



ホワイトペーパー(和訳版)

## クラウド・コンポーネントの世界における アプリケーション指向監視

Application-Oriented Monitoring in a Cloud Component World

トム・ルビンスキ著

2011年2月発行

### 要約:

「我が社の重大なソフトウェア・アプリケーションの性能は実際どうなのだろうか」という質問に答えることは、この10年の間にますます難しくなってきました。コンポーネント化とサービス指向アーキテクチャによってアプリケーション開発は平易になりましたが、それに応ずるように複雑さ、アプリケーション環境内での依存の数が増え、重大なアプリケーションの健全性の状態監視を極めて煩雑にしています。

クラウド革命はITコスト削減で大いに期待されていますが、初期においては新たなる統合ポイントやインタフェースによってアプリケーション監視はさらに困難になります。本書では、コンポーネント・アーキテクチャをベースにする複雑なアプリケーションの監視に固有の課題を取り上げ、最終的にはクラウドそのものの性質によっていかに適切に監視できるかを、議論します。

---

## 目次

アプリケーション指向監視 .....	3
コンポーネント指向アプリケーションを監視する難題 .....	3
クラウド革命がコンポーネント化を推進 .....	4
リアルタイムでのアプリケーション健全性ステートとデータ・フロー .....	5
構成データからの自動化監視 - 聖杯 .....	7
SL 社について .....	8

### 著作権

© Sherrill-Lubinski Corporation. All Rights Reserved.

### 商標

SL Corporation, SL-GMS, RTView, SL のロゴは米国ならびに各国での Sherrill-Lubinski Corporation の商標あるいは登録商標です。

---

## アプリケーション指向監視

今日の大企業は、全社が依存する重大アプリケーションの可用性と性能を維持することがいかに重要かを、極めてよく理解しています。たくさんの仕事が自動化され、かつ、あるいはオンラインで行われているため、たった一回の短い停止でも何百万ドル(何億円)もの収益が失われます。

大企業でこうした価値を提供しているアプリケーションは通常とても複雑で、多数の独立したサービス・コンポーネントを含んでおり、それらがメッセージング・システムや、アプリケーションやウェブ・サービス技術あるいは単なるファイル転送を使って、データをやり取りしています。これらのアプリケーションが共有しているサブシステムへの依存はこの10年間に顕著に増えており、ベースとなっているコンポーネントを共有している任意のアプリケーションの性能を知ることは、ますます難しくなっています。

とても大きな難題であるため、社内の開発リソースを使ってこれを解決できる企業は多くありません。その結果、この問題に取り組もうとする商用のアプリケーション性能監視(APM)ソリューションが激増することになります。しかし、これらのソリューションの多くはベンダー特有で、特定のコンポーネントに限られたり、ネットワーク・トラフィックのプロファイルといった1つの側面のみに取り組むものです。

本書の著者と SL 社は、リアルタイムな可視化と監視のアプリケーションで25年以上の経験があり、特に Java には専門知識を有しています。SL 社の RTView 製品はコンポーネント・ベースのアプリケーションの性能監視を取り巻く広範囲な要求条件に応えるために開発されたもので、特に共有サービスに依存する重要なアプリケーションの健全性の状態を可視化して把握し、管理することに重点を置いています。

アプリケーションをサポートするグループは担当しているアプリケーションを監視する必要があり、同じコンポーネント・サービスを共有している他のアプリケーションについては気にも掛けません。よって効果的な監視には、多数のサブシステムからの情報を一括化し、かつそれをアプリケーション指向で表示することが欠かせません。

事態をさらに複雑にするのは、IT 管理にまつわるコスト削減で有望視されている、アプリケーションの「クラウド」運用です。やがてそうなることではありますが、大きな変化には時間がかかるものです。これらのアプリケーションは本来複雑なので、移行は段階的に行われることになります。つまり、監視しなくてはならない「統合ポイント」=「レガシー・コンポーネントと新しいクラウド・ベースのモジュールとの間のデータ転送でエラーが起こり得る場所」が、さらに多くなることを意味します。

## コンポーネント指向アプリケーションを監視する難題

この10年間、アプリケーションが洗練されて複雑になり、さらにコンポーネント指向になるにつれて、アプリケーション監視の難しさは増すばかりです。アプリケーション・サーバ、エンタープライズ・ビーン、メッセージング・ミドルウェア、さらに最近ではインメモリ分散キャッシング・システムなどの共有サービス・コンポーネントにより、アプリケーション開発者の生産性は大いに向上してきました。しかし同時にこれらに頼れば頼るほど、アプリケーションが実際にどう動いているのかを理解するのはますます難しくなっています。

---

メッセージング・ミドルウェアがよい例です。Oracle Fusion Middleware, TIBCO EMS, IBM MQ は、アプリケーション・サブシステム間で重要なデータを確実に転送するため、たくさんの企業組織に使われています。あちこちにある何十ものアプリケーションが同じメッセージ・ブローカを使ってデータを転送することもあるかもしれません。そのうちのいずれかがブローカに負荷をかけ過ぎてしまった場合、すべてのアプリケーションに影響を及ぼします。誰が犯人で誰がその悪影響を受けたのかを判断するのは簡単ではありません。

さらに悪いことに、リソースを共有しているため、1つのアプリケーションでトラブルがあっても共有リソースにはアラートがトリガされないかもしれません。普段は2秒で済むアプリケーションのジョブ完了に2分もかかっていると、ユーザが悲鳴を上げてサポート・チームに連絡してきて、初めて問題があるとわかることは良くあることです。

どのアプリケーションがどのリソースを使っているのかは誰も知らないため、結果はおなじみの「喧々諤々」で、真夜中に全員が呼び出されて何が悪いのかを調べることになります。企業組織にはメジャーな監視ツールを一通りすべて揃えているかもしれませんが、ほとんどは連携しない単独な項目に限定されており、1つのコンポーネントについてのデータを示すだけです。全員が右往左往し、すべてのデータが理にかなうようにしようと、最終的に問題の原因を見つけるのは何時間も後になることがあります。

RTView は、この原因究明に特化して設計されたもので、コンポーネント・サブシステムから収集した健全性状態の情報を、中央集約化、組織化されたアプリケーション指向式で表示します。

ですがその前に、クラウドについて少々…

## クラウド革命がコンポーネント化を推進

「クラウド革命」についての話は混乱を招くことがあります。この用語は人によって意味が異なります。ある人たちは Amazon や ebay などの企業が 90 年代から「クラウド」だったと主張し、クラウドが新しいものだという意見を一笑に付します。

別の人たちは、新しい技術のおかげで「仮想」サーバが誕生し、システム構成が何百回も複製可能になったことを認識しています。あちこちに動かすのはハードウェアではなく、ソフトウェアだけです。つまり、デプロイメントは前世代より桁違いに速くできるようになったということです。これは、手作業で回路図を書くことから、プログラミング言語を使ってソフトウェアでロジック構成を制御することへの移行に等しいことです。

IBM WebSphere 部門の CTO であるジェリー・クオモのようなビジョンを持つ人たちは、この目覚ましい変化を何年も前に察知し、自らの組織内で積極的にクラウド・ベースのソリューションを推進してきました。2008 年の InfoQ のインタビューで、ジェリーは「原子的仮想化」と「分子的仮想化」について説明し、それはサーバの構成と情報を「フリーズ・ドライ」する技術であり、それを使ってサーバをプリインストールして構成し、ホット・デプロイできるようにするものだとしています。彼いわく、「これらを集合で作れるようになると、本当に凄いことになります…これによって、実際のサーバ・トポロジーすなわちサーバのグループをフリーズ・ドライできるのです。」このインタビューは次のサイトにあります：

<http://www.infoq.com/interviews/jerry-cuomo-cloud-computing>

プライベートおよびパブリックのクラウド・コンピューティングは、これらの仮想化技術で共有計算機リソースを利用できるようにすれば、自然ななりゆきとして生まれてきます。これは大規模な運用を支援するだけのハードウェアを管理する時間やリソースのない多数のビジネス・ユーザの手に、膨大な量の計算機力をもたらすものです。これは真に革命です。

---

アプリケーションは今や基礎を成すホスト・ハードウェアから完全に切り離すことができるようになりました。かつて不変のインフラと見なされていたハードウェア・サーバが、現在では構成可能なコンポーネントでもあるのです。アプリケーションが特定のホストやネットワークのアドレスで実行されているとは、もはや想定することができません。

環境がプライベート・クラウドであれ、パブリック・クラウドであれ、アプリケーションの健全性状態の監視問題はさらに複雑になっています。デプロイメント・アーキテクチャ全体がコンポーネント化され、考慮すべき別の間接性レベルが導入されます。サーバからミドルウェア・スタック全体まで、アプリケーションを構築している基盤のすべてをコンポーネントとして捉え、監視しなくてはなりません。

短期的には、アプリケーション全体がクラウドで運用されることはなく、アプリケーションの一部はレガシー・システム内に残るため、多くの新しい統合ポイントを監視する必要があります。標準化とコンポーネント化がユートピア的な未来のアプリケーションの光景を約束する一方、短期的な現実はかなり異なっています。前述のコンポーネント指向アプリケーションを監視する難題は、さらに難しくなっていきます。

## リアルタイムでのアプリケーション健全性ステートとデータ・フロー

ほとんどのアプリケーションは、クラウドで実装されているか否かを問わず、真の健全性状態と性能レベルへのリアルタイムな可視性がないという、共通の問題に悩まされています。

CPU やメモリ使用などのインフラ指標に終始する従来のシステム管理ツールは普通、共有コンポーネントの世界では不十分だとわかります。コンポーネント・ベンダが提供するポイント・ソリューションは、特定のサブシステムの性能を監視することを目的としており、起こっているアプリケーションの問題原因を特定するには、複数のサブシステムの指標をそれらに依存する多数のアプリケーションに関連付けるという気の滅入る作業を、ユーザが行わなければなりません。

ネットワーク・パケット分析やメソッド・レベルでの VM 計装といった他の技術は、アプリケーションのコードベース内、すなわち外部世界へのインタフェース内のホット・スポットを特定する助けにはあります。しかし、必要なフィード・プロセスがダウンするときや、ファイルが見つからないときには、問題は1つのコンポーネントから別のコンポーネントへのデータ・フローのエラーの結果として起こることがよくあります。このタイプの問題では、こうしたツールで特定するのは困難です。

アプリケーションの健全性状態を真に可視化するには、依存する複数のコンポーネントの状態を網羅的にかつ統合的に見る必要があります。さらにもっと重要なのは、個別のアプリケーションが使用する共有リソース量をユーザが知ることのできるフィルタリング・システムです。

市場には、この問題に取り組むための APM ソリューションがいくつか出ています。SL 社では、同社の RTView 製品によって、アプリケーションの健全性状態のリアルタイムな監視と履歴ベースライン分析で、多くの APM 機能を提供しています。次にあげる機能などは、複数コンポーネント・サブシステムをベースにしたアプリケーションを監視する場合、特に重要です：

- 1) 関連するリソースすべてから監視データを収集し、それを見る必要があるユーザがリアルタイムで利用できるようにします。これにはモジュール化されたスケーラブルな監視システムが必要であり、複数のインスタンスを地域やサブネット全体に運用し、サマリ情報に加えてコンポーネントの詳細へのドリルダウンも提供します。

この部分のソリューションは、特に重大なシステム停止やユーザから性能の遅さについてクレームがあがっている場合に重要で、どのサブシステムが故障しているのかをすばやく判断しなければなりません。全コンポーネントからの全データを自由自在に操れることが、この問題を解決するのに不可欠です。

- 2) 特定のアプリケーションに関連するすべてのコンポーネントからのデータの流れを単一画面の「コンテキストに沿った」ビューで監視データを表示します(図1)。ユーザはこれを、各コンポーネントのスループット指標や健全性状態あるいは可用性を表示する、リアルタイムなデータ・フローの入った「ライブの」Visio ダイアグラムと考えます。そして、ユーザはこれらのダイアグラムをカスタマイズして構築でき、ダイアグラム内のオブジェクトを特定の性能指標と関連付けられるようにする必要があります。

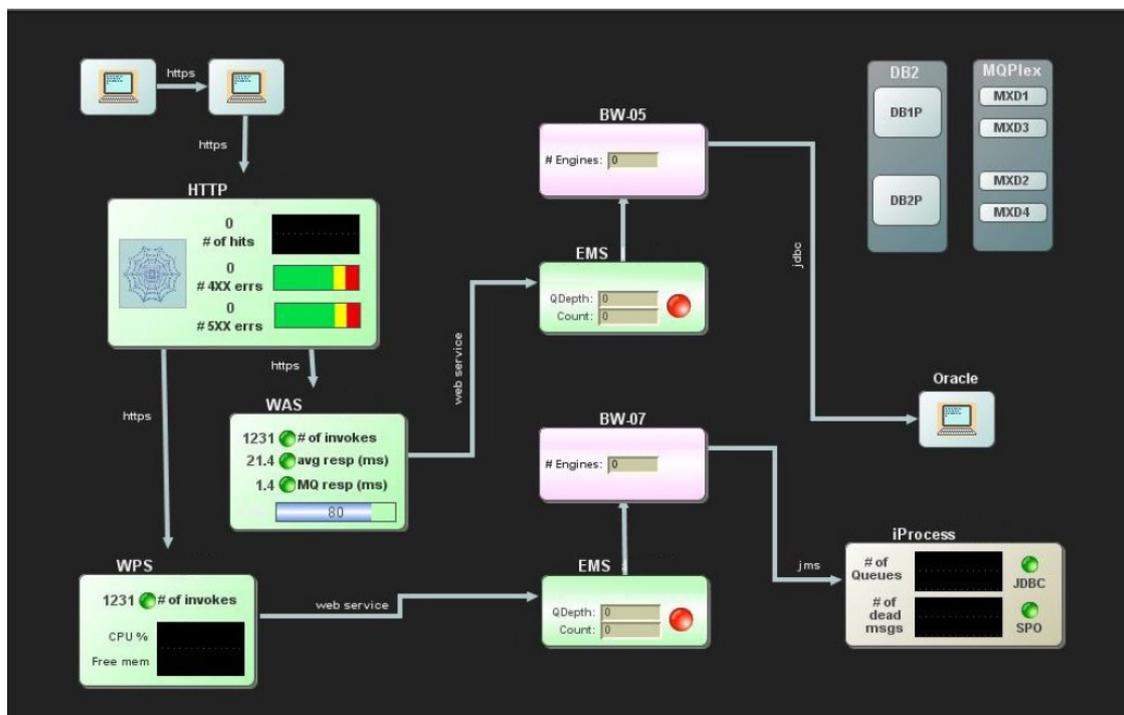


図1: いくつかのコンポーネントからの指標を表示するアプリケーション・データ・フロー画面

この機能が重要なのは、アプリケーションをサポートしているスタッフは、アプリケーションを互いにデータを渡し合うデータ・フィードとサービスの観点で考えることが、よくあるからです。アプリケーションが正常に動作するには、各コンポーネントが起動していて実行中であって、それらのコンポーネント間でデータが流れていなければなりません。

- 3) 共有コンポーネントで起きた問題がどのアプリケーションに影響するのかを迅速に判断できるよう、アプリケーションで監視データをフィルタリングします。この機能は、たとえば1台のサーバを10以上のアプリケーションが共有し、かつ各アプリケーションが2、3の特定のトピックやキューを使っているようなメッセージング・システムには欠かせません。ほとんどのユーザは、自分のアプリケーションに関連した指標だけを見たいからです。

こうしたテクニックは、包括的なアプリケーション指向監視ソリューションに要求されるいくつかの重要な性質を明確にします。RTView は、数々の企業組織に強力なアプリケーション指向のビューを提供する機能で、大変な成功を収めてきました。複数でかつ／または極端に大規模なアプリケーションを監視する場合、多くの監視画面と情報を可能な限り自動化できることが欠かせません。

## 構成データからの自動化監視 - 聖杯

多くの企業組織では所有するハードウェアとソフトウェア資産のインベントリ情報のデータベースを保持しています。こうした情報を利用して自動的に一連の画面データを集め、そのインベントリ情報からアプリケーションの健全性状態を表示することができます。

このタイプの自動化を実現するには、監視システムの主要機能をデータ駆動にできる柔軟さがが必要です。RTViewは高度なデータ駆動型システムの一例であり、その構成のあらゆる側面はデータソース、画面スタイル、レイアウト、オブジェクト関係、その他を表すデータ・テーブルで駆動できます。このような柔軟性を使って RTView を運用することができ、インベントリ情報データベースで保持されているデータから、データ収集パラメータを自動的に構成することができます。同様に依存性情報を監視画面(図2)に自動的に表示することが可能で、そこに重要なアプリケーション／コンポーネントの関係やリアルタイム指標を表示できます。そして、これらはすべてカスタム・プログラム開発を要しません。

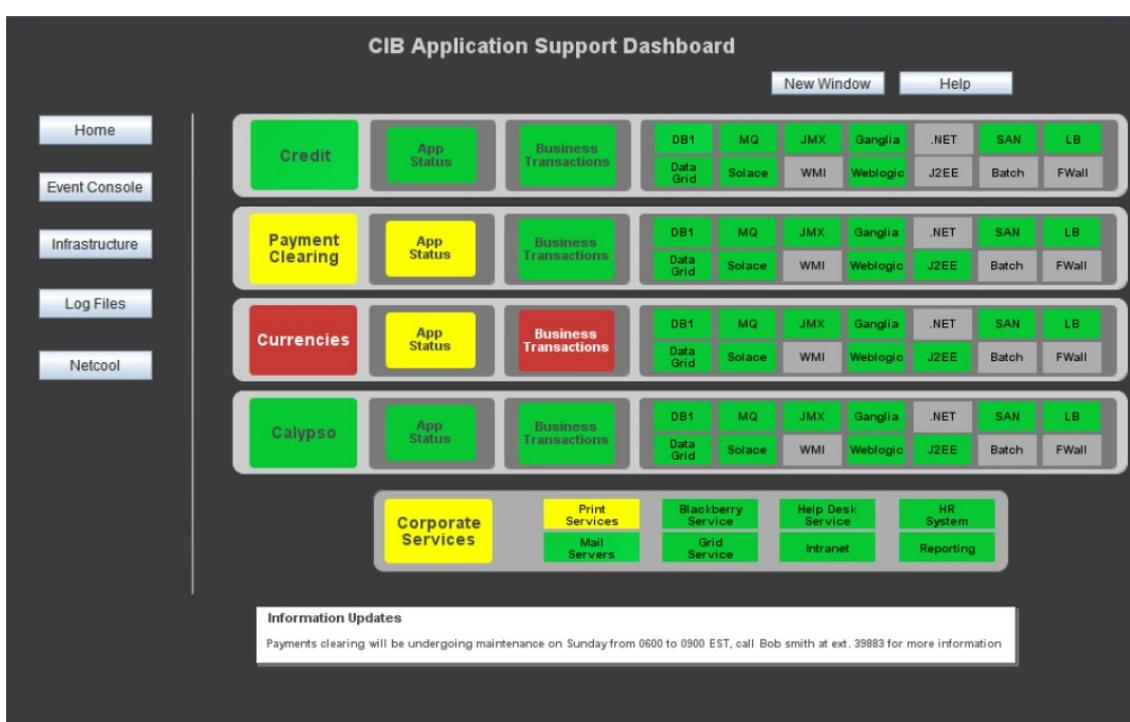


図2: 各アプリケーションに関連のあるコンポーネントを表示するアプリケーション依存性画面

ユーザがアクセスする構成情報のタイプには広い幅があります。ミドルウェア・コンポーネントやサーバ上のアプリケーション依存性を単なるスプレッドシートで保持している場合もあります。また、より洗練された「構成管理データベース (CMDB)」を使用している場合もあります。CMDB 内のデータが正確に現行のシステム状態を表している限り、データ駆動型の構成と監視画面は、真のシステム健全性状態を表示する有用なビューを提供します。

しかし、サーバが常時取り替えられたり再割り当てされるなどの場合、構成情報のデータベースを手作業で維持するのは難しくなります。手作業で管理するデータベースはすぐに古くなり、それに基づいた監視システムは信頼できません。

この理由から、多くの企業組織ではさまざまなソフトウェア・コンポーネントやサブシステム上のネットワークのトポロジーやアプリケーションの依存性を「自動検出」するように設計されているツールを利用しようとしてきました。これは優れたアイデアですが、実際には検出機構が不十分で、常に更新していないと新しいコンポーネントや構成内の変化を把握できません。

---

RTView は何社ものお客様が、自動検出ツールで得られる情報から監視画面を自動的に作ることに成功されてきました。ほとんどの場合、多数の例外があるので、モデルを確定してプロセスを自動化する前に、まず手作業でレビューする段階を経ることが必要でした。

アプリケーションをクラウド環境で運用する場合、インフラ、ミドルウェアやアプリケーション自体の情報はすべて構成データベースに含まれています。基礎を成すコンポーネントはすべて仮想化されデプロイされているので、それに対する参照もすべてパラメータ化されています。たとえば、ネットワーク・アドレス、サーバ名、デプロイしたアプリケーション・サーバ名、メッセージ・システムについての情報は、すべてこのデータベースに含まれています。

このシナリオでは、この情報を「発見」する必要はありません。依存性の情報はすでにデプロイメント・データベースに含まれています。この点から自然に導かれることは、監視はもはやアドオンや後付けとは見なされないということです。すべてのサブシステムのデプロイメントはクラウドのプロビジョニングで自動化されるため、監視アプリケーション自体も自動的にデプロイして構成できます。RTView は、こうした環境での監視システムの選択として急速に頭角を現しています。

## SL 社について

過去25年以上にわたり、SL 社はリアルタイム・データの監視・分析・可視化ソフトウェアの提供で、最も知識豊富で鋭敏であり続けました。SL 社の旗艦製品である RTView は、エンドツーエンドのアプリケーション性能管理 (APM) やビジネス・アクティビティ監視 (BAM)、コンポーネント・レベルのインフラ監視にわたる広範なオペレーションの可視化に取り組んでいます。また RTView は、複合イベント処理 (CEP) エンジン、TIBCO などのメッセージング・ミドルウェア、Oracle Coherence データ・グリッドの可視化で使用されているデファクト・スタンダードです。SL 社の可視化ソリューション技術と専門性、コミットメントと顧客実績、パートナー重視のビジネス・カルチャーが、世界の業界でリードする何千にもおよぶ最もクリティカルなアプリケーションとビジネスの支援で SL 社が選択されてきた理由です。

株式会社 SL ジャパンは 1998 年に設立された、SL 社の日本法人です。



**株式会社 SL ジャパン**

〒107-0062 東京都港区南青山3-8-5 アーバンプレム南青山 3階

Tel. 03-3423-6051 Fax. 03-3423-6070 info@sl-j.co.jp [www.sl-j.co.jp](http://www.sl-j.co.jp)