WebSphere Application Server V9

パフォーマンス管理

© 2016 IBM Corporation

Agenda

1. パフォーマンス管理の必要性

2. WASにおけるパフォーマンス管理

3. システム全体の統合管理・監視

© 2016 IBM Corporation

なぜパフォーマンス管理が必要か？

- 昨今の現状
  - 例えば、Web系なら
    - TV番組で紹介された際に、
      膨大なユーザーが検索するためアクセスが増える
    - 結果、サーバーダウン
    - なぜか？みんなWebを利用できる時代
      - スマートフォンですぐその場で検索！
      - PCですぐ検索！
  - 他にも、システムに必要な処理量は増加傾向
    - API
    - モバイルアプリ
    - IoT
    - 社内のシステム間連携

© 2016 IBM Corporation

なぜパフォーマンス管理が必要か？

- 画面表示の遅さによる影響

ページの読込みに **3秒** かかると **40%のユーザーが離脱**

Source :Forrester Consulting “eCommerce Web Site Performance Today”

18%は表示が遅いことを理由に **決済を途中でやめる**

Source :Forrester Consulting “eCommerce Web Site Performance Today”

表示時間が **0.1秒** 増えると **売上の1%が影響** を受ける

Source :Amazon.com “Make Data Useful”

表示時間が **1秒** 増えると **顧客満足度が16%低下**

Source :Aberdeen Group “The Performance of Web Applications: Customers Are Won or Lost in One Second”

モバイル・アプリケションの**80%**は、一度起動された後、

二度と使われることはない

Source :IBM

© 2016 IBM Corporation

なぜパフォーマンス管理が必要か？

- 他にも、
  - アプリ障害に気づかなかったために、積荷および輸送処理が**丸一日停滞**し、
    結果、**120億円の利益損失**になった
  - 取引システムが**全面停止**したことで、株価が急落し、
    3週間のうちに**2900億円の金融価値を損失**
  - その他

© 2016 IBM Corporation

なぜパフォーマンス管理が必要か？

- 大前提: **問題が起きないようにする**
  - パフォーマンス設計を綿密に行う
    - パフォーマンス要件を正確に洗出す
    - チューニング・テストを繰り返し、最適な環境を整える
  - パフォーマンス監視をしっかり行う
  - 問題が起きる前に調整する

- しかし、問題が起きてしまったら、、、
  - 問題箇所を迅速に切り分ける
  - 必要な報告をスムーズに行い、担当者に連携する
  - 発生した問題を迅速に解決する
    - 適宜チューニングを行う

© 2016 IBM Corporation

本日お話したいこと

- WASにおけるパフォーマンス管理
  - チューニング
  - データ取得
  - 監視
- システム全体の統合管理・監視

© 2016 IBM Corporation

WebSphere Application Server V9

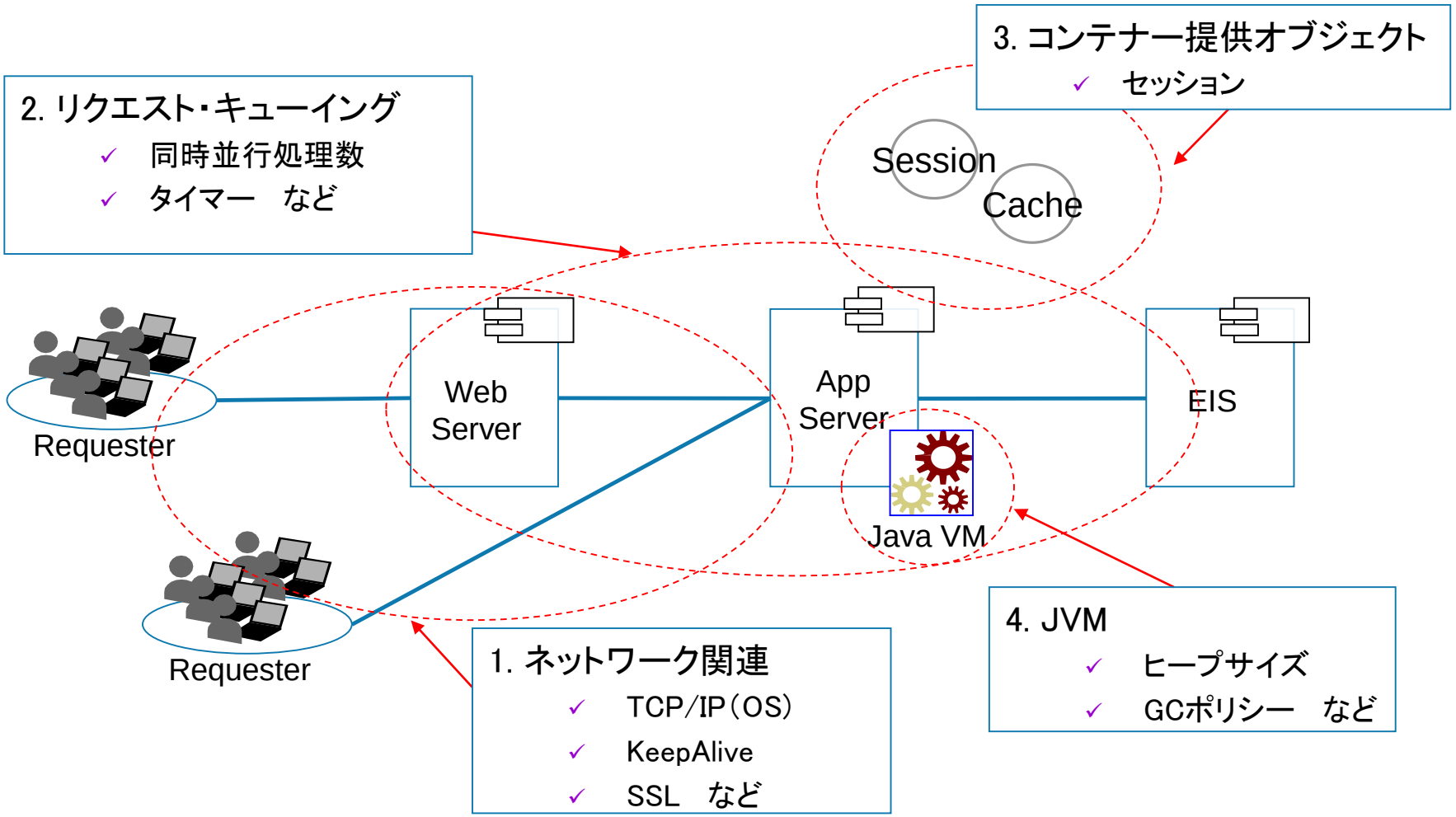
2. WASにおけるパフォーマンス管理

2-1　パフォーマンス・チューニング
2-2　パフォーマンス・データ取得
2-3　パフォーマンス監視

© 2016 IBM Corporation

チューニング概要

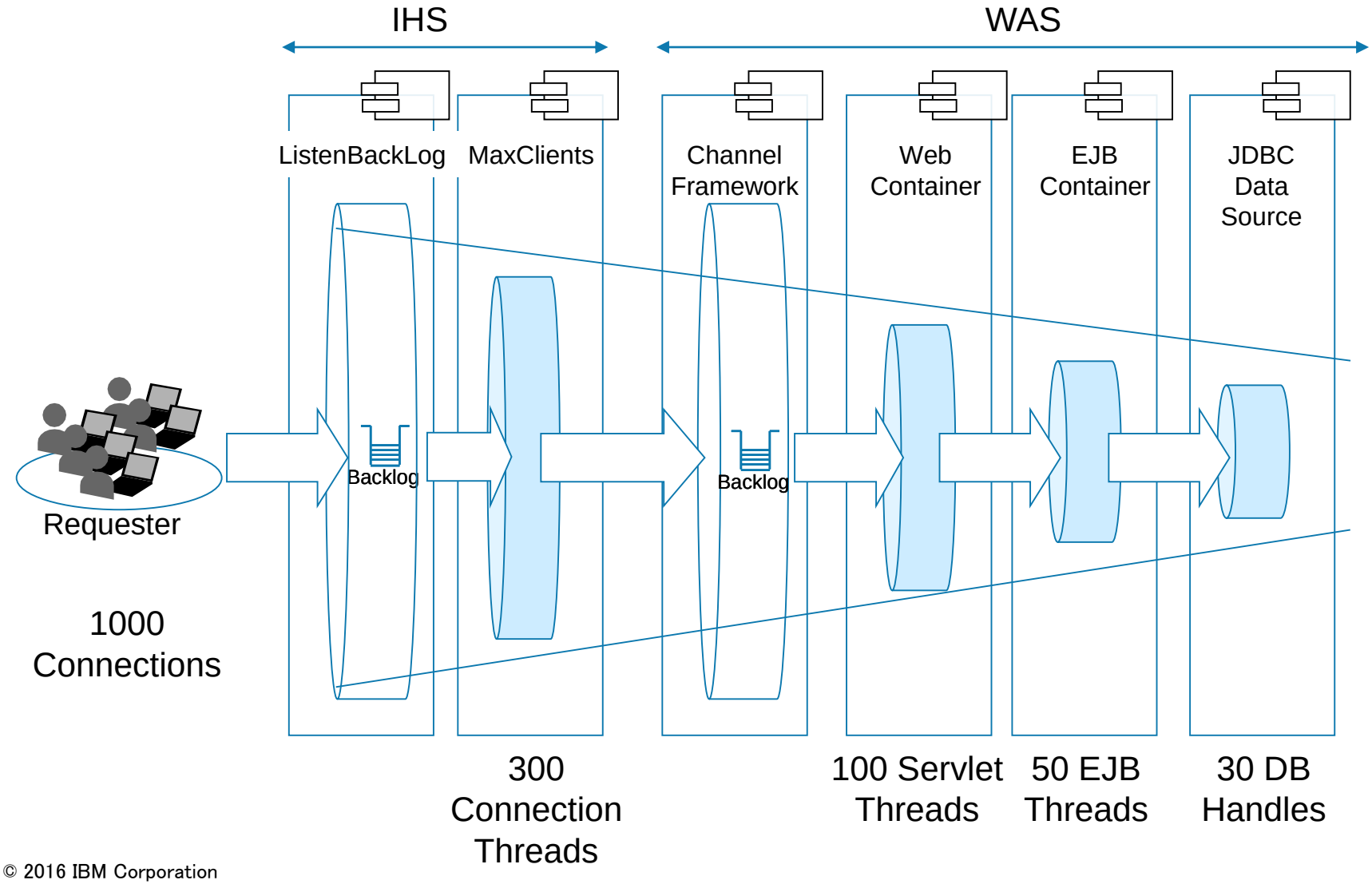
- チューニングの主要ポイント



© 2016 IBM Corporation

リクエストのキューイング方針（1/2）

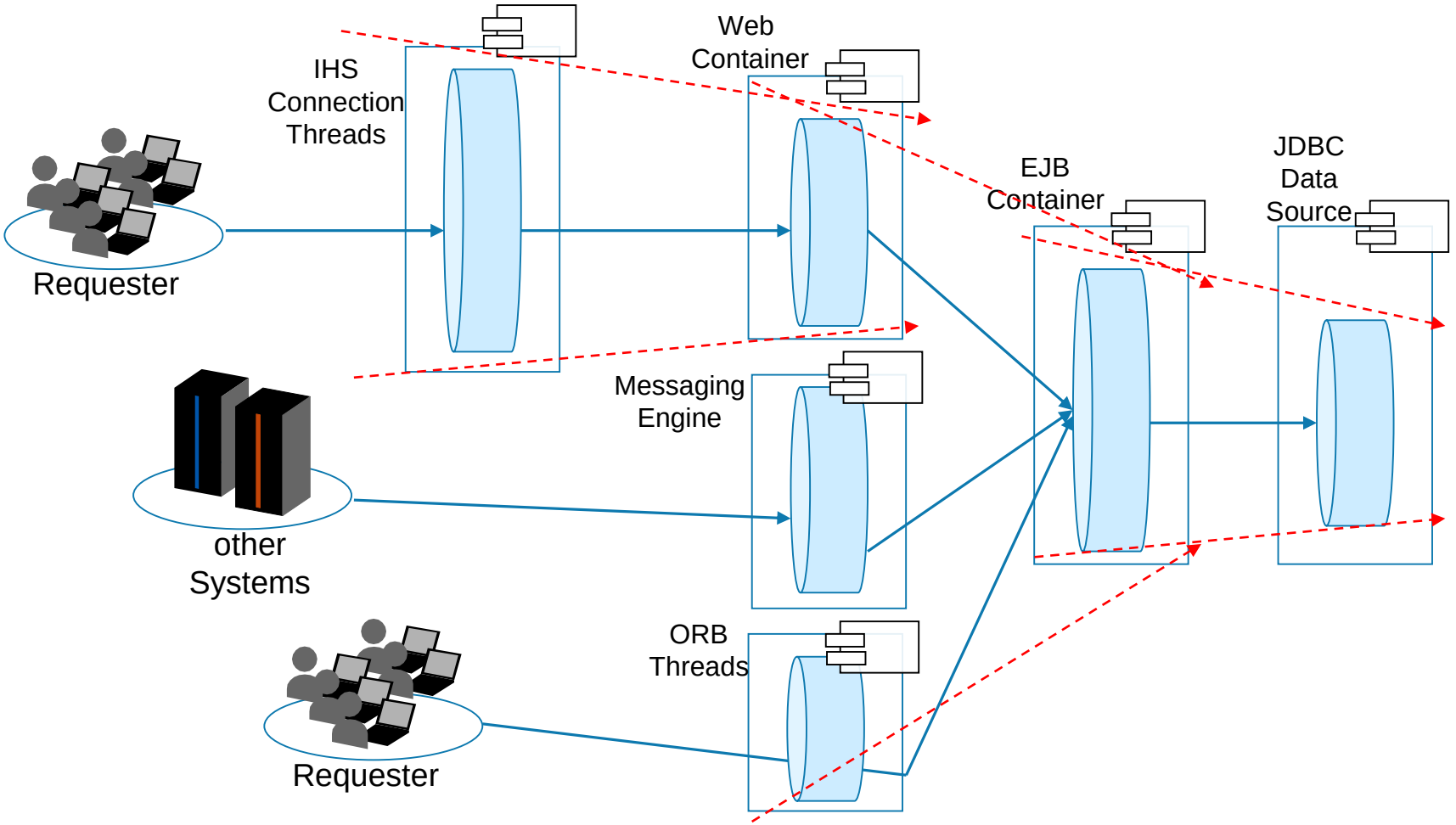
- 基本形： “Funnel（漏斗：ろうと） model”



© 2016 IBM Corporation

キューイング方針(2/2)

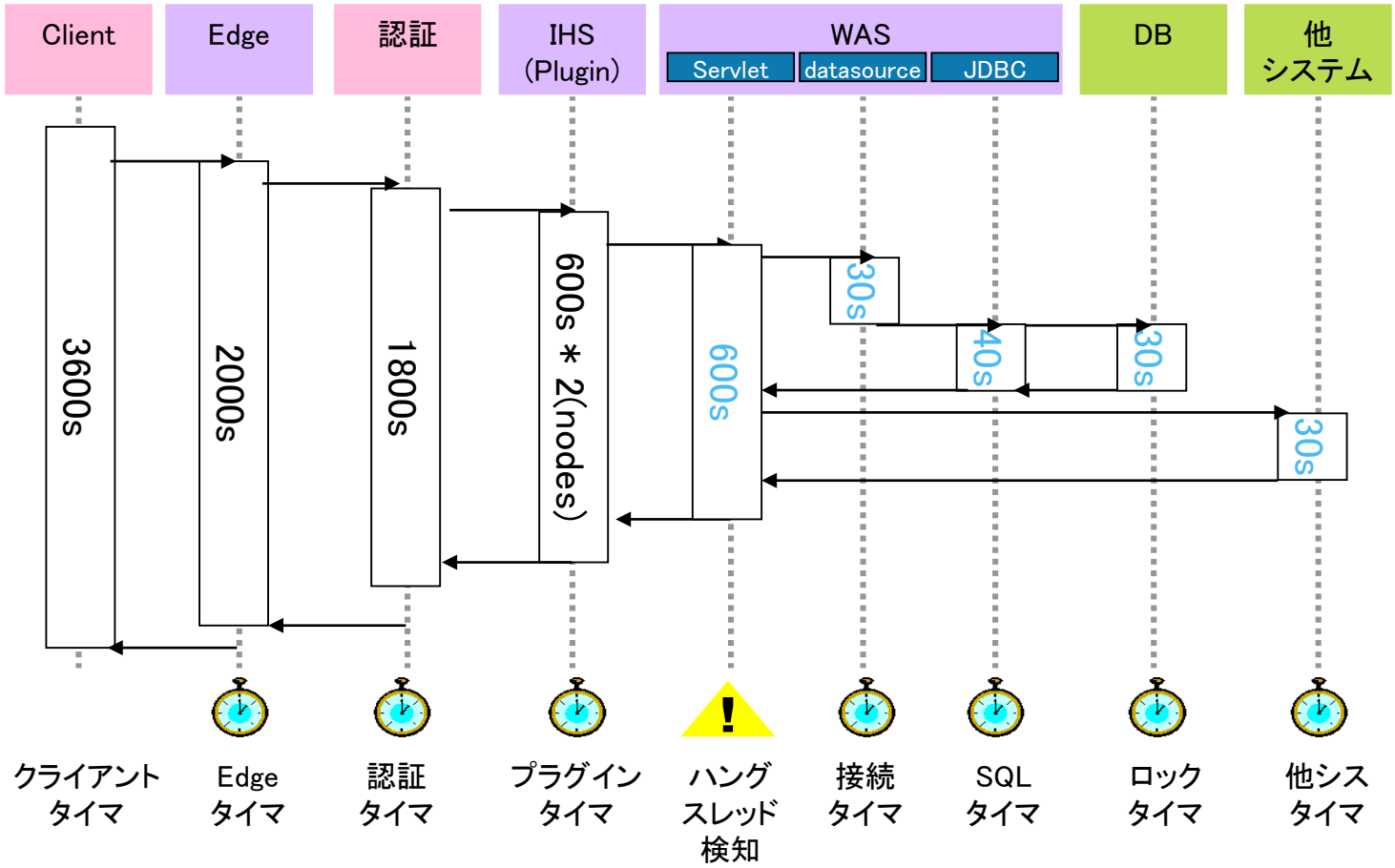
- 必ずしも「ろうと」の形にならないことも・・・
  - どこからどのコンポーネントにリクエストが入ってくるか経路を把握



© 2016 IBM Corporation

タイマー設計

- タイマー設計全体図の例



© 2016 IBM Corporation

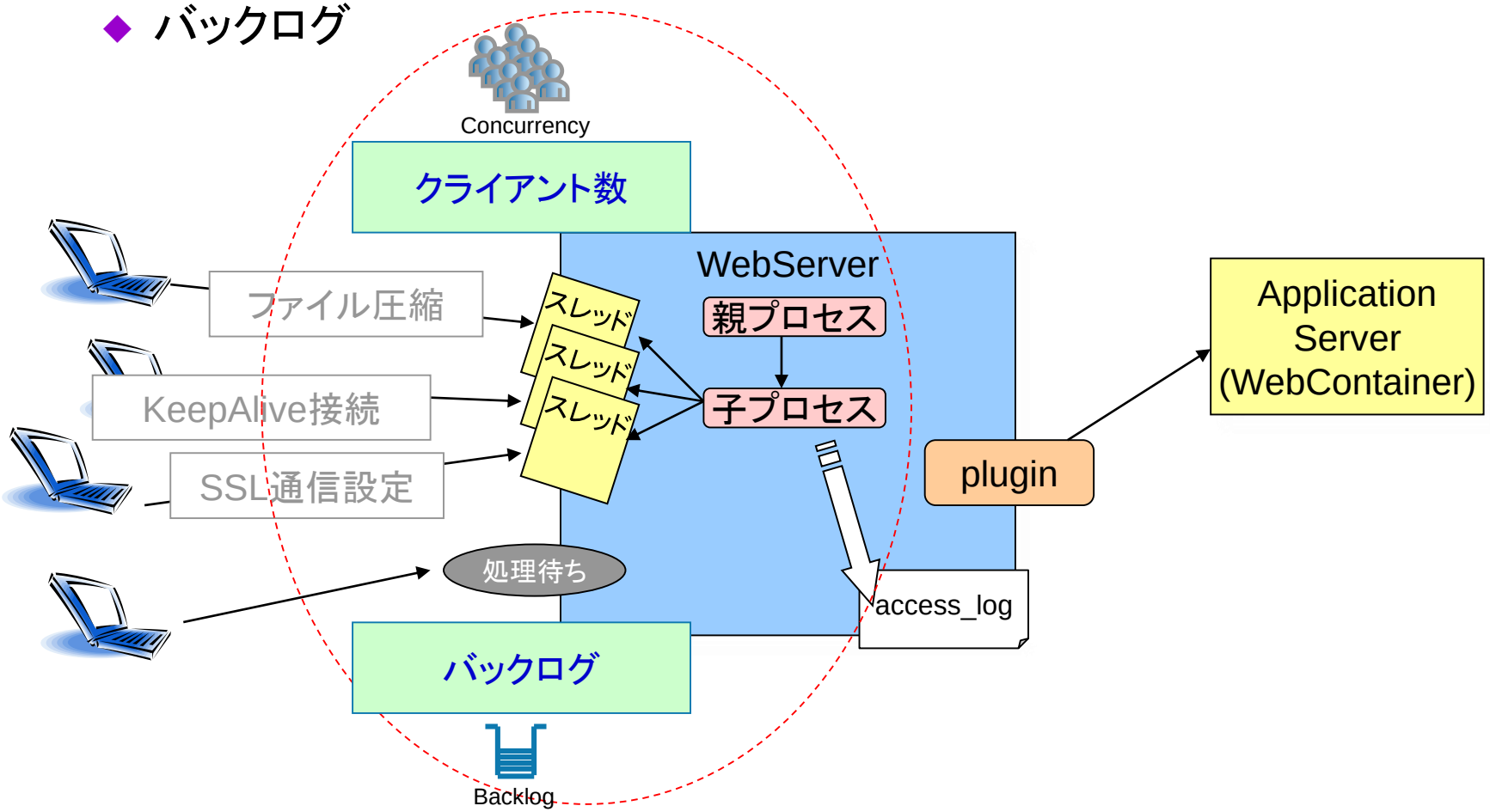
コンポーネント別チューニング

- 主な検討項目
  - Webサーバー
    - クライアント数
    - バックログ
  - Webサーバー・プラグインとWebコンテナー
    - Webコンテナーのスレッド・プール
    - アプリケーション・サーバーが処理できる接続の最大数
    - 構成ファイルの再ロード間隔
    - セッション・オブジェクト
  - JVM
    - ヒープ
    - GC
  - JDBCデータソース
    - 接続プール
    - preparedStatementCacheサイズ
  - メッセージング
    - 接続プール
    - MDBの同時並行数
    - パーシスタント・メッセージ

© 2016 IBM Corporation

Webサーバーのチューニング

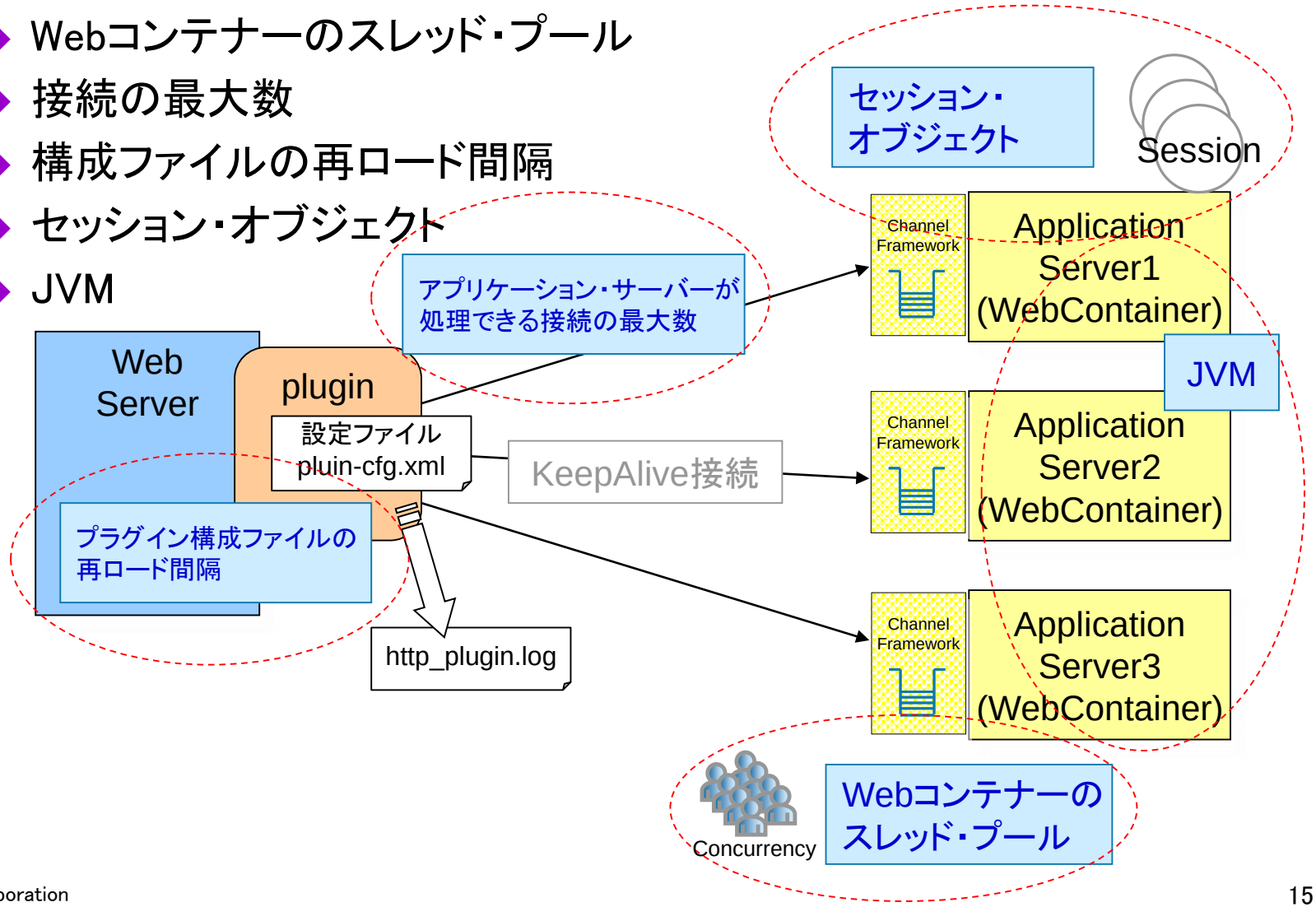
- 主要なチューニング・ポイント
  - クライアント数
  - バックログ



© 2016 IBM Corporation

プラグインとWebコンテナーのチューニング

- 主要なチューニング・ポイント
  - Webコンテナーのスレッド・プール
  - 接続の最大数
  - 構成ファイルの再ロード間隔
  - セッション・オブジェクト
  - JVM



© 2016 IBM Corporation

パフォーマンス・チューニングのポイント

- 事前確認
  - コンポーネント毎にどの要素のパラメーターが存在するかを確認
- パフォーマンス・テスト時のチューニング
  - システム全体を考慮
  - 1度に 1つのパラメーターを変更
  - チューニングを始める前にフォールバック手順を確認
  - ハードウェアをアップグレードする前に問題を理解

© 2016 IBM Corporation

WebSphere Application Server V9

2. WASにおけるパフォーマンス管理

2-1　パフォーマンス・チューニング
2-2　パフォーマンス・データ取得
2-3　パフォーマンス監視

© 2016 IBM Corporation

パフォーマンス・データ取得の概要

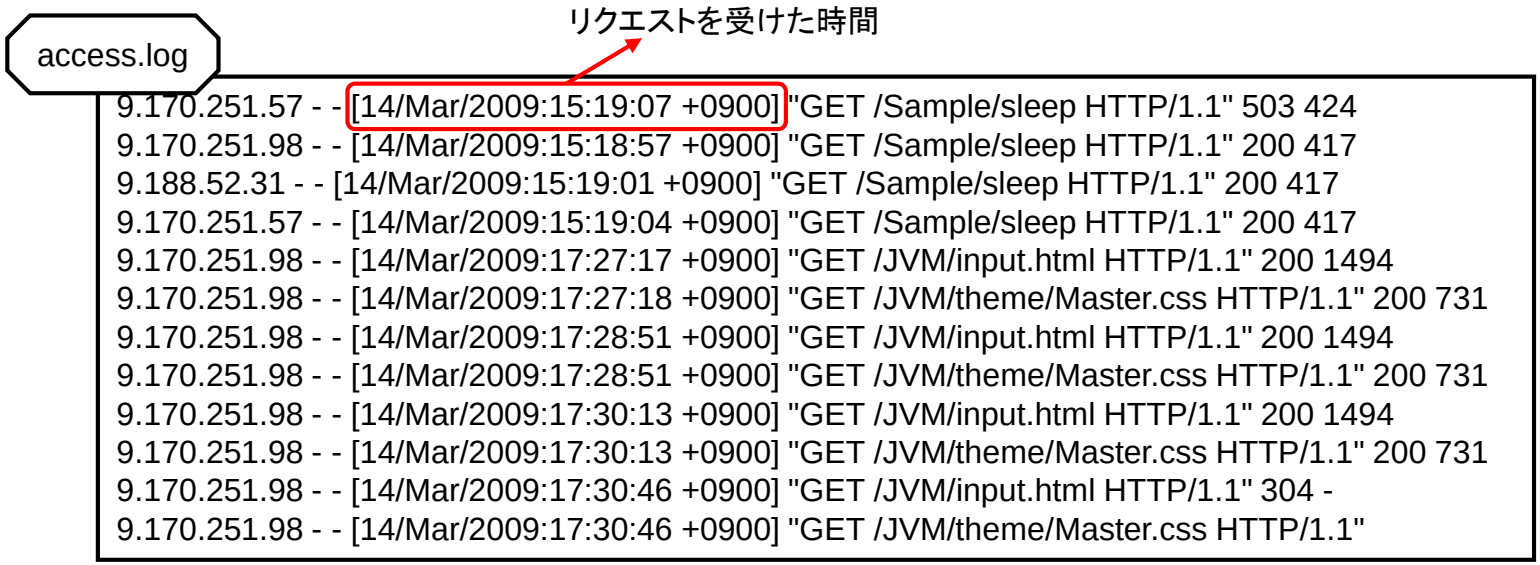
- パフォーマンス・データはさまざまな種類をさまざまな方法で取得可能
  - OS
    - サーバーのリソース使用状況
      - CPU、メモリー、ネットワークI/O、ディスクI/O、など
  - IHS
    - ユーザーアクセス数
    - レスポンスタイム
  - WAS
    - レスポンスタイム
    - ヒープ使用状況(GC発生状況)
    - Webコンテナー・スレッド状況
    - DB接続状況

- パフォーマンス・データ取得のポイント
  - 取得すべきパフォーマンス・データの明確化
  - 各機能の特徴や出力形式を理解した上でデータ取得方法を選択

© 2016 IBM Corporation

IHSアクセス・ログからアクセス数およびスループットの取得

- アクセス数
  - IHSのアクセス・ログからアクセス数を確認可能
  - IHSアクセス・ログの時間表示はリクエストを受けた時間
  - アクセス数からスループットを算出可能



IHSのアクセス・ログから処理したリクエスト数を時間間隔で割ると、
スループット（アクセス件数/秒）が計算可能

© 2016 IBM Corporation

IHSアクセス・ログからレスポンスタイムの取得

- レスポンスタイム
  - IHSのアクセス・ログからレスポンスにかかっている時間を確認可能
  - デフォルトでは記載されないので手動で追加する必要あり
  - LogFormatディレクティブ
    - %D ： レスポンスタイムをマイクロ秒単位で記録
    - %T ： レスポンスタイムを秒単位で記録

httpd.conf

```
LogFormat “%h %l %u %t ¥”%r¥“ %>s %b [%D] ”  common
```

レスポンスタイムをログに記録させる設定が必要

access.log

```
9.170.251.57 - - [20//Mar/2009:14:18:31 +0900] "GET /JVM/index.html HTTP/1.1" 404 61 [1234375]
9.170.251.57 - - [20/Mar/2009:14:18:46 +0900] "GET /JVM/input.html HTTP/1.1" 200 1494 [62500]
9.170.251.57 - - [20/Mar/2009:14:18:46 +0900] "GET /JVM/theme/Master.css HTTP/1.1" 200 731 [31250]
9.170.251.57 - - [20/Mar/2009:14:18:52 +0900] "POST /JVM/TableViewer HTTP/1.1" 200 8924 [5718750]
```

アクセス・ログにレスポンスタイムが記録される

表計算ソフトなどを用いて、レスポンスの遅いリクエストを判断

A	B	C	D	E	F
クライアントIPアドレス	アクセス時間	HTTPリクエスト	HTTPコード	バイト数	レスポンスタイム
9.170.251.57	[20//Mar/2009:14:18:31 +0900]	GET /JVM/index.html HTTP/1.1	404	61	1234375
9.170.251.57	[20/Mar/2009:14:18:46 +0900]	GET /JVM/input.html HTTP/1.1	200	1494	62500
9.170.251.57	[20/Mar/2009:14:18:46 +0900]	GET /JVM/theme/Master.css HTTP/1.1	200	731	31250
9.170.251.57	[20/Mar/2009:14:18:52 +0900]	POST /JVM/TableViewer HTTP/1.1	200	8924	5718750
9.170.251.57	[20/Mar/2009:14:20:00 +0900]	GET /JVM/index.html HTTP/1.1	304	-	0
9.170.251.57	[20/Mar/2009:14:20:29 +0900]	GET /JVM/index.html HTTP/1.1	200	228	0

© 2016 IBM Corporation

要求メトリックからレスポンスタイムの取得

- 要求メトリック
  - 要求メトリックをONに設定すると各コンポーネントのレスポンスタイムがログに出力
  - どのコンポーネントの処理で遅延が発生しているかを判断

SystemOut.log

```
[09/03/15 14:19:07:266 JST] 00000030 PmiRmArmWrapp I   PMRM0003I:
parent:ver=1,ip=9.170.251.57,time=1129785202984,pid=2720,reqid=1,event=1 -
current:ver=1,ip=9.170.251.57,time=1129785202984,pid=2720,reqid=2,event=1 type=JDBC
detail=SELECT * from staff elapsed=3125
[09/03/15 14:19:07:469 JST] 00000030 ServletWrappe A   SRVE0242I: [JvmHandsOn] [/JVM]
[/output.jsp]: 初期化が正常に行われました。
[09/03/15 14:19:08:172 JST] 00000030 PmiRmArmWrapp I   PMRM0003I:
parent:ver=1,ip=9.170.251.57,time=1129785472812,pid=2968,reqid=0,event=1 -
current:ver=1,ip=9.170.251.57,time=1129785202984,pid=2720,reqid=1,event=1 type=URI
detail=/JVM/TableViewer elapsed=14313
```

DB接続のレスポンスタイム

Webコンテナーでのレスポンスタイム

各コンポーネントのレスポンスタイムが記録されるので、その差に注目
差が大きいところに多くの時間を取られていることになる

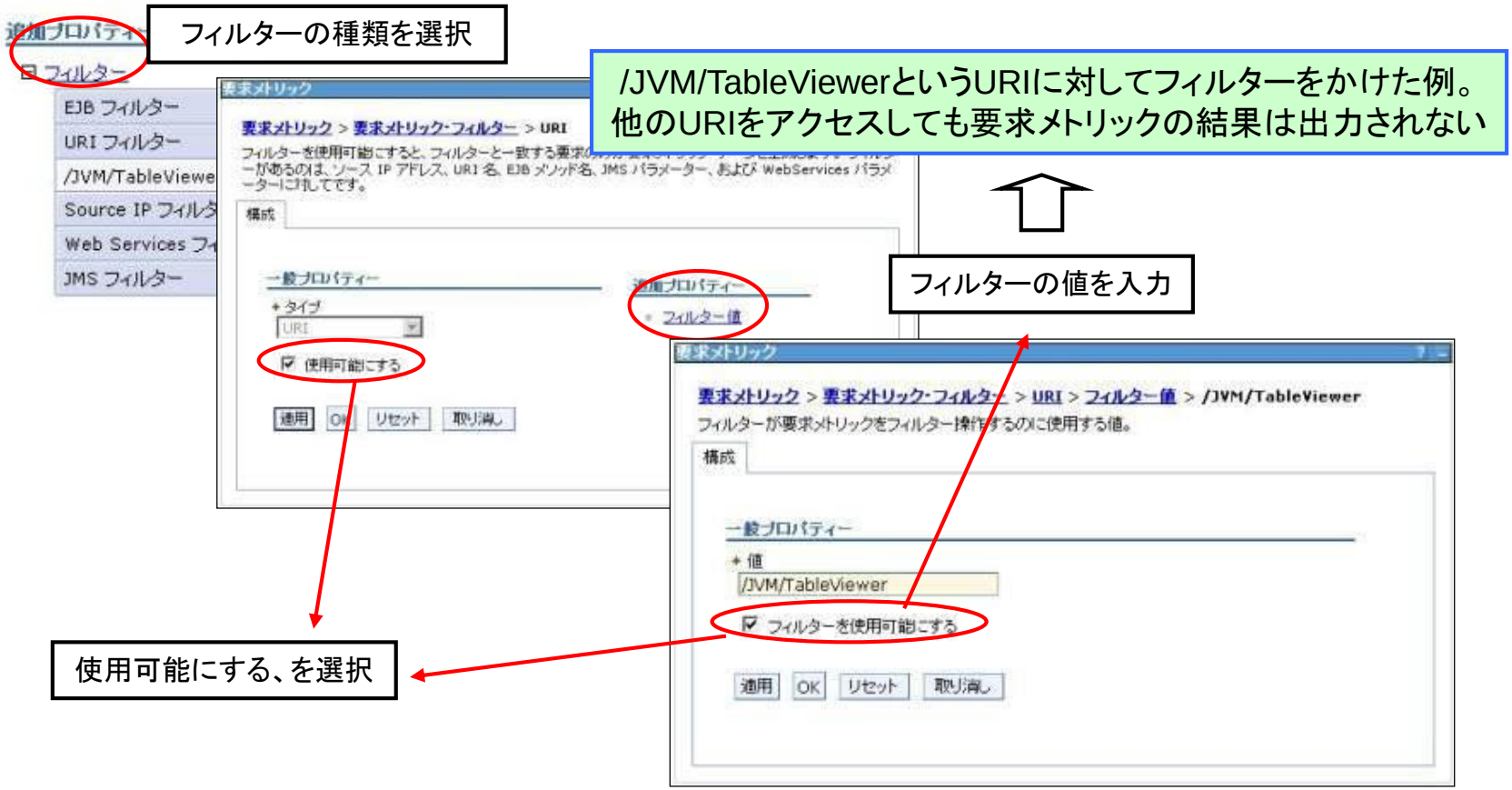
A	B	C	D	E	F
日付/時刻	parent	current	type	detail	elapsed
[09/03/15 14:19:07:266 JST]	parent:ver=1,ip=9.170.251.57,time=11	current:ver=1,ip=9.170.	type=JDBC	detail=SELECT * fro	3125
[09/03/15 14:19:08:172 JST]	parent:ver=1,ip=9.170.251.57,time=11	current:ver=1,ip=9.170.	type=URI	detail=/JVM/TableV	14313

リクエスト全てに書き込みを行うと、要求メトリックによる出力自体がパフォーマンスに影響を与える可能性があるので、フィルターにより対象となるリクエストのみ出力させる

© 2016 IBM Corporation

要求メトリックのフィルタリング機能

- 要求メトリックのフィルター
  - 要求メトリックの出力によるパフォーマンスダウンを抑えるために特定データのみを出力（EJB/URI/SourceIP/WebService/JMS）



© 2016 IBM Corporation

パフォーマンス・データ取得のポイント

- 1. パフォーマンス・テスト内容の明確化
  - キャパシティー・テスト
  - 限界値の確認
  - 高負荷状態での長時間耐久確認
- 2. 取得すべきデータおよび目標値の明確化
  - レスポンスタイム
  - スループット
  - システム・リソース
- 3. データ取得方法およびデータ分析方法の検討
  - OS/ミドルウェアの機能を使用
  - 使用実績のあるツールを使用
  - できるだけシステムのパフォーマンスに影響を与えないツールを使用

© 2016 IBM Corporation

WebSphere Application Server V9

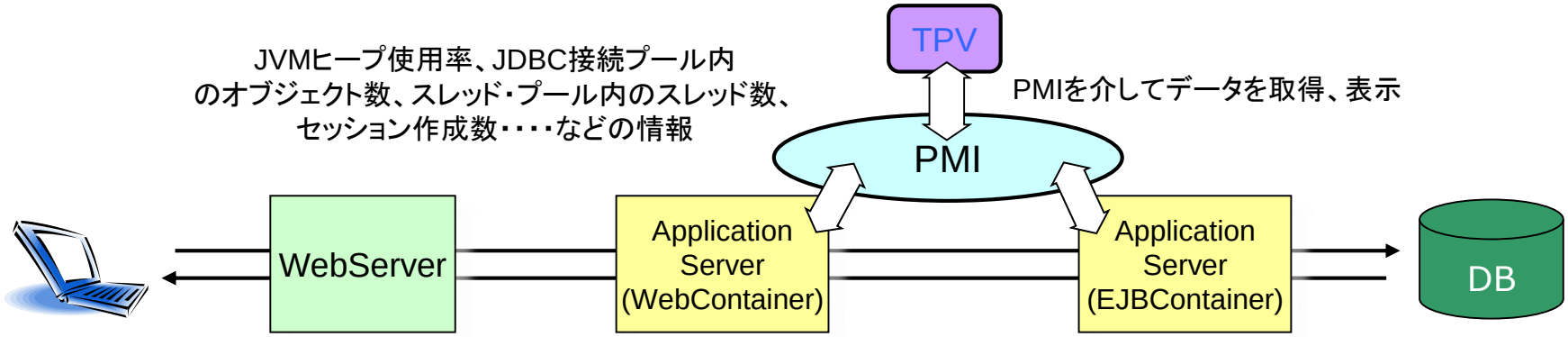
2. WASにおけるパフォーマンス管理

2-1　パフォーマンス・チューニング
2-2　パフォーマンス・データ取得
2-3　パフォーマンス監視

© 2016 IBM Corporation

PMI（Performance Monitoring Infrastructure）

- PMI（Performance Monitoring Infrastructure）
  - WASのコンポーネントのデータをモニターするインターフェース
  - パフォーマンスをモニターするツールはPMIを介してデータを取得して表示



デフォルトでOn



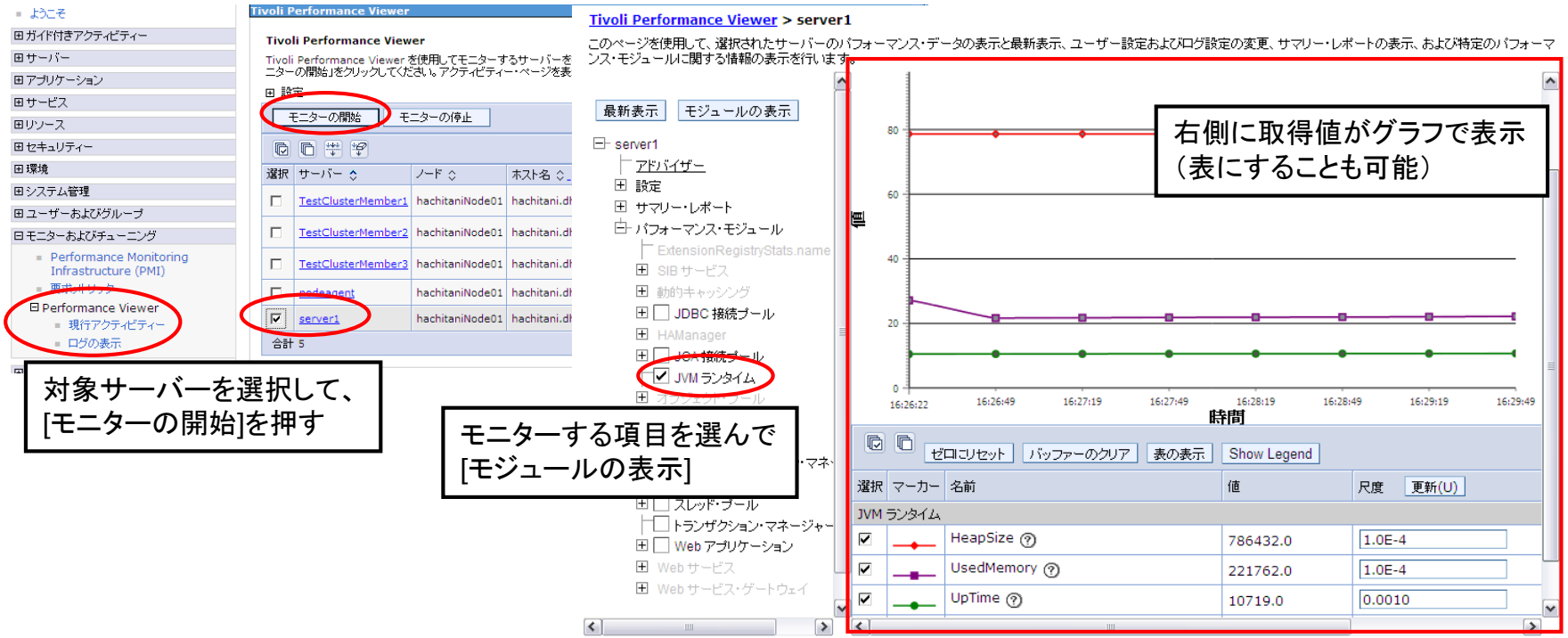
PMIで取得する情報のレベル
（統計セット）を「基本」、「拡張」、
「すべて」、「カスタム」から選択

「＋」をクリックすると、そのレベルで
取得できる情報の一覧が表示される

© 2016 IBM Corporation

Tivoli Performance Viewer

- Tivoli Performance Viewer
  - WASが提供しているパフォーマンス・モニター・ツール
  - 管理コンソールに統合
    - [モニターおよびチューニング]→[Performance Viewer]
  - リアルタイムでデータを表示
  - ログの取得と再生が可能



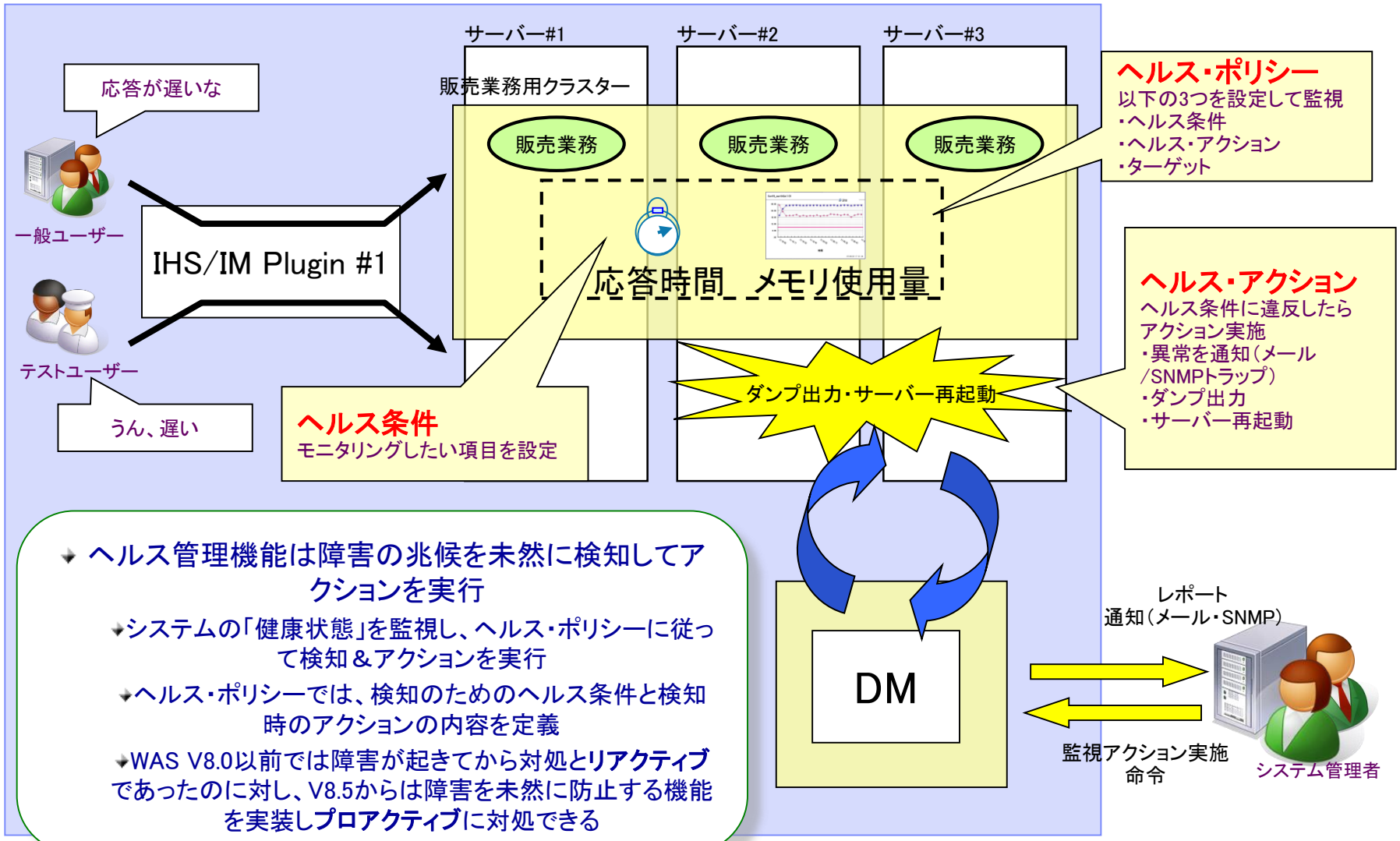
© 2016 IBM Corporation

Tivoli Performance Viewer で取得できるデータ

カテゴリー	取得情報
エンタープライズBean	応答時間やライフサイクルのアクティビティー、Enterprise Beanメソッドおよびエンタープライズ Beanによって使用されるリモート・インターフェースの情報
JDBC接続プール	新規に作成された接続数など、データソースの接続プールについての使用情報
JCA接続プール	JCAについての使用情報
JVMランタイム	使用メモリーなどJVMに関する情報
サーブレット・セッション・マネージャー	アクティブなセッションの平均数など、HTTP セッションの使用状況
スレッド・プール	オブジェクト・リクエスト・ブローカー (ORB) スレッドのスレッド・プール、HTTP要求を処理するWebコンテナー・プールについての情報
トランザクション・マネージャー	アクティブ・トランザクションの平均数など、トランザクションマネージャーのパフォーマンス情報
Webアプリケーション	ロードされたサーブレットの数、サーブレット要求の応答時間など、選択されたサーバーの情報
ORB	オブジェクト参照ルックアップ時間などORBに関する情報
ワークロード管理	IIOP要求の数など、クライアントとサーバーのワークロード管理に間する情報
動的キャッシング	メモリー内のキャッシュ・サイズなど、動的キャッシュ・サービスに関する情報
Webサービス	ロードされたWebサービスの数など、Webサービスに関する情報
Webサービス・ゲートウェイ	同期要求の数など、Webサービス・ゲートウェイに関する情報
システム・データ	CPU使用率など、マシン(ノード)に関する情報（複数のサーバー・バージョンのノード・エージェントでのみ使用可能）
アラーム・マネージャー	アラームに関する情報
オブジェクト・プール	オブジェクト・プールに関する情報
スケジューラー	スケジューラー・サービスに関する情報
HA マネージャー	HAマネージャーに関する情報
DCS統計	DCSスタックを使用した送信メッセージ数など、DCSに関する情報
SipContainerModule	アプリケーション・サーバーのSIPContainerni関する情報
ポートレット・アプリケーション	ポートレット・アプリケーションに関する情報
SIBサービス	SIBサービスに関する情報

© 2016 IBM Corporation

ヘルス管理とは：障害の兆候検知と復旧



- ヘルス管理機能は障害の兆候を未然に検知してアクションを実行
  - システムの「健康状態」を監視し、ヘルス・ポリシーに従って検知＆アクションを実行
  - ヘルス・ポリシーでは、検知のためのヘルス条件と検知時のアクションの内容を定義
  - WAS V8.0以前では障害が起きてから対処と**リアクティブ**であったのに対し、V8.5からは障害を未然に防止する機能を実装し**プロアクティブ**に対処できる

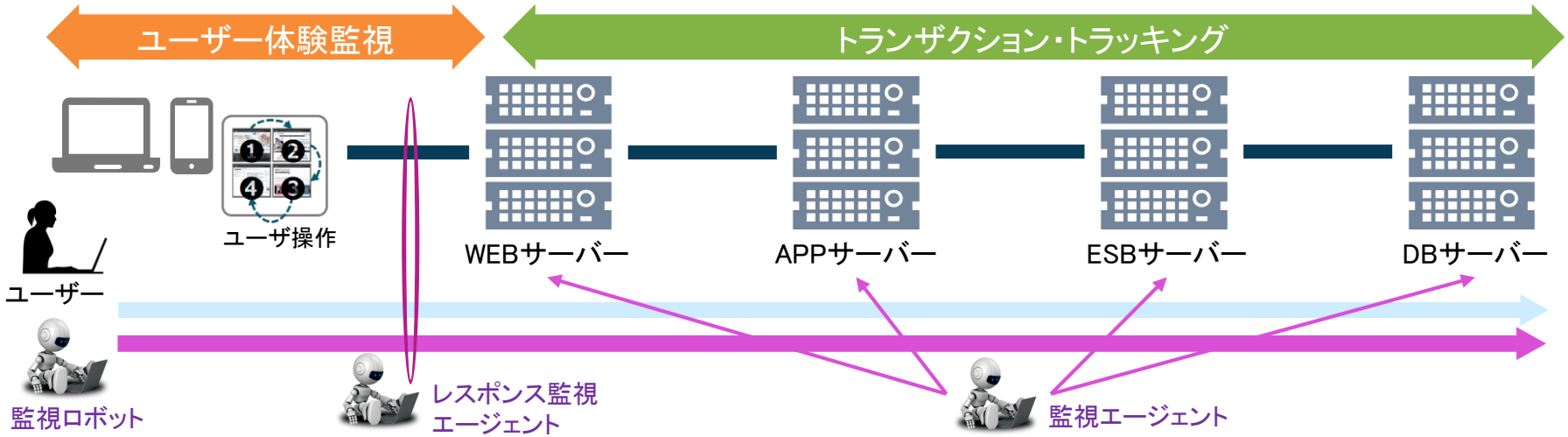
© 2016 IBM Corporation

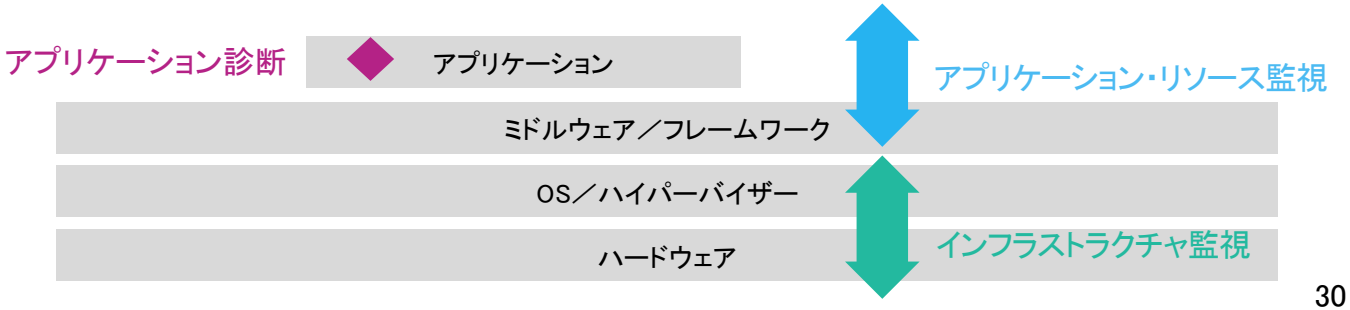
2. システム全体の統合管理・監視

© 2016 IBM Corporation

IBM Performance Management（IPM）

- IT環境を360°監視する 包括的なアプリケーション性能管理ソリューション
  - オンプレミス上のプラットフォーム（OSや仮想化インフラ）のリソース監視だけでなく、パブリック・クラウド環境で稼動するミドルウェアも含め、様々なコンポーネントの可用性や性能情報を一元的に監視
  - リソース監視だけでなくユーザーが体感しているアプリケーション性能を、監視ロボットやレスポンス監視エージェントを用いて監視





© 2016 IBM Corporation

オファリング毎の機能とエージェントの違い (1/2)

IBM Performance Management

Agent / Feature	IBM APM Advanced	IBM APM	IBM Monitoring	IBM App Diag
Features	**All features** – The best option for managing your critical applications in production	• Transaction tracking • End User Experience • IBM Monitoring features	• Application-Aware Infrastructure Monitoring for huge array of infrastructure & resources • Agent Builder	• Find performance bottlenecks in application code • IBM Monitoring features
Agents				
Monitoring Agent for MongoDB	✓	✓	✓	✖
Monitoring Agent for MySQL	✓	✓	✓	✖
Monitoring Agent for Node.js	✓	✓	✓	✓
Monitoring Agent for Oracle Database	✓	✓	✓	✖
Monitoring Agent for PHP	✓	✓	✓	✖
Monitoring Agent for PostgreSQL	✓	✓	✓	✖
Monitoring Agent for Python	✓	✓	✓	✖
Monitoring Agent for Ruby	✓	✓	✓	✓
Monitoring Agent for SAP Applications	✓	✓	✖	✖
Monitoring Agent for Tomcat	✓	✓	✓	✖
Monitoring Agent for UNIX OS	✓	✓	✓	✓
Monitoring Agent for VMware VI	✓	✓	✓	✖
Monitoring Agent for WebSphere® Applications	✓	✓	✓	✓
Monitoring Agent for WebSphere Infrastructure Manager	✓	✓	✓	✓
Monitoring Agent for WebSphere MQ	✓	✓	✖	✖
Monitoring Agent for Windows OS	✓	✓	✓	✓
Response Time Monitoring Agent	✓	✓	✓	✖

© 2016 IBM Corporation

オファリング毎の機能とエージェントの違い (2/2)

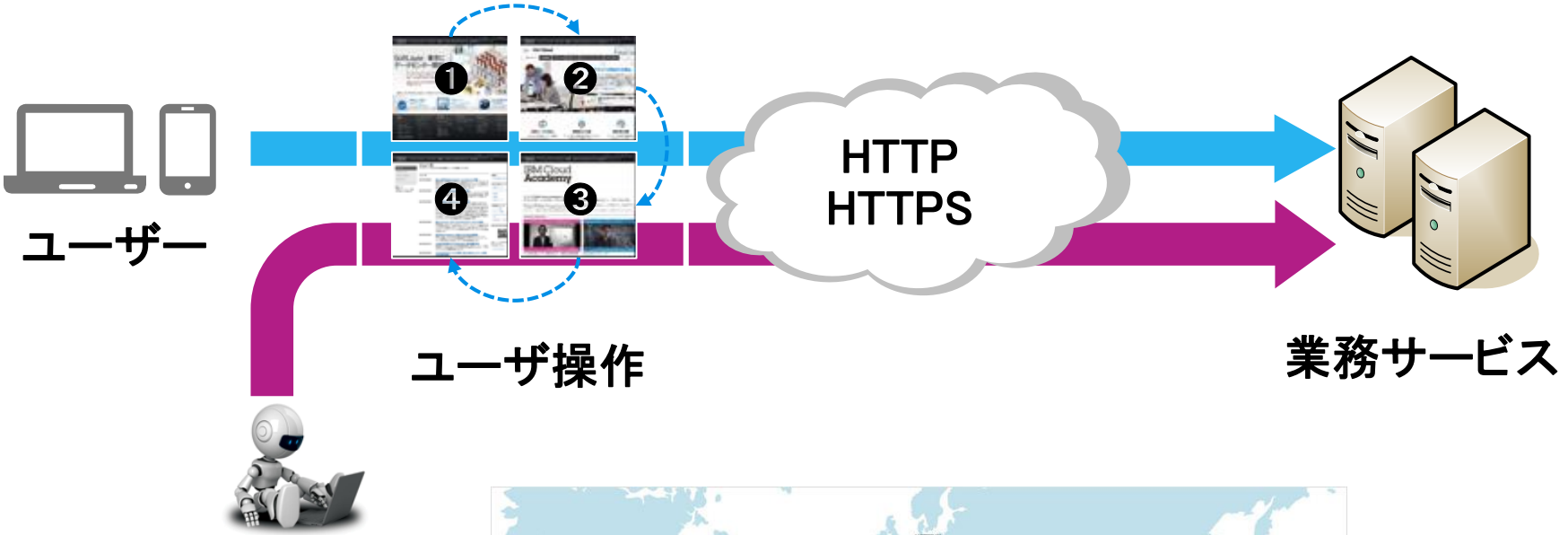
IBM Performance Management

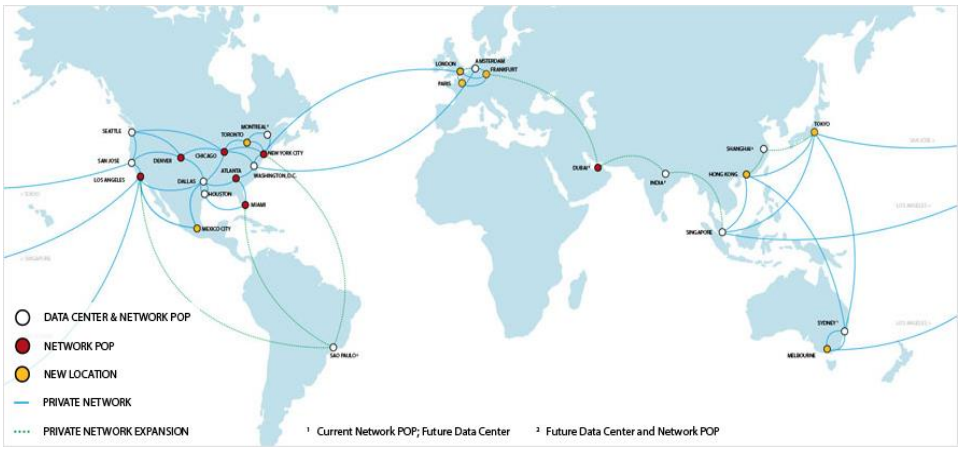
Agent / Feature	IBM APM Advanced	IBM APM	IBM Monitoring	IBM App Diag
Features	**All features** – The best option for managing your critical applications in production	• Transaction tracking • End User Experience • IBM Monitoring features	• Application-Aware Infrastructure Monitoring for huge array of infrastructure & resources • Agent Builder	• Find performance bottlenecks in application code • IBM Monitoring features
Agents				
Monitoring Agent for Cisco UCS	✓	✓	✓	✖
Monitoring Agent for DataPower®	✓	✓	✓	✖
Monitoring Agent for DB2®	✓	✓	✓	✖
Monitoring Agent for HMC Base	✓	✓	✓	✖
Monitoring Agent for HTTP Server	✓	✓	✓	✖
Monitoring Agent for IBM Integration Bus	✓	✓	✖	✖
Monitoring Agent for JBoss	✓	✓	✓	✖
Monitoring Agent for Linux KVM	✓	✓	✓	✖
Monitoring Agent for Linux OS	✓	✓	✓	✓
Monitoring Agent for Microsoft Active Directory	✓	✓	✓	✖
Monitoring Agent for Microsoft Cluster Server	✓	✓	✓	✖
Monitoring Agent for Microsoft Exchange Server	✓	✓	✓	✖
Monitoring Agent for Microsoft Hyper-V Server	✓	✓	✓	✖
Monitoring Agent for Microsoft Internet Information Services	✓	✓	✓	✖
Monitoring Agent for Microsoft .NET	✓	✓	✓	✓
Monitoring Agent for Microsoft SharePoint Server	✓	✓	✓	✖
Monitoring Agent for Microsoft SQL Server	✓	✓	✓	✖

© 2016 IBM Corporation

ユーザー体験監視

世界各地のSoftLayer上にホストされたサーバーからリクエストを送信し、
合成トランザクションでサービスの正常性と性能を24時間365日監視



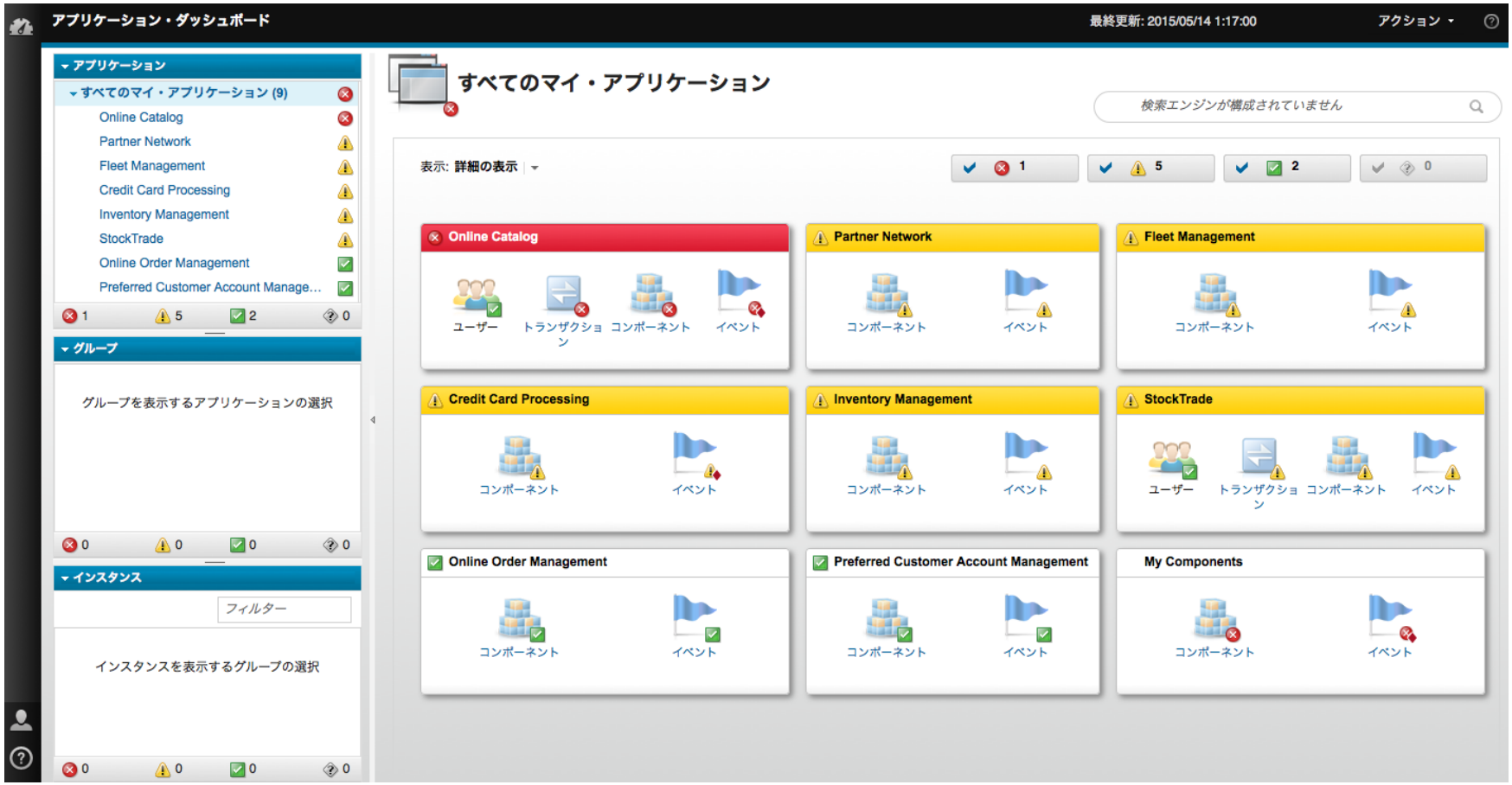


ダラス
ワシントンDC
サンノゼ
トロント
アムステルダム
メルボルン
シンガポール
香港
東京

© 2016 IBM Corporation

ダッシュボード

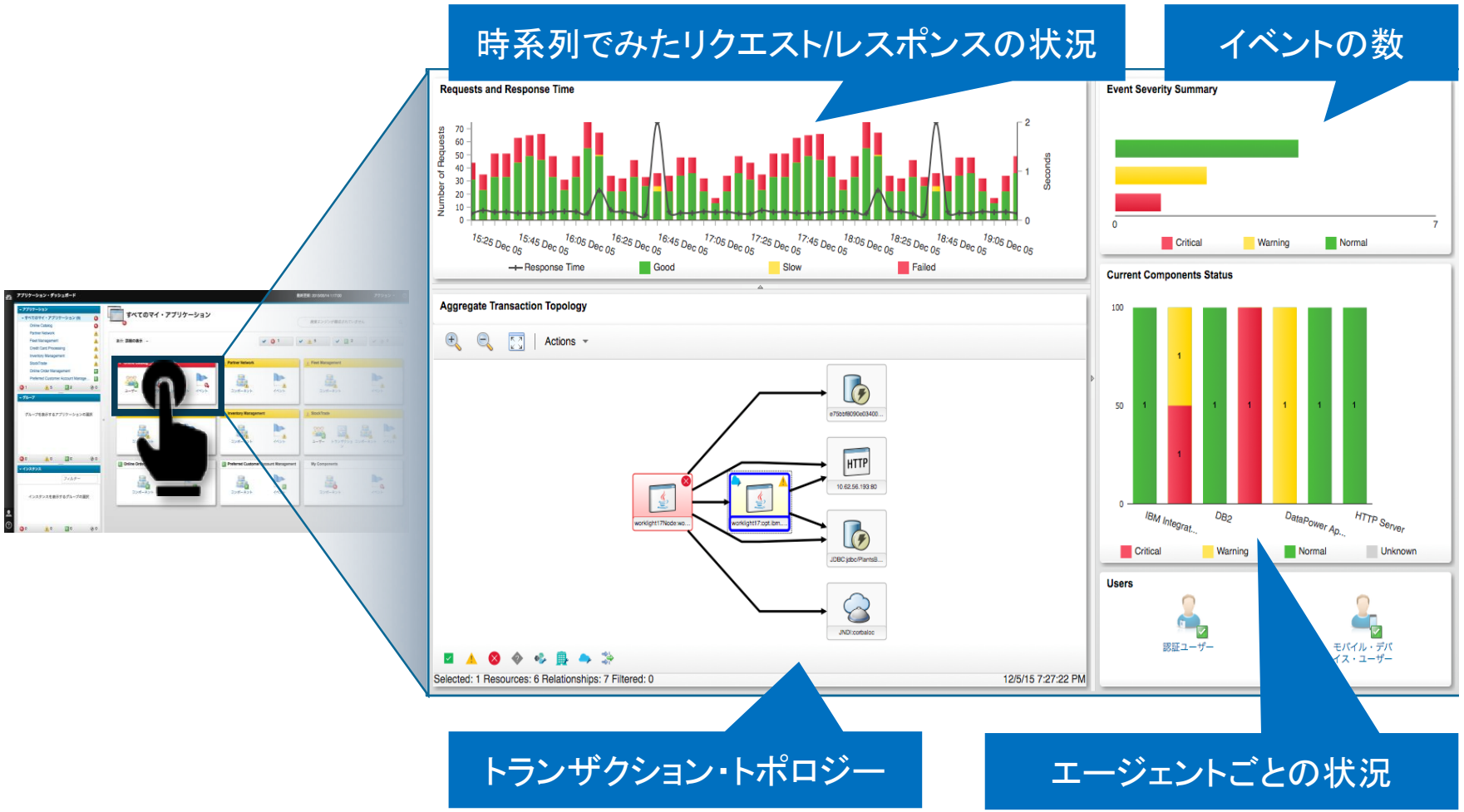
サービスの状態をひと目で把握できるダッシュボード



© 2016 IBM Corporation

アプリケーションのステータス概要

個別のアプリケーション毎のリクエスト数、応答時間、発生イベント、
トランザクション・トポロジーがわかる



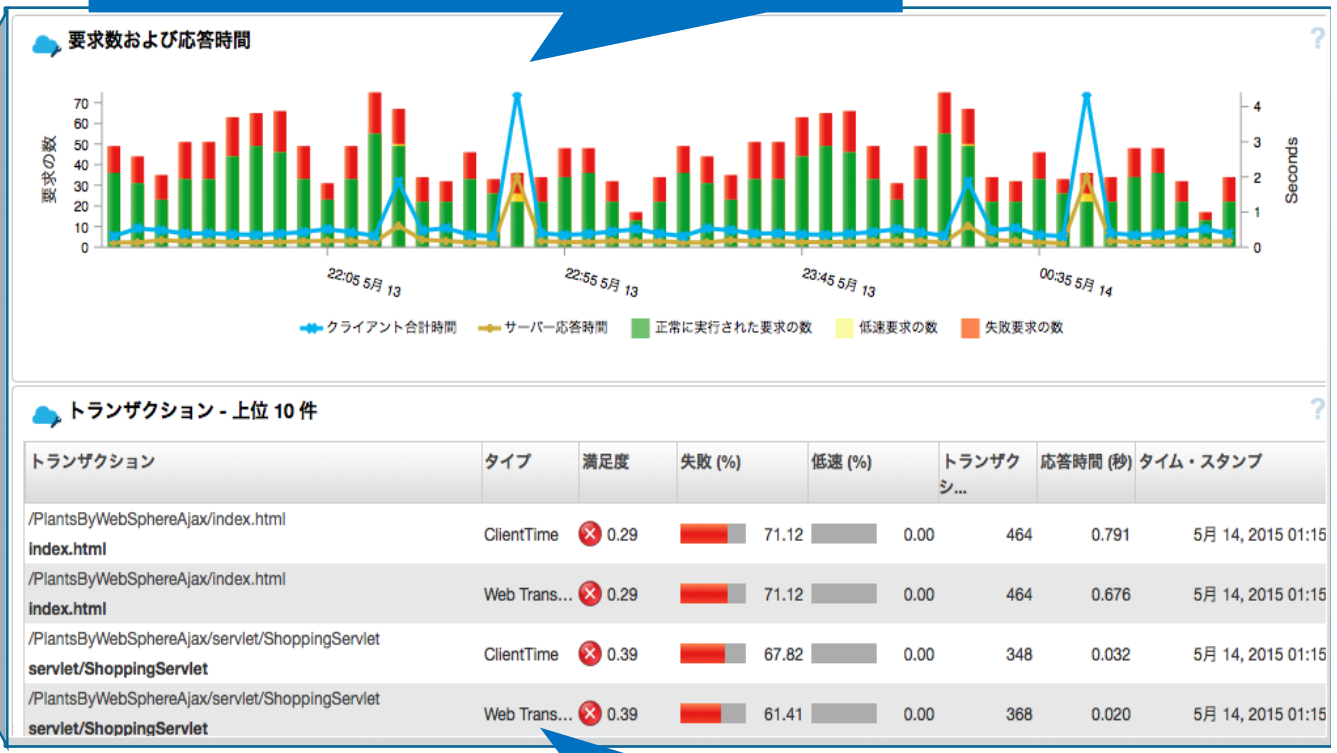
© 2016 IBM Corporation

トランザクションの要求数と応答時間

トランザクションのリクエスト数と応答時間、
個別のページ・トランザクションの失敗率や応答時間がわかる

時系列でみたリクエスト/レスポンスの状況





失敗率の高いトランザクション上位10件

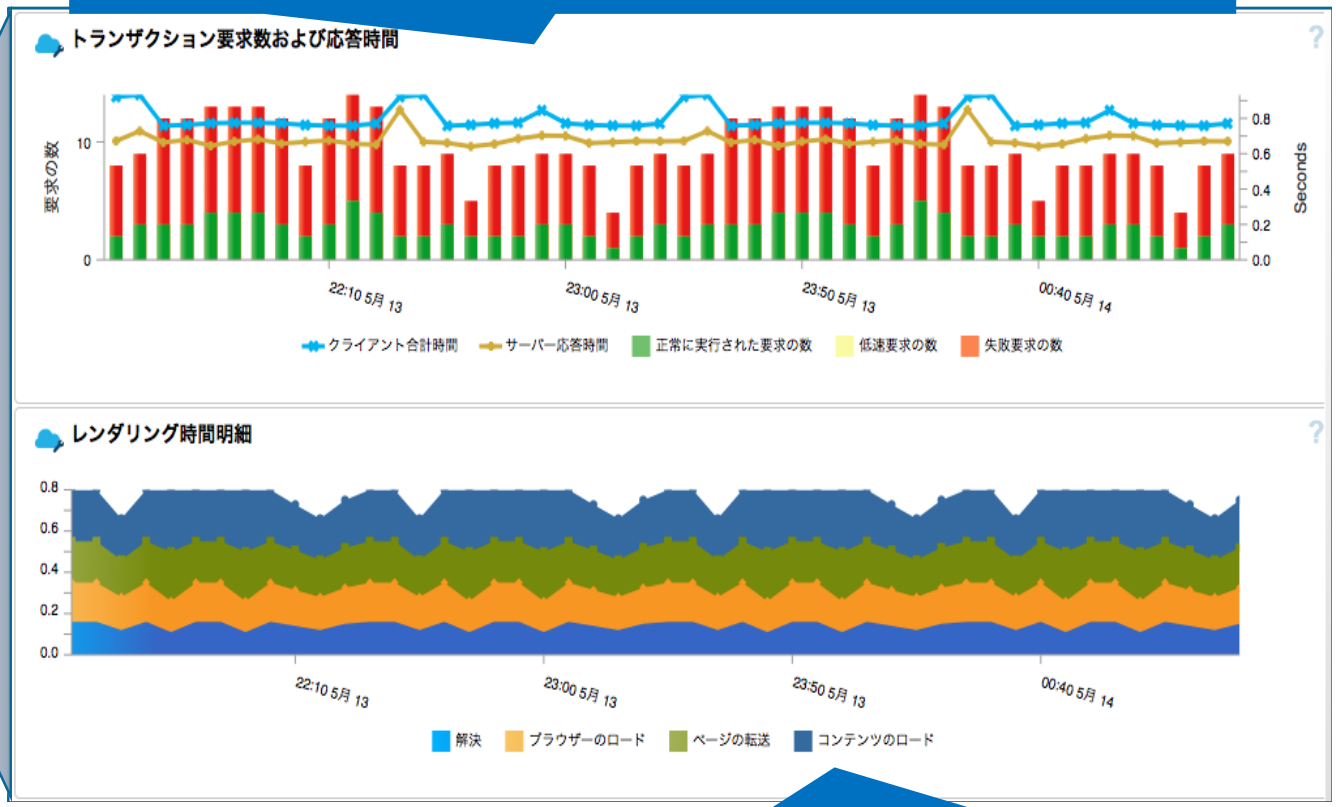
© 2016 IBM Corporation

ページ・トランザクションの詳細

個別のページ・トランザクション毎のリクエスト数、応答時間、
ページのレンダリング時間の内訳がわかる

クライアント側、サーバー側のリクエスト数とレスポンスタイム



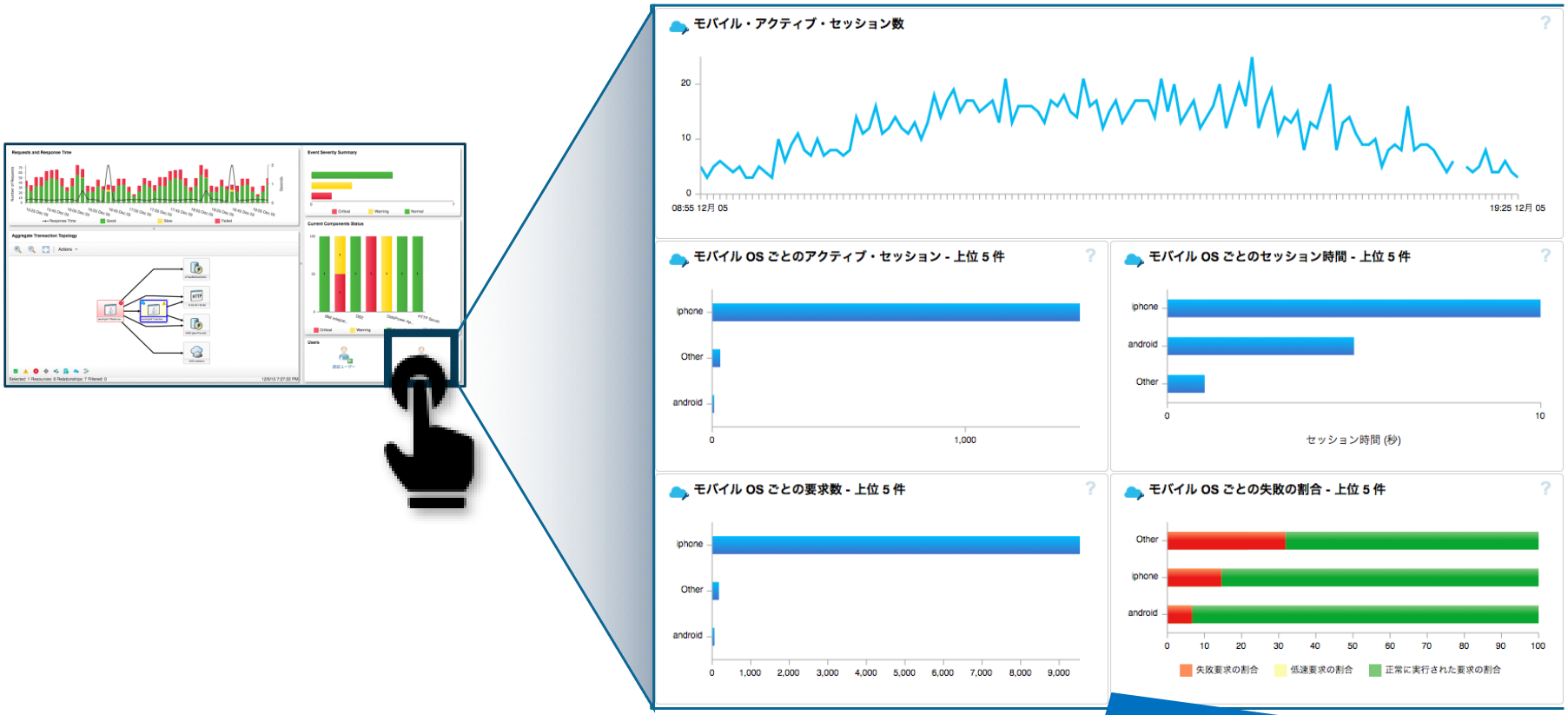


ブラウザのロード、ページ転送、コンテンツのロードの
レンダリングに要した時間

© 2016 IBM Corporation

モバイル・デバイスからのアクセスの状態

モバイル・デバイスからのアクセスのセッション数や、
OS毎のセッション数、リクエスト数、成否の割合がわかる



モバイルOSごとのリクエスト数やセッション数、失敗の割合

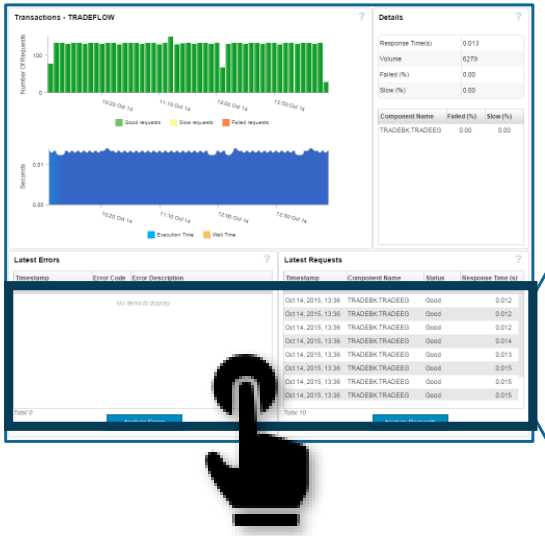
© 2016 IBM Corporation

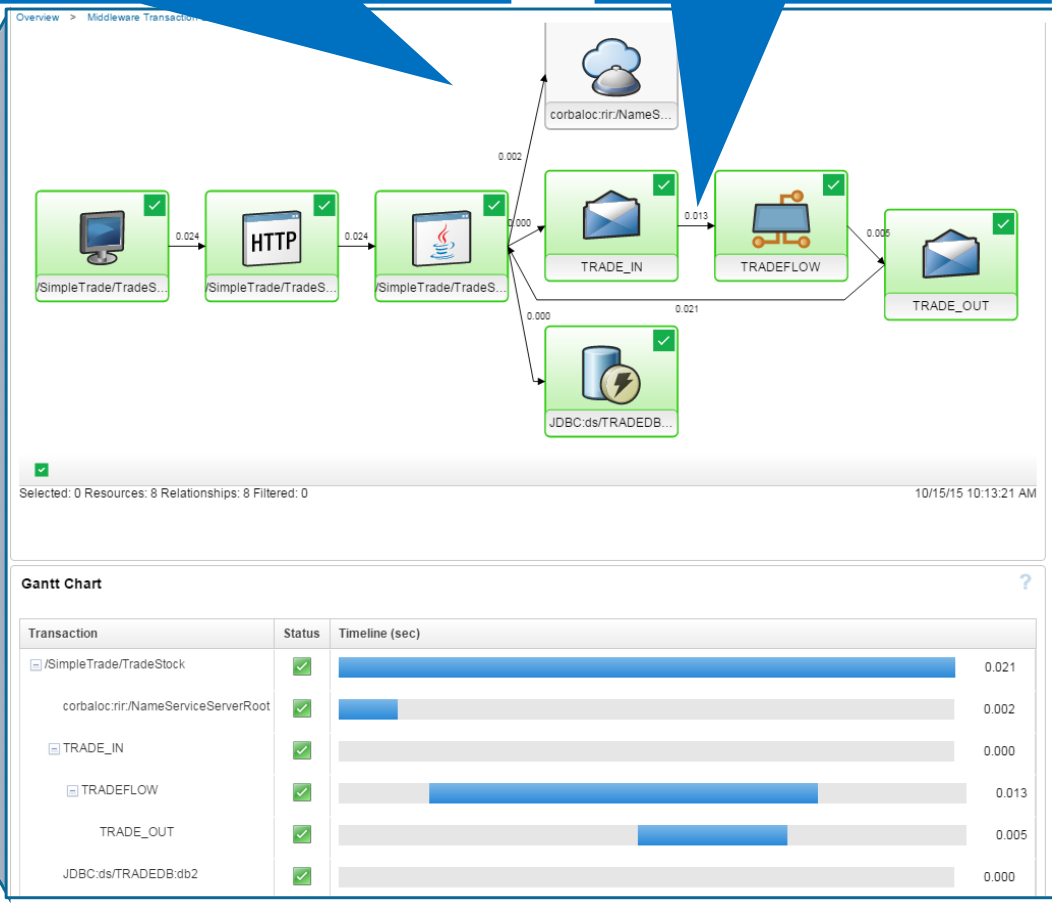
トランザクション・トラッキング

個別のリクエストのトポロジーと処理に要した時間がわかる

個別リクエストのトランザクション・トポロジー

かかった時間

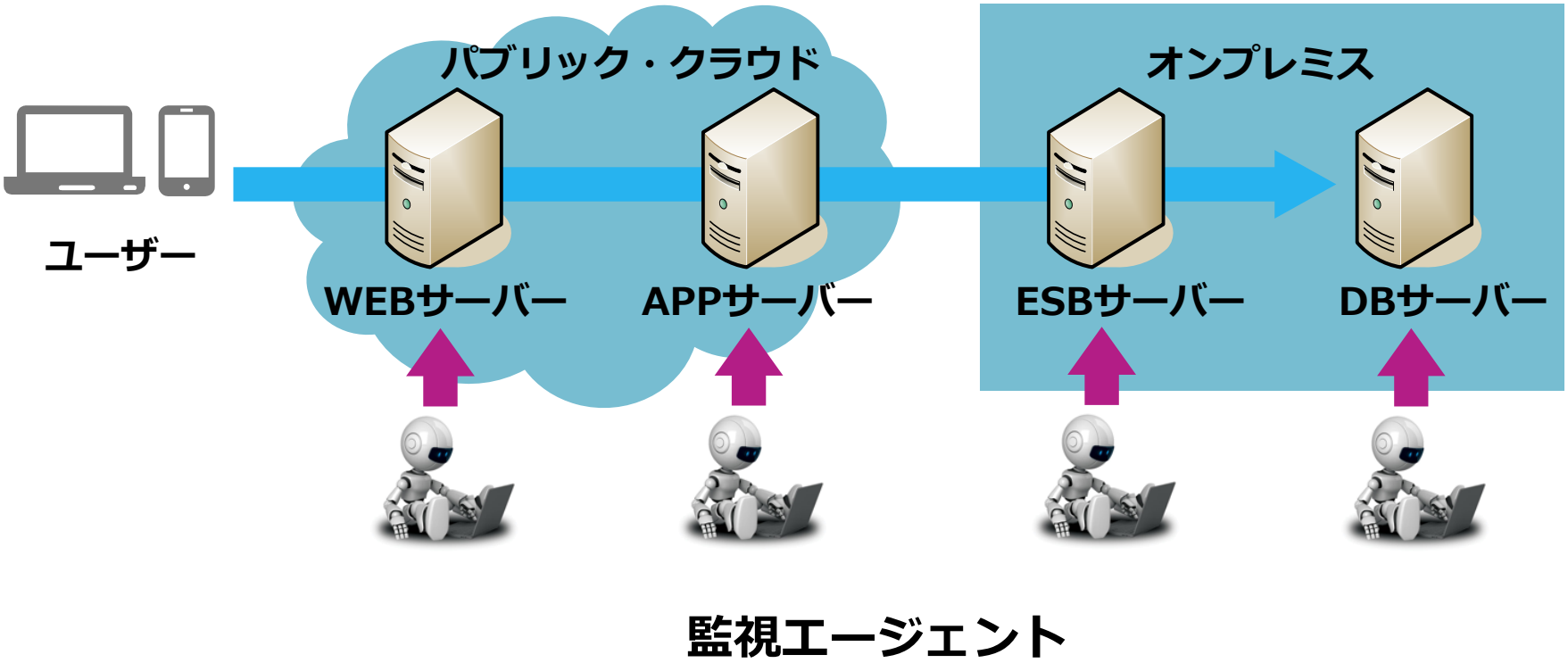




© 2016 IBM Corporation

アプリケーション・リソース監視

ハイブリッド・クラウド環境でもOS、ミドルウェアの可用性や性能情報を網羅的に監視可能

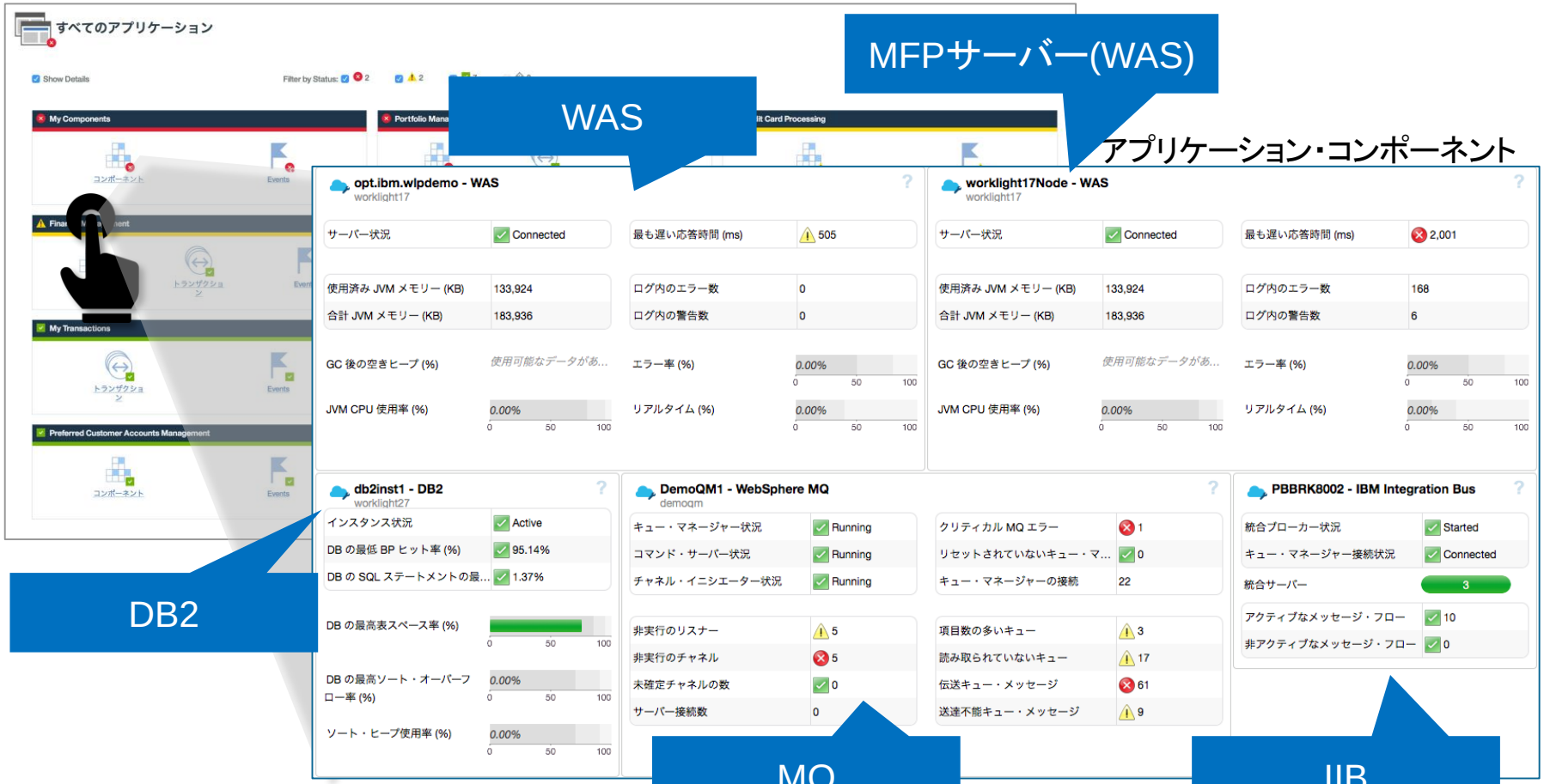


© 2016 IBM Corporation

アプリケーション・コンポーネント概要

WebSphere だけでなく
サービスを構成するDB、OSなどのコンポーネントの状態がわかる

アプリケーション全体を把握できるダッシュボード



© 2016 IBM Corporation

アプリケーション・リソース（WAS）1/2

WAS のモニター機能が提供する
GC、EJBコンテナ、スレッドプールなどの状態を見やすくグラフ化

レスポンスが最も遅かったリクエスト

WASのサーバー名、セル名などの情報





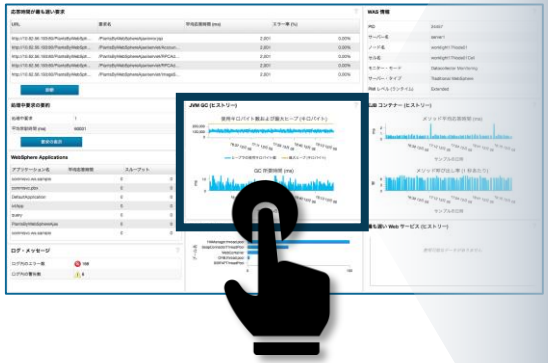
使用率の高いスレッドプール

時系列のGC所要時間

© 2016 IBM Corporation

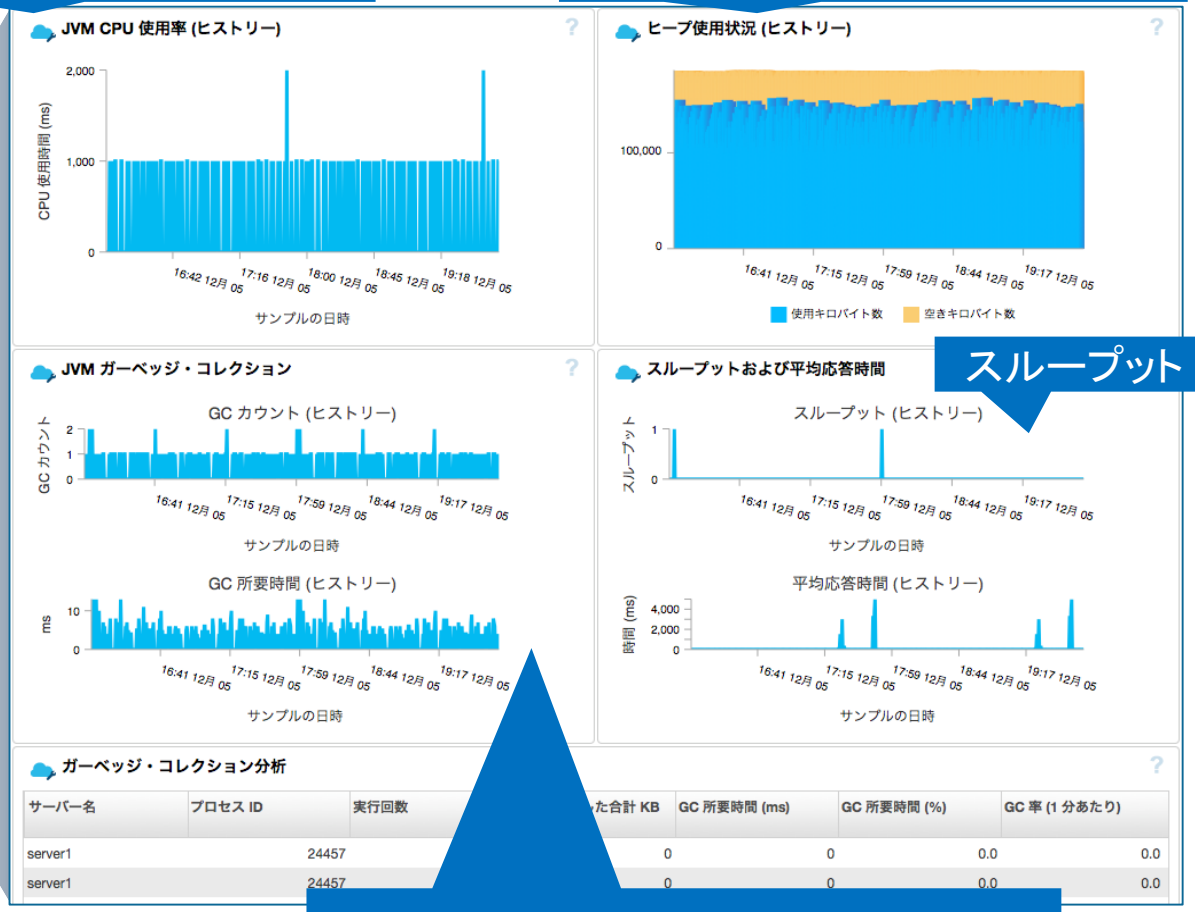
アプリケーション・リソース（WAS）2/2

JVMのヒープ使用状況、GC実行回数・所要時間などの詳細な状態もわかる



JVMのCPU 使用率

ヒープ使用状況



スループット

時系列のGC所要時間

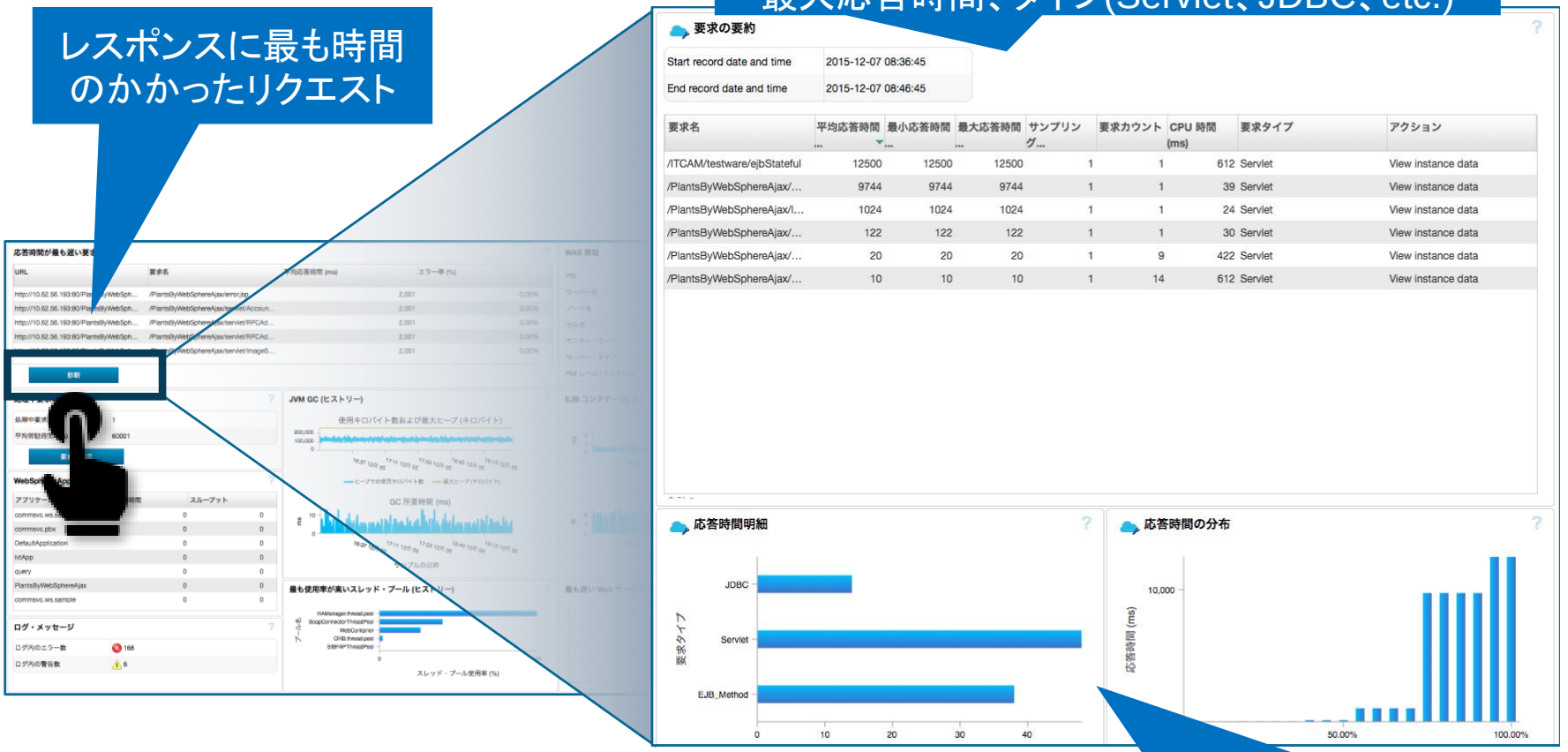
© 2016 IBM Corporation

アプリケーション診断（WAS）

実行された個別のリクエスト毎の応答時間や実行回数などがわかる

リクエストごとの平均応答時間、最小応答時間、最大応答時間、タイプ(Servlet、JDBC、etc.)

レスポンスに最も時間のかかったリクエスト



JDBC、Servlet、EJBメソッドごとのレスポンスタイム

© 2016 IBM Corporation

アプリケーション診断（WAS スタックトレース）

リクエストのシーケンス、スタックトレース、実行したSQL文などがわかる

リクエストのシーケンス



要求シーケンス - 1

順序 ID	深さ	イベント名	開始日時	イベント・タイプ	応答時間 (ms)	イベント・タイプ	CPU 時間 (ms)
1	1	/ITCAM/testware /ejbStateful	Dec 6, 2015 5:46:05 PM	Servlet	12612	Servlet	600
2	2	TextStringStatefulBean.fi	Dec 6, 2015 5:46:08 PM	EJB_Method	10000	EJB_Method	12
3	3	executeQuery	Dec 6, 2015 5:46:12 PM	JDBC	6250	JDBC	0

要求コンテキスト - executeQuery

名前	値
requestType	JDBC
requestName	executeQuery
dataSourceName	JDBC:jdbc/PlantsByWebSphereAjaxDat…
sqlText	CALL QATEST(?)

要求スタック・トレース - executeQuery

完全修飾メソッド名
org.apache.derby.impl.jdbc.EmbedPreparedStatement.execute:0
com.ibm.tivoli.itcamdemo.database.handleRequest:478
com.ibm.tivoli.itcamdemo.database.handleRequest:178

executeQuery()の詳細

リクエストのスタックトレース

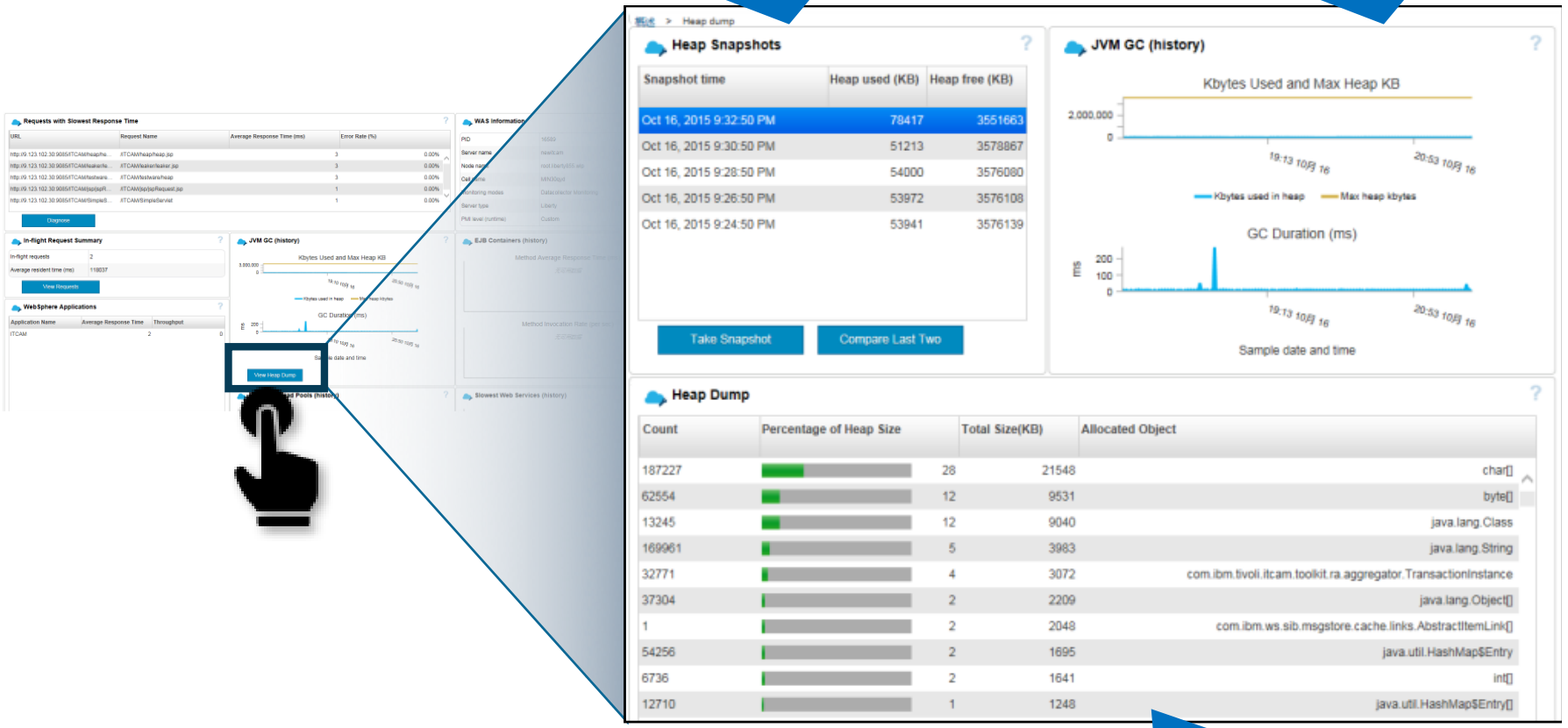
© 2016 IBM Corporation

アプリケーション診断（WAS ヒープダンプ）

ヒープダンプを取得し、参照や比較ができる

ヒープのスナップショット

時系列のGC間隔



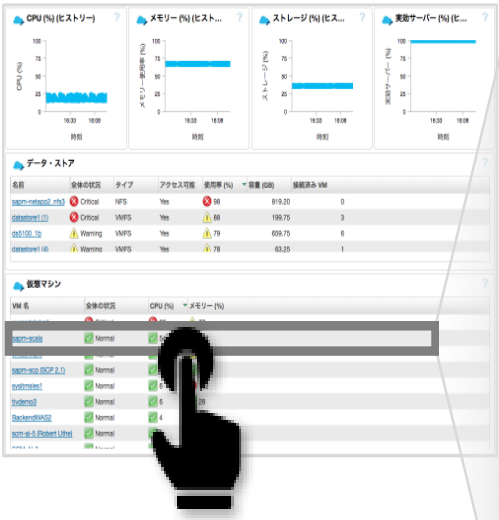
ヒープダンプ

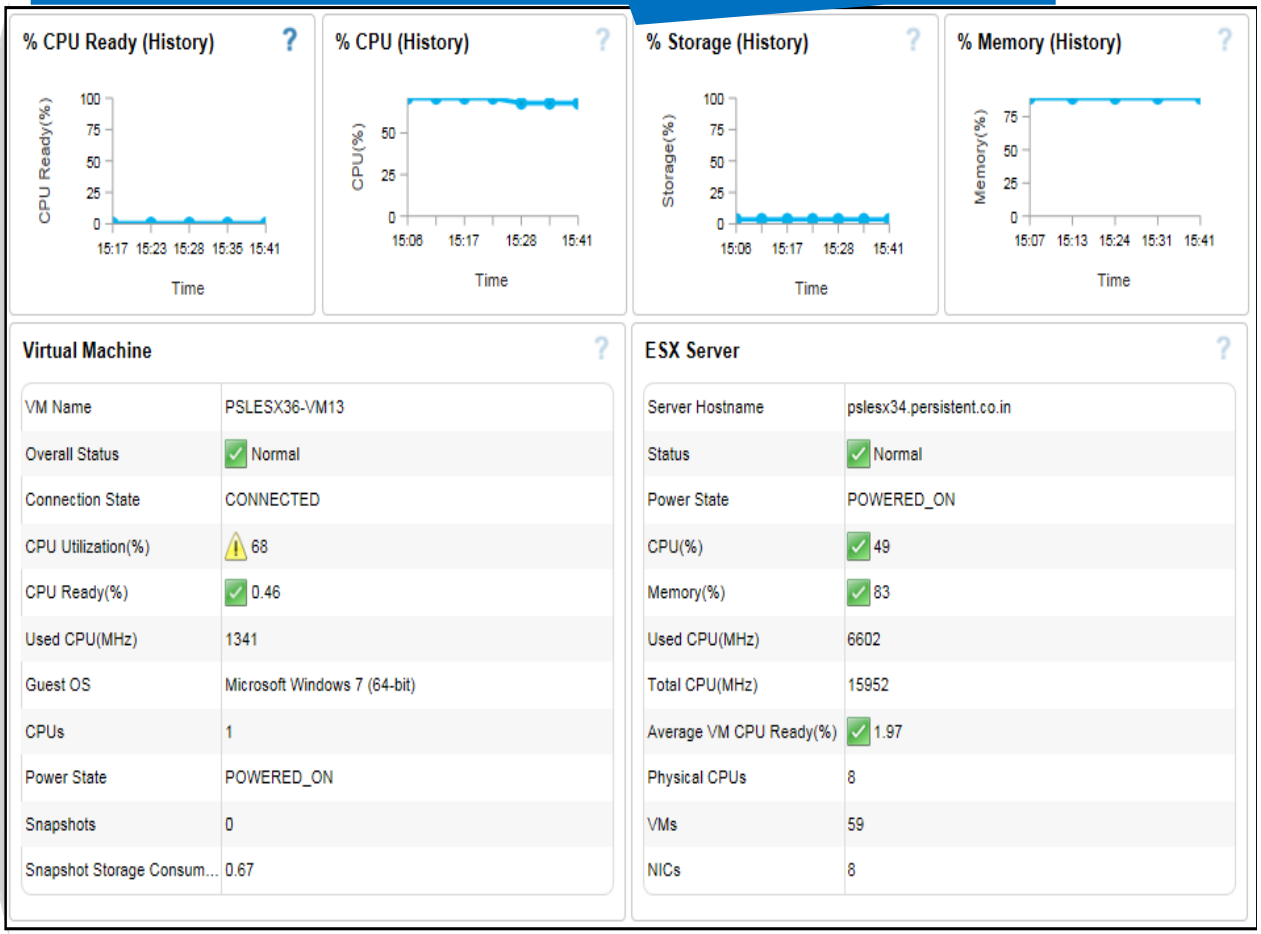
© 2016 IBM Corporation

インフラストラクチャ・リソース（VMware 概要）

仮想マシンのリソース使用状況がわかる

CPU、ストレージ、メモリなどの使用率





© 2016 IBM Corporation

エージェントの導入・構成

- 例えば、WASの場合 (Linux)
  - WASをインストールしているOSに対して、エージェント導入を行う
    - エージェント: Monitoring Agent for WebSphere Applications
  - 手順は簡単 3ステップ

1. エージェント導入用のtarファイルを展開する
2. 任意のエージェントを選択
   - $ ./installAPMAgent.sh を実行し、インタラクティブ形式で選択
3. データ・コレクターを構成する
   - $ ./simpleconfig.sh

  - これだけで、IPMサーバー上で、WAS監視が可能

Free Trial

IBM Performance Management (SaaS) はこちらから ==>
https://www.ibm.com/marketplace/cloud/application-performance-management/us/en-us

© 2016 IBM Corporation

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律的またはその他の指導や助言を意図したものではなく、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、「現状のまま」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だすことを意図したものでも、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もなく、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も、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をご覧ください。

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。

© 2016 IBM Corporation