

Addressing IT operations analytics (ITOA)
use cases with AppDynamics

APPDYNAMICS

Addressing IT operations analytics (ITOA) use cases with AppDynamics

If you can't see it, you can't manage it - ITOA use case #1	4
Stop troubleshooting based on hunches and intuition - ITOA use case #2	9
Prioritize with business context - ITOA use case #3.....	15
What is the business impact? - ITOA use case #4	17
Create an attack plan - ITOA use case #5	19
Conclusion	23

If you can't see it, you can't manage it
- ITOA use case #1

If you can't see it, you can't manage it - ITOA use case #1

“There was 5 exabytes of information created between the dawn of civilization through 2003, but that much information is now created every 2 days, and the pace is increasing...,” - Eric Schmidt, Former CEO, Google.

If IT leaders hadn't already heard Schmidt's famous quotation, today they are definitely facing the challenge he describes. Gone are the days when IT leaders were tasked with just keeping an organization running, now IT teams are charged with driving innovation. As businesses become defined by the software that runs them, IT leaders must not only collect and try to make sense of the increasing amount of information these systems generate, but leverage this data as a competitive advantage in the marketplace. This type of competitive advantage may come in many forms, but generally speaking, the more IT leaders know about their environments and the ways end users interact with them, the better off they (and the business) will be. Gleaning this type of insight from IT environments is what analysts refer to as IT Operations Analytics (ITOA). ITOA solutions collect the structured and unstructured data generated by IT environments, process that data, and display the information in an actionable way so operations teams can make better informed decisions in real-time. I'd like to discuss five common ITOA use cases we see across our customer base in this series, starting with visualizing your environment. In the rest of this series I'll examine each of the other use cases and describe how a solution like the Application Intelligence Platform can address each and in turn provide value for operations teams.

The five common ITOA use cases I'll delve into are:

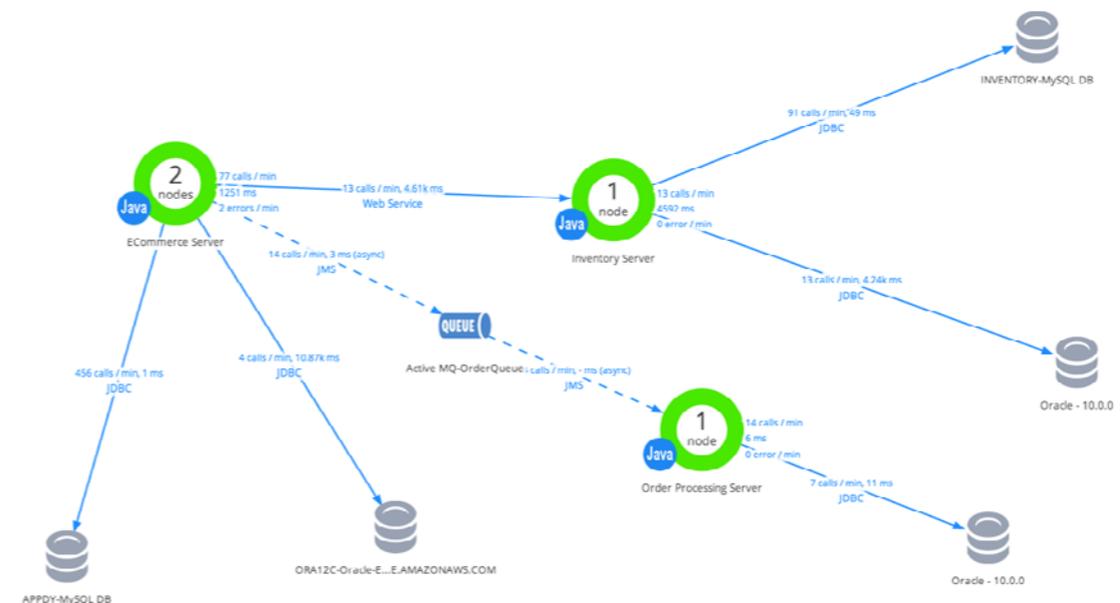
- Visualize the environment
- Rapid troubleshooting
- Prioritize issues and opportunities
- Analyze business impact
- Create action plans

Visualizing the environment

The first use case refers to the ability for an ITOA system to model infrastructure and / or the application stack being monitored. These models vary in nature but oftentimes are topological representations of the environment. Being able to visualize the application environment and see the dependencies is an important foundation for the rest of the use cases on this list.

In the [Summer '14 release announcement blog](#), we highlighted the enhancements we've made in regard to our [flow maps](#), which is the visual representation of the application environment, including application servers, databases, web services, and more.

What's great about the AppDynamics approach is that this flow map is discovered automatically out of the box, unlike legacy monitoring solutions that require significant manual configuration to get the same kind of view. We also automatically adjust this flow map on the fly when your application changes (re-architected app, code release, etc.). Because we know all the common entry and exit points of each node, we simply tag and trace the paths the different user requests take to paint a picture of the flow of requests and all the interactions between different components inside the application. Most customers see something like the flow map below within minutes of installing AppDynamics in their environment.

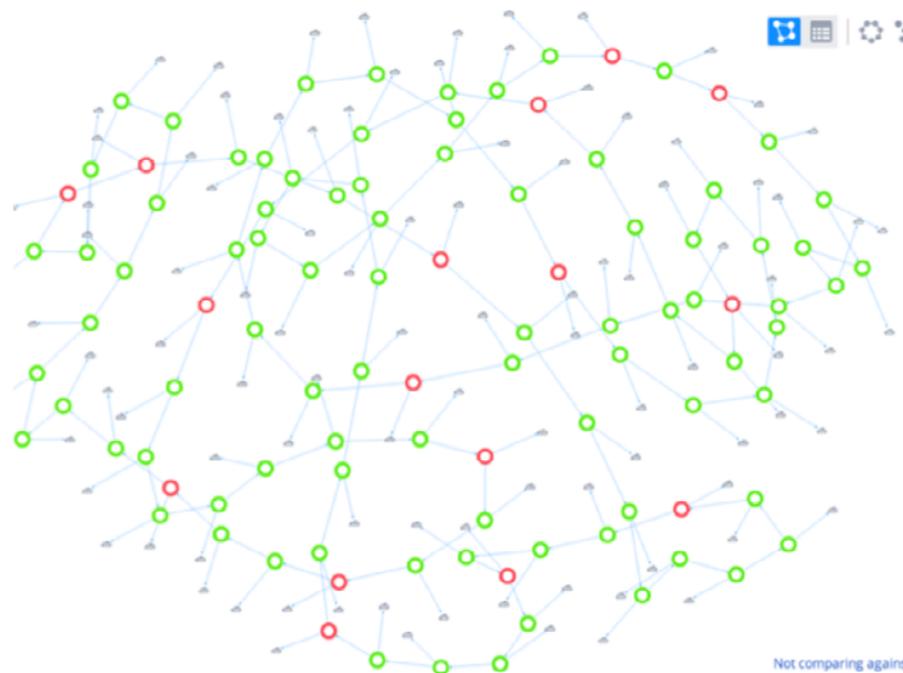


If you can't see it, you can't manage it - ITOA use case #1 (cont'd)

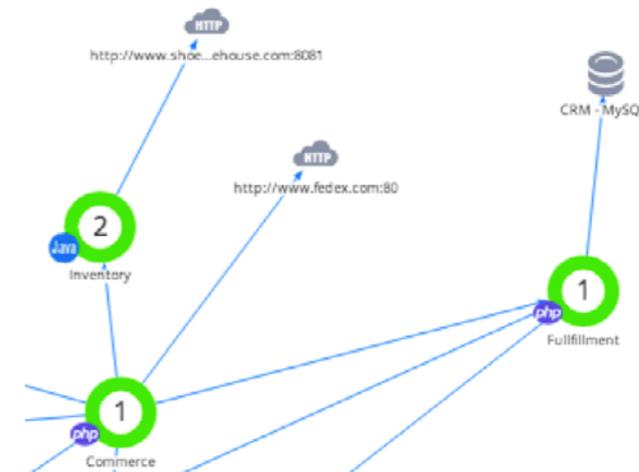
Now a flow map like this is obviously very valuable, but what happens when the application environment is very large and complex? How does this kind of view scale for the kinds of enterprise applications many AppDynamics customers have deployed? Environments with thousands of nodes and potentially hundreds of tiers? Luckily for our customers, the Application Intelligence Platform was built from the ground up to handle these kinds of environments with ease. There are two characteristics of our flow maps that enable operations teams to manage flow maps of large-scale application performance management deployments; self-aggregation and self-organizing layouts.

Self-aggregation refers to our powerful algorithms that make these complex environments more manageable by condensing and expanding the visualization to enable intelligent zooming in and zooming out of the topology of the application. This allows us to automatically deliver the right level of application health indicators to match the zoom level.

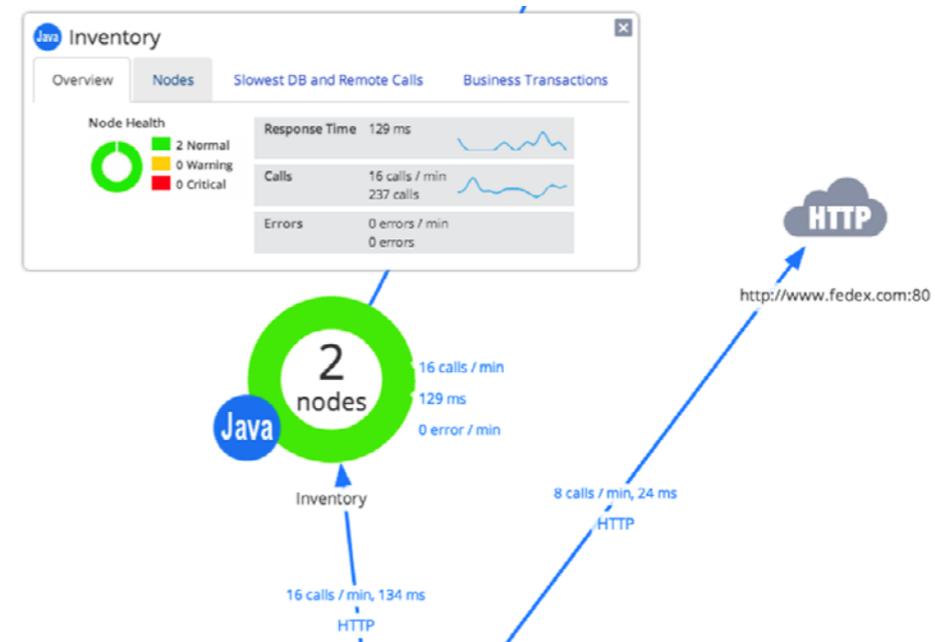
For example, this is what a complex application could look like when zoomed all the way out:



As one zooms in, relevant metrics information becomes visible:



Until you are zoomed all the way in on a particular tier and can see all of the associated metrics you'd care about:

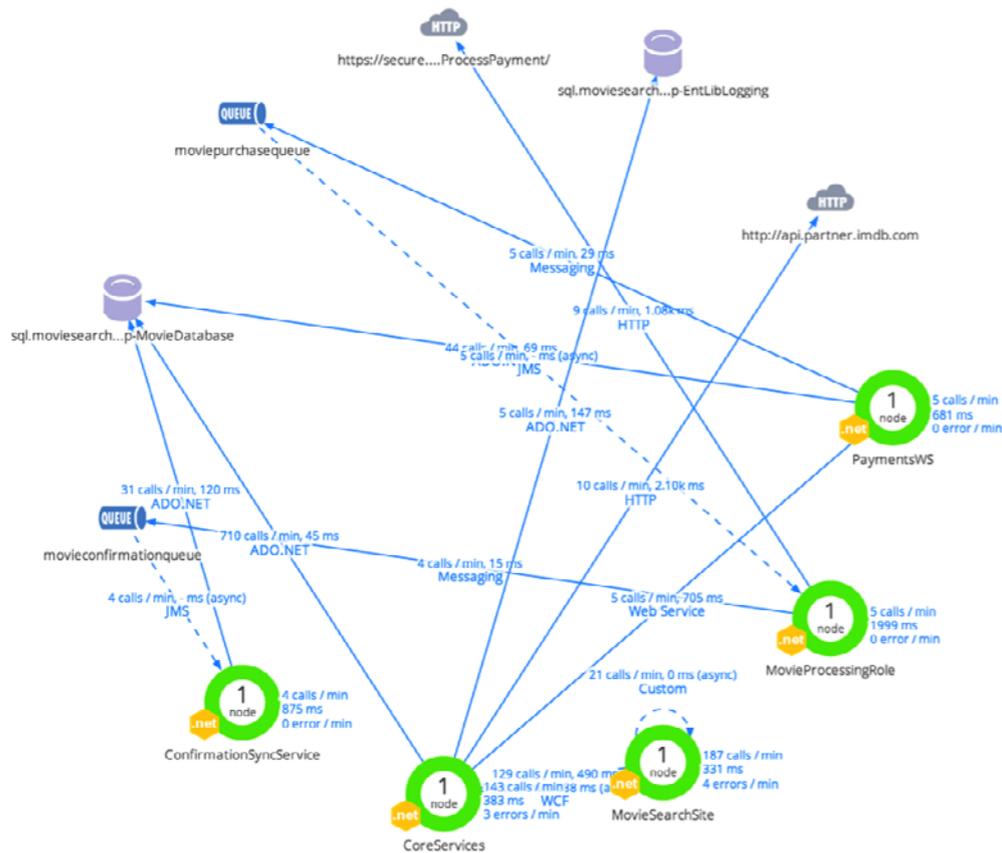


If you can't see it, you can't manage it - ITOA use case #1 (cont'd)

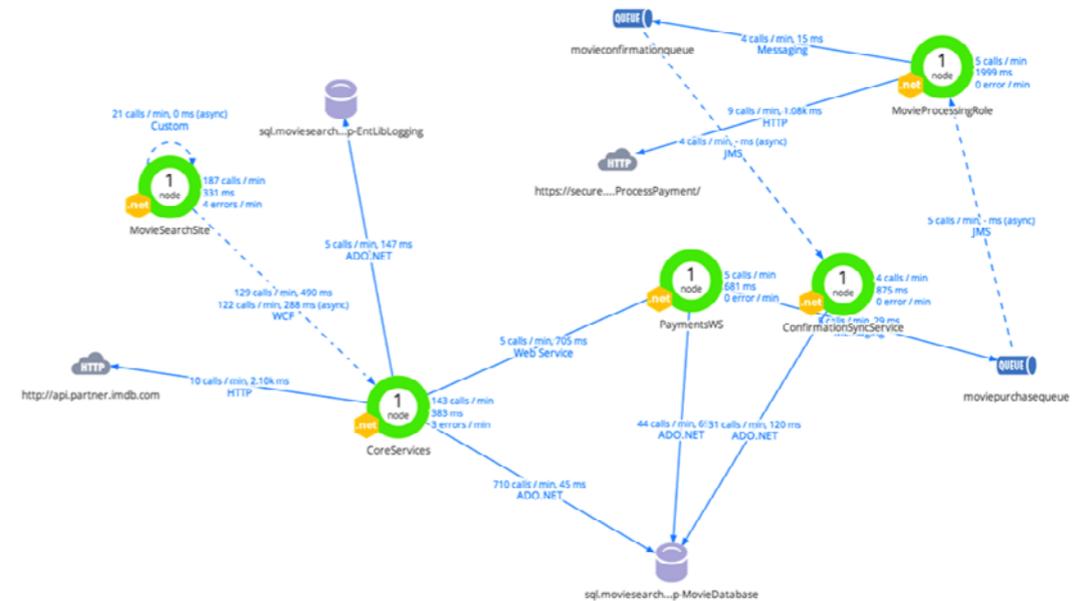
The ability to iterate back and forth between a macro-level view of the application and a close-up of a particular part of the environment gives operations teams the visibility they need to understand exactly how an application functions and how the different components interact with each other.

Self-organizing layouts relates to our ability to automatically format the service and tier dependencies by using auto-grouping heuristics to dynamically determine tier and node weightages. By leveraging static data (like application tier profiles) and dynamic KPIs (like transaction response times) we organize the business-critical tiers in a way that brings the most important parts of the application to the forefront depending on the type of layout you prefer.

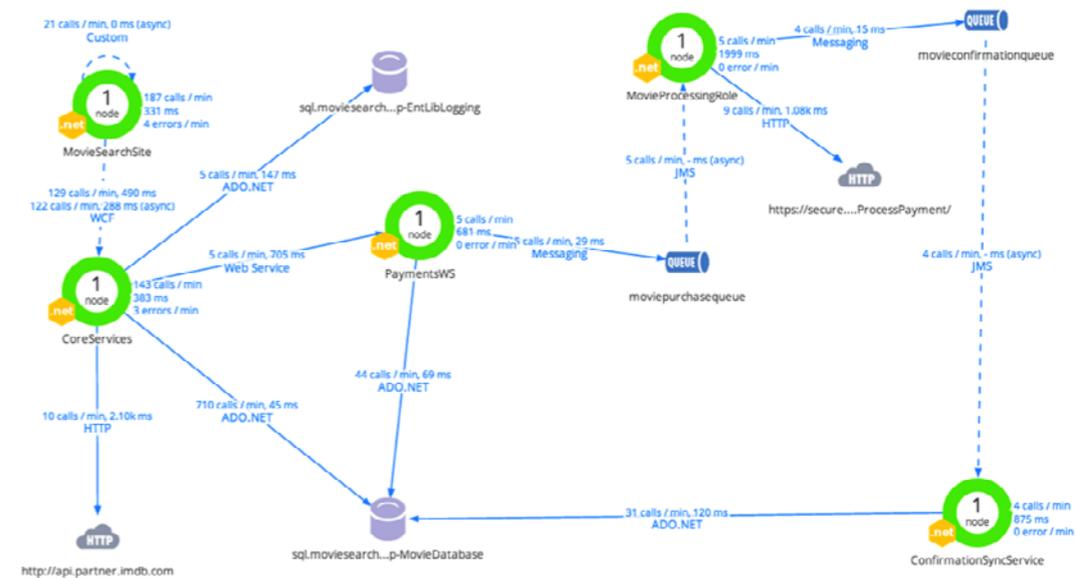
One can automatically group the flow map into a circular view:



You can let AppDynamics suggest a layout:

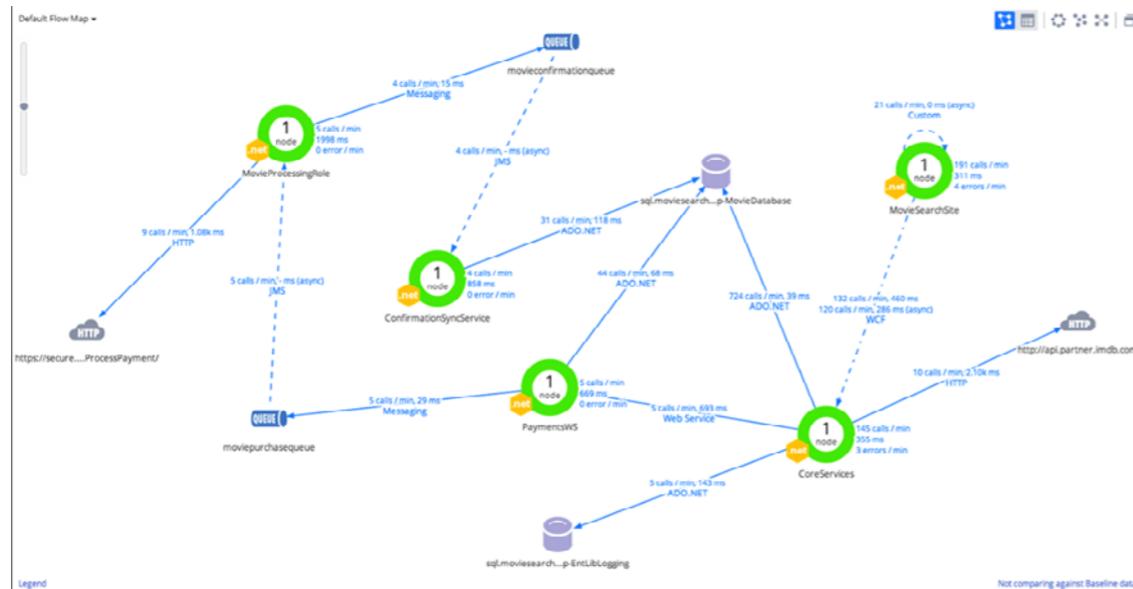


You can create a custom layout just by dragging and dropping individual components:



If you can't see it, you can't manage it - ITOA use case #1 (cont'd)

And you can auto-fit your layout to the screen for efficient zooming in / out:



As you can see, AppDynamics provides a great way for IT Operations teams to discover and visualize their application environment. We automatically map the application out of the box, we provide flexible layout options so customers can customize the view to their liking, and offer a way for Ops teams to understand how different applications interact with each other.

You've seen how AppDynamics can visualize individual applications, but what if, like many of our large enterprise customers, you have many different complex applications that have dependencies on one or more other applications? How does one obtain a data-center view to understand, at a high level, what application health looks like across all applications?

With the cross-app business flow feature, customers can do just that. AppDynamics even supports role-based access control (RBAC) so administrators can limit user access to a particular application. We allow customers to group, define, and limit access to applications however makes the most sense for their individual environments and for their business.

Stop troubleshooting based on hunches
and intuition - ITOA use case #2

Stop troubleshooting based on hunches and intuition - ITOA use case #2

In this chapter I'm going to focus on the second common ITOA use case, rapid troubleshooting.

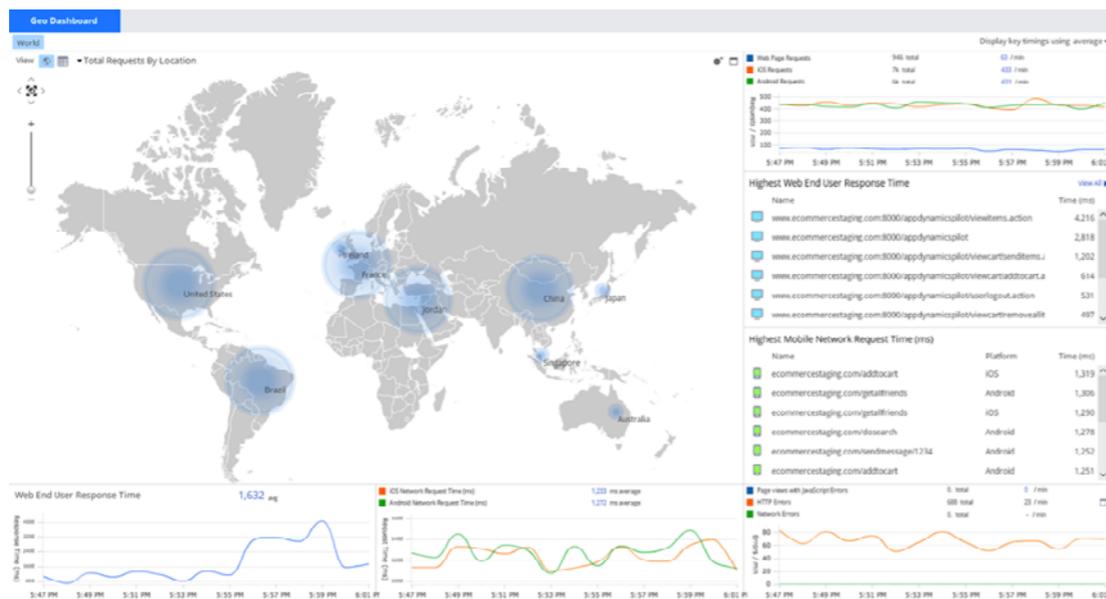
Rapid troubleshooting

Once you can visualize the inter-dependencies inside your environment, the second common ITOA use case involves leveraging this visibility to solve operational problems. Being able to pinpoint root cause of an issue using data generated from your application environment is of the utmost importance for businesses whose livelihood depends on these types of applications. Typically Operations teams troubleshoot issues that arise using their instincts and knowledge of past events, however with AppDynamics' unified monitoring approach, Operations teams can triage and isolate issues faster than ever before using a common view of application performance. Quickly knowing what an issue **ISN'T** is almost as valuable as knowing what it **IS** in these situations where every second counts.

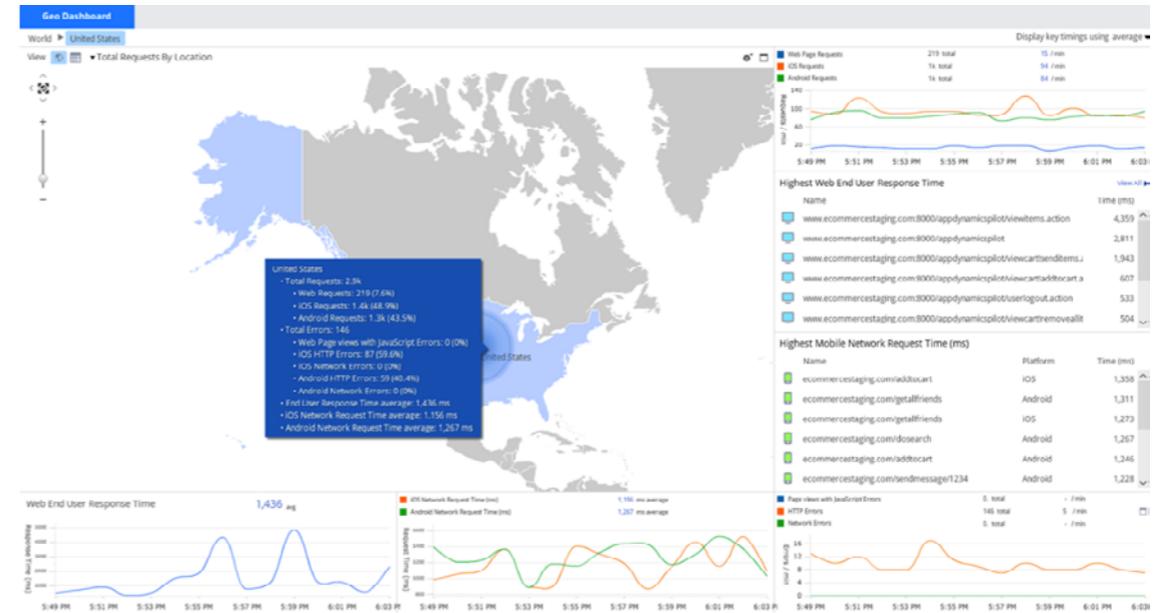
AppDynamics offers multiple ways for customers to isolate and identify the root-cause of application performance issues in real-time. I'll walk through some examples of how customers can discover root cause using several different views inside the **Application Intelligence Platform** starting the investigation on the front end, going through to the server side, and wrapping things up with the database backends.

Real-User Monitoring (RUM):

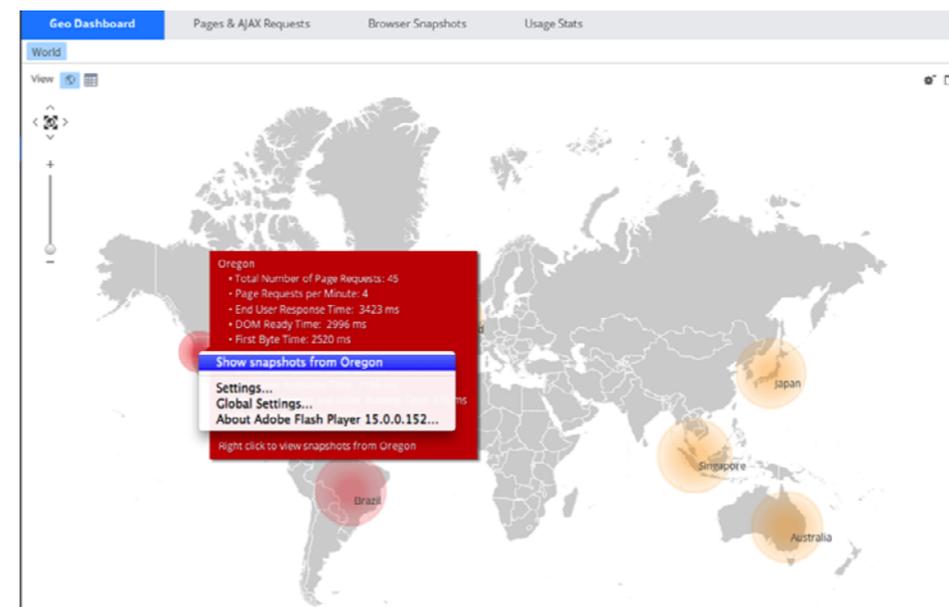
Customers can view the high-level real-user load and health by geography:



By clicking on a geo, one can filter the data associated with just that geography.

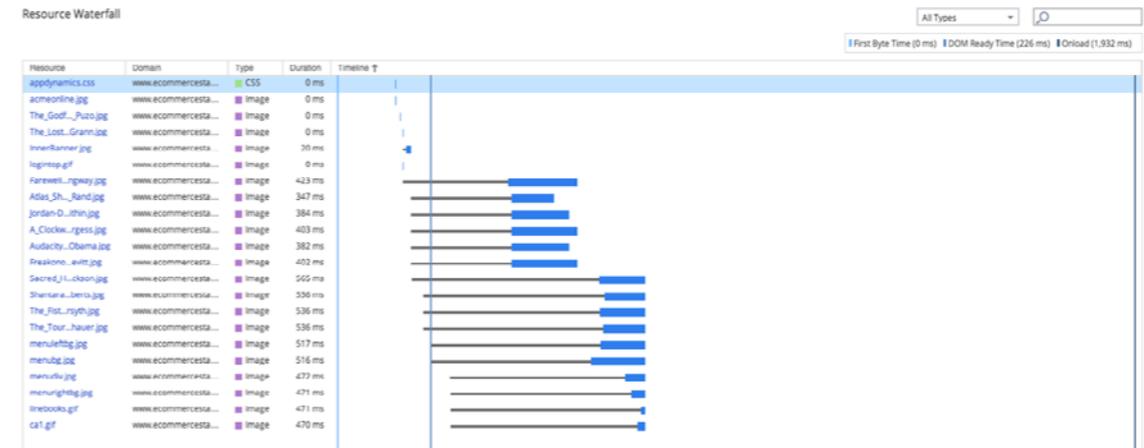
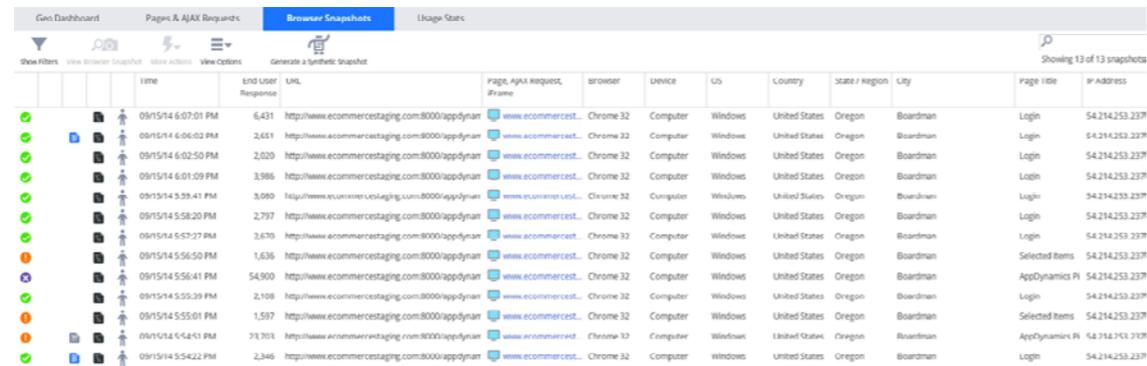


When viewing Browser RUM data, one can right click and view all **browser snapshots** from a particular region:

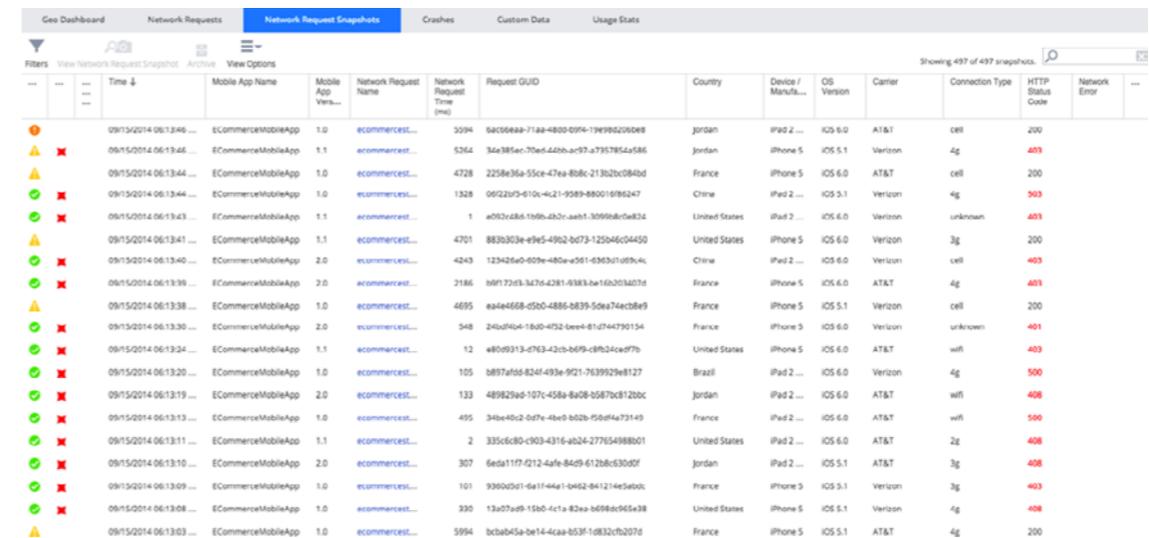
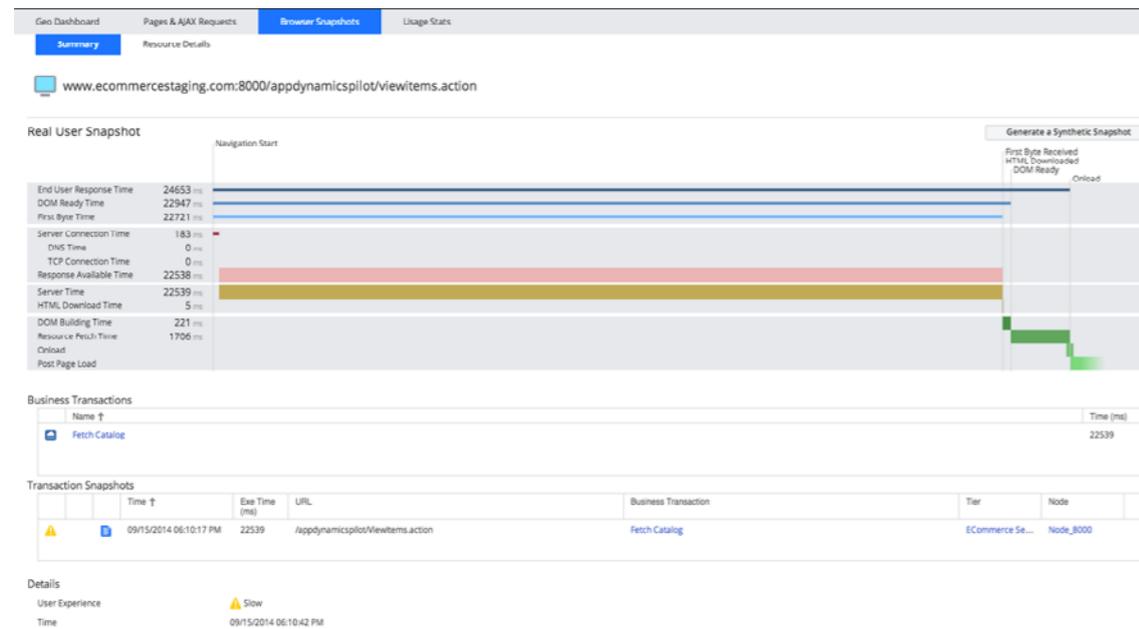


Stop troubleshooting based on hunches and intuition - ITOA use case #2 (cont'd)

These browser snapshots provide a wealth of information including the performance of the different components of the browser (like pages, AJAX requests, and iFrames), associated business transactions, and resource timing for the snapshot.

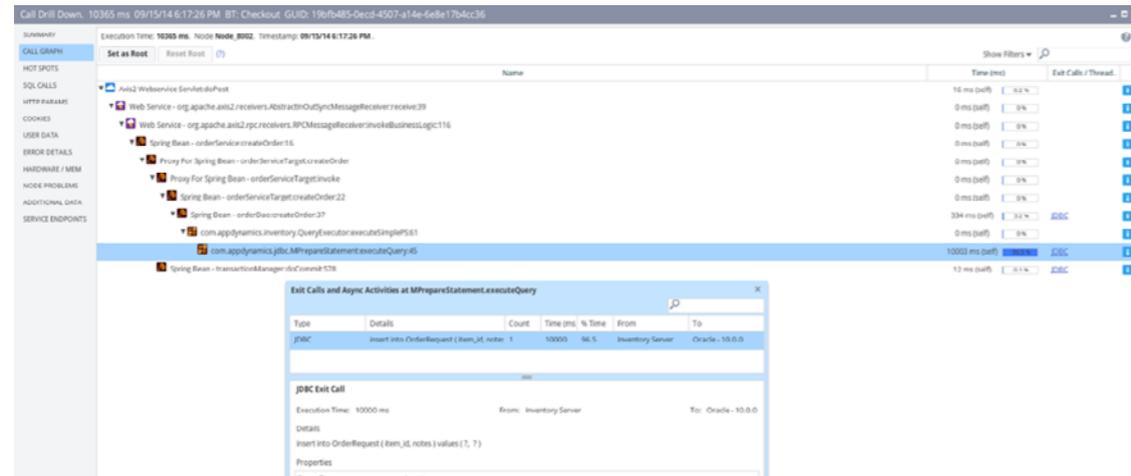


When looking at Mobile RUM data, you can view network request snapshots to analyze what happened when the device called out to the application over the network...

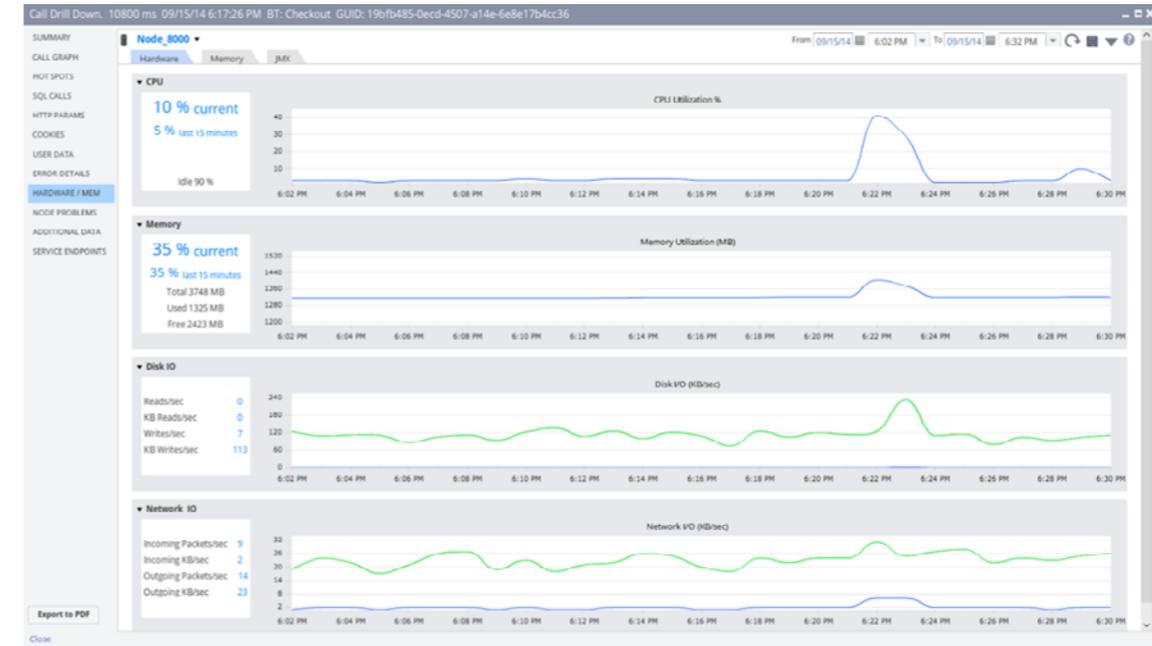


Stop troubleshooting based on hunches and intuition - ITOA use case #2 (cont'd)

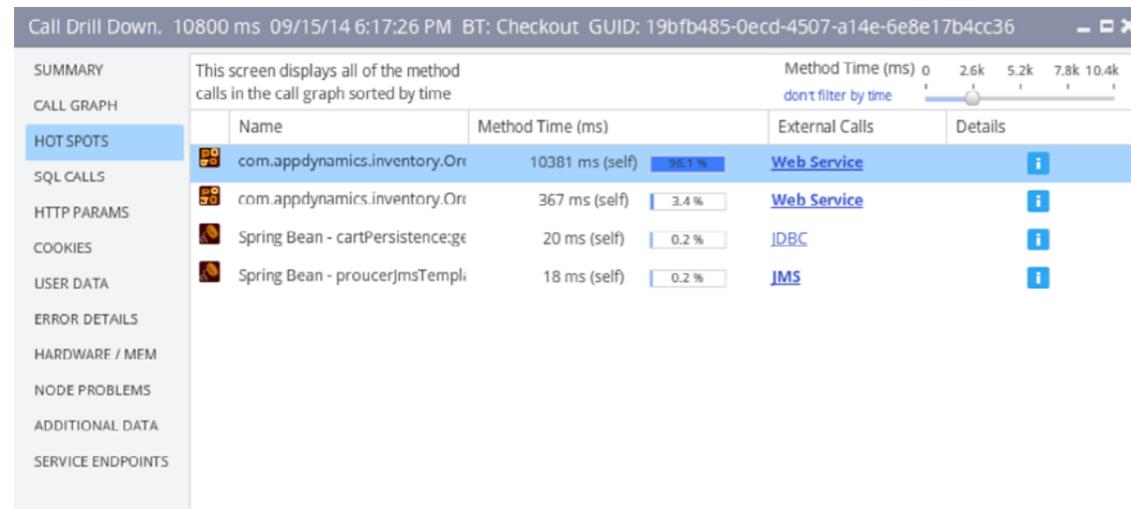
By drilling down into the call graph, one can isolate the exact class, method, or web service that is a performance bottleneck for a particular user request:



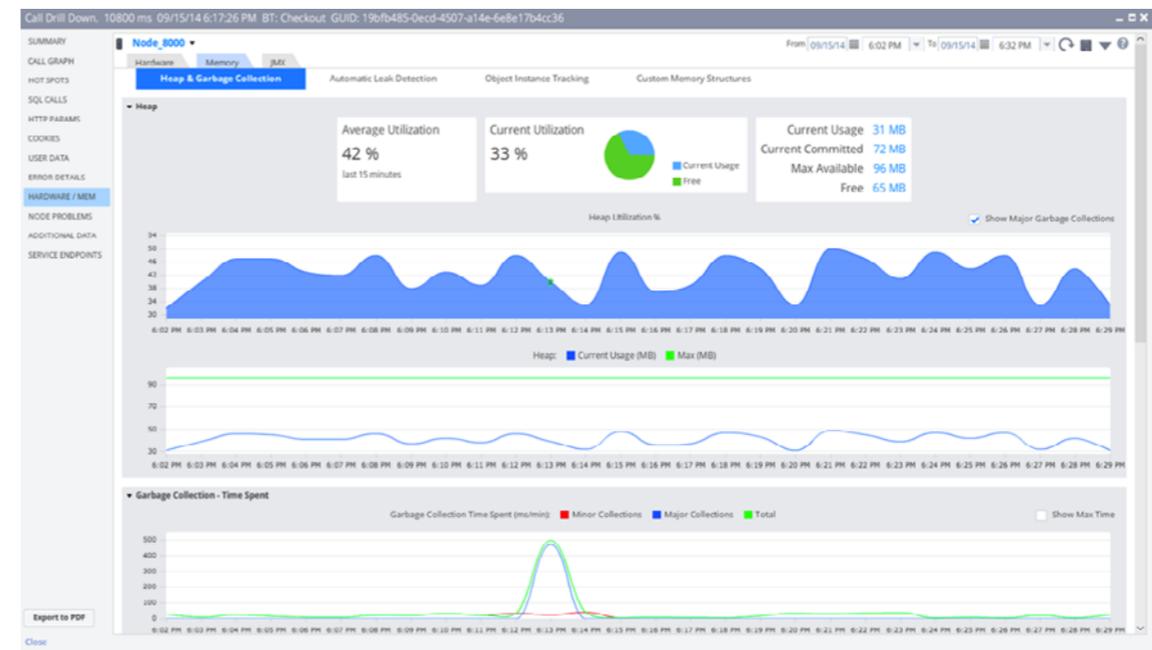
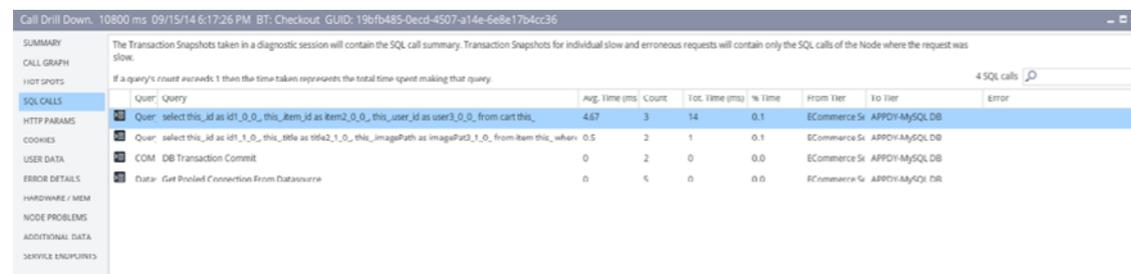
What was going on in the hardware and memory during this time? We automatically correlate and show those metrics too:



We also isolate the hot spots of this particular snapshot:



And show things like top SQL calls that could potentially be slowing down the application from a database perspective:



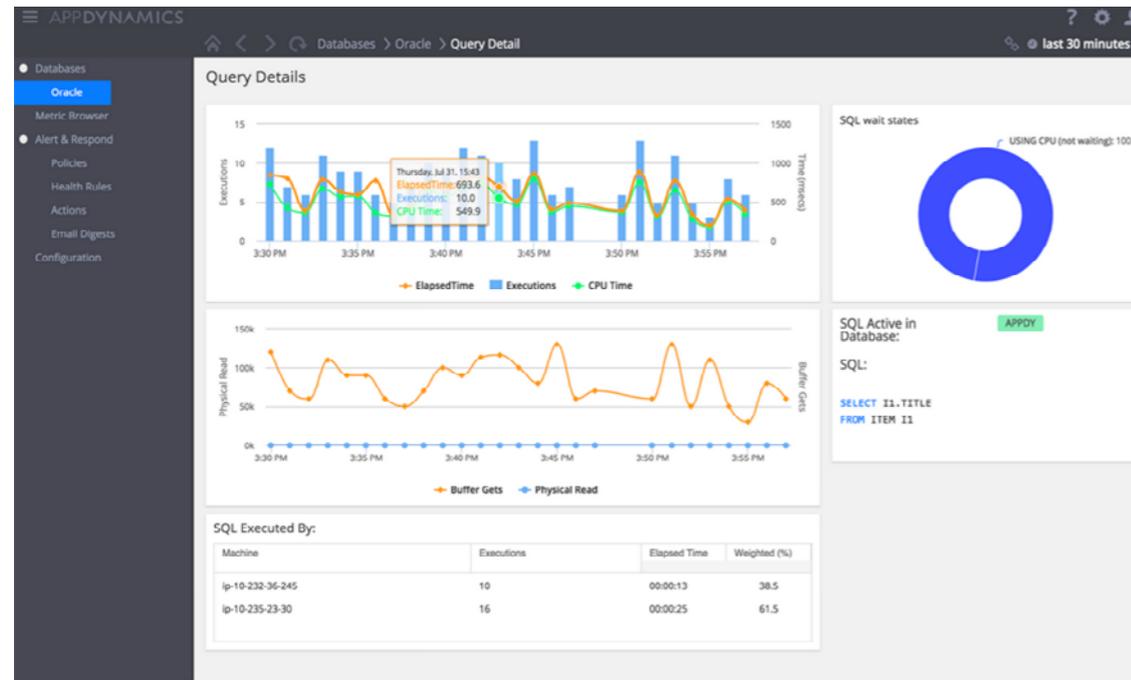
Stop troubleshooting based on hunches and intuition - ITOA use case #2 (cont'd)

Database Monitoring:

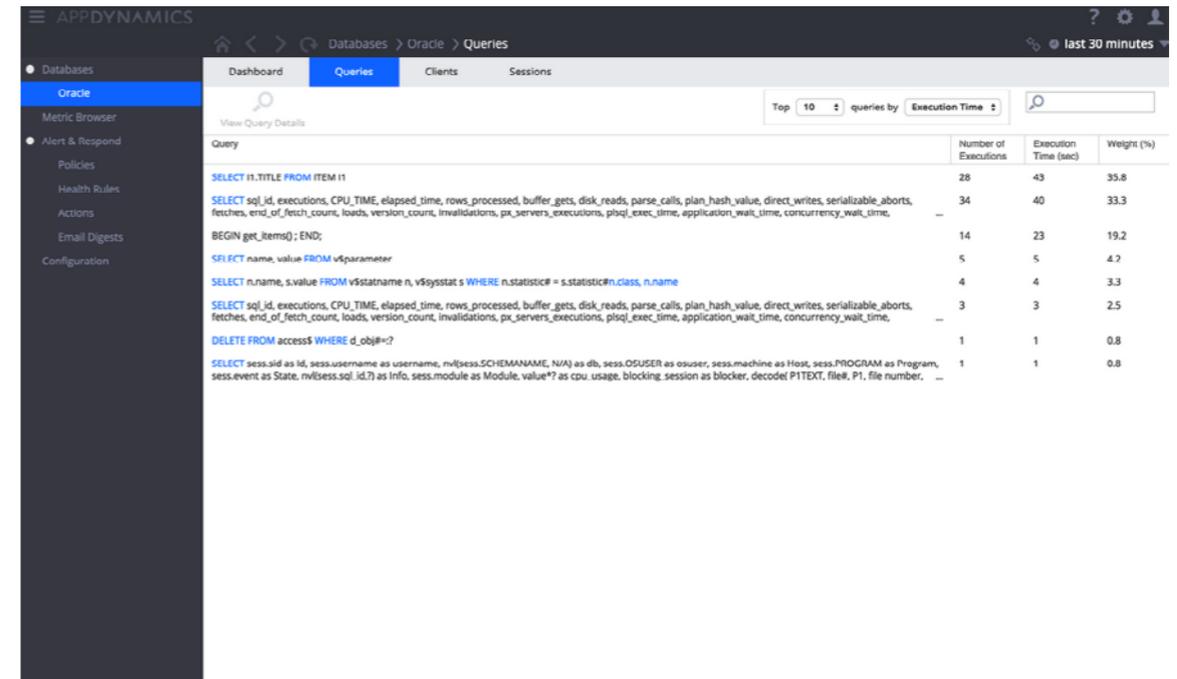
What if all signs point to an issue in the database as the root cause? AppDynamics also offers a [database monitoring](#) product module that provides granular visibility inside your database environment.

With our database monitoring product module, ops teams can now drill into the databases themselves to gain information about:

Detailed query analysis:



Top query analysis:



As you can see, the Application Intelligence Platform provides multiple ways for customers to leverage operational analytics to isolate issues and drill down to root cause from multiple dimensions. No matter where you start in the troubleshooting process, AppDynamics provides the flexibility to iterate quickly and drill up, down, and across the application environment, with all data automatically correlated, so Operations teams can find the root-cause as fast as possible.

Prioritize with business context - ITOA
use case #3

Prioritize with business context - ITOA use case #3

I recommend spending some time familiarizing yourself with the first three use cases I've highlighted, they lay some important groundwork for the rest of this series. With that, let's dive into use case #3 — prioritizing issues and opportunities.

Prioritizing issues and opportunities

Once operations teams isolate the root-cause of an application performance issue, the next step is to determine how to rank and prioritize these different issues. Oftentimes these fixes are simply prioritized in the order in which they came into the support team. However, what if the most recent root-cause that's been discovered is one that is directly impacting revenue, not the support ticket that was filed first?

Let's look at an example to illustrate this point. Put yourself in the shoes of an application support person and look at this short list of open issues that are in your queue:

- November 1, 2014 - Issue #1075 - Open for 14 days
- November 10, 2014 - Issue #1091 - Open for 4 days
- November 12, 2014 - Issue #1087 - Open for 2 days

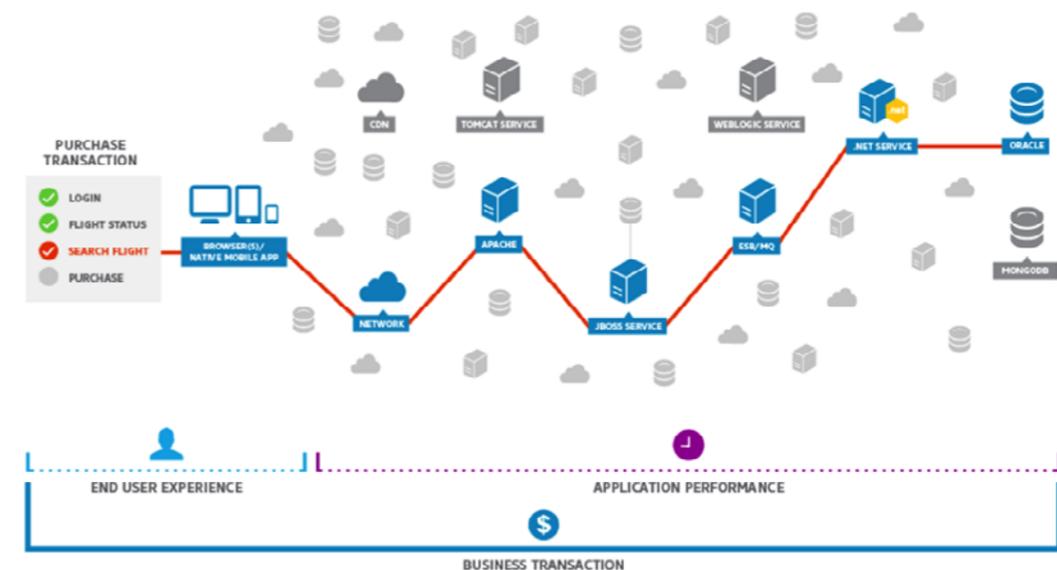
Which one do you tackle first? Most likely the one that has been open the longest, right? Prioritizing that ticket will help you, as an application support person, improve the KPIs you get measured on like average time taken to close out a ticket.

Now let's take another look at that same list but this time we will add in one data point that indicates the business context of the particular issue.

- November 1, 2014 - Issue #1075 - Open for 14 days - Related to 'update profile' function
- November 10, 2014 - Issue #1091 - Open for 4 days - Related to 'search products' function
- November 12, 2014 - Issue #1087 - Open for 2 days - Related to 'checkout' function

Now which one would you prioritize? A segment of users not being able to update their profiles or search for certain products is obviously not good for the end user experience, but if people are having issues checking out of your application, that means revenue is being directly impacted. Updating preferences in a profile is not nearly important to the business as making sure money is flowing in. That business context allows operations teams to focus their efforts on the issue with the largest potential business impact. In the Application Intelligence Platform, that business context is what we refer to as a business transaction.

We Understand Transactions From the Start



Simply put, a business transaction is a type of user request. AppDynamics automatically buckets these different types of user requests and baselines their normal behavior out of the box. Actions like 'submit order', 'get quote', 'login', 'add to cart' are all examples of business transactions.

Getting this business context, and automatically correlating it with the different performance issues that occur in the application, give operations teams that additional data point they need to stack rank remediation effort and tackle the large problems that are impacting the business first. And because AppDynamics dynamically baselines each business transaction separately, you'll know when a certain business transaction breaches what we've defined as 'normal' performance. This allows Operations teams to create alerts and other automated actions when a business transaction has abnormal performance. For example, you could automatically create an incident in your ticketing system ([ServiceNow](#), [PagerDuty](#), [VictorOps](#), [Jira](#), etc.) anytime a really important business transaction deviates from the baseline, ensuring it gets proper prioritization within your team so they can proactively address the issue before it affects other end users.

By leveraging business transactions, Operations teams can easily understand how their technical perspective of the health of the environment relates to what really matters — the health of the business.

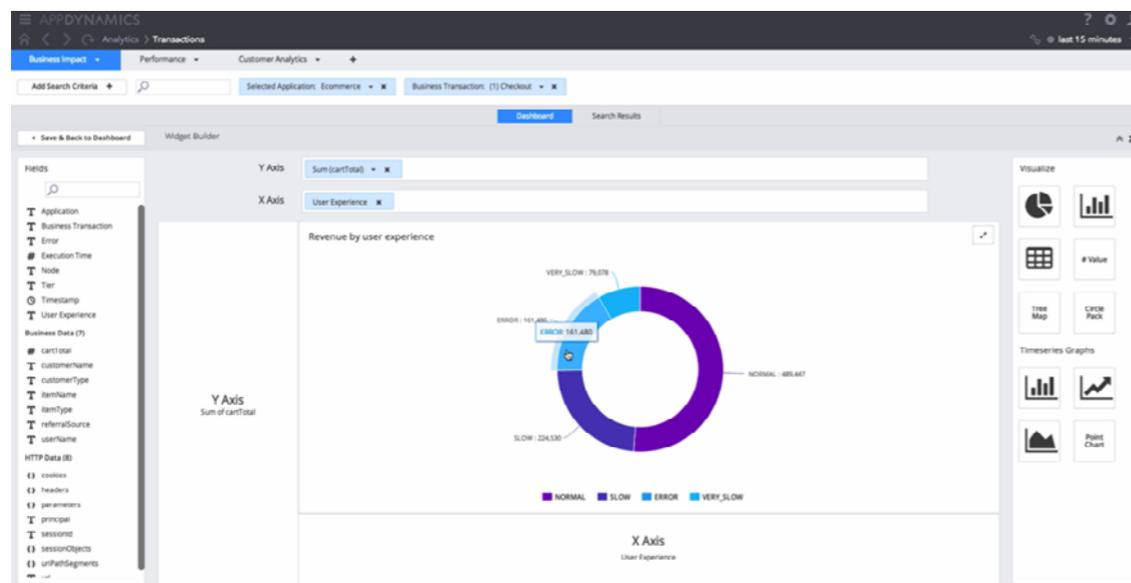
What is the business impact? - ITOA use case #4

What is the business impact? - ITOA use case #4

Analyze the business impact

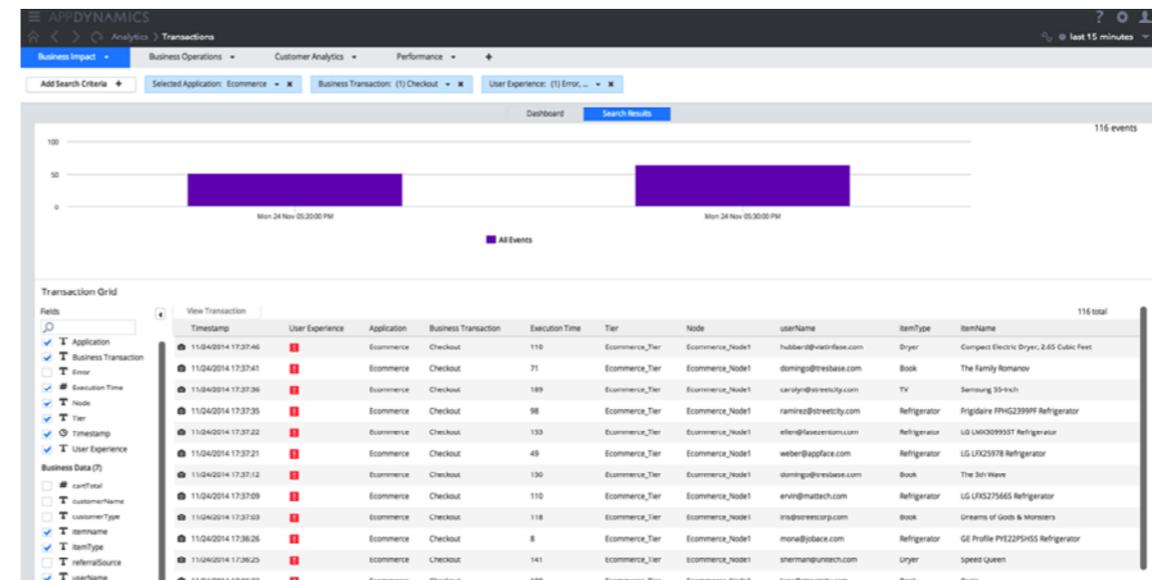
The fourth use case, analyzing business impact, is a way to quantify the value of all the work IT Operations professionals put in to making application environments run as smoothly as possible. Most of the time, IT is viewed as a cost center because when a revenue generating application has an outage for example, IT Ops is an easy team to point a finger at. However, by leveraging our new offering, Application Analytics, Operations teams now have a solution that enables them to have a more intelligent discussion about operational data and the associated business outcomes. This enables Ops to have more positive conversations that revolve around the added value that the team provides to the business.

Consider the example below; the Ops team was able to build a dashboard showing the dollar amount associated with normal, slow, very slow, and error transactions. The error transactions are the requests that did not complete as expected and resulted in no revenue. This is the money the business is leaving on the table due to an application that isn't performing well. If the dollar value associated with these error transactions (\$161K) is half of what it was during the same time period yesterday due to the work the Ops team completed by finding and fixing performance bottlenecks... they've added a very quantifiable number to the company's top line.



Suppose there was an unplanned outage that affected a large number of users - wouldn't it be great if marketing could get a report showing all the users that were affected during the checkout process and the exact items that were in their cart at the time so they could kick off a 'win back' campaign? With Application Analytics, Ops teams can make that wish a reality.

By filtering the set of transactions to only those with errors during a checkout process for a particular time range, and adding in the user IDs, product category, and product names, Ops teams can come out of a negative situation (like an outage) looking like a hero to their line of business peers.



Not only can Ops teams analyze the business impact of the troubleshooting they do on a regular basis, they can also use Application Analytics to identify anomalies and use this information to start the triage and troubleshooting process instead of only analyzing the results of their work. For example, Ops teams can leverage Application Analytics to discover the answers to these kinds of questions and proactively address issues:

- Why is the number of TVs sold in the past 15 minutes far below normal?
- Why are there a large number of slow transactions associated with trying to add a book to an online shopping cart?
- Why did a particular customer's package not get delivered?

Application Analytics provides a real-time view into how the business is correlated with the operational metrics they interact with on a daily basis, all with automatic propagation of business context in the form of business transactions and no application code changes required.

Create an attack plan - ITOA use case #5

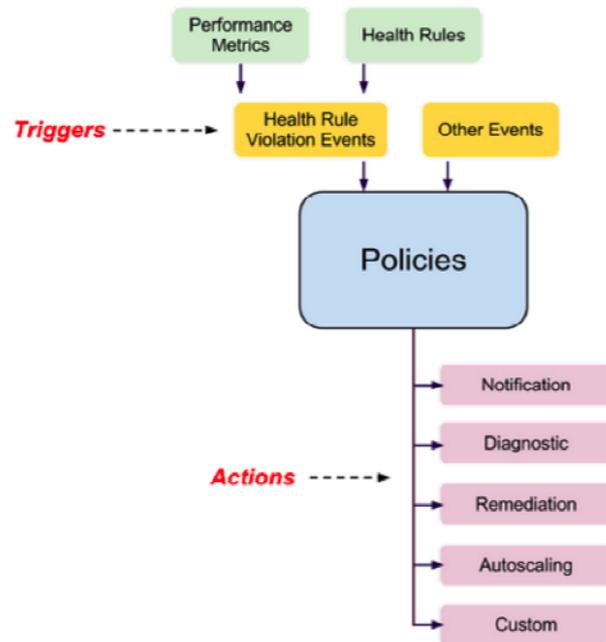
Create an attack plan - ITOA use case #5

If you haven't already done so, I recommend taking some time to read the info we already covered, all four of these use cases lay important groundwork for arguably the most important use case, creating an action plan. Let's dive in and take a closer look at how AppDynamics empowers Operations teams to take action with AppDynamics.

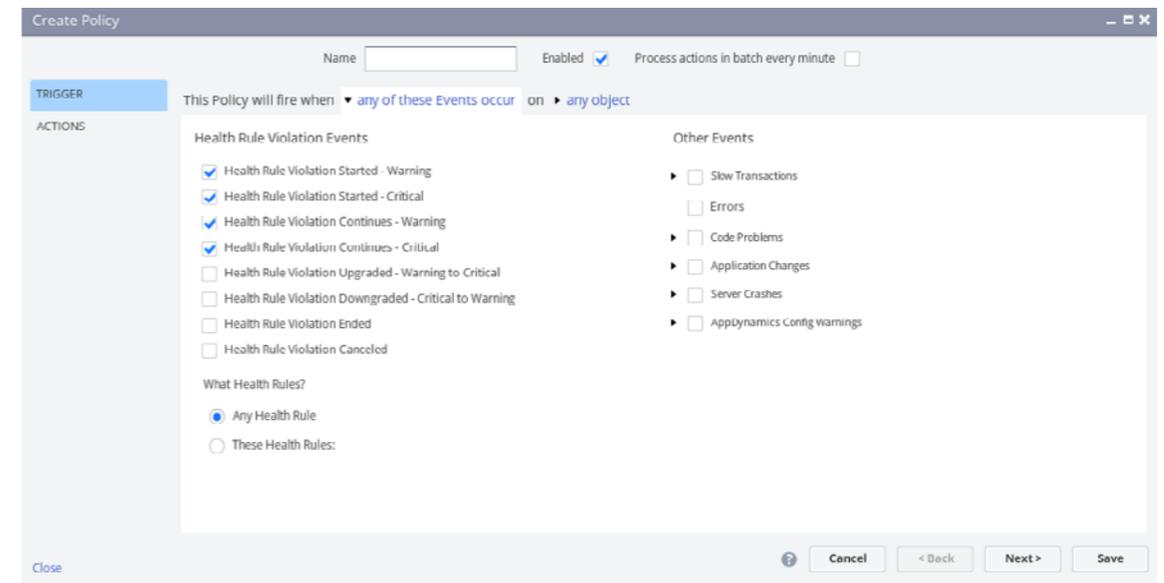
Creating an action plan

Visualizing the application environment, finding the root cause of performance issues, and stack rankings these fixes are all integral parts of how IT Operations professionals are using analytics solutions to solve real-world problems. Creating an action plan is all about taking the correct action once you have this information at your fingertips.

The notion of taking action in AppDynamics is governed by what we call **policies**. Policies are workflows that are made up of two things: **triggers** and **actions**.



Triggers consist of things like health rules violations or other events like slow transactions, errors, code problems, application changes, server crashes, and configuration warnings.



Once you've defined what rules or events will trigger your action, next you define exactly what you want done when that condition is met.

Actions consist of a number of different things in AppDynamics including notifications, diagnostics, remediation, custom actions, or cloud auto-scaling. This automated response to changing conditions in your application gives customers the flexibility to build out sophisticated workflows that can address different application conditions.

Create an attack plan - ITOA use case #5 (cont'd)

Create Action

Select what type of action to create:

Notifications

- Send an email
- Send a SMS message

Diagnostics

- Start a Diagnostic Session to collect Snapshots
- Take a thread dump

Remediation

- Run a script or executable on problematic Nodes

Custom Actions

- Run a Custom Action that has been uploaded to the Controller

Cloud Auto-Scaling

- Run a Workflow to scale up/down your Application

Cancel OK

Diagnostic actions include starting a diagnostic session (as the name implies) or taking a thread dump.

Create Diagnostic Session Action

Name

Duration 10 minutes

Snapshot rate 5 snapshots per minute

Business Transactions to run on

- Affected Business Transactions
- These Business Transactions:

Cancel OK

Create Diagnostic Session Action

Name

Duration 10 minutes

Snapshot rate 5 snapshots per minute

Business Transactions to run on

- Affected Business Transactions
- These Business Transactions:

Cancel OK

Let's take a closer look at these five different kinds of actions you can configure in AppDynamics.

The notification actions are pretty straightforward; simply get an email or an SMS text message when a particular trigger happens.

AppDynamics Notification
E-Commerce

Checkout Transaction is Slow

Summary of events occurring during the 1+ minute(s) prior to Wed Apr 02 7:15:02 EDT 2013:

Count	Event Type
1	Health Rule

New Critical Health Rule Violation

Wed Apr 03 16:03:55 EDT 2013

Health rule violation status changed from None to Open (Critical level) for health rule 'Checkout Transaction is Slow' of type Custom. All of the following conditions were evaluated on Application E-Commerce and were found to be violating thresholds.

1) Avg Response Time Violation
The observed value 12048 was greater than the threshold 200 for the last 1 minute.

Create an attack plan - ITOA use case #5 (cont'd)

Remediation actions run a local script in a node. You can use this kind of action to automate your internal run book procedures. To protect against misconfiguring this kind of action and initiating an action that could negatively affect the application, we've also built in an optional approval step into this workflow. That way, you can either fully automate the remediation or require human intervention for better peace of mind and control over your environment.

Create Remediation Script Action

You can specify any script or executable and the Java Machine Agent will execute it, and upload the results to the controller. You can download the script output on the Events screen (for events that trigger Policies).

Name: IncreasePool

Relative path to script: `${machine.agent.directory}/local-scripts/` runbook.sh

Absolute paths to log files:

Path
/tmp/script.out

Script timeout in minutes: 2

Require approval before executing this ...

E-mail address for approver: _____

[Configure Email / SMS settings](#)

[View machines with Java machine-agent installed](#) Cancel OK

Custom actions are typically used to integrate third party alerting and ticketing systems with AppDynamics and consist of a script and a custom .xml file. Usually these alerts / tickets are automatically created and the full context of what was happening inside the application is automatically appended to the ticket in the third party system. For more information on how to leverage custom actions to integrate with alerting extensions like [ServiceNow](#), [PagerDuty](#), and [Jira](#), check out the [AppDynamics eXchange](#).

Problem

Activity >>

2012-10-04 13:57:52 System Administrator - Changed: Assigned to, Description, Impact, Opened by, Priority

Assigned to: Beth Anglin

Description:

Application Name: ACME Book Store Application
Policy Violation Alert Time: Thu Oct 04 13:57:55 PDT 2012
Severity: 1
Name of Violated Policy: SamplePolicy
Affected Entity Type: APPLICATION
Name of Affected Entity: ACME Book Store Application

EVALUATION ENTITY #1:
Evaluation Entity: APPLICATION
Evaluation Entity Name: ACME Book Store Application

Triggered Condition #1:

Scope Type: APPLICATION
Scope Name: ACME Book Store Application
Name of Condition: Avg Response Time
Avg Response Time: > 0
Violation Value: 592

Incident URL: <http://127.0.0.1:8080/controller/#location>

Impact: 1 - High
Opened by: System Administrator
Priority: 1 - Critical

Create an attack plan - ITOA use case #5 (cont'd)

The final **action is cloud** auto-scaling. AppDynamics provides customers the ability to automatically scale up / scale down cloud computing resources from major cloud providers like **Amazon Web Services (AWS)**, **Windows Azure**, **Rackspace**, and OpenStack to name a few. Check out the **AppDynamics eXchange** for more information on cloud connectors.

The two most common cloud auto-scaling use cases are 1) creating new virtual machines and 2) terminating virtual machines when certain criteria are met. For example, if load breaches a certain threshold, you can create more cloud computing resources. Once load falls back down to a normal level, you can automatically terminate those newly created virtual machines.

The screenshot shows a 'Create Workflow Step' dialog box. At the top, there is a 'Name' input field. Below it, the 'Step Type' section offers five options: 'Create new Machines and Configure them' (highlighted with a blue background and a plus icon), 'Terminate Machines' (with a red X icon), 'Configure existing Machines' (with a wrench icon), 'Configure a specific Machine' (with a wrench icon), and 'Manual' (with a hand icon). A descriptive text below the step types reads: 'Create one or more machines and execute Tasks on each machine after is has started.' Below this, there is a checked checkbox for 'Setup Default Parameters for this Step' with a help icon. The configuration section includes: 'Compute Cloud' with a dropdown menu showing 'Select a Compute Cl...' and a link 'Register a Compute Cloud'; 'Launch instances of Image' with a dropdown menu showing 'Select an Image:' and a link 'Register an Image'; 'Number of Machines to Create' with an input field containing '1'; and 'Timeout (1 - 60 min)' with an input field containing '30'. At the bottom right, there are 'Cancel' and 'Create Workflow Step' buttons.

As you can see, leveraging a powerful solution like AppDynamics to automate actions in an application environment can foster a better end user experience via proactive identification and resolution of application performance issues.

Conclusion

IT Operations Analytics (ITOA) solutions provide the actionable information today's software-defined businesses need in order to truly optimize their applications. By optimizing these mission-critical applications, Operations teams end up proactively addressing most of the issues that could ruin the end user experience for their customers. In today's day and age, having a great experience while interacting with a digital business is of the utmost importance in terms of customer retention and lifetime value. Instead of being seen as a cost center, IT teams in the know are leveraging analytics solutions to become a binary differentiator in their respective industries and really drive business agility and profitability. Former CEO of GE, Jack Welch, summarizes it best, "An organization's ability to learn, and translate that learning into action rapidly, is the ultimate competitive advantage."

APPDYNAMICS

appdynamics.com