# 困難で時間のかかる本番環境の問題を dynaTrace で解決

**ケーススタディ – 本番環境のトラブルシューティング - Intuit**

企業
Intuit, Inc.（NASD：INTU）

業種
財務ソフトウェアおよびサービス

所在地
世界各地

課題
社内の Production Readiness Center による、オンライン Billing Manager サービスの可視化と、パフォーマンス低下の根本原因の切り分けを実現

アプリケーション環境
Java EE / Spring / Hibernate / Oracle 11g

要件
以下を実現することにより、本番環境におけるトラブルシューティングを推進
- → コードレベルで可視化を実現し、問題の根本原因を迅速に特定
- → Web サービスおよびデータベースコールの可視化
- → 複雑な Spring ／ Hibernate をベースとする環境の可視化
- → 各トランザクションに対して「スライス＆ダイス」を行い、様々な視点から調査し問題を迅速に切り分け

ソリューション
***本番環境に dynaTrace を導入することにより***、Intuit は、困難なパフォーマンス問題を 8 時間以内に解決し、パフォーマンスを 10 倍向上。その結果、以下の成果を達成。
- → 他のベンダーから入手したシステムのビジネスロジックを可視化
- → 冗長なデータベースコールを引き起こし、ガベージコレクション停止のトリガとなっていたフレームワークを可視化
- → 冗長コールの原因となるフレームワークとアプリケーション間のやり取りを切り分け、ログイン時間や処理時間を短縮

***「dynaTrace によって、システムを詳細に可視化することができました。これにより、ビジネスロジックを理解するための出発点が提供されたのです。」***

Intuit は、中小企業、銀行や信用金庫などの金融機関、一般消費者や経理のエキスパートを対象とした、ビジネスおよび財務管理ソリューションの大手プロバイダです。 QuickBooks®、Quicken®、TurboTax® といった代表的な製品やサービスは、小規模ビジネスの管理と給与処理、個人の資産管理、税金の確定申告書類作成および申告作業を簡略化するものです。1983 年に設立された Intuit は、2009 年度には 32 億ドルの売上を計上しています。従業員は約 7,800 人で、主要なオフィスは米国、カナダ、英国、インド、などに存在します。

**課題：**Intuit は中小企業を対象とした金融サービス製品の大手ソフトウェアプロバイダであり、SaaS をベースとしたサービスでは業界をリードしています。Intuit の製品の 1 つである Intuit FMS Billing Manager は、中小企業を対象に、オンラインでの請求書発行およびクレジットカードによる支払い機能を無料で提供します。

Billing Manager の顧客企業は小規模ですが拡大傾向にあり、5 万人以上の個人事業主を中心に形成されています。その顧客企業の中に、ある小規模ホスティングサービス事業者がいました。この事業者は、Billing Manager を請求および支払い管理システムとして使用しながら、30 の中小企業にサービスを提供していました。ある時、この事業者は、作成済みの未払い請求書 300 件を処理しようとしていましたが問題が発生してしまいました。ログインの初期化と処理に 5 ～ 10 分かかり、各請求書の処理にさらに 10 ～ 20 分かかってしまったのです。

当時、これらの問題の解決には、膨大な労力が必要であることがわかりました。Billing Manager チームは、ほぼ 3 週間かけてこの問題の解決を試みましたが、以前から使用しているツールでは根本原因を切り分けることができませんでした。この問題はついに、本番環境の機能停止にまでいたりました。サイトには、初めて「パフォーマンス問題を調査中です」と書かれたページを掲載するはめとなりました。これは事実上、「サービス停止」を意味していました。一方、問題解決にあたっている Billing Manager チームは、システムのコードやアーキテクチャを可視化できていませんでした。というのも、このシステムは別のベンダーからテクノロジや顧客基盤を買収し、引き継いだものだったからです。

チームが Billing Manager の開発バージョンと本番運用バージョンを比較すると、非常に矛盾した結果が出ました。精査の後、エンジニアリング担当取締役はチーム全員（DBAs Operations、Production Readiness Center、開発担当）を呼び、問題をできるだけ早く切り分けるように指示しました。「同時にトラブルシューティングも行う必要がある」取締役は言いました。「コストはいくらかかってもいい。」

**ソリューション：**Intuit には、以前から「Production Readiness Center（PRC）」と呼ばれるチームが編成されていました。このチームは Intuit のすべての開発事業部門に対し、社内のパフォーマンスベンチマークとコンサルティングサービスを提供していました。PRC 所属の設アーキテクトである Ray Lai 氏は、チームが達成できていないコードの可視化を実現するために、本番環境に dynaTrace をインストールすることを提案しました。Lai 氏はパフォーマンス診断とトラブルシューティングのために、dynaTrace を購入していましたが、Intuit の本番システムには導入したことがありませんでした。そのため、当初、この提案はあまり歓迎されませんでした。なぜなら、dynaTrace の導入により、現行の Intuit のルールや手順が乱されることになるからです。

パフォーマンスが低下した状態が続き、ほぼ機能停止になると、チームは問題を特定するために dynaTrace を本番環境にインストールする一方、CA Wily Introscope や Oracle データベースレポート機能など、他のベンダーのテクノロジも並行して使用し、解決を急ぐことにしました。

*Lai氏とチームのメンバーは、dynaTraceを迅速にインストールし、数時間のうちにシステムを可視化することができました。その結果、アプリケーションとHibernate間のやり取りによって大量の冗長データベースコールが発生していることが判明したのです。*

*「ユーザー1人あたり、750以上のデータベースコールがトリガされていました。ユーザーが3人になると、14,550以上のデータベースコールがトリガされます。これらすべてのコールがリソース競合を起こしたので、ガベージコレクションの停止がトリガされました。」*

*dynaTraceを利用することで、Lai氏とチームメンバーは、冗長コールの根本原因を8時間以内に特定し、パフォーマンスを10倍向上させることができました。*

*「dynaTraceを利用すると、データを様々な角度から精査することができます。スライス＆ダイスにより、個々のトランザクションを「外側から内側へ」トラッキングし、最も時間のかかるトランザクションのパフォーマンスを低下させている原因を把握することができます。」*

*「個々のWebサービスやデータベースコールを「内側から外側へ」確認することも可能です。このため、データを詳細レベルで理解し、問題を迅速に切り分けることができます。」*

*Lai氏とチームメンバーは、dynaTraceをインストールしてから24時間以内に、問題の根本原因を特定し、解決策を示すことができました。*

*運用チームは、並行してIntroscopeのインストールも行っていましたが、dynaTraceよりもずっと長い時間がかかりました。実際、Lai氏とチームメンバーが問題の根本原因をグループに提示したとき、Introscopeは、まだインストール中でした。*

**結果：**インストール後1時間も経たないうちに、システムはかなり標準的な構造に基づいており、一般的なJSPページを表示するためにSpringやHibernateなどのフレームワークが使用されていることがわかりました。「dynaTraceによって、システムを詳細に可視化することができました。」Lai氏は続けます。「これにより、ビジネスロジックを理解するための出発点が提供されたのです。」

Hibernateは迅速なアプリケーション開発を実現するポピュラーなフレームワークですが、アプリケーションとフレームワークの間のやり取りがよく理解されていない場合に、ロード時に膨大なデータベースコールを要求するデータオブジェクトツリーにより、パフォーマンス問題を引き起こす可能性があります。Lai氏とチームのメンバーは、dynaTraceを迅速にインストールし、数時間のうちにシステムを可視化することができました。その結果、アプリケーションとHibernate間のやり取りによって大量の冗長データベースコールが発生していることが判明しました。

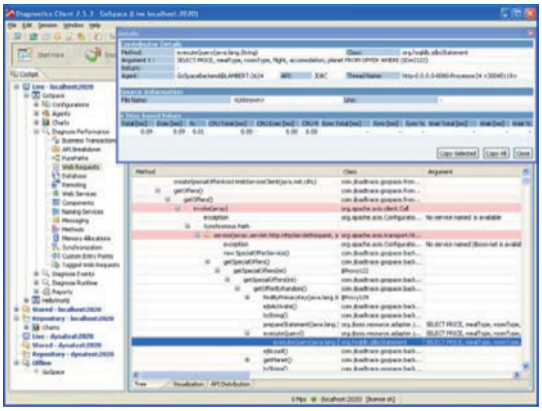
**[dynaTrace System Architect Dashboard]画面**

dynaTraceを利用することにより、システムがログイン時に受信ボックス内の未払い請求書フォルダをすべてスキャンし、請求書に関する情報を取得していることがわかりました。見積もりや発注書は各請求書と関連づけられているため、Hibernateから膨大な量のオーバーヘッドが発生し、ファミリーツリーに大量のオブジェクトリレーションが生成されていました。また、既存のアーキテクチャにより、ユーザーがページの操作を開始する前には必ず、ビジネス管理アプリケーションがメモリへ一括読み込みを行うようになっていました。

一度に1～2件の請求書を処理する小規模個人事業者の場合であれば、パフォーマンスは、やや低下する程度で済んでいましたが、300件の請求書の処理となると、アプリケーションは、ほぼ停止状態に追い込まれました。「ユーザー1人あたり、750以上のデータベースコールがトリガされていました。」Lai氏は説明します。「ユーザーが3人になると、14,550以上のデータベースコールがトリガされます。これらすべてのコールがリソース競合を起こしたので、ガベージコレクションの停止がトリガされることになります。」

dynaTraceを利用することで、Lai氏とチームメンバーは、冗長コールの根本原因を8時間以内に特定することができました。パフォーマンスを10倍向上させ、ログイン時間も250秒から20秒に短縮しました。「dynaTraceを利用すると、データを様々な角度から精査することができます。」とLai氏は述べています。「スライス＆ダイスにより、個々のトランザクションを「外側から内側へ」トラッキングし、最も時間のかかるトランザクションのパフォーマンスを低下させている原因を把握することができます。一方、個々のWebサービスやデータベースコールを「内側から外側へ」確認することも可能です。このため、データを詳細レベルで理解し、問題を迅速に切り分けることができます。」

非常に詳細なパフォーマンスデータに対してスライス＆ダイスを行い、分析の出発点を提供してくれるdynaTraceの機能がなければ、データベースコールに関するこの問題を見つけることはできなかったでしょう。Oracle Automated Workload Repository（AWR）のレポートによると、各SQLコールにかかる時間は100ミリ秒未満となっていました。SQLステートメントの実行速度が遅いことがあり、ガベージコレクションの停止が原因なのか、他に原因があるのかは不明なまま、SQLを調整することによって高速化できることがあります。しかし、ステートメントの調整では、大量に発生する問題を解決することはできないでしょう。



**データベースコールを示した画面**

Lai氏とチームメンバーは、dynaTraceをインストールしてから24時間以内に、問題の根本原因を特定し、解決策を示すことができました。運用チームは、並行してIntroscopeのインストールも行っていましたが、dynaTraceよりもずっと長い時間がかかりました。実際、Lai氏とチームメンバーが問題の根本原因をグループに提示したとき、Introscopeは、まだインストール中だったのです。