し、新しいログ・ファイルが作成されます。ログを保存したい場合は、再起動する前にログ・ファイルを保存する事をおすすめします。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## IHSのログ

---

- access\_log
  - ◆ IHSに届いたリクエストの処理記録
  - ◆ LogFormatディレクティブで、記録項目はカスタマイズ可能(→後述)
  - ◆ 出力先はディレクティブで指定(管理コンソールからも可能)
  
- error\_log
  - ◆ IHSのリクエスト、もしくは処理中に発生したエラーやwarningを記録
  - ◆ LogLevelディレクティブで、ログ・レベルは変更可能(→後述)
  - ◆ 出力先はディレクティブで指定(管理コンソールからも可能)

35

IHSで取得されるログは、access.logとerror.logの2つになります。

access.logは、クライアントからIHSに届いたリクエストに対するリプライの記録であり、アクセス数の集計によく用いられるログです。access.logに記述される内容は、LogFormatディレクティブでカスタマイズが可能です。

error.logは、リクエストもしくはIHSがリクエストを処理する中で発生したエラーやWarningを記録するログです。出力される内容はLogLevelディレクティブで変更することが可能です。

これら2つのログはデフォルトでは<IHS\_ROOT>/logsに出力されますが、CustomLog や ErrorLog ディレクティブで出力先やログ名を変更することも可能です。またV6からはノード・エージェントの管理下であれば、管理コンソールから出力先を変更することも可能です。

問題判別をされる際には、access\_log でのリクエストの HTTP ステータスやレスポンス・タイム、error\_log にエラーメッセージが出力されていないか等を確認して下さい。

WASV8.0によるWebシステム基盤設計Workshop

## IHSアクセス・ログのカスタマイズ

■ LogFormatディレクティブ

- ◆ 指示子でもってログに記述する内容を定義
- ◆ 各々のフォーマットに識別のための名前をつけ、そのフォーマット名でログに結びつける

httpd.conf

```
LogFormat "%h %l %u %t %r" "%s %b" "%{Referer}i" "%{User-Agent}i" combined
LogFormat "%h %l %u %t %r" "%s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
CustomLog logs/access.log common
```

フォーマット名commonを  
ログのフォーマットに指定。

ログの出力ディレクトリー、ファイル名を指定。

◆ IHSのレスポンス・タイムはデフォルトでは記載がないので注意

- %Dもしくは%TをLogFormatに追加

httpd.conf

```
LogFormat "%h %l %u %t %r" "%s %b (%D)" common
```

手動でレスポンス・タイムをログに記録させる設定が必要。

36

アクセスログのカスタマイズは、LogFormatディレクティブで行います。デフォルトでは以上4パターンのLogFormatが指定されており、それぞれに名前が付けられています。LogFormatディレクティブの最後に指定されている内容(上記のcombined、common、referer、agent)がそれぞれのフォーマット名になります。アクセスログの出力先はCustomLogで指定しますが、その際、最後にフォーマット名を記述します。そうすることで、指定されたフォーマットでアクセスログに出力されます。

LogFormatでは、ログに記述する内容を指示子と呼ばれる特殊記号で表現します。デフォルトで指定されているLogFormatではIHSのレスポンス・タイムは記録されません。IHSのレスポンス・タイムを記録するには、LogFormatディレクティブに%Dもしくは%Tを追加する必要があります。%Dはマイクロ秒単位、%Tは秒単位でIHSがリクエストを受け取ってレスポンスを返すまでにかかる時間をアクセス・ログに記録します。また、リクエストがIHSに入って来た時ではなく、IHSから出て行った時にaccess\_logにログが書き込まれます。従いまして、access\_logでは、リクエスト順にログが書き込まれるわけではないことにご注意下さい。

問題判別の観点からは、このレスポンス・タイム出力を設定される事をお勧め致します。

・IHSのLogFormatディレクティブ

[http://httpd.apache.org/docs/2.2/mod/mod\\_log\\_config.html#formats](http://httpd.apache.org/docs/2.2/mod/mod_log_config.html#formats)

WAS V8.0によるWebシステム基盤設計Workshop

## IHSエラー・ログのログ・レベル

- LogLevelディレクティブ
  - ◆ エラー・ログに記録される内容のレベルを指定
  - ◆ デフォルト : warn
  - ◆ debug, info, notice, warn, error, crit, alert, emergの8種類

httpd.conf

LogLevel **debug**

error.log

LogLevelをWarnからdebugに変更してIHSを起動したときの例  
デフォルトでは記録されないinfoやdebug情報が出力される

```
[Sat Apr 11 11:47:46 2009] [info] mod_unique_id: using ip addr 9.188.198.192
[Sat Apr 11 11:47:47 2009] [info] mod_unique_id: using ip addr 9.188.198.192
[Sat Apr 11 11:47:48 2009] [notice] PLUGIN: mod_was_ap22_http: apache_log_header: WebSphere Plugins loaded.
[Sat Apr 11 11:47:48 2009] [notice] PLUGIN: mod_was_ap22_http: apache_log_header: -----Plugin Information-----
[Sat Apr 11 11:47:48 2009] [notice] PLUGIN: mod_was_ap22_http: apache_log_header: Bld version: 7.0.0
[Sat Apr 11 11:47:48 2009] [notice] PLUGIN: mod_was_ap22_http: apache_log_header: Bld date: Nov 11 2008, 22:32:03
[Sat Apr 11 11:47:48 2009] [notice] PLUGIN: mod_was_ap22_http: apache_log_header: Webserver: IBM_HTTP_Server
[Sat Apr 11 11:47:48 2009] [notice] PLUGIN: mod_was_ap22_http: apache_log_header: -----
[Sat Apr 11 11:47:48 2009] [notice] Using config file /opt/IBM/HTTPServer/conf/httpd.conf
[Sat Apr 11 11:47:48 2009] [debug] mod_mpmstats.c(189): mpmstats daemon started (pid 25709)
[Sat Apr 11 11:47:48 2009] [notice] IBM_HTTP_Server/7.0.0.0 (Unix) configured -- resuming normal operations
[Sat Apr 11 11:47:48 2009] [info] Server built: Aug 6 2008 17:10:22
[Sat Apr 11 11:47:48 2009] [debug] worker.c(1837): AcceptMutex: sysvsem (default: sysvsem)
[Sat Apr 11 11:47:48 2009] [notice] Core file limit is 0; core dumps will not be written for server crashes
```

IHSのエラー・ログに記録される内容は、LogLevelディレクティブで指定することが出来ます。ログ・レベルにはdebug、info、notice、warn、error、crit、alert、emergの8種類がありdebugからemergになるにつれ重要度が高くなります。ログ・レベルはデフォルトではwarnに設定されているため、warnより上の重要度の情報が記載されます。さらにdebugやinfoといった詳細な情報をログに記録させたい場合には、LogLevelディレクティブの指定をdebugに変更すると、全ての情報がエラーログに記載されるようになります。問題判別を行う際には、このログ・レベルを上げて下さい。

WASV8.0によるWebシステム基盤設計Workshop

## プラグインのログ

- http\_plugin.log
  - ◆ プラグインの処理内容を記録
  - ◆ WebサーバーとWAS間の通信に問題があるときに参照
  - ◆ plugin-cfg.xmlファイル内でログに関する設定を行う(→後述)
    - 設定は管理コンソールからも実施可能

plugin-cfg.xml

```
<Config ASDisableNagle="false" AcceptAllContent="false"
AppServerPortPreference="WebserverPort" ChunkedResponse="false" FIPSEnable="false"
IISDisableNagle="false" IISPluginPriority="High" IgnoreDNSFailures="false" RefreshInterval="60"
ResponseChunkSize="64" VHostMatchingCompat="false">
  <Log LogLevel="Error" Name="/opt/IBM/HTTPServer/Plugins/logs/webserver1/http_plugin.log"/>
</Config>
```

ログレベル

ログの出力先

プラグイン・ログ設定:

\* ログ・ファイル名  
/opt/IBM/HTTPServer/Plugins/logs/w

ログ・レベル  
エラー

トレース  
状態  
警告  
エラー  
デバッグ  
詳細

適用 リセット 取り消し

管理コンソール  
「サーバー」→「Webサーバー」→  
<WebServer name>→「プラグイン・プロパティ」

プラグインが出力するログがhttp\_plugin.logになります。http\_plugin.logはプラグインの処理内容が記録されており、WebサーバーとWASとの通信に問題があると思われるときに参照して問題解析を行ってください。

プラグイン・ログに関するログレベルや出力先の設定はplugin-cfg.xmlで記載されており、plugin-cfg.xml内のLogLevel、Nameといった設定箇所を編集することで変更することが可能です。またV6からは管理コンソールからの「プラグイン・プロパティ」からも設定が可能です。管理コンソールから設定を行うためには、Webサーバーがノード・エージェントの管理下である必要があります。またplugin-cfg.xmlの位置は、IHSの設定ファイルであるhttpd.confで指定します。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## プラグイン・ログのログ・レベル

---

- Trace (→後述)
  - ◆ リクエストを受けた際にプラグインが行った全過程の詳細が記録
- Stats
  - ◆ リクエストに対し選択されたサーバー、ロード・バランシング情報を記録
- Warn
  - ◆ プラグインの処理中に発生したエラーとWarningを記録
- Error (デフォルト)
  - ◆ プラグインの処理中に発生したエラーを記録
- Debug
  - ◆ リクエストの処理中に実行された重大なステップが記録
- Detail
  - ◆ リクエストおよびリプライに関するすべての情報を記録

http\_plugin.log

```
[Sat Apr 11 12:23:49 2009] 00006903 f461ab90 - STATS: ws_server_group: serverGroupCheckServerStatus:
Checking status of WSClientIse135Custom_server1, ignoreWeights 0, markedDown 0, retryNow 0, retryInSec --,
wlbAllows 1 reachedMaxConnectionsLimit 0
[Sat Apr 11 12:23:49 2009] 00006903 f461ab90 - STATS: ws_server: serverSetFailoverStatus: Server
WSClientIse135Custom_servers: pendingRequests 0 failedRequests 0 affinityRequests 0 totalRequests 1.
```

39

プラグイン・ログの出力レベルは、デフォルトではErrorになっています。つまり、エラーが発生しない限り、プラグインがリクエストを正しく割り振ってもその記録は残りません。より詳細な記録をプラグイン・ログに記載させるためにはログレベルを変更する必要があります。

ログ・レベルにはError以外にも、Debug、Detail、Warn、Stats、Traceがあり、Traceがもっとも詳細に処理内容が記載されます。Statsではプラグインの割り振り先やロードバランシング情報が記載されます。IHSとWASが1対N構成時に、IHSのプラグインが割り振ったサーバーを確認したい場合には一般的にはStatsを設定します。

WASV8.0によるWebシステム基盤設計Workshop

## プラグイン・ログのトレース

### ■ ログ・レベルを「Trace」に設定

◆ リクエストを受けた際にプラグインが行った全過程を記録

http\_plugin.log

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - DETAIL: ws\_common: websphereShouldHandleRequest: trying to match a route for: 'n.com'; uri=/XATestWeb/index3.html'

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - DEBUG: ws\_common: websphereVhostMatch: Found a match '\*:80' to 'ise135:80' in pre 1, exact match 0

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - DEBUG: ws\_common: websphereUriMatch: Found a match '/XATestWeb/\*' to '/XATestWeb/index3.html' in UriGroup: default\_host\_WSClntCluster\_URIIs with score 11, exact match 11

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - TRACE: mod\_was\_ap22\_http: as\_translate\_name: WebSphere will handle: /XATestWeb/index3.html

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - TRACE: lib\_htrequest: htrequestSetMethod: Setting the method [GET]

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - TRACE: lib\_htrequest: htrequestSetURL: Setting the url [/XATestWeb/index3.html]

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - TRACE: ws\_server\_group: serverGroupCheckServerStatus: Checking status of markedDown 1, retryNow 1, retryInSec -121, wlbAllows 0 reachedMaxConnectionsLimit 0

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - TRACE: ws\_server\_group: lockedServerGroupUseServer: Server WSClntTitanCustom\_server2 picked, Weight 1

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - TRACE: ws\_server\_group: serverGroupIncrementConnectionCount: Server WSClntTitanCustom\_server2 picked, pendingConnectionCount 1 totalConnectionsCount 1.

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - DETAIL: lib\_htrequest: htrequestWrite: Writing the request reqInfo is OK: GET /XATestWeb/index3.html HTTP/1.1

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - DETAIL: ws\_common: websphereExecute: Wrote the request; reading the response

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - DETAIL: lib\_htresponse: htresponseRead: Reading the response: 97bf674

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - DETAIL: HTTP/1.1 200 OK

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - DETAIL: Date: Sat, 11 Apr 2009 03:38:39 GMT

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - TRACE: mod\_was\_ap22\_http: cb\_write\_headers: In the write headers callback

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - TRACE: mod\_was\_ap22\_http: cb\_write\_headers: Adding header Date value is Sat, 11 Apr

[Sat Apr 11 12:38:39 2009] 00006923 f641db90 - DETAIL: ws\_common: websphereEndRequest: Ending the request

40

プラグイン・ログのレベルをTraceに設定すると、プラグインが処理した詳細な内容がログに記載されますので、問題判別時には非常に有用です。上記のログは、プラグイン・ログのレベルをTraceにしたときのログを抜粋して見やすくした例です。リクエストがプラグインに来てから、仮想ホストやURIを比較しWASに処理を渡すことを判断したことを示す記述や、リクエスト・ヘッダーの生成、リクエストを処理するサーバーの選択、リクエストの送信とレスポンスの受信、さらにクライアントに返すレスポンスのヘッダーを作成してクライアントに処理を返すまでの内容を確認することが出来ます。



Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## WASのログ

---

- JVMログ (SystemOut.log / SystemErr.log)
  - ◆ JVMの標準出力、標準エラー出力
  - ◆ アプリケーションの挙動やプロセスの稼働状況
- プロセス・ログ (native\_stdout.log / native\_stderr.log)
  - ◆ ネイティブ・コードやJVM自身の出力が書き込む情報
- FFDCログ (<process\_name>\_<timestamp>.txt)
  - ◆ エラー情報を発生時に自動で記録される
  - ◆ サポート・センター、開発部門が障害対応時に見るログ
- HTTPトランスポート・ログ (http\_access.log / http\_error.log)
  - ◆ WebコンテナのHTTPトランスポートに対するアクセスを記録するログ
  - ◆ IHSを介さないアクセスに対するアクセス・ログ、エラー・ログ
  - ◆ デフォルトでは出力されない
- IBM保守ログ (activity.log)
  - ◆ JVMの詳細な情報を記録したログ
  - ◆ showlogコマンドを使用
  - ◆ デフォルトでは出力されなくなった

41

WASで取得できるログは以上になりますので、ご確認下さい。

・JVMログ

SystemOut.log、SystemErr.logという名前で出力され、アプリケーションの挙動やプロセスの稼働状況を記載されます。

・プロセス・ログ

native\_stdout.log、native\_stderr.logという名前で出力され、ネイティブ・コードやJVM自身の出力が記載されます。

・FFDC(First Failure Data Capture)ログ

全てのシステムエラーを発生時点で自動的に記録されるもので、サポート・センターや開発部門が障害対応時に確認するログです。

・HTTPトランスポート・ログ

WebコンテナのHTTPトランスポートに対するアクセスを記録するログで、IHSを介さないアクセスに対するアクセス・ログとエラー・ログです。

・IBM保守ログ

JVMログにさらに詳細な情報を記録したログで、バイナリー・ファイルのためエディターで開くことはできません。確認するためにはshowlog.bat(sh)というコマンド実行が必要になります。

V8 からデフォルトでは出力されなくなりました。出力させるには管理コンソール上で「トラブルシューティング」>「ログおよびトレース」>「*server\_name*」>「IBM® 保守ログ」と進み、「使用可能にする」チェック・ボックスを選択します。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## JVMログ

■ SystemOut.log / SystemErr.log

- ◆ 問題判別を行う上で、最も重要なログ
- ◆ アプリケーションの挙動やプロセスの稼動状況が記載されているログ
- ◆ アプリケーション・サーバー、ノード・エージェント、デプロイメント・マネージャーなど各プロセス単位で存在

サーバー→サーバー・タイプWebSphere Application Server  
 →<サーバー名>「ロギングおよびトレース」→「JVMログ」

出力先ファイル名

ファイルのフォーマット(基本or拡張)

＜ログ・ファイルの循環＞  
 ファイルサイズ(指定サイズが来たとき)や  
 時間(指定時間に達したとき)で新規ファイル  
 を作成可能

アプリケーション内のprint文をログに書き込むか、  
 また、その文を指定したフォーマットに従って  
 フォーマットするか(デフォルト両方ON)

42

JVMログは、アプリケーションの挙動やプロセスの稼動状況が記載されているログで、問題判別を行う上でもっとも重要なログの一つです。JVM ログは各アプリケーション・サーバー、ノード・エージェント、デプロイメント・マネージャーなど、各プロセス単位で <PROFILE\_ROOT>/logs/<process\_name> ディレクトリ以下に出力されます。JVM ログに関する設定は管理コンソールから行います。

WAS V8.0によるWebシステム基盤設計Workshop

## JVMログに出力されるメッセージ

Designing Web System Infrastructure with WAS V8.0

- メッセージID
  - ◆ メッセージを発行したコンポーネントやその重要度を示す
  - ◆ エラー・メッセージが出た際、メッセージIDでInformation Centerを検索

SystemOut.log

```
[12/01/06 19:24:03:487 JST] 00000000 WsServerImpl A WSVR0001I: e-business に
対してサーバー member01 はオープンな状態です
```

メッセージID  
最初の4文字(もしくは5文字)がコンポーネントを示す  
続く数字がそのコンポーネントでメッセージを識別するための番号  
最後の1文字が重要度(I=情報、W=警告、E=エラー)を示す

Infocenterから「参照」→「メッセージ」より、  
メッセージIDとその内容と対応を見ることができる



43

JVM ログに出力されるメッセージにはメッセージ ID と呼ばれる ID が付けられており、メッセージが分類できるようになっています。メッセージ ID は最初の 4 文字(もしくは5文字)がコンポーネントを示しており、続く数字がそれらのメッセージを識別するための番号、最後の一文字が重要度を示しています。重要度は、Iが情報、Wが警告、Eがエラーを示しており、一番重要度が高いのはエラーです。

WAS V8.0 の Information Center ではメッセージIDごとにその内容と対応を確認することが出来ます。出力されたメッセージに関して情報が知りたい場合は、InformationCenterでメッセージ ID をもとに検索して下さい。

・WAS V8.0 Information Center

<http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp>

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## プロセス・ログ

■ native\_stdout.log / native\_stderr.log

- ◆ ネイティブ・コードやJVM自身の出力が書き込まれるログ
- ◆ SystemOut.logへの出力は、プロセス・ログへは出力されない(WAS V5より)
- ◆ JVMのverbose:gcを設定した場合、その出力はnative\_stderr.logに記録。
- ◆ 循環はしないが、JVMログに比べファイルの増加率は小さい。

「ログGINGおよびトレース」→「プロセスログ」

「プロセス定義」→「Java仮想マシン」

アプリケーションにサービス > member01 > member01 > プロセスログ

このページを使用して、標準出力および標準エラーのストリームが書き込まれるファイル名を指定する設定を変更します。プロセスの標準出力および標準エラー・ストリームを個別ログファイルに出力すると、プロセスログはネイティブ・コードが書き込まれます。このログには、ネイティブ・コードの標準出力、標準エラー、および JVM によって書き込まれる標準出力も含まれます。それぞれアプリケーション・サーバーとものすべてのアプリケーション・プロセスに 1 組のプロセスログが作成されます。プロセスログは、プロセス・ログ・マネージャーによって自動的に作成されます。構成ファイルでの変更は、サーバーの再起動時に適用されます。ランタイムでの変更は、即時に適用されます。

構成 ランタイム

一般プロパティ

STDOUT ファイル名

{SERVER\_LOG\_ROOT}/native\_stdout.log

STDERR ファイル名

{SERVER\_LOG\_ROOT}/native\_stderr.log

適用 OKのリセット キャンセル

出力先ファイル名

アプリケーションにサービス > server1 > プロセス定義 > Java 仮想マシン

Java 仮想マシンの詳細設定。

構成

一般プロパティ

クラスパス

ブートクラスパス

追加プロパティ

カスタム・プロパティ

☐ 冗長クラスロード

☒ 冗長ガーベージコレクション

☐ 冗長 JNI

冗長ガーベージコレクションをONにすると、GCの詳細な情報がnative\_stderr.logに出力される

44

プロセス・ログはネイティブ・コードや JVM 自身の出力が書き込まれるログであり、WAS V5からは SystemOut.log の出力はプロセス・ログには出力されなくなりました。JVM ログと違い、プロセス・ログには循環させるための仕組みは組み込まれていませんが、JVM ログに比べファイルの増加率は非常に小さいです。また、JVM の verbose:gc の設定をした場合、GCの結果の出力はデフォルトでは native\_stderr.log に記載されます。ガーベージ・コレクションの状況を確認されたい場合には、当ログを確認して下さい。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## FFDCログ(<process\_name>\_<timestamp>.txt)

---

- **First Failure Data Capture**
  - ◆ 全てのシステム・エラー情報を発生時点で自動的に記録するツール
- サポート・センターや開発部門が障害時に見て分析
- <WAS\_PROFILE\_ROOT>/logs/ffdcディレクトリーに出力
  - ◆ <process\_name>\_<timestamp>.txt
- <process name>\_exception.log
  - ◆ サーバー起動中に発生した例外情報の統計を管理する
  - ◆ ffdcディレクトリーに作成
- デフォルトでは 7 日後に自動的に消去される。
  - ◆ <WAS\_root>/properties 内の以下のファイルで日数を設定可
    - ffdcRun.properties, ffdcStart.properties, ffdcStop.properties
    - ExceptionFileMaximumAge=7 を所要の日数に変更
    - 3つのファイルで同じ値に設定

45

FFDCとはFirst Failure Data Captureの略で、全てのシステム・エラー情報を発生時点で自動的に記録するツールになります。このツールによって作成されたFFDCログは、サポート・センターや開発部門が障害時に確認して分析を行うためのものです。ログはプロセス単位のディレクトリーではなく、<WAS\_PROFILE\_ROOT>/logs/ffdcというディレクトリーに出力されます。ファイル名は<process\_name>\_<timestamp>.txt (例えば、member01\_237f13d\_12.01.19\_15.03.50.8545244359570673761648.txt)となります。また、サーバー起動中に発生した例外情報の統計を管理する <process\_name>\_exception.log というログファイルもffdcディレクトリーに作成されます。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

HTTPトランスポートログ

■ http\_access.log / http\_error.log

◆ WebコンテナのHTTPトランスポートに対するアクセスの記録

▶ Webサーバーを介さない場合に有用なアクセスログ

◆ デフォルトでは取得されない

Webコンテナ・トランスポート・チェーン  
→「Transport\_name」→「HTTPインバウンド・チャネル」

サーバー→サーバー・タイプWebSphere Application Server→  
<Server\_name>→NCSA アクセス・ロギングおよび HTTP エラー・ロギング

一般プロパティ

☐サーバー起動時にロギング・サービスを使用可能にする

サーバーの起動時に  
ロギングのサービスを使用可能にする

NCSA アクセス・ロギング

☒アクセス・ロギングを使用可能にする

※アクセス・ログのファイル・パス  
\${SERVER\_LOG\_ROOT}/http\_

アクセス・ログの最大サイズ  
500 MB

ヒストリー・ファイルの最大数  
1

※NCSA アクセス・ログ・フォーマット  
共通

エラー・ロギング

☒エラー・ロギングを使用可能にする

※エラー・ログのファイル・パス  
\${SERVER\_LOG\_ROOT}/http\_

内蔵接続

☒ユーザー・バースタント・キーアラブ接続

☐接続当りの無制限のバースタント要求数

☒接続当りの最大のバースタント要求数

接続当りの最大のバースタント要求数  
100

サービス結合からの保護

最大ヘッダー・フィールド・サイズ  
32768

最大ヘッダー数  
50

☐要求全体の制限バッファ・サイズ  
要求全体の最大のバースタント・サイズ

ロギング

☐ロギング可能化

☒NCSA アクセス・ロギング

☐エラー・ロギング

適用 OK リセット 取り消し

指定したトランスポートに対して  
ロギングを使用可能にする

46

HTTP トランスポート・ログはアプリケーション・サーバーの Web コンテナの HTTP トランスポートに対するアクセスを記録するログで、Web サーバーのアクセス・ログに相当するものになります。これは Web サーバーを介さないアクセスに対して、Web サーバーのアクセス・ログの代わりとなるログです。ファイル名は `http access.log` および `http error.log` で、JVM ログと同じディレクトリに出力されます。

HTTPトランスポートのログはデフォルトでは取得されないので、取得するためには設定を変更する必要があります。

管理コンソールの「サーバー」→「サーバー・タイプ」→WebSphere Application Server からアプリケーション・サーバーを選択し、「NCSA アクセス・ロギングおよび HTTP エラー・ロギング」を選択します。設定画面では、アクセス・ロギングやエラー・ロギングを使用可能にする、といった項目にはチェックが付いていますが、「サーバー始動時にサービスを使用可能にする」にチェックが付いていませんので、チェックを入れます。他にもファイルの最大サイズやフォーマット (共通 (Common) or 結合 (Combined))、ファイル名などはここで指定します。

実際にどの HTTP トランスポートのリクエストおよびエラーをロギングするかは、チャンネルに対して行います。アプリケーション・サーバーのページより、「Web コンテナ・トランスポート・チェーン」で該当トランスポート・チェーンを選択し、「HTTP インバウンド・チャンネル」をクリックします。設定画面の一般プロパティの下に「ロギング可能化」にチェックを付けると、このチャンネルに対してリクエストされたアクセス・ログとエラー・ログが出力されます。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## IBM保守ログ

■ activity.log

- ◆ JVMログに詳細情報を追加したログ
- ◆ showlog.bat(sh)を使用して読み取る。
- ◆ デフォルトでは出力されなくなった。 New v8
- ◆ 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〉→IBM保守ログ→保守ログを使用可能にする をチェック。

**ロギングおよびトレース > member01 > IBM 保守ログ**

このページを使用して、IBM 保守ログを構成します。これはアクティビティ・ログとも呼ばれます。IBM 保守ログは、System.out ストリームに書き込まれるアプリケーション・サーバーのメッセージに加えて、問題の分析に使用できる拡張サービス情報が入っている特別なメッセージを含んでいます。1 つのノード上の、すべてのアプリケーション・サーバーおよびそれらのノード・エージェント (ある場合) も含めたすべての Java 仮想マシン (JVM) に対して、1 つの保守ログがあります。デプロイメント・マネージャーに対しては、その独自の logs ディレクトリに、アクティビティ・ログが別々に作成されます。IBM 保守ログは、バイナリー・フォーマットで保守されます。IBM 保守ログを表示するには、ログ・ナビゲーターまたは Showlog ツールを使用します。

構成

一般プロパティ

☒ 保守ログを使用可能にする

ファイル名:

最大ファイルサイズ:  MB

☒ 相関 ID を使用可能にする

ここをチェックする

ここをチェックする

47

IBM保守ログのファイル名はactivity.logという名前でノード単位に作成されます。これは〈WAS\_PROFILE\_ROOT〉/logsディレクトリに出力されます。このIBM保守ログはJVMログに詳細情報を追加したログで、showlog.bat(sh)コマンドにてactivity.logの内容を確認する事が出来ます。

WAS V7 まではデフォルトで出力されていましたが、V8 ではデフォルトでは出力されなくなりました。

出力させるには管理コンソール上で、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〉→IBM保守ログと遷移し、「保守ログを使用可能にする」をチェックします。



## WASのトレース

- `trace.log`

- ◆ WASコンポーネントの実行時の詳細な記録
- ◆ 障害発生箇所特定の大きな手がかり
- ◆ 移動中のサーバーでも設定が可能

稼働中のサーバーでトレースの設定を行うにはランタイム・タブを開き、「ランタイム変更も構成に保管する」をチェックし、適用する。

サーバー→サーバー・タイプ→WebSphere Application Server→  
 <Server name>→「診断トレース・サービス」

このページを使用して、診断トレースとサービスのポリティシーを表示および変更します。診断トレースでは、この管理対象プロセス内でアプリケーションサーバーとゴブナーがどのように実行されるかについての詳細な情報が提供されます。構成ファイルでの変更は、サーバーの再起動時に適用されます。ランタイムファイルでの変更は、即時に適用されます。

構成 ランタイム

Trace Output

● メモリー・バッファ  
・ 最大バッファ・サイズ  
8 x 1000 件

● ファイル  
● 最大ファイルサイズ  
20 MB

\* ヒストリーファイルの最大数  
 5

\* ファイル名  
 履歴ファイル名: C:\Program Files\Microsoft Office\Office12\Excel12\Excel12.exe

トレース出力形式  
基本 (デフォルト) \*

(最大サイズに達した履歴・ファイルを削除)

構成 ランタイム

一般加印行

✓ ランタイム変更も構造的に保存する

一、書き出すコンポーネントの指定

細は次ページ)

\_\_\_\_\_

・ファイルの最大数

ース・ファイルが新規に作成され、  
だけ残される)

ファイルのフォーマット

トレースに書き出すコンポーネントの指定  
(詳細は次ページ)

トレースの出力ファイル名、  
 最大ファイルサイズとヒストリー・ファイルの最大数  
 (最大サイズに達したとき、トレース・ファイルが新規に作成され、  
 ヒストリー・ファイル最大数の数だけ残される)

ファイルのフォーマット  
(基本or拡張orログアナライザ)

48

トレースを用いると、アプリケーション・サーバーや環境内の他のプロセスなど、WASコンポーネントの実行に関する、より詳細情報を取得することが出来ます。トレースを使用することにより、障害発生箇所特定の大きな手がかりとなります。トレースの収集は、多くの場合テクニカル・サポート担当者の依頼の元で行います。トレースはJVMログ同様にプロセス単位でディレクトリにtrace.logという名前取得されます。またランタイム・タブを使用することで稼動中のサーバーでもトレースの取得を設定することが可能です。

トレースの設定画面は、管理コンソール左メニューから「サーバー」→「サーバー・タイプ」→WebSphere Application Server でサーバーを選択し、「診断トレース・サービス」をクリックすると表示されます。この画面からは、トレースの出力先やバッファ・サイズ、最大ファイル・サイズやフォーマットを指定することが出来ます。トレースのログ出力はデフォルトでもONになっています。しかし、ログ詳細レベルがデフォルトでは\*=infoとなっているため、実際には出力されません。

以前のバージョンでは、トレースの取得は負荷の高い処理であるため、問題判別の際に限って使用することがベスト・プラクティスでした。

しかし、V8.0 では HPEL の登場により、常時取得が現実的になっています。



WASV8.0によるWebシステム基盤設計Workshop

## ログ詳細レベルの変更

- コンポーネント単位でトレースに出力させるものを指定
- 下のGUIから選択orログ詳細レベル仕様を直接入力

アプリケーションサーバ > member01 > 設定 | サービス > ログ詳細レベルの変更

ログレベルを指定することによって、どのイベントが、Java ログによって記録されるのかを制御します。個々のコンポーネントのログ詳細レベルを指定するログコンポーネントを指定し、事前定義済みのログレベルを選択します。ログ

機密の可能性のあるデータをログしない。

ログ詳細レベルの変更

コンポーネント

グループ

ログ詳細レベル仕様 <component>=<level>を直接書き込んでも構わない

右クリックして選択

メッセージレベル

- 致命的
- 重大
- 警告
- 監査
- 情報
- 構成

トレースレベル

- 詳細 - 低
- 詳細 - 中
- 詳細 - 高
- 詳細 - 最高

49

トレース設定画面右の追加プロパティの「ログ詳細レベル仕様」をクリックすると、どのコンポーネントに対しログまたはトレースをどのレベルで取得するかを設定することが出来ます。画面下からコンポーネントを選択しクリックすると、レベルを選択するポップアップが現れますので、レベルを選択すると、それらが上記画面に反映される仕組みになっています。デフォルトでは全てのレベルが情報(info)になっています。

また、上記画面からコンポーネントとレベルを<component>=<level>という形で直接書き込んでも構いません。複数のコンポーネントに対しレベルを指定したい場合は「:」で区切ります。

V8 新機能として、「機密の可能性のあるデータのログおよびトレースを使用不可にする」オプションが追加されました。ここをチェックすることで、HTTPリクエストに含まれるユーザーの入力をログしないように設定することができます。

## ロギング・レベル

- 致命的から詳細-最高 までの10段階 (+ off + all)
- trace.log に書き込まれるのは詳細-中 (fine) 以降を設定した場合

System.out.logログに書き込まれる情報量が変化

トレースとしてtrace.logに書き込まれる

| ロギング・レベル<br>(ログ詳細レベルでの指定) | 管理コンソール上の<br>呼び方 |
|---------------------------|------------------|
| off                       | －                |
| fatal                     | 致命的              |
| severe または error          | 重大               |
| warning                   | 警告               |
| audit                     | 監査               |
| info                      | 状況               |
| config                    | 構成               |
| detail                    | 詳細-低             |
| fine                      | 詳細-中             |
| finer または entryExit       | 詳細-高             |
| finest または debug          | 詳細-最高            |
| all                       | －                |

重大度高  
-----  
低

50

ロギング・レベル 致命的 (fatal) から詳細-低 (detail) までの情報は、SystemOut.log に出力されます。ロギング・レベルを詳細-中 (fine) 以上とした場合、fine とそれより詳しい (= 重要度の低い) メッセージが trace.log に出力されます。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## WASのトレースの取得例

■ 例：データベース接続の状況を確認する際のトレース

\*=info:WASj2c=all:RRA=all:WAS.database=all:Transaction=all

◆ WAS-DB間での問題判別に有用

- 接続プールの使用状況
- リソース参照の情報(アイソレーション・レベル・共用接続の有無、など)
- トランザクションの開始・終了

trace.log

コネクション・オブジェクトは接続プール内で、1つはInUse、もう1つはFree。その接続は非共用接続

```

18:13:47:049 JST] 00000039 PoolManager 3 release(), Pool contents ==> PoolManager name
PoolManager object:867644343
Total number of connections: 2 (max/min 10/1, reap/unused/aged 180/1800/0, connectiontimeout/purge 180/EntirePool)
(testConnection/interval false/0, stuck timer/time/threshold 0/0/0, surge time/connections 0/-1)
Shared Connection information (shared partitions 200)
No shared connections

Free Connection information (free distribution table/partitions 5/1)
(2)(0)MCWrapper id 5ff15ff1 Managed connection WSRdbManagedConnectionImpl@6ba36ba3 State:STATE_ACTIVE_FREE

Total number of connection in free pool: 1

UnShared Connection information
MCWrapper id 32e332e3 Managed connection WSRdbManagedConnectionImpl@408d408d State:STATE_ACTIVE_INUSE Thread
Id: 00000039 Thread Name: Web
Container : 0 Handle count 0
Total number of connection in unshared pool: 1

```

51

ここでは実際にトレースを取得した際のサンプルを紹介します。

発生した問題に対してどのようなトレースを取得するかは、IBM サービス・ラインの指示や、先述したMustGatherのサポート・サイトなどを参考にして取得を行います。上記はデータベース接続の状況を確認する際に有用な、J2C に関するトレースを取得した例になります。このトレース結果から、接続プールの使用状況や、アイソレーション・レベルや共用接続の有無といったリソース参照の情報、トランザクションに関する情報などがわかります。WASとデータベースとの間で問題が発生した際には、問題判別のために非常に有用です。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## JDBCトレース・ログ

---

- JDBCトレース・ログとは
  - ◆ JDBCドライバーが出力するトレース
  - ◆ アプリケーションとデータベース間のデータの流れを確認できる
  
- 効果的なケース
  - ◆ ① SQLExceptionの原因を探る
  - ◆ ② ドライバーの内部エラーの原因を探る
  - ◆ ③ JDBC APIの実行時刻とreturn時刻を知る
  - ◆ ④ パラメーター・マーカへの設定値を確認する
  - ◆ ⑤ 分離レベルを確認する
  
- 例1: 分離レベルを確認する
  - ◆ トレースに出力されていたTransactionIsolationにより確認する (0=None = UR, 2 = CS, 4 = RS, 8 = RR)

[省略] setTransactionIsolation (2) called
  
- 例2: JDBC APIの実行時刻とreturn時刻を知る
  - ◆ calledとreturnedを抽出する
  - ◆ 更に、ネットワークに要した時間、Driver内の処理時間を算出できる

[省略][Time:2007-05-17-14:05:51.794][省略] executeUpdate (insert into Table (sampleColumn) values (1)) called  
 [省略][Time:2007-05-17-14:05:51.888][省略] executeUpdate () returned  
 [省略][SystemMonitor:stop] core: 94.63ms | network: 48.102ms | server: 0.0ms

| java.sql.Connection     |                              |   |
|-------------------------|------------------------------|---|
| public static final int | TRANSACTION_NONE             | 0 |
| public static final int | TRANSACTION_READ_COMMITTED   | 2 |
| public static final int | TRANSACTION_READ_UNCOMMITTED | 1 |
| public static final int | TRANSACTION_REPEATABLE_READ  | 4 |
| public static final int | TRANSACTION_SERIALIZABLE     | 8 |

52

JDBCトレース・ログとは、JDBCドライバーが出力するトレースになり、ドライバーのプロパティの設定値や実行しているSQLステートメント・JDBC APIなど、アプリケーションとデータベース間のデータの遣り取りの流れを確認する事が出来ます。JDBCトレースログを取得する効果的なケースとしては、次が挙げられます。

- ① SQLExceptionの原因を探る
- ② ドライバーの内部エラーの原因を探る
- ③ JDBC APIの実行時刻とreturn時刻を知る
- ④ パラメーター・マーカへの設定値を確認する
- ⑤ 分離レベルを確認する

・分離レベルとは

実行するトランザクションが、RDBMSによってどの程度のデータ保証すべきかを定義したもの。

UR: Uncommitted Read(未コミット読み取り)

CS: Cursor Stability(カーソル固定)

RS: Read Stability(読み取り固定)

RR: Repeatable Read(反復可能読み取り)

Designing Web System Infrastructure with WAS V8.0

WAS V8.0によるWebシステム基盤設計Workshop

## JDBCトレース設定 (DB2)

- JDBCドライバーとデータベース間でやり取りされるデータの流を取得
  - ◆ ドライバーのプロパティ値、実行しているSQLステートメント、実行しているJDBC API等を確認できる
- 設定指針
  - ◆ 以下のようなWASトレース設定では取得できない場合
    - ドライバーの内部エラー
    - JDBC APIの実行時刻とreturn時刻
    - パラメーター・マーカの設定値
- 設定方法
  - ◆ 管理コンソールより、JDBC → データ・ソース → カスタム・プロパティを選択し、以下を設定する

|                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |       |
|-------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| <input type="checkbox"/> <code>traceLevel</code>      | -1 Specifies the level of trace, determined by a bitwise combination of contents: TRACE_NONE=0, TRACE_CONNECTION_CALLS=1, TRACE_STATEMENT_CALLS=2, TRACE_RESULT_SET_CALLS=4, TRACE_DRIVER_CONFIGURATION=16, TRACE_CONNECTS=32, TRACE_OGSA_FLOWS=64, TRACE_RESULT_SET_META_DATA=128, TRACE_PARAMETERS_META_DATA=256, TRACE_DIAGNOSTICS=512, TRACE_SQL=1024, TRACE_META_CALLS=4096, TRACE_DATA_SOURCE_CALLS=16384, TRACE_LARGE_OBJECT_CALLS=32768, TRACE_9151EN_MONITOR=131072, TRACE_TRACEPOINTS=262144, TRACE_ALL=1. | false |
| <input type="checkbox"/> <code>traceFileAppend</code> | Specifies whether to append to or overwrite the file specified by the tracefile property. The default is false, which means the file specified by the tracefile property is overwritten.                                                                                                                                                                                                                                                                                                                             | false |
| <input type="checkbox"/> <code>traceFile</code>       | The file to store the trace output. If you specify the trace file, the DB2 JCC trace will be logged in this file. If this property is not specified and the WAS database or com.ibm.was.db2.logwriter trace group is enabled, then both WebSphere Application Server trace and DB2 trace will be logged into the WebSphere Application Server trace file.                                                                                                                                                            | false |
| <input type="checkbox"/> <code>traceDirectory</code>  | Specifies the directory where the trace file will be created.                                                                                                                                                                                                                                                                                                                                                                                                                                                        | false |

`[ibm][db2][jcc][省略]PreparedStatement@39eae493 setInt(1, 1) called`  
`[ibm][db2][jcc][省略]PreparedStatement@39eae493 setString(2, COL2-1) called`

トレースレベルを設定する。  
 整数値を加算することで、複数のトレースが取得可能。  
 (例)1+2=3 (CONNECTION\_CALL, STATEMENT\_CALL)

trueに設定すると、トレースファイルが上書きされない。

トレースファイルを設定する。

接続オブジェクト毎にトレースファイルが出力される。  
 (例) \_cpds\_1, \_cpds\_2...  
 traceFileを設定すると、traceDirectoryは無効になる。

53

JDBCトレースとは、JDBCドライバーとデータベース間でやり取りされるデータの流を取得したのになります。ドライバーの内部エラー、JDBC APIの実行時刻とreturn時刻やパラメーター・マーカの設定値を確認したい場合等に、取得することを検討して下さい。

JDBCトレースは、管理コンソールより、JDBC → データ・ソース → カスタム・プロパティを選択し、`traceLevel`、`traceFileAppend`、`traceFile`、`traceDirectory`を設定して下さい。`traceLevel`については、下記リンク先をご確認下さい。

・DB2 Information Center - 「サポートされるすべてのデータベース製品に共通の IBM Data Server Driver for JDBC and SQLJ のプロパティ」

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.apdv.java.doc/doc/r0052038.html>

(抜粋)

`com.ibm.db2.jcc.DB2BaseDataSource.TRACE_CONNECTION_CALLS= 0x1 (= 1)`

`com.ibm.db2.jcc.DB2BaseDataSource.TRACE_STATEMENT_CALLS= 0x2 (= 2)`

(※)パラメーター・マーカとは

通常、疑問符(?)で示され、ステートメント実行中に値が取得されるSQLステートメント内のプレースホルダーです。何度も実行する必要のあるSQLステートメントの場合、SQLステートメントを一回だけ準備し、パラメーター・マーカを使って実行時に入力値を置換することにより、照会プランを再利用することが出来ます。

○ `SELECT * FROM T1 WHERE C1 = ? ;`

× `SELECT * FROM T1 WHERE C1 = 100;`

・DB2 V9.5 Information Center - 「パラメーター・マーカ」

<http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/index.jsp?topic=/com.ibm.db2.luw.apdv.routines.doc/doc/c0020295.html>

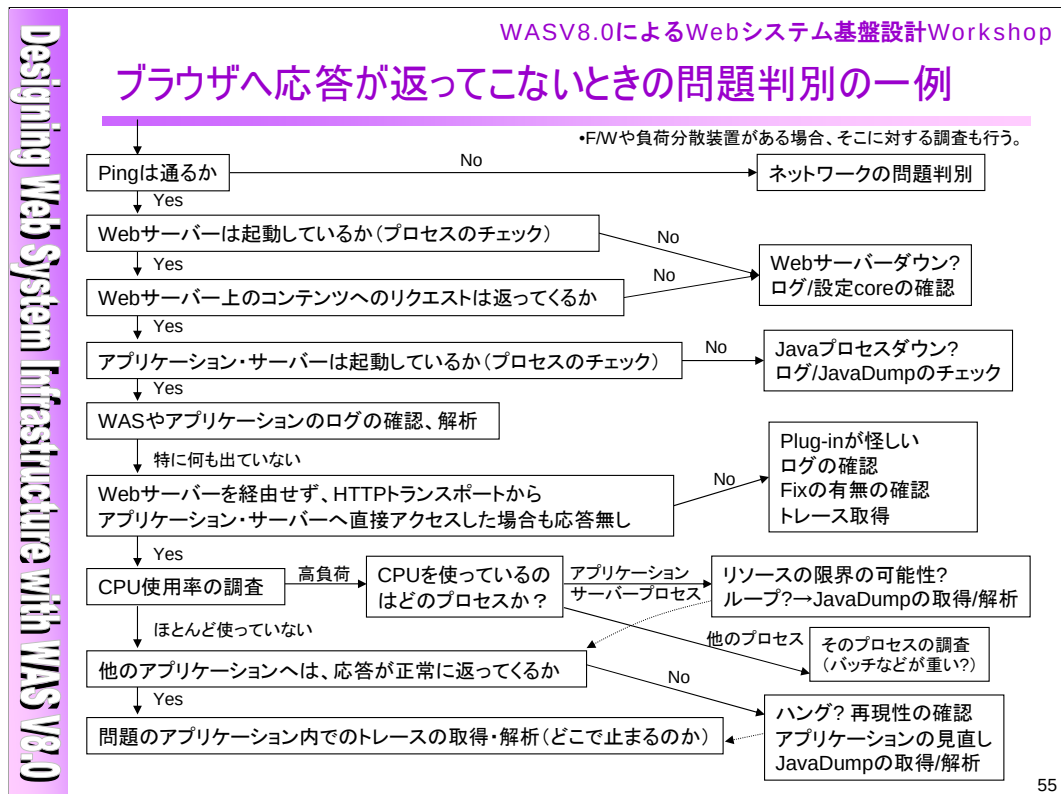
Designing Web System Infrastructure with WAS V8.0

## 2.問題判別

---

2-1 ログ、トレース  
2-2 症状別問題判別

54



上記はブラウザから応答が返ってこない時の問題判別手順の一例を示したのになります。

ping は正常に通るか、Web サーバーは起動しているか、静的コンテンツへのリクエストは返ってくるか、など一つ一つの問題を切り分けて問題判別を行います。また負荷分散装置やファイアウォールなどがある場合は、それに対する調査も必要になります。

最近、セキュリティの強化に伴ってか、ファイアウォールによる通信の遮断を原因とするお問い合わせを受けています。

ファイアウォールで特定のポートに向けた通信のみが遮断されている場合、ping は通っても目的の通信が通っていないことがありますので、特に注意が必要です。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## ハング/ループ

---

- プロセスは存在しているが、応答が無い
  - ◆ リソースの処理待ちであったり、パフォーマンス劣化によるスローダウン状態ではない
- 主な原因
  - ◆ アプリケーションや製品の問題
    - デッドロック
    - 再帰呼び出しによる無限ループ
- 対応: Java ダンプの取得が有効
  - ◆ ハードウェアやWASのリソースを使い切っていないかを確認
  - ◆ 十分なリソースがあるにもかかわらず問題が発生している場合  
Java ダンプを取得し、デッドロックや無限ループの存在を確認する

56

ハングは、プロセスは存在しているにもかかわらず応答がない現象になります。

主な原因としては、アプリケーションまたは製品のバグに起因する場合があります。スレッドがデッドロックを起こし停止してしまう場合や、再起呼び出しによる無限ループが発生し、結果的に応答がなくなる場合があります。最終的にStackOverflowでJVMがダウンしてしまう可能性もあります。ハングが発生した際、CPU使用率があまり高くない場合にはデッドロック、CPU使用率が高い場合には再起呼び出しによる無限ループの状態になっていることが考えられます。この場合は、Java ダンプの取得が有効です。

また、後方リソースの処理待ちに起因する場合があります。この場合、DB や MQなど、後方リソース処理やネットワーク処理などにより、スレッドがI/O 処理待ちの状態となります。この場合、後方のリソースについて問題判別を行う必要があります。

また、パフォーマンスの劣化により、レスポンスが返ってこないように感じられる場合です。この場合、遅くてもレスポンスが帰ってくる状態であれば、後方リソースまたはパフォーマンスの問題と考えられます。(次ページ「スローダウン」参照)



Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## スローダウン

---

- レスポンスが遅い
  - ◆ ハング/ループではなく、やがてはレスポンスが返ってくる状態
- 主な原因
  - ◆ マシンのリソースの限界: CPU使用率やディスク I/Oなどの調査
    - 事前に負荷テスト・パフォーマンスチューニングを行い、リクエストをどこまで受付可能か、限界値を見極めておく
    - 負荷分散装置などでそれ以上のリクエストがこないように流量制限
  - ◆ 情報収集や、その他のアプリケーション(バッチなど)
    - トレースがONになっていたり、情報収集自体の負荷はないか
    - grep/find/tailなどのコマンドもシステムに負荷を与えることがあるので注意
- 対応: ボトルネックの特定が必要
  - ◆ OS、ネットワーク、ソフトウェアなど、全体を見て検出
  - ◆ Webサーバーでレスポンス・タイム記録 + 要求メトリックが有用  
(→「IHSアクセス・ログのカスタマイズ」)

57

スローダウンは、個々の構成要素は正常に稼動しているにも関わらず、レスポンスが返ってくるのが非常に遅い現象になります。

主な原因は、マシンのリソースの限界や他のアプリケーションの影響が考えられます。マシンのリソースはCPU使用率などを調査し、必要以上に負荷がかかっていないかを確認します。その場合には、負荷分散装置などを利用し、それ以上のリクエストがこないように流量制限することも考慮にいれなくてはなりません。また、事前に負荷テストやパフォーマンスチューニングを行い、リクエストをどこまで受付可能か、限界値を見極めておくことも重要になります。

また、他のアプリケーションやバッチの影響もあります。見落としがちですが、grepやfind、tailなどのコマンドもシステムに負荷を与えることがあるので注意が必要です。

スローダウンを解決するには、そのボトルネックがどこにあるのかを検出することが非常に重要です。考えられるボトルネックとしては、OSレベル、Network、ソフトウェアなどシステム全体からそのボトルネックを検出することが重要です。OSレベルであればvmstatコマンド、ソフトウェアであれば、Webサーバーでレスポンスタイム記録を取得し、要求メトリックを利用してコンポーネントごとのレスポンスタイムを取得することが有用です。(→「IHSアクセスログのカスタマイズ」)

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

## OutOfMemory

---

- メモリ領域を使い切った状態
- 主な原因
  - ◆ Java ヒープ領域の不足
  - ◆ Native ヒープ領域の不足
- 対応: GCやメモリ領域のチェック
  - ◆ verbose:gc
    - 出来る限り、通常時から取得して下さい
    - より詳細なGCの挙動を確認したい場合には、GCトレースを取得して下さい
  - ◆ ヒープ・ダンプの解析
    - 正常時に1回取得しておく、障害発生時の分析に便利である
    - 再現可能な状況で、PDを行う際には、問題発生までの時間の経過に合わせて（もしくは、状況に応じて）2回程取得することを推奨

58

OutOfMemoryは、メモリー領域を使い切ってしまう、ログにOutOfMemoryエラーが出力されJVMがダウンする現象になります。

主な原因としては、Java heap領域の不足の場合とNative heap領域の不足の場合があります。OutOfMemoryが発生した時には、GCがどのように行われ、メモリー領域がどのように使われているのかチェックする必要があります。そのために取得すべき情報としては、verbose:gc、HeapDumpがあります。（→「verbose:gc」）また、より詳細なGCの挙動を確認したい場合には、GCトレースを取得して下さい。

正常時の状況のヒープ・ダンプを事前に取得しておき、問題発生時に比較することにより問題判別がよりスムーズに進みますので、正常時に一度、ヒープ・ダンプを取得しておくことをお勧めします。また、問題判別を行う場合には、問題発生までの時間の経過に合わせて2回程度ヒープ・ダンプを取得することを推奨します。

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

クラッシュ

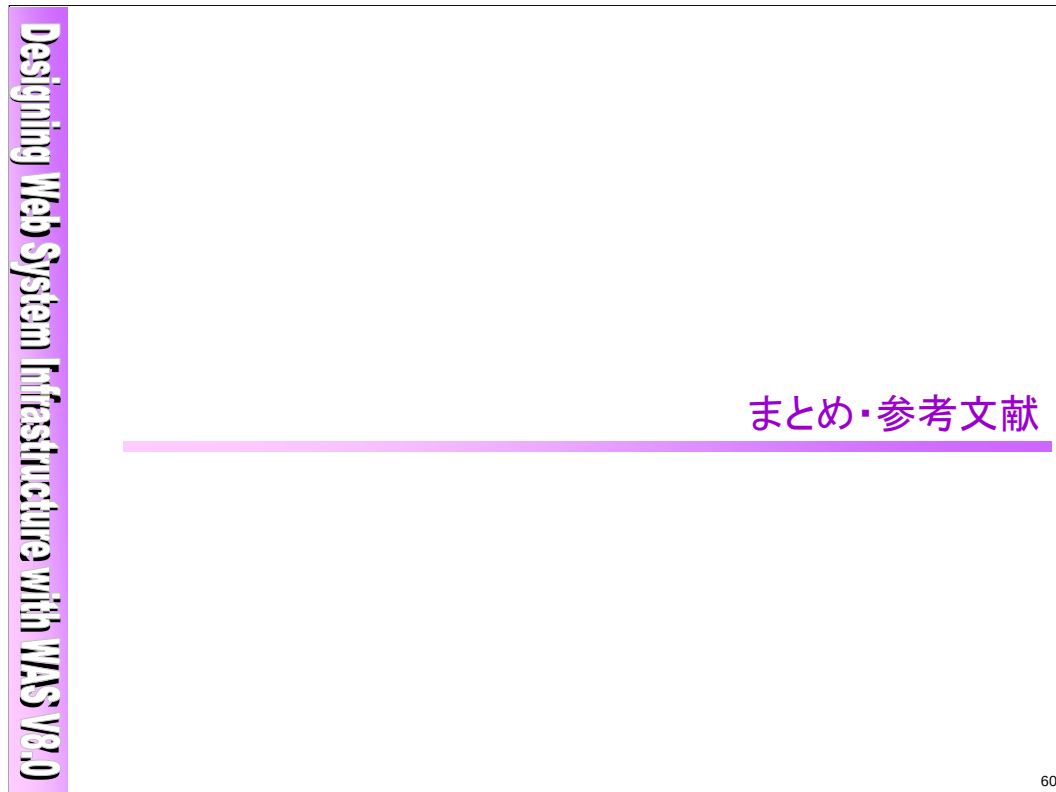
- JVMの異常終了
- 主な原因
  - ◆ ネイティブ・コード(JNI)の問題
  - ◆ JVM そのものの問題
- IBMサービス・ラインに依頼
  - ◆ クラッシュ時は Java コア、Java ダンプは自動で出力

59

クラッシュは、SIGSEGV (incorrect access to memory) 、SIGILL (illegal instruction) などJVMが異常終了する現象になります。

主な原因としては、ネイティブ・コード (JNI) の問題や JVM そのものの問題が考えられます。JNI の問題には JDBCドライバーの問題なども含まれます。

この場合には、Java コアやJava ダンプが自動で取得されます。その原因が JVM の問題と考えられる場合には、Java コアと Java ダンプ・ファイルを取得し、IBMサービス・ラインに報告します。



Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

まとめ

- 問題判別の心得
  - ◆ 基本手順、確認すべきサイト、情報収集方法について確認しました。
  - ◆ IBM Support Assistant (ISA) について確認しました。
- 問題判別
  - ◆ 以下の項目に分類し、問題判別方法について確認しました。
    - 各コンポーネントのログ・トレース
      - Edge Component、IHS、プラグイン、WASのそれぞれのログと、トレース取得方法
    - 症状別問題判別方法
      - ブラウザへ応答が返ってこないときの問題判別の一例、ハング/ループ、スローダウン、OutOfMemory、クラッシュなど、代表的な症状における問題判別方法

61

Designing Web System Infrastructure with WAS V8.0

WAS V8.0によるWebシステム基盤設計Workshop

参考文献

- Information Center
  - ◆ WAS V8.0 InformationCenter
    - <http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp>

62

Designing Web System Infrastructure with WAS V8.0

WASV8.0によるWebシステム基盤設計Workshop

ワークショップ、セッション、および資料は、IBMまたはセッション発表者によって準備され、それぞれ独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる参加者に対しても法的またはその他の指導や助言を意図したのではなく、またそのような結果を生むものでもありません。本講演資料に含まれている情報については、完全性と正確性を期するよう努力しましたが、「現状のまゝ」提供され、明示または暗示にかかわらずいかなる保証も伴わないものとします。本講演資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。本講演資料に含まれている内容は、IBMまたはそのサプライヤーやライセンス交付者からいかなる保証または表明を引き出すことを意図したもので、IBMソフトウェアの使用を規定する適用ライセンス契約の条項を変更することを意図したものでなく、またそのような結果を生むものでもありません。

本講演資料でIBM製品、プログラム、またはサービスに言及していても、IBMが営業活動を行っているすべての国でそれらが使用可能であることを暗示するものではありません。本講演資料で言及している製品リリース日付や製品機能は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、いかなる方法においても将来の製品または機能が使用可能になると確約することを意図したものではありません。本講演資料に含まれている内容は、参加者が開始する活動によって特定の販売、売上高の向上、またはその他の結果が生じると述べる、または暗示することを意図したもので、またそのような結果を生むものでもありません。パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。

記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴ、ibm.com、WebSphereは、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。他の製品名およびサービス名等は、それぞれIBMまたは各社の商標である場合があります。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)をご覧ください。

Linuxは、Linus Torvaldsの米国およびその他の国における登録商標です。  
Windowsは、Microsoft Corporationの米国およびその他の国における商標です。  
JavaおよびすべてのJava関連の商標およびロゴは、Oracleやその関連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。

63