

APPDYNAMICS

WEBアプリケーション性能を可視化！ ～ 次世代APM AppDynamicsご紹介～

* APM : Application Performance Management

株式会社サムライズ
ビジネスアナリティクスグループ

SAMURAIZ



株式会社サムライズ

サムライズ会社概要

会社名：株式会社サムライズ (Samuraiz Corporation)

住所：東京都品川区大崎一丁目6番4号 新大崎勧業ビル

創立：2006年7月

事業内容：

- ソフトウェア販売事業（国内外ソフトウェア製品販売）
- サービス&サポート事業（保守、教育サービス、各種サービス）
- IT関連の事業開発・事業開発支援

取扱商品

- ◆ アドビサーバソリューション
- ◆ ビジネスインテリジェンスソリューション
- ◆ Webマーケティングソリューション
- ◆ アプリケーションパフォーマンスソリューション
- ◆ セキュリティ対策ソリューション

webtrends™

InsightNavi

ca.
technologies

AppDynamics

copy
fun & smart captcha

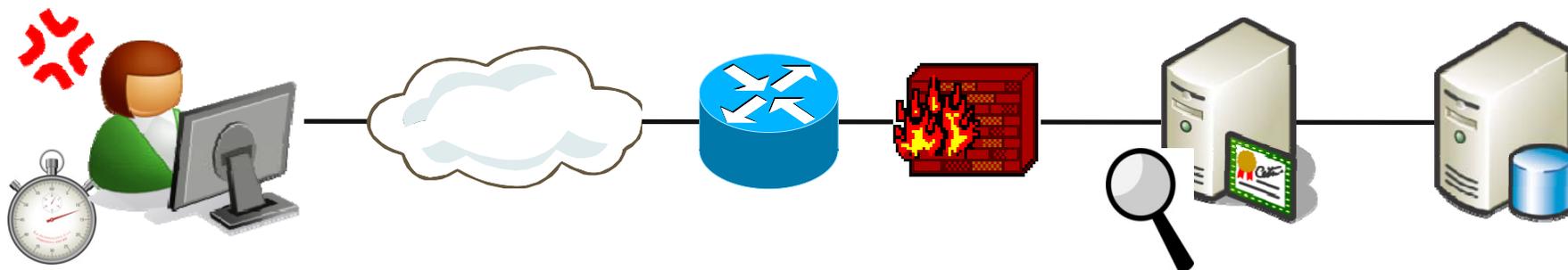
GoodData

SCASTA

dynatrace

OPENTEXT

アプリケーションパフォーマンス管理(APM)とは



アプリケーションパフォーマンス管理(APM)

アプリケーション管理: アプリケーション動作をソースコードレベルで管理

レスポンス管理: ユーザ体感レスポンス管理

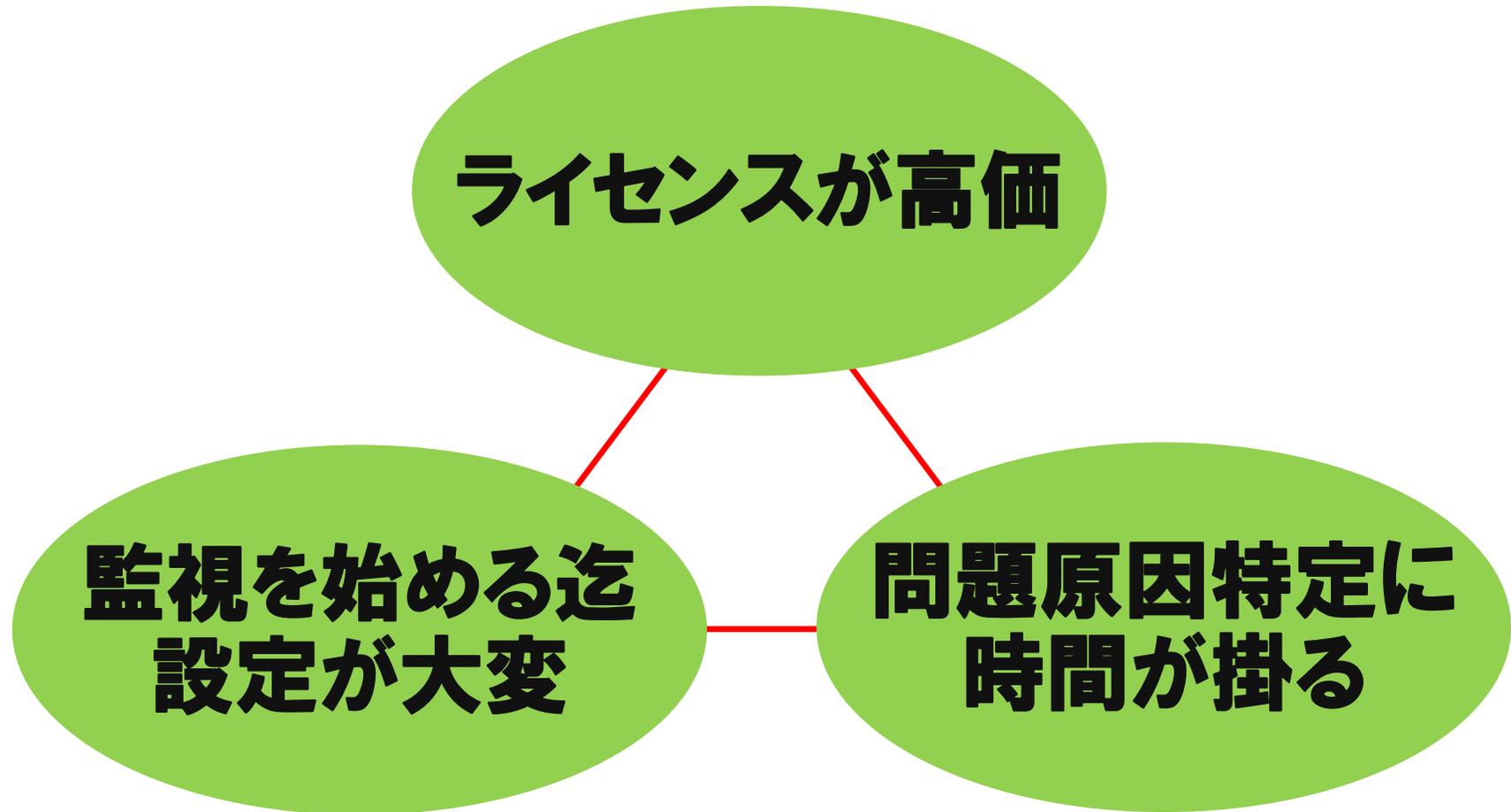
トラフィック管理: 通信トラフィックやプロトコル管理

サービス監視: 稼働するサービスやプロセス状態

リソース監視: CPUやメモリなどリソース利用状況

死活監視: Pingによる応答監視

これまでの運用監視ツールは



APPDYNAMICS

次世代APM AppDynamicsとは？

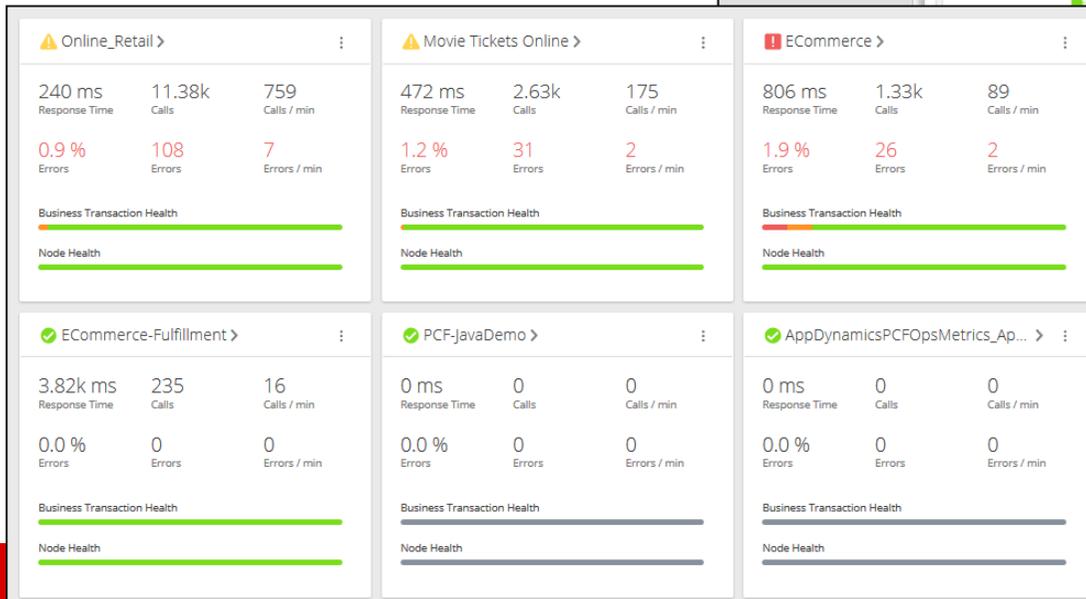
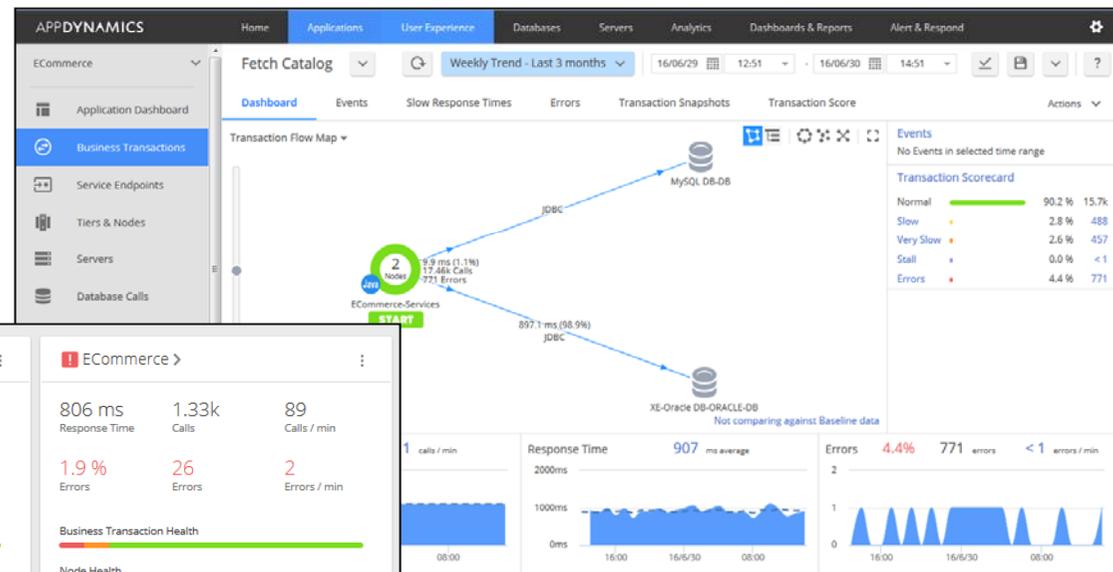
SAMURAIZ



株式会社サムライズ

アップダイナミクスとは？

- JavaやPHPなどウェブアプリケーションシステムのモニタリングソフトウェア
- エージェントを導入し、ソースコードレベルで稼働状況を低負荷でモニタ
- アプリケーションがスローダウンした原因を迅速に特定します



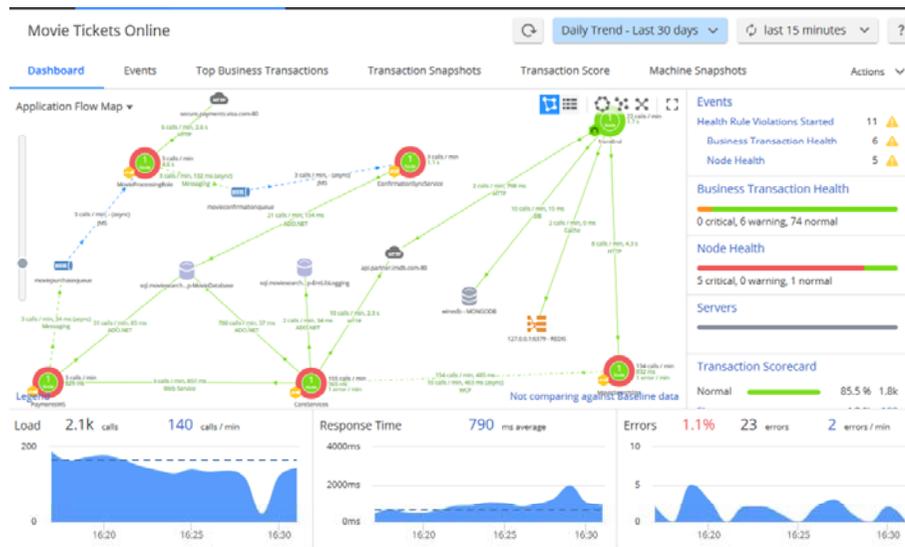
アップダイナミクスのメリット



ライセンス
が手軽

データを自動
計測・集計

問題点を
容易に特定



必要な時に必要な所へ

アップダイナミクスは何が出来るのですか？



1

平均2%以下の低オーバーヘッドにより、商用環境のオンライン・サービス全体の可視化を可能とし、サービス品質問題の原因を即座に特定します

2

システム上の全てのビジネス・トランザクション（サービス）を自動検知し、従来のシステム毎の監視からは検出できなかった問題を検出します

3

個々のサービスに付随したアプリケーション処理、データベース処理、クライアント処理を監視し、サービス品質低下時の状況を詳細に記録します

4

個々のサービスの処理時間から平常時のパフォーマンスを自動測定し 平常時からの乖離率によりスローダウンの判定基準を自動設定します

5

個々のサービス品質とシステム・リソースの使用状況を監視し、不要なハードウェア投資を抑制します（リソース・バランスを把握）

アプリケーション・インテリジェンスのリーダー

APP DYNAMICS について

その強みとメリット



前年比
100%成長
(2015年度)



APM市場の
リーダーとして
評価



Webから簡単に
ダウンロードできる
ソフトウェア



特別な
実装作業
やスキルが不要



オン・プレミス、
SaaS、または
ハイブリッド何れも可



既存顧客の
ライセンス契約
更新率*



世界
2,000社以
上のお客様



スケーラビリティ
大規模APM
導入企業の
20社中18社が利用



ビジネス部門、
運用、開発など
全社にわたって
共通的に活用



低いTCO
(ライフタイム
における
総保有コスト)



短期間で
導入が可能で
すぐに効果を発
揮

*2015年度, 5万ドル以上のライセンス料金のお客様

なぜアップダイナミクスが評価されているのか？

	AppDynamics
本番運用環境の監視に強い実績	全世界2,000社を超える実績 本番運用環境でAPMを大規模導入しているトップ20社中18社が採用
広範な監視対象範囲	サーバーアプリケーション、データベース、ハードウェア/OS、Webエンドユーザー体験、モバイルアプリケーションを、単一のwebコントローラー画面から監視可能
柔軟な提供形態	APMの提供形態： オン・プレミスへの導入 と SaaSでの利用 の双方をご提供 APMの監視対象： オン・プレミス、クラウド(Amazon AWS, Azure, Softlayer 等)、そして、それらとのハイブリッド環境をサポート
低TCO、早期回収	事前のビジネス・バリュー・アセスメントの投資回収仮説と、 実際のProof of Value (PoC) 結果による裏打ちで、費用対効果と回収期間を予測 → 70%以上のお客様が予算措置無しでご導入
圧倒的な顧客満足度	年間サブスクリプションの更新率99%、Net Promoter Score 87ポイント 導入後18ヶ月以内の追加ライセンス購入 → 導入時の2.75倍のライセンスを追加

世界100カ国以上で 1,800社を超えるお客様

<p>金融</p>	<p>小売、旅行</p>	<p>SaaS、クラウド</p>	<p>メディア、エンターテイメント</p>
<p>製造、輸送機器</p>	<p>官公庁、教育</p>	<p>ハイテク、通信</p>	<p>製薬、ヘルスケア</p>

実現したビジネスの価値 — お客様の実例より

コストの削減



- アプリケーションの利用規模が10倍に拡大
- 340万ドルのハードウェア投資を回避
- 2年間で480万ドル削減



- ハードウェア、ソフトウェアの構成を最適化することに成功
- 1ヶ月で2,500以上のビルドを本番環境に投入することに成功

効率的な運用



- サポート・チケットの94%削減
- 年間 \$200,000 以上のコスト削減



- 本番環境におけるMTTRの65%改善
- テスト運用における障害解決時間の75%改善



顧客満足度の向上



- 顧客のトランザクション時間を10秒から1秒未満にまで改善
- コールセンター・アプリケーションの待ち時間を大幅に改善



- トランザクションのパフォーマンスを25%改善
- 約1年間で投資を回収

売上の保全

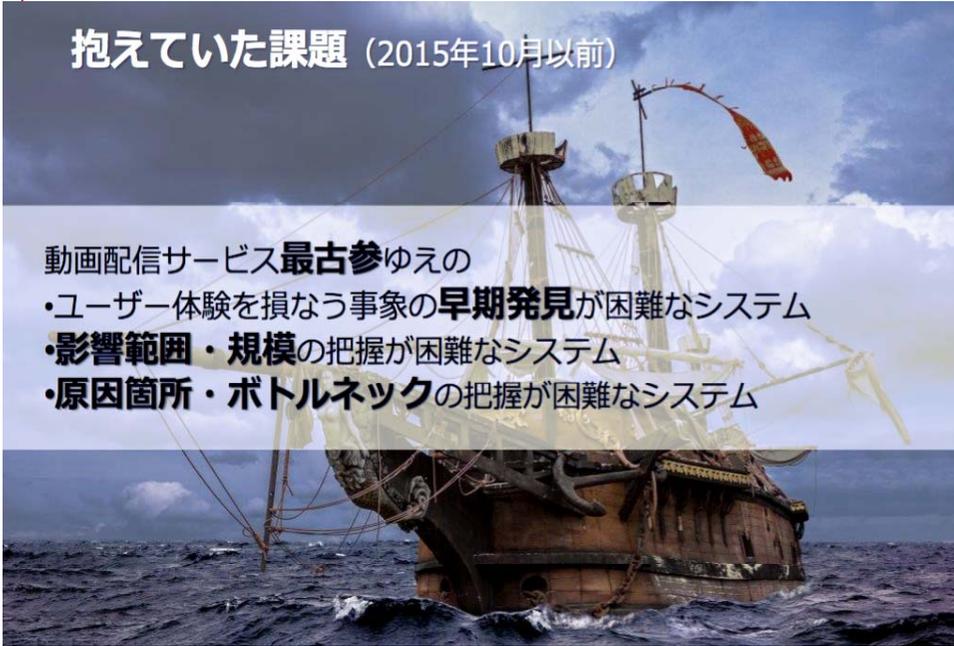


- MTTRを2時間から30分に短縮
- 2012年にはダウン時の売上損失が135万ドルもあったものが改善



- 本番環境の稼働率が99.95%まで改善
- 16万7千ドルの売上損失を回避
- 80万ドルの生産性改善を実現

事例：オンラインレンタル事業 U-NEXT 様

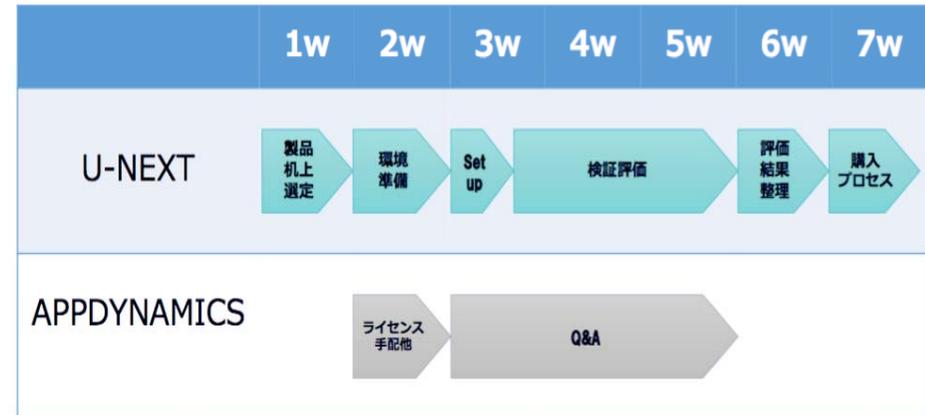


抱えていた課題 (2015年10月以前)

動画配信サービス最古参ゆえの

- ユーザー体験を損なう事象の**早期発見**が困難なシステム
- **影響範囲・規模**の把握が困難なシステム
- **原因箇所・ボトルネック**の把握が困難なシステム

APM選定・検証・評価 (スケジュール)



導入後の効果事例 (アニメ放題)



★★★★☆ 評価件数：81

最新バージョンは
平均4.5点以上
が日常に

現在のバージョン 全てのバージョン

平均評価 ★★★★★ 評価件数：1,020

評価値は
単調増加傾向



導入後の効果 (定性)

期待効果	実績 (2015年12月時点)	今後の期待
UXを損なう事象を早期発見したい	Good (APM配下)	アラート機能使い込みによるプロアクティブ監視体制の拡大余地あり (U-NEXT)
UXを損なう事象の影響範囲・規模を速やかに特定したい	Great (APM配下)	APM監視対象サーバー・プロセスの拡大 (APPDYNAMICS&U-NEXT)
UXを損なう事象の原因箇所・ボトルネックを速やかに特定したい	Great (APM配下)	APM監視対象サーバー・プロセスの拡大 (APPDYNAMICS&U-NEXT)
継続的にバックエンドシステムのパフォーマンスを改善したい	Great	特になし
End to Endでパフォーマンスを監視したい	So-so	さらなる改良求む (APPDYNAMICS)

アップダイナミクスが実現する機能

ビジネス・トランザクション



SEE
状態監視



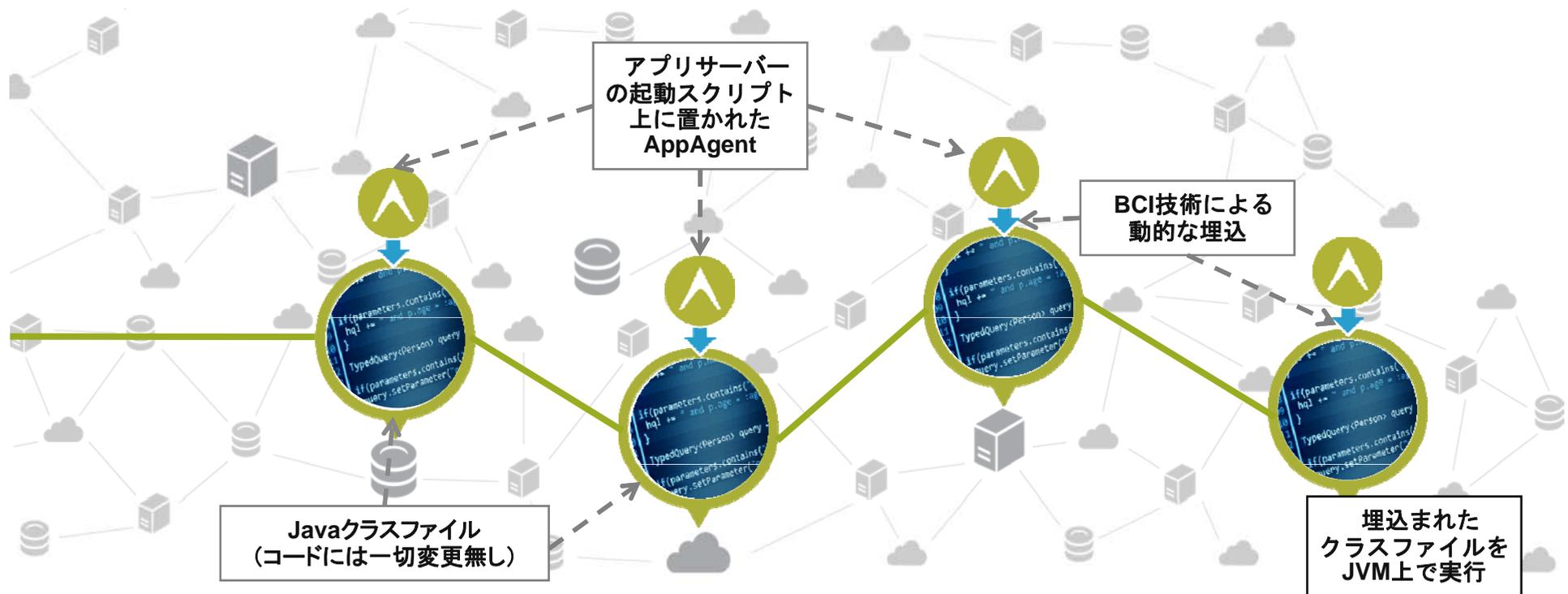
ACT
対策、対処



KNOW
影響把握

SEE | 状態監視: エージェント導入による性能監視

- 常に全てのユーザトランザクションを把握
- 把握したトランザクションデータを自動で関連付け
- 収集したトランザクションデータを自動学習し閾値や傾向を設定



インテリジェントな コード監視: 自動的、自己学習、低オーバーヘッド

ACT | 対策、対処: より早い問題解決を実現

(Fast MTTR: Mean Time To Resolution)

マウスクリックによるナビゲーションで迅速に根本原因を把握

エンド・ユーザー
レスポンスが遅い!



開発



QA



運用

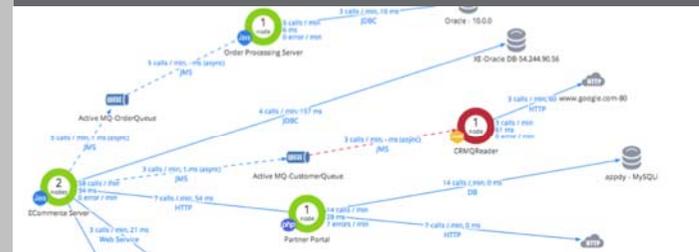


営業

モニター

Name ↑	Health	Response Time (ms)	Calls
Checkout	!	10	2313
Search	!	50	4476
UserLogin	!	9	2327
UserLogout	✓	13	2314

トラブル・シューティング



対処、解決

Name	Time (ms)	CPU time
StrutsActionProxy:execute	1 ms (self)	0.2 %
Struts Action - ViewCarts:sendItems:157	0 ms (self)	0 %
Proxy For Spring Bean - cartServiceTarget:getAllCartItems	0 ms (self)	0 %
Proxy For Spring Bean - cartServiceTarget:invoke	0 ms (self)	0 %
Spring Bean - cartServiceTarget:getAllCartItems:25	0 ms (self)	0 %
Spring Bean - cartPersistence:getAllCartItems:26	104 ms (self)	20.5 %
Proxy For Spring Bean - cartServiceTarget:getCartSize	304 ms (self)	59.8 %

KNOW | 影響把握: ビジネスの状態を知る インテリジェンス

- リアルタイムな ビジネス・ダッシュボード

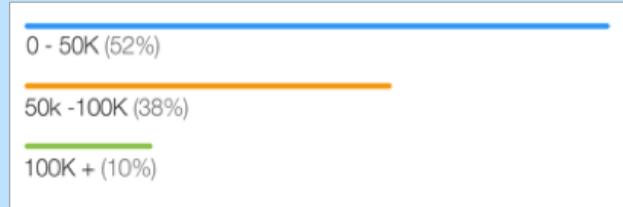
“過去24時間における チェックアウト・プロセス
でのエラーに起因する逸失売上額は?”

\$2,491



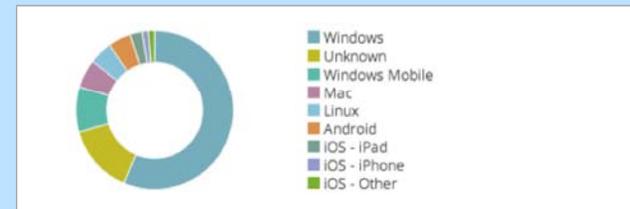
“本日の平均売上単価は?”

\$42,910



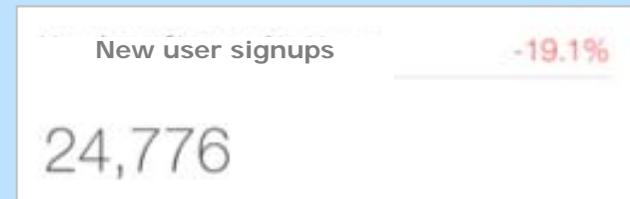
“最も売上に貢献しているデバイスは?”

iOS iPhone

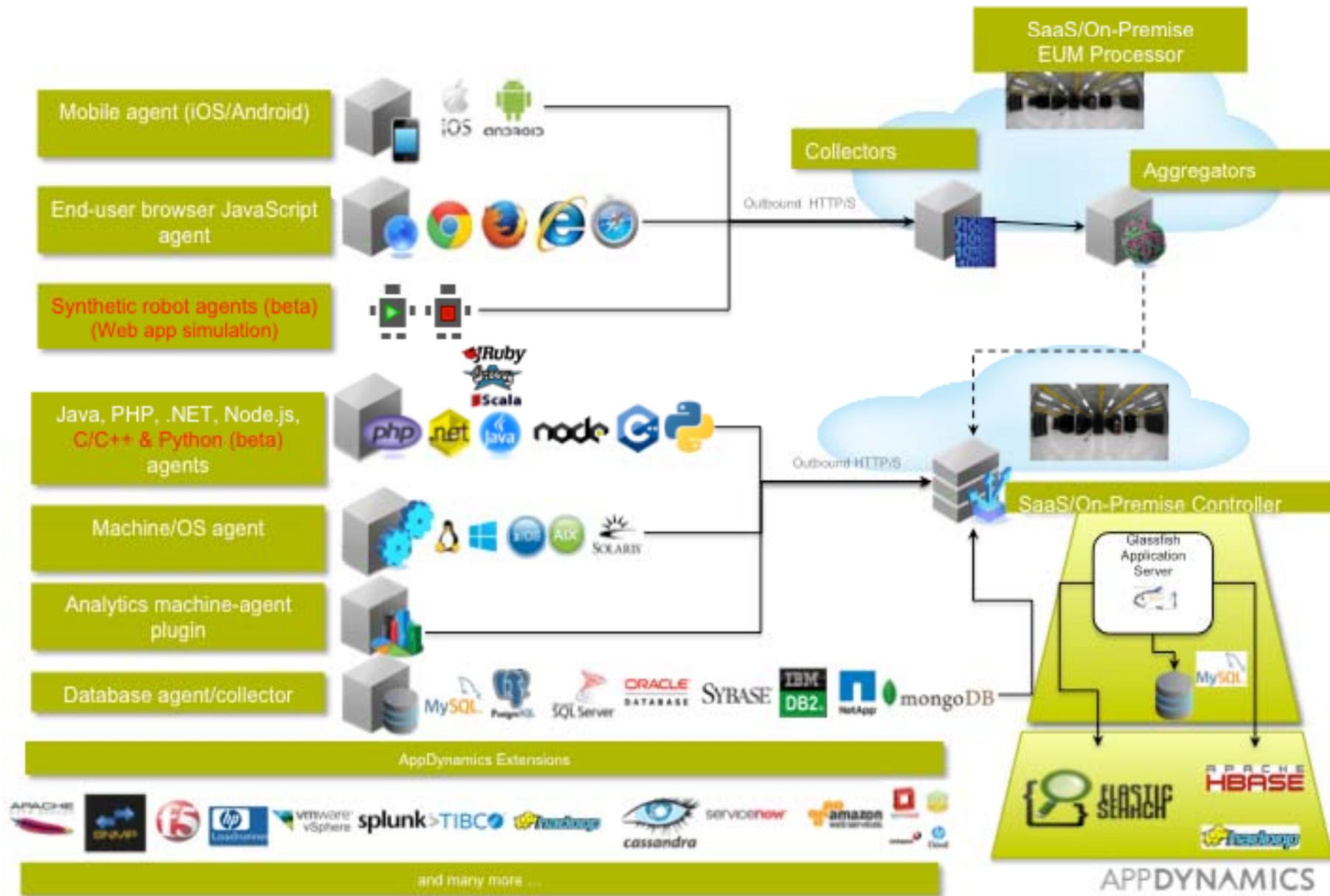


“今期、新たに会員登録したお客様数は?”

24,776



AppDynamics コンポーネント・アーキテクチャ (4.0)



AppDynamics の製品アーキテクチャ



アップダイナミクスの主な特徴

システムが提供するサービス（ビジネス・トランザクション）を監視



統合された
プラットフォーム

単一のUI

単一の
データプラットフォーム



本番環境の
モニタリング

低オーバーヘッド

全てのユーザー
トランザクションを監視



自動検知と
可視化

複雑なトランザクション
フローでも自動的に検知

手動での構成は不要



閾値を自動設定
し自己学習する

標準的な性能値を把握

大きく乖離したレスポンス性能
を検知してアラートを発出

コントローラー配置の 自在な選択肢



SaaS



オン・プレミス



プライベート・クラウド



パブリック・クラウド

本日はありがとうございました



SAMURAI Z

Real Solutions for Enterprise Web

◆ 製品に関するお問い合わせ ◆

**株式会社サムライズ
ビジネスアナリティクスグループ
APMセールス**

〒141-0032

東京都品川区大崎1-6-4 新大崎勸業ビル10F

TEL:03-5436-2040

<http://www.samuraiz.co.jp>

E-mail:y_koide@samuraiz.co.jp

参考資料

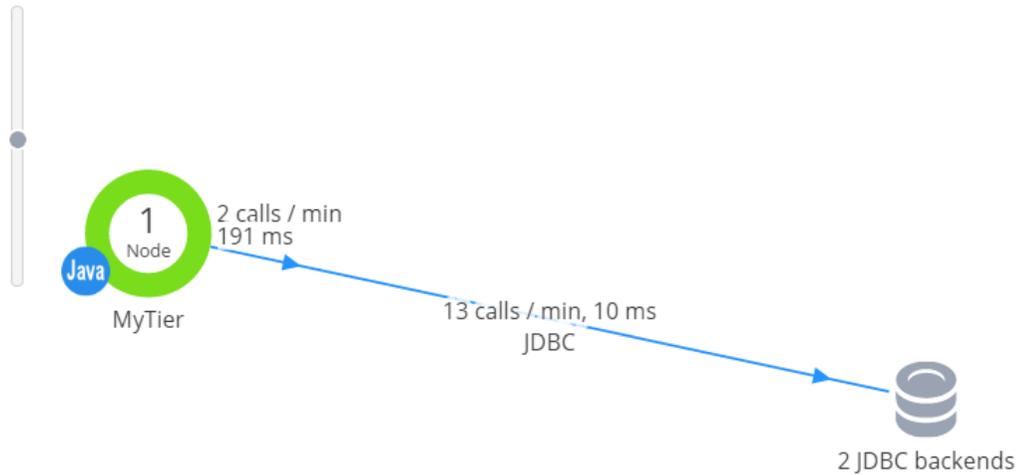
Application Flow マップ

サポート・システム

🔄
Baseline... ▾
last 6 hours ▾
?

Dashboard
Events
Top Business Transactions
Transaction Snapshots
Transaction Score
Actions ▾

Application Flow Map ▾



Events
No Events in selected time range

Business Transaction Health

 0 critical, 0 warning, 55 normal

Node Health

 0 critical, 0 warning, 1 normal

Transaction Scorecard

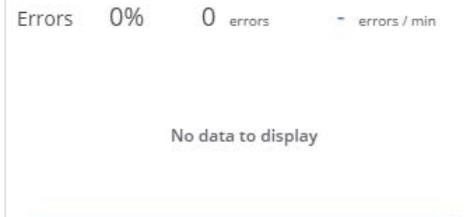
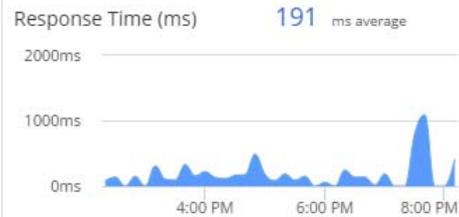
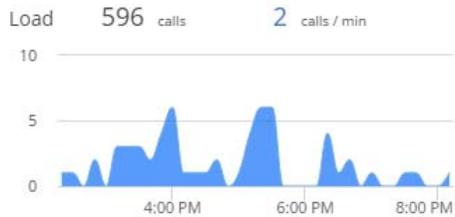
Normal		97.7 %	582
Slow		0.5 %	3
Very Slow		1.8 %	11
Stall		0.0 %	-
Errors		0.0 %	<1

Exceptions

Exceptions	- total	- / min
HTTP Error Codes	- total	- / min
Error Page Redirects	- total	- / min

Legend

Not comparing against Baseline data



トランザクションスナップショット

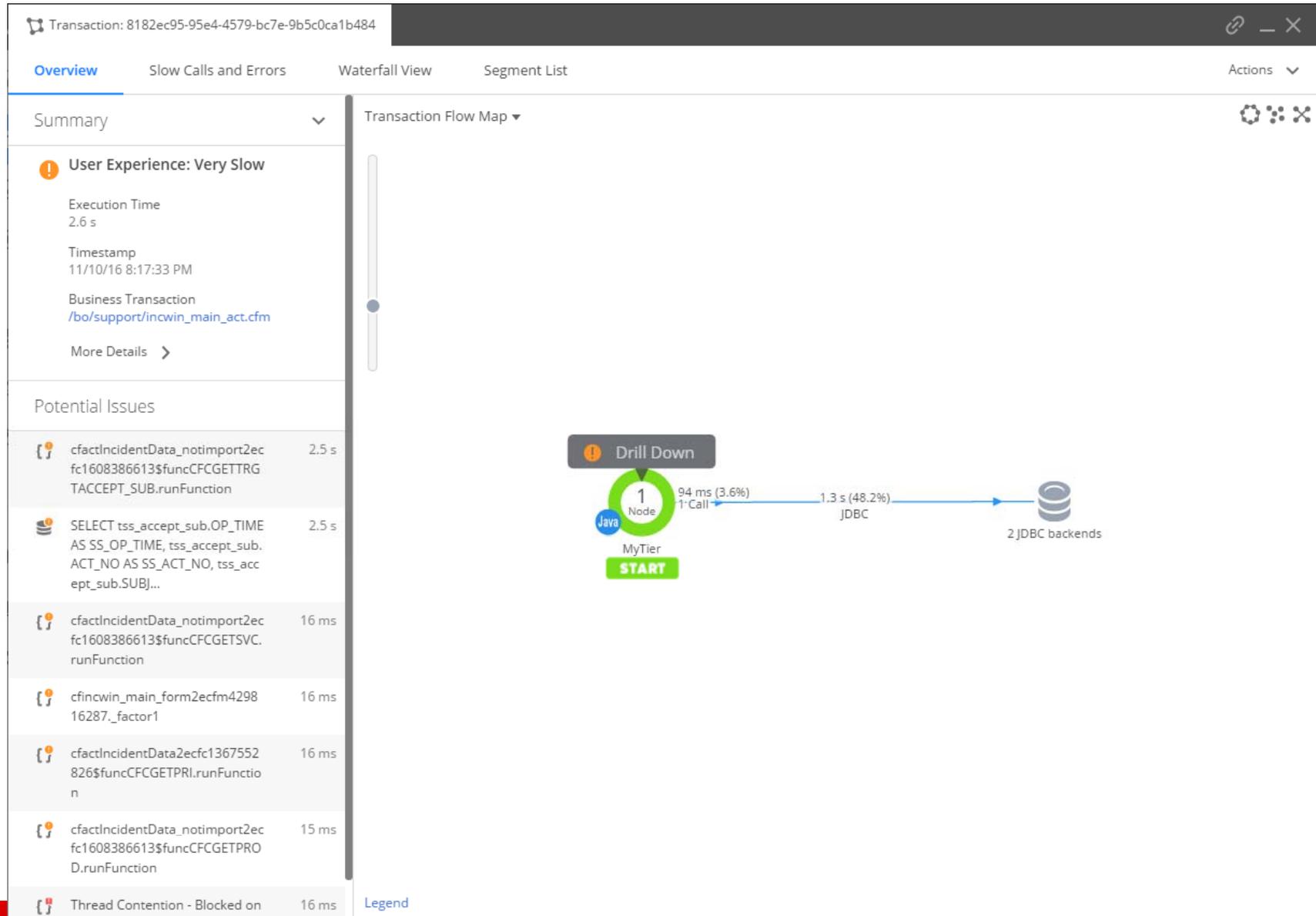
Dashboard Events Top Business Transactions **Transaction Snapshots** Transaction Score

All Snapshots Slow and Error Transactions Diagnostic Sessions Periodic Collection

 Details  Filters  Analyze  Actions  Configure

		Time ↓	Exe Time (ms)	URL	Business Transaction	Tier
		11/10/16 8:17:38 PM	0	/bo/support/template/cfc/actIncident...	/bo/support/template/cfc/actIncident...	MyTier
		11/10/16 8:17:33 PM	2,594	/bo/support/incwin_main_act.cfm	/bo/support/incwin_main_act.cfm	MyTier
		11/10/16 8:17:25 PM	141	/bo/support/template/cfc/actIncident...	/bo/support/template/cfc/actIncident...	MyTier
		11/10/16 8:17:23 PM	140	/bo/support/template/cfc/actIncident...	/bo/support/template/cfc/actIncident...	MyTier
		11/10/16 8:17:18 PM	172	/bo/support/template/cfc/actIncident...	/bo/support/template/cfc/actIncident...	MyTier
		11/10/16 8:17:18 PM	62	/bo/support/lo_center_act.cfm	/bo/support/lo_center_act.cfm	MyTier
		11/10/16 8:17:16 PM	15	/bo/support/lo_left_home.cfm	/bo/support/lo_left_home.cfm	MyTier
		11/10/16 8:17:14 PM	94	/bo/support/lo_left_contract.cfm	/bo/support/lo_left_contract.cfm	MyTier
		11/10/16 7:46:16 PM	0	/CFIDE/GraphData.cfm	/CFIDE/GraphData.cfm	MyTier

トランザクションフローマップ



Call Graph

Transaction: 8182ec95-95e4-4579-bc7e-9b5c0ca1b484 ! 2,594 ms - MyTier

Overview **Call Graph** Slow Calls & Errors DB & Remote Service Calls Server Data Collectors More Actions

Execution Time 2.6 s Wait Time 1 ms Block Time 0 ms Node SVR-DC-INT-003:134 Timestamp 11/10/16 8:17:33 PM

Details Filters Set Root Reset Root Export Showing 27 of 27

Name	Time (ms)	Percent %	Thread State	Exit Calls / Threads...
Servlet - BootstrapServlet:service	0 ms (self)	0%		
Servlet - CfmServlet:service:200	0 ms (self)	0%		
Application.cfc:ONREQUEST:222	0 ms (self)	0%		
incwin_main_act.cfm:35	0 ms (self)	0%		
incwin_main_var1.cfm:6	0 ms (self)	0%		
actIncidentData.cfc:CFCGETPRI:8	16 ms (self)	0.6%		JDBC (2)
actIncidentData.cfc:CFCGETSTAT:17	0 ms (self)	0%		JDBC
actIncidentData.cfc:CFCGETSTAFF:38	0 ms (self)	0%		JDBC
actIncidentData.cfc:CFCGETCFG:50	0 ms (self)	0%		JDBC
actIncidentData.cfc:CFCGETRES:63	0 ms (self)	0%		JDBC
actIncidentData.cfc:CFCGETACPT:76	0 ms (self)	0%		JDBC
actIncidentData.cfc:CFCGETACT:91	0 ms (self)	0%		JDBC
actIncidentData_notimport.cfc:CFCGETSVC:681	0 ms (self)	0%		JDBC
actIncidentData.cfc:CFCGETSC:111	0 ms (self)	0%		JDBC
actIncidentData_notimport.cfc:CFCGETPROD:637	15 ms (self)	0.6%		JDBC
actIncidentData_notimport.cfc:CFCGETTRGTACCEPT_SUB:188	0 ms (self)	0%		JDBC
actIncidentData_notimport.cfc:CFCGETSVC:681	0 ms (self)	0%		JDBC

Some packages have been excluded from this Call Graph

データベース・アクセスに時間がかかっている

【発生箇所】

インシデント編集ウィンドウのアクション一覧を取得するDB処理で発生している。

【原因】

インシデントのアクション履歴が多い(何度もユーザーやメーカーとやり取りしている)為、SQLの実行に、時間がかかっていた。

Execution Time 2.6 s Wait Time 1 ms Block Time 0 ms Node SVR-DC-INT-003:134 Timestamp 11/10/16 8:17:33 PM

👁 Details | 🗑 Filters | 📄 Set Root | ⌛ Reset Root | 📤 Export

Showing 27 of 27

Name	Time (ms)	Percent %	Thread State	Exit Calls / Threads...
actIncidentData.cfc:CFCGETACPT:76	0 ms (self)	0%		JDBC
actIncidentData.cfc:CFCGETACT:91	0 ms (self)	0%		JDBC
actIncidentData_notimport.cfc:CFCGETSVC:681	0 ms (self)	0%		JDBC
actIncidentData.cfc:CFCGETSC:111	0 ms (self)	0%		JDBC
actIncidentData_notimport.cfc:CFCGETPROD:637	15 ms (self)	0.6%		JDBC
actIncidentData_notimport.cfc:CFCGETTRGTACCEPT_SUB:188	0 ms (self)	0%		JDBC
actIncidentData_notimport.cfc:CFCGETSVC:681	0 ms (self)	0%		JDBC
incwin_main_var2.cfm:4	0 ms (self)	0%		
actIncidentData_notimport.cfc:CFCGETTRGTACCEPT_MAIN:161	0 ms (self)	0%		JDBC
actIncidentData_notimport.cfc:CFCGETSVC:681	0 ms (self)	0%		JDBC
actIncidentData_notimport.cfc:CFCGETTRGTACCEPT_SUB:188	2,531 ms (se...)	97.6%		JDBC
actIncidentData_notimport.cfc:CFCGETSVC:681	16 ms (self)	0.6%		JDBC
incwin_main_form.cfm:1	0 ms (self)	0%		
incwin_main_form.cfm:1	0 ms (self)	0%		
incwin_main_form.cfm:2	0 ms (self)	0%		
incwin_main_form.cfm:3	0 ms (self)	0%		
incwin_main_form.cfm:42	16 ms (self)	0.6%		

Some packages have been excluded from this Call Graph

時間のかかっているSQLの内容

runFunction:188 ×

<p>Name actIncidentData_notimport.cfc:CFCGETTRGTA CCEPT_SUB</p> <p>Type COLD_FUSION_FUNCTION</p> <p>Class cfactIncidentData_notimport2ecfc160838661 3\$funcCFCGETTRGTACCEPT_SUB</p> <p>Method runFunction</p> <p>Line Number 188</p>	<p>Self Time 2,531ms 97.6%</p> <div style="width: 100%; height: 10px; background: linear-gradient(to right, blue 97.6%, gray 97.6%);"></div> <p>Total Time 2,547ms 98.2%</p> <div style="width: 100%; height: 10px; background: linear-gradient(to right, orange 98.2%, gray 98.2%);"></div> <p>Details Block: 0 ms, Wait: 0 ms</p> <div style="width: 100%; height: 10px; background: linear-gradient(to right, gray 100%, gray 100%);"></div>
--	---

Exit Calls

Type	Details	Count	Time (ms)	% Time	From	To
JDBC	SELECT tss_accept_sub.OP_TIME AS SS_OP_TI...	1	2500	96.4%	MyTier	SUPPORT-...

JDBC Exit Call

Time 2,500ms

From MyTier

To SUPPORT-MySQL-192.168.103.6-5.5.39

Details SELECT tss_accept_sub.OP_TIME AS SS_OP_TIME, tss_accept_sub.ACT_NO AS SS_ACT_NO, tss_accept_sub.SUBJECT AS SS_SUBJECT, tss_accept_sub.ACCEPT_SUB_NO, tss_accept_sub.CONTENT AS SS_CONTENT, CONCAT(' ', tss_accept_sub. b.ANS_EM_NO) AS SS_ANS_EM_NO, DATE_FORMAT(tss_accept_sub.ANS_DATE, '%Y/%m/%d') AS SS_ANS_DATE, tss_acc ept_sub.DEL_FLAG, (select count(*) from tss_accept_sub_file where tss_accept_sub_file.DEL_FLAG < 1000 AND tss_acc ept_sub_file.ACCEPT_NO = tss_accept_sub.ACCEPT_NO AND tss_accept_sub_file.ACCEPT_SUB_NO = tss_accept_sub.A CCEPT_SUB_NO) AS EXT_ATTH FROM tss_accept_sub WHERE tss_accept_sub.ACCEPT_NO = ? AND tss_accept_sub.DEL _FLAG <? AND tss_accept_sub.DEL_FLAG >= ? AND tss_accept_sub.SUPPORT_NO = ? ORDER BY ACCEPT_SUB_NO

Properties

Query Type Query

Statement Type Prepared Statement

複数回データベースにアクセスしている

【発生箇所】

レポートページの表示に時間が掛っている。

【原因】

チャートの処理に時間が掛っているのかと思ったら、チャートの表示処理はそれほどでもなく、チャートの数値データを抽出しているクエリが1秒近くかかるものが7個あったのが原因でトータル10秒程度かかっていた。

Execution Time 10.3 s Node SVR-DC-INT-003:134 Timestamp 10/17/16 9:18:15 AM

Details Filters Set Root Reset Root Export Showing 623 of 623

Name	Time (ms)	Percent %	Exit Calls / Threads...
lo_center_report_inner_var1.cfm:65	0 ms (self)	0%	
actIncidentData.cfc:CFCGETCFG:50	15 ms (self)	0.1%	JDBC (2)
actIncidentData_notimport.cfc:CFCGETPROD:637	0 ms (self)	0%	JDBC
actIncidentData.cfc:CFCGETCFG:50	0 ms (self)	0%	JDBC
actIncidentData.cfc:CFCGETSTAFF:38	0 ms (self)	0%	JDBC
actIncidentData_report.cfc:CFCREPTTRGTACCEPT_MAIN:26	32 ms (self)	0.3%	JDBC
actIncidentData_report.cfc:CFCREPTTRGTACCEPT_RE:51	1,000 ms (se...)	9.7%	JDBC
actIncidentData_report.cfc:CFCREPTTRGTACCEPT_QA:76	984 ms (self)	9.6%	JDBC
actIncidentData_report.cfc:CFCREPTTRGTACCEPT_QA_2:102	969 ms (self)	9.4%	JDBC
actIncidentData_report.cfc:CFCREPTTRGTACCEPT_QA_6:127	968 ms (self)	9.4%	JDBC
actIncidentData_report.cfc:CFCREPTTRGTACCEPT_QA_186:152	954 ms (self)	9.3%	JDBC
actIncidentData_report.cfc:CFCREPTTRGTACCEPT_QA_ESC:177	1,015 ms (se...)	9.9%	JDBC
actIncidentData_report.cfc:CFCREPTTRGTACCEPT_MAIN_CLOSE:231	31 ms (self)	0.3%	JDBC
actIncidentData_report.cfc:CFCREPTTRGTACCEPT_QA_5:274	1,016 ms (se...)	9.9%	JDBC
lo_center_report_inner_2.cfm:49	203 ms (self)	2%	
lo_center_report_inner_2.cfm:73	94 ms (self)	0.9%	

約1秒かかっている7個のSQL

Some packages have been excluded from this Call Graph

サポートインシデントのレポート出力処理に4秒近く掛っている

【原因】

SQL処理は20ミリ秒以下で処理が終わっているが、その後で1秒ほど処理に時間が掛っている

<!-- X営業日未対応インシデント検索クエリ-->

<!-- X営業日未対応インシデント(エスカレーション)検索クエリ-->

<!-- 対応件数が多いリクエストの検索クエリ-->

⇒スプレッドシートオブジェクトを作成している所で時間が掛っている???

(現行設定では、Excel (POI) 処理は計測結果に出てこない模様)

ユーザー定義関数(SetListHeader)で1.7秒ほど時間が掛っている

Excelのセルにデータ埋め込み & セルのフォーマット処理で時間が掛っている??

こちらも Excel(POI)処理は計測結果に出てこない(設定次第?)

【今後】

・Excel(POI)処理の詳細を取れるようになればもう少し処理の詳細が分かる?

Transaction: 5df5db05-6fb2-4ab5-9d6f-2ce4cc2f9186 3,109 ms - MyTier

Overview **Call Graph** Slow Calls & Errors DB & Remote Service Calls Server Data Collectors More Actions

Execution Time 3.1 s Wait Time 1 ms Block Time 0 ms Node SVR-DC-INT-003:134 Timestamp 10/03/16 11:28:33 AM

Details Filters Set Root Reset Root Export Showing 14 of 14

Name	Time (ms)	Percent %	Thread State	Exit Calls / Threads...
Servlet - BootstrapServlet:service	16 ms (self)	0.5%		
Servlet - CfmServlet:service:200	0 ms (self)	0%		
Application.cfc:ONREQUEST:222	0 ms (self)	0%		
lo_center_act.cfm:195	62 ms (self)	2%		
lo_center_supreport.cfm:44	1,187 ms (se...)	38.2%		JDBC (5)
com.rsa.jsafe.provider.JS_SecureRandom:engineNextBytes	0 ms (self)	0%		
com.rsa.jsafe.crypto.c6:c	0 ms (self)	0%		
com.rsa.jsafe.crypto.c6:a	0 ms (self)	0%		
com.rsa.jsafe.crypto.c6:a	0 ms (self)	0%		
com.rsa.jsafe.provider.c7:engineGenerateSeed	16 ms (self)	0.5%		
lo_center_supreport.cfm:SETLISTHEADER:238	1,687 ms (se...)	54.3%		
lo_center_supreport.cfm:SETCELLQUERY:169	32 ms (self)	1%		
lo_center_supreport.cfm:SETLISTHEADER:245	93 ms (self)	3%		
lo_center_supreport.cfm:SETCELLQUERY:159	16 ms (self)	0.5%		

Some nodes may have been excluded from this Call Graph.

POIを使用してエクセルシートへの書き込み

POIの処理が表示できるように設定を変更して、再度取得してみた。

Transaction: cda88f64-572e-40b6-95cb-c6050865999c 3,469 ms - MyTier

Overview **Call Graph** Slow Calls & Errors DB & Remote Service Calls Server Data Collectors More Actions

Execution Time 3.5 s Node SVR-DC-INT-003:134 Timestamp 11/10/16 3:11:10 PM

Details Filters Set Root Reset Root Export Showing 108 of 108

Name	Time (ms)	Percent %	Exit Calls / Threads...
▶ Constructor of org.apache.poi.hpsf.Section:289	15 ms (total)	0.4%	
▼ lo_center_supreport.cfm:SETLISTHEADER:238	0 ms (self)	0%	
▼ org.apache.poi.hssf.usermodel.HSSFSheet:createDrawingPatriarch:68	0 ms (self)	0%	
▼ org.apache.poi.hssf.usermodel.HSSFSheet:createDrawingPatriarch:1492	0 ms (self)	0%	
org.apache.poi.hssf.model.Sheet:aggregateDrawingRecords:1481	15 ms (self)	0.4%	
org.apache.poi.hssf.usermodel.HSSFWorkbook:getCreationHelper:1689	16 ms (self)	0.5%	
▼ org.apache.poi.hssf.usermodel.HSSFWorkbook:createFont:86	0 ms (self)	0%	
▼ org.apache.poi.hssf.usermodel.HSSFWorkbook:createFont:1034	0 ms (self)	0%	
org.apache.poi.hssf.usermodel.HSSFWorkbook:getFontAt:1092	15 ms (self)	0.4%	
▼ org.apache.poi.hssf.util.HSSFColor:getIndexHash:56	0 ms (self)	0%	
▼ org.apache.poi.hssf.util.HSSFColor:createColorsByIndexMap:60	0 ms (self)	0%	
org.apache.poi.hssf.util.HSSFColor:getAllColors:119	63 ms (self)	1.8%	
▼ org.apache.poi.hssf.usermodel.HSSFSheet:autoSizeColumn:1683	1,640 ms (se...	47.3%	
org.apache.poi.hssf.usermodel.HSSFWorkbook:getFontAt:1087	16 ms (self)	0.5%	
▼ org.apache.poi.hssf.util.HSSFColor:...	0 ms (self)	0%	
▼ org.apache.poi.hssf.util.HSSFColor:...	0 ms (self)	0%	

Some packages have been excluded from this Call Graph

列幅を自動調節するメソッドで時間がかかっていた