



Scout APM

Data Dog Comparison Guide



Table of Contents

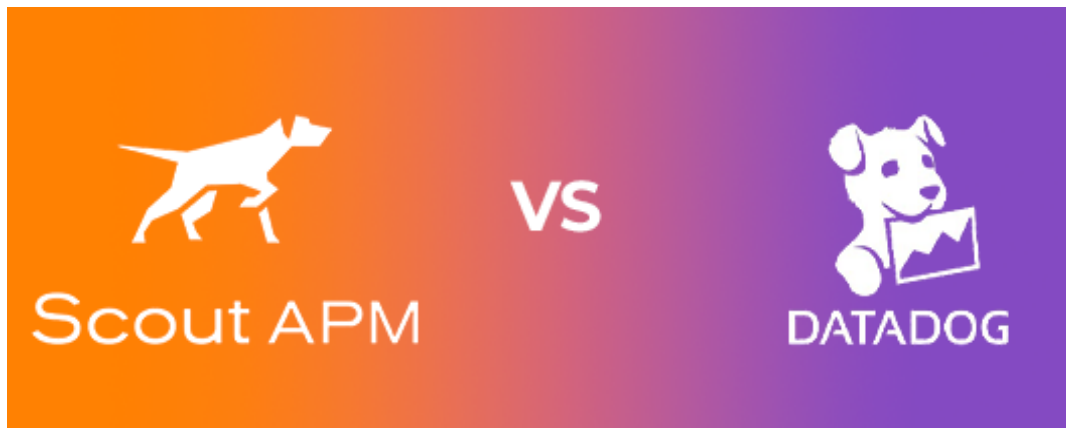
01 **What Triggers More Problems?**

02 **The Transaction Trace**

03 **Daily Digest**

04 **Agent**

05 **Conclusion**



Scout vs Data Dog

You're probably confused where Data Dog and [Scout](#) overlap. You're not alone! They've each offered similar products since their inception, and come on, they both have dogs in their logos!

Here's a timeline to clear things up:

- Both Scout (2009) and Data Dog (2010) started around the same time and focused on system-level monitoring (vserver resource usage, custom metrics via StatsD, etc).
- In late 2015, Scout launched a dedicated Application Performance Management (APM) product. Data Dog launched their own flavor of APM nine months later.
- In the summer of 2017, Scout's original server monitoring product was acquired by Pingdom, becoming Pingdom Server Monitor. Scout now focuses on application monitoring.

Data Dog offers both infrastructure and application monitoring. Scout offers a focused APM product, so their offerings overlap on application monitoring.

Both Data Dog's and Scout's APM offerings are solid, but they are aimed at different teams. How are they different, and what's the best fit for you?

CHAPTER 1

Does your code or infrastructure trigger more problems?

Let's start with the slogans.

Data Dog:

Data Dog is bringing sanity to **DevOps teams** with our Infrastructure & Application Performance Monitoring platform.

Scout:

App monitoring that **developers** love.

Scout is developer-centric: **this makes Scout a great tool if many of your time-consuming performance problems are triggered by product code written by your engineering team.** Data Dog is DevOps-centric: it's an extensive platform for charting just about any type of metric, from many different services, in a variety of different ways.

DevOps engineers write code, but it's typically infrastructure code and orchestration code. They are far less likely to modify the functionality of a web endpoint on their own.

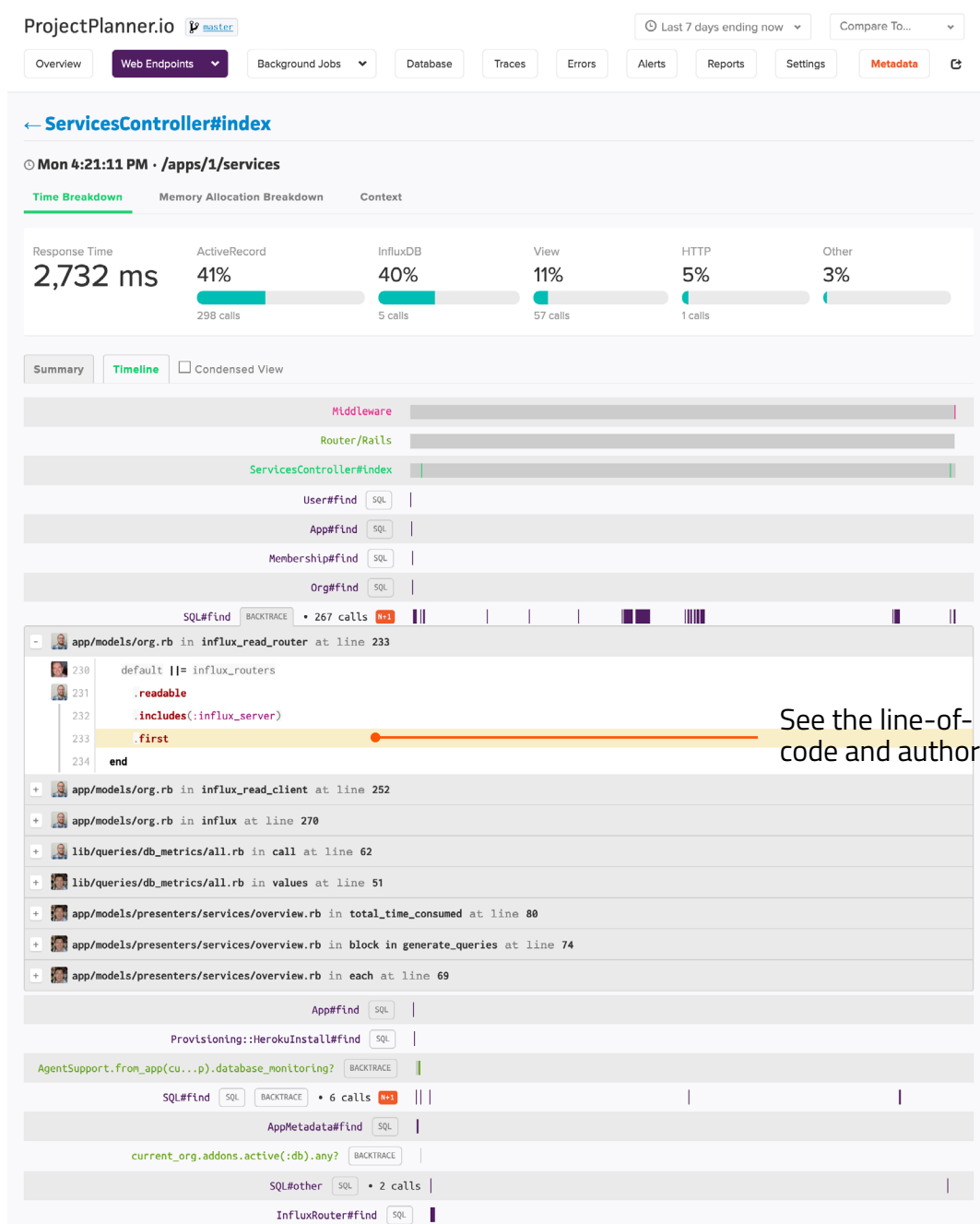
Let's take a look at three areas of the Data Dog and Scout APM products so you can see the difference.

CHAPTER 2

The Transaction Trace

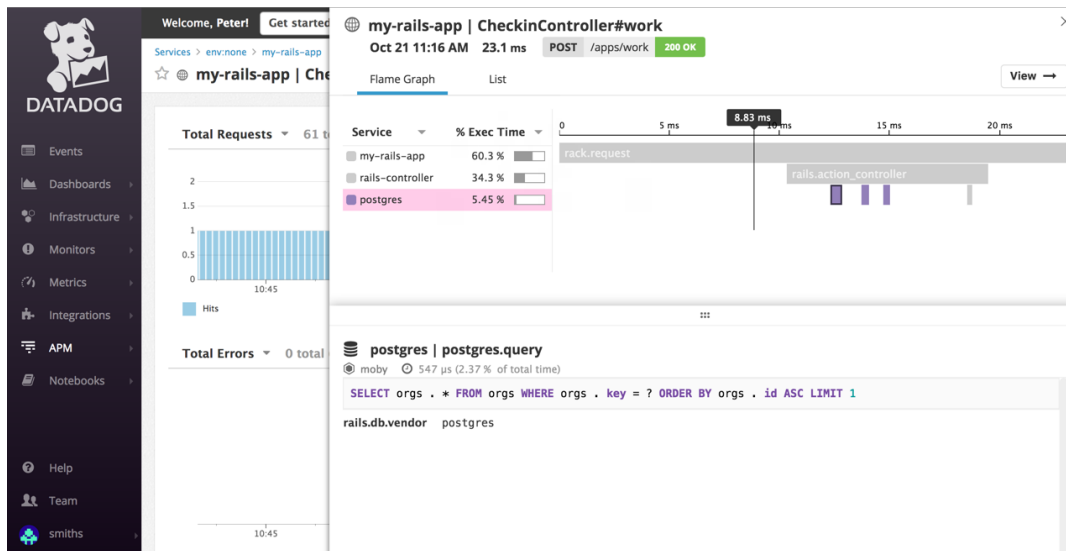
The soul of app monitoring is the transaction trace: all the metrics and analysis an APM platform delivers originates from data collected in a trace. A transaction trace is a single web request or execution of a single background job.

The screenshot below shows a transaction trace for the same web endpoint in Scout and Data Dog:



See the line-of-code and author

DataDog Transaction Trace



Highlights of the developer “special sauce” in the Scout trace:

- Scout provides backtraces to method calls and database queries. These aren’t available in Data Dog, which makes sense: a DevOps engineer is unlikely to make a change to product code. A developer, however, needs this backtrace to fix an issue.
- Scout injects the actual code into the trace, including the git blame metadata on the developer that wrote the code. This is critical to identify the best person for the fix (usually the developer that wrote it).
- Scout’s profiler, ScoutProf, breaks down time spent in custom code without instrumentation. This eliminates the painful step of custom instrumentation. This code is unlikely to be familiar to a DevOps team - it’s custom product code.
- Scout identifies factors that can trigger longer web requests, like SQL queries returning lots of rows or N+1 queries. Fixing these frequently requires code changes.

CHAPTER 3

Daily Digest

Both Data Dog and Scout provide a Daily Digest of your app's health via email. How this is displayed looks quite a bit different:

DataDog Digest Email

DATADOG

The Daily Digest for SmithOps

Here's what happened in Datadog on **October 20**.

Metric Alerts

2 Alerts Triggered

[Triggered] Service postgres has an abnormal change in throughput on env:none postgres throughput deviated too much from its usual value.
To troubleshoot, check the postgres [service page](#) or [recent traces](#).

At least **50%** of `sum:trace.postgres.query.hits{env:none,service:postgres}` values have been more than **2** deviations from the predicted values during the **last 10m**.
The monitor was last triggered at Fri Oct 20 2017 23:52:55 UTC (**3 secs ago**).


[\[Monitor Status\]](#) · [\[Edit Monitor\]](#)
20 Oct, 23:52:58 UTC

[Triggered on (host:moby)] [Auto] Clock in sync with NTP
Triggers if any host's clock goes out of sync with the time given by NTP. The offset threshold is configured in the Agent's `ntp.yaml` file.
Please read the [KB article](#) on NTP Offset issues for more details on cause and resolution.
Offset 392.522018433 secs higher than offset threshold (60 secs)
The monitor was last triggered at Fri Oct 20 2017 23:14:04 UTC (**3 secs ago**).

[\[Monitor Status\]](#) · [\[Edit Monitor\]](#) · [\[View moby\]](#)
20 Oct, 23:14:07 UTC

To manage your Datadog subscriptions, click [here](#).

Scout Digest Email

Scout  Dec 8 Performance Digest

Across 5 apps, web and background job execution times improved by 20% Thursday vs. the prior week.

We've highlighted ProjectPlanner.io below.

Summary of web & background job performance for the app.

WEB		BACKGROUND JOBS	
Response Time	Throughput	Execution Time	Throughput
↓10% 50 ms	↑16% 500 /min	↓10% 25 ms	↓2% 1,000 /min

Response times were slowest from **9:00 AM - 11:00 AM**. During this period, **ActiveRecord call times increased 30%** vs. the rest of the day.

Were there any deploys yesterday?


Insignificant changes are faded for less attention.

Ties performance trends to a specific deploy, if possible.

Drills deep into app performance, looking for specific categories of metrics with slower response times.

Identifies endpoints that are triggering an abnormal number of slow requests.

Calls out the source of the bottleneck.

- UsersController#show** is 30% faster, with a mean response time of 200 ms.
Performance improved following a deploy with commits from  chris@scoutapp.com at 11:34 AM.
- PhotosController#index** is generating a number of slow requests, including a **7 second request** vs. a mean response time of 355 ms.
A single Photos#find call took 6 seconds.
- CitiesController#show** is 26% slower, with a mean response time of 47 ms.
Response times began increasing around 8:00 PM.

Highlights of the developer "special sauce" in the Scout trace:

- Scout starts from named web endpoints - like PhotosController#index - then drills into factors - like slow database calls - that trigger problems.
- Scout identifies developers associated with slow code and when the code was committed.

CHAPTER 4

Agent

Scout is installed by a developer - it's typically a one-line Git commit to the product code base. Data Dog requires a separate agent install, which would typically be managed via Chef/Puppet/Ansible/etc.

CHAPTER 5

Conclusion

So, why don't we just tell you which to use?

First, there's no reason you can't use both: it's easy to pull Scout's APM metrics into Data Dog for your ops team [via the Scout API](#). This can give an ops team the high-level health view and engineering teams the code-level visibility they need.

What would I do? Try Scout APM and Data Dog at the same time (they don't conflict), then ask: "which tool helps our team solve performance issues faster?"

[Start your free 14-day trial of Scout APM today.](#)