

予兆検知による重大障害の未然防止

オンラインサービスを提供するうえでの課題

サービス利用者からの問い合わせで障害を認識

気づいたときにはサービス停止の重大障害

過度な閾値監視によるアラートストーム対応に疲弊

AppDynamicsが解決します！

サービス利用者が感じる体感の変化を**障害の予兆**と捉え、システム運用者が**プロアクティブ**に対処していくことで**重大障害を未然に防止**します。

初動対応



CPUやメモリ、ディスクI/Oの異常、データベースや外部システムの応答遅延などから発生するサービス利用者が感じるレスポンスの変化を監視、普段との乖離を予兆と捉え、影響が広がる前に初動対応に着手できます。

影響調査



複雑に連携するアプリケーションや外部システム連携、マルチ/ハイブリッドクラウド環境など、システム全体を俯瞰し、どのアプリケーションで、どの業務処理で、問題が起こっているか即時に把握できます。

原因調査



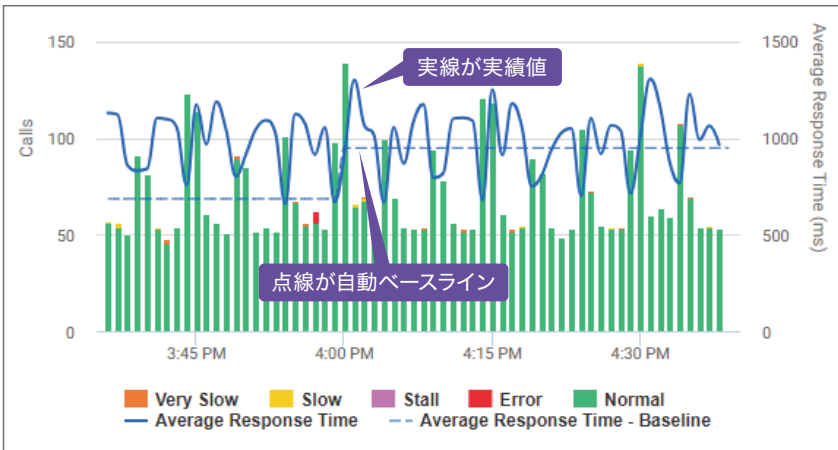
プログラム内部を可視化し、コード単位での処理時間やSQL実行内容、インフラリソース状況まで、事象発生時の情報を記録、数クリックで原因箇所を特定できます。

障害対応



障害が拡大する前にシステム運用者がプロアクティブに復旧作業にあたることで、ビジネスインパクトになる重大障害の未然防止に役立ちます。

自動ベースラインによる予兆検知でシステムの変化を見逃さない

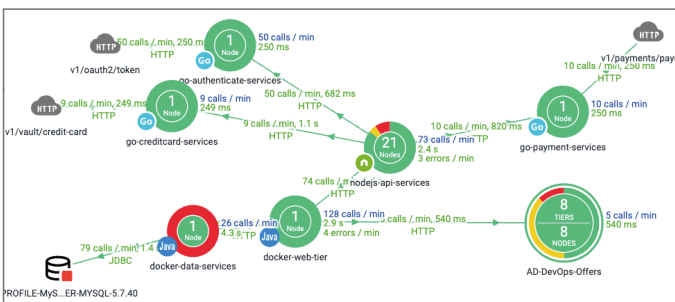


AppDynamicsは、機械学習により自動でベースラインを調整します。

日次、週次、月次、カスタム期間でレスポンスタイムの「正常時」を計測、ベースラインからの乖離を処理ごとに捕捉し、Normal/Slow/Very Slowと自動分類し、アラートを発報。

レスポンスタイム以外にも、CPUやメモリ、ネットワーク、ディスクI/Oなど検知できるメトリックをすべてベースラインとして利用可能。

システム全体から影響箇所を把握



Agentを導入するだけで、複雑な構成のアプリケーションでも全体的な監視が可能。赤やオレンジで問題が発生している箇所、冗長化されたノードの障害も容易に特定。

ビジネスランザクションごとに影響を把握

Name	Health
ログイン	🟢
サービス検索	🔴
予約入力	🟡
ユーザーデータ照会	🟢
支払い方法指定	🟢
最終確認	🟢
受付完了	🔴

アプリケーションが処理するランザクションを自動識別し、影響を受けている業務をヘルス状態として可視化。ランザクション名に日本語名を設定することで視認性も向上。

発生した事象をもれなく記録

Time	Exe Time (ms) ↓
10/06/23 2:06:30 PM	49,149
10/06/23 9:59:52 AM	46,447
10/06/23 2:53:22 PM	44,773
10/06/23 2:05:07 PM	44,352
10/06/23 10:35:56 AM	
10/06/23 11:16:14 AM	
10/06/23 10:00:50 AM	
10/06/23 8:54:37 AM	
10/06/23 3:42:45 PM	

遅いランザクションをスナップショットとして詳細に記録。再現テストをせず、発生事象を確認可能。



コードレベルで原因箇所を特定

Name	Time (ms)	Percent %	Thread State	Exit Calls / Threads *
Servlet - javax.servlet doPost	2 ms (self)	0%		
Web Service - com.sun.xml.ws.server.incomingTube2Invoke149	0 ms (self)	0%		
Spring Bean - orderService.createOrder27	0 ms (self)	0%		
Spring Bean - Order2processor.createOrder59	0 ms (self)	0%		
Spring Bean - queryExecutor.executeTraceQuery51	8,937 ms (self)	100%		JDBC

JDBC Exit Call

Time 8.934ms

From InventoryServices

To XE-Oracle DB-ORACLE-DB-1521

Details

```
SELECT COMPANY, CITY, COUNTRY, CONTACT, PHONE, TO_CHAR(SUM(ORDERTOTAL), '0999999999999999') AS VALUE, SHIPPED FROM | SELECT C.COMPANYNAME AS COMPANY, C.CITY AS CITY, C.COUNTRY AS COUNTRY, C.CONTACTNAME AS CONTACT, C.PHONE AS PHONE, O.ORDERS AS ORD FROM WORLD.ORDERS O, JOIN WORLD.CUSTOMERS C ON C.CUSTOMERID = O.CUSTOMERID WHERE C.CUSTOMERID = ? JOIN (SELECT O.ORDEROID AS OID2, SUM(O.QUANTITY * O.DUNITPRICE * O.DISCOUNT) AS ORDERTOTAL FROM WORLD.ORDERS O, JOIN WORLD.CUSTOMERS C ON C.CUSTOMERID = O.ORDEROID WHERE O.CUSTOMERID = (SELECT * FROM TABLE(WORLD.GET_NUM_TP(?))) GROUP BY O.ORDEROID) O ON O.ID1 = O.OID2 GROUP BY COMPANY, CITY, COUNTRY, CONTACT, PHONE
```

問題箇所をコードレベルで特定可能。事象発生時のSQLや外部リクエスト先、インフラリソースまで一気通貫で確認、原因切り分けを支援。

※AppDynamicsは、Cisco Systems, Inc.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。※本カタログ中の会社名、商品名は各社の商標、または登録商標です。※本文中および図中では、TMマーク、®マークは表記していません。※製品の仕様は、改良のため、予告なく変更する場合があります。※本製品を輸出される場合には、外国為替及び外国貿易法ならびに米国の輸出管理関連法規などの規制をご確認のうえ、必要な手続きをお取りください。なお、ご不明な場合は、当社担当営業にお問い合わせください。※本カタログ中の情報は、カタログ作成時点のものです。

