EPISODE 1367

[INTRODUCTION]

[00:00:00] KP: Modern business applications are complex. It's not enough to have raw logs or some basic telemetry. Today's enterprise organizations require an application performance monitoring solution, or APM. Today's applications are complex distributed systems whose performance depends on a wide variety of factors. Every single line of code can affect production, and teams need insights into the health of their systems and how to improve them. In this episode, I interview Omri Sass and Hugo Kaczmarek from Datadog, a provider of cloud monitoring as a service. We discussed the APM space and the challenges faced by modern enterprise teams, we also get into some details about their new Live Search option, and why that solution was an important addition to their product suite.

[INTERVIEW]

[00:00:54] KP: Omri and Hugo, welcome to Software Engineering Daily.

[00:00:56] OS: Thanks for having us, Kyle.

[00:00:58] HK: Thank you.

[00:01:00] KP: So you're both at Datadog. Tell me a little bit about what you guys do and the nature of your collaboration.

[00:01:05] OS: So I think I can go first here. Hugo and I are both Senior Product Managers working in our APM department. It's fairly large. And our focus kind of spans everything that starts with how to detect and keep track of service health, all the way down to keeping track of code performance in exactly the impact of every line of code with our continuous profiler.

[00:01:31] KP: Well, could we drill down a little bit on APM? What should a company who is not yet really embracing these sorts of tools be thinking about as they plan to ramp up?

[00:01:41] OS: I think the best thing to do here is to start with what exactly is APM and how it's useful. And for that, APM is a broad suite of tools, right? In most definitions, they bundle user experience monitoring, whether we call it digital experience, or real user monitoring on the frontend, capabilities for detection and localization for backend services. So how do we know where an issue is happening? And the ability to tie all of these things together in the form of distributed tracing that can tell you, “Okay, in this particular request, what exactly is going on for this user? Where exactly is a performance bottleneck happening?” So for any company that is in a distributed architecture where monitoring becomes slightly harder than just looking at the health of your one monolithic service, anything like an APM tool, or any tool that falls under the APM umbrella, would be very valuable?

[00:02:40] KP: And can you describe or contrast how APM is the same as – What's the Venn diagram, I guess, with the general topic of observability?

[00:02:49] OS: So observability is a base building block in this, I'd say, a pyramid where at the top we have analytics, and machine learning and all those fun capabilities. Underneath we have monitoring. And at the base, we have observability, the ability to have complete or accurate knowledge into the performance of your entire stack. APM is a critical part of this, because it's one of the only ways to assess the health across multiple services simultaneously. And it doesn't require any upfront knowledge about any one place in particular.

[00:03:27] KP: And where does APM fall under observability specifically? How can companies use something like data dog to achieve better observability?

[00:03:37] OS: So that's a great question. And the idea is that APM is one of the core building blocks within observability. It gives you really like a core type of visibility into service health, into your dependencies. It's the sort of thing that lets you say, “Okay, something is wrong with my service. Let me dig deeper into the code. And we have ways of making sure that you know you need to go into the code.” Or I need to provision more hardware for my service. We have ways of letting you know that that's the case. Or alternatively, my system is suffering, but it's because of some downstream dependency. What is going on downstream of me in the tools and services that I rely on and that I consume from? Has anyone made a code change there? Are their infrastructure running properly? These are the types of insights that our APM tool can provide.

[00:04:38] KP: And can you speak a little bit to how organizations leverage that? Is there a sentinel who monitors all day? Or do collaborators all look at it? Are there alerts going out? What's the general operating procedure for people using Datadog?

[00:04:52] OS: So this is one of those fun examples for me as a product manager because the answer is all of the above. What tends to happen is that centralized groups, so SRE teams, or sys admins and organizations that have adopted that practice, they use APM to understand and to impose governance across all engineering teams. That means that they can set up monitoring. They can create the alerts. They can make sure that people get paged when they should. But then for application teams, they can use APM when they optimize their services, when they assess the health of their services, and when they investigate outages, right? And that understanding is my service now performing, or is it a dependency. Or alternatively, is someone abusing my service upstream from that? So we basically provide value for everyone in the space from those centralized SRE or governance groups, backend application engineering, frontend application engineering. Everyone gets to enjoy the benefits of these cross service insights.

[00:06:01] KP: And I'm wondering if we could explore the user experience a bit further. I'm imagining maybe a director of technology at a mid-sized company. Not writing as much code anymore, but responsible for a lot of code and wants to just maybe have a weekly period of time where they go and review things. They can log into data dog. What sort of things are available that they should be looking at?

[00:06:25] OS: So this truly is the beauty of Datadog as a platform. Our dashboarding capabilities have been around for a very long time. And they're trusted and loved by thousands, if not more of our users. Everything that we build within the APM suite uses the Datadog platform. We use what we call standard metrics that can be used to display on dashboards. They can be used in monitors. And most relevant to your question, they can be used for SLOs, right? So service level objectives can be defined using APM primitives.

Basically, that means that if you're trying to define uptime for a service or a user journey, you have all of the relevant information out of the box with our APM solution, and you can use it. So if I am this engineering director of a midsize company, I don't need to go into any particular part of the APM experience. I can have my own custom dashboard that has the things that I care about, like the SLO, the high-level business metrics, our overall costs or anything like that. Whereas all of the engineering teams that roll up to me can have views that are substantially more granular and can go as deep as an individual request or an individual line of code.

[00:07:43] KP: And can we talk a little bit about live search for listeners who aren't familiar with that feature of the product? What is it?

[00:07:49] HK: So Live Search is a capability that we released I think around a year ago, and it allows customers to view all their APM events, so APM related events for 15 minutes. And when I say view, I mean search across any dimensions. As well as something that is very important, do analytics. So when we say analytics, we mean do counts, group buys, plot distributions, plot time series, etc.

And so I think it changed a lot the way our customers view our products. When you think about observability, in general, for a lot of things, it usually means sampling. Meaning you cannot observe everything, because it would cost you too much, and especially when you talk about APM and distributed tracing. If you imagine when you do distributed tracing, for every request, every web request that you receive as a customer, you will send around 100 events to a monitoring platform like Datadog. And for some customers, sometimes it's billions per minutes. And at some point, keeping it all is cost prohibitive. And a lot of thought leaders on APM always say, and you can find some videos about that, you need to sample APM. So you need to, for example, keep one out of 1 million events. And that generates a lot of friction with customers because customers are always wondering, "What if I'm losing something?" What if I'm losing an error?" What if I'm losing slow request? I won't be able to dig into that."

When we offer Live Search, basically, the value proposition changed a lot. You can view everything for 15 minutes, sure, but everything, and especially in outage scenarios. So that means customers feel safer releasing applications. They feel safer doing any type of migrations. And in the end, they feel safer buying Datadog.

[00:09:59] KP: Is there anything special about 15 minutes? Why not 10 or 20?

[00:10:03] HK: So 15 minutes was kind of the minimal time that was acceptable in terms of technology and to allow customers to view enough data. We are definitely thinking about extending that to one hour, if not more.

[00:10:21] KP: Are there any particular use cases that have really been opened up that you couldn't tackle before Live Search was available?

[00:10:29] HK: Definitely. I think the most common use case is the outage response. When you're in an outage, it can be scoped to something very minimal. Can be one specific endpoint of a given application that suddenly answers in 10 seconds instead of regular 100 milliseconds. And in most cases, if it's something that is infrequent, you don't have any data. And customers would not use that adult to solve their use case. They would use something else. And because we have Live Search, now Datadog is usually the primary tool to install incidents.

[00:11:14] KP: Well, outage response is going to be critical. Any major outage with a big company, you can measure the dollars lost in milliseconds some of the time. How does Datadog play a role in a team in a critical state like that? How do they use it to get back on track?

[00:11:30] OS: Datadog offers a very wide array of tools that supports this incident response. Coming from the APM side of the house for Hugo and myself, the easiest steak is that, in a war room situation, one of the first places a user should go to is live search and live analytics were, being able to localize what exactly is going on? How can we see the impact of something? And can we drill down into an example request that helps us figure out, “Okay, someone deployed bad code somewhere. Let's go and revert that.”

But then there are a lot of additional things that we can use. We've recently released incident management that helps us manage the entire lifecycle of this incident. We have a lot of custom metrics and calculations that we can do to make sure that we capture the dollar amount or any other type of impact in the machine learning side of the house, right? So under the umbrella of what we call Watchdog, our AI ops engine, we're also working on automatic impact analysis, which will be able to tell you, “Okay, here's how many users that were impacted by a particular incident. How many services? Here's the impact on your error budget,” or anything like that. And the idea is that Datadog is here to support you, when things are currently on fire, when you need to figure out what is the thing that needs to be fixed. And then later when you conduct your post mortem, when you do the optimization, when you understand the root cause, we are there with a very wide array of tools to make sure that you have everything that you need for that. And to do it fast.

[00:13:07] KP: Well, you'd mentioned that Live Search went live about a year ago. So you've had a great amount of time to see it in field. I'm wondering if we could rewind a little bit and explore the story of releasing it. How do you spend time, I guess, first off, beta testing something like this?

[00:13:23] HK: That's a good one. I think, at Datadog, for any new feature or initiative, we have a very fast iteration process. Let me tell you a little bit more. At Datadog right now – So it wasn't the case a year ago, but you can imagine we have now I think over 1000 engineers. And all of them are using most of the Datadog suite. So in a way, it's kind of cheating for us because it's simpler to develop a new product. We can assess the wheel internally to use it. We can access the product market fit. And we can very fastly iterate on it.

So in a way, whenever we have an idea, we can start from scratch and build a prototype that we use internally. And that can be used by thousands of users internally, and iterate very fast. And I'd like to add something else. We have some trusted partners in our customers, some customers that have loved our products for the past 5 to 10 years. And these customers are always very excited to test and iterate with us. So because of that, when we wanted to test Live Search, it was quite simple. We just had to hack something, give it to a couple thousand people and instantly know the value or not.

[00:15:00] KP: Well, when you're providing a critical service like application performance monitoring, I mean, in some sense, I can use it for looking ahead and improving my process. But I'm also coming back to the outage monitoring situation where something is critical and I need to learn it. I have to absolutely depend on you guys to be up and to be reliable and all that sort of thing. Does that extra pressure change the way you release the product and develop it as well?

[00:15:27] HK: We have a lot of automatized systems for releases, for SLA, and we share SLAs on all our products with customers. And whenever there's a breach of those, we trigger an outage on our side and we investigate. And usually within minutes, it's solved. So I would say we take that very carefully. And it's just a matter of how efficient our release process can be. And I think over the past 10 years or even more since the company was created, we've become pretty good at that.

[00:16:28] KP: Are there any recent milestones we could highlight in changes in the way people are using performance monitoring tools? Have things like live search changed the game in significant ways?

[00:16:39] OS: I'd like to say that the answer is yes. But it's not only milestones as much as we'd like for it to be. It's not just the features and the products that Datadog has been releasing. The market itself is undergoing some transformation, right? 10 years ago when Datadog was founded, DevOps was a brand new kind of concept. And over time, it's become the standard. The same thing applies to complicated architectures where we've been talking about breaking down monoliths for years. But the reality is that even companies that still have monoliths have very complicated architectures around them. Microservices that have spun up in support or in replacement of parts of a monolith, all of these combined, and the kind of constant migration. Someone is always migrating from – Historically, it was from on-prem to the cloud, or from host-based environments, to containers or to serverless. The reality is that almost everyone right now runs some form of hybrid environment, hybrid in some sense.

And then all of the tools that we've provided, and we are providing, and obviously we're going to announce a few more shiny tools over the next few months. But all of these work together. The idea that you have a multi-cloud or architecture that relies on actual virtual machines, and on containers, and on some form of serverless environment, all of those needs to be monitored together. And this single pane of glass that we provide is really, really powerful for those cases. If I want to take all of this combined, and to answer your question, what has changed? Everything is more complicated, is more distributed, less centralized. And for that, we've had to build more and more solutions allowing everyone. Be it people in DevOps or SRE roles, application engineers on frontend or backend, database administrators, all of them can come to one place, see all of their metrics and all of their observability data in one place and speak one common language.

And I can't tell you that this is one milestone that's like this is what happened. It's an ongoing process. But every release works towards this shared vision, where all data and all telemetry that relates to your applications and your infrastructure lives in one place. And all of its users, regardless of their role, regardless of the language that they speak, can all work with it together.

[00:19:27] KP: With a robust offering like that, a smaller simpler company doesn't necessarily take advantage of everything at the start. Maybe as they grow, they adopt new services. Do you see common adoption patterns like that? Or the way in which companies use application performance monitoring evolve over time?

[00:19:46] OS: I have to say, and, Hugo, I don't know if you have a different perspective. My take is that a lot of it has to do with habits. And even young people who are starting their own startup and don't have a lot of professional experience. Even for them, there is some desire to understand the health of their code. And there is an entry point to that. And some people go at it from the perspective of logging, right? They know that they use some design pattern or some library that automatically logs their application. So they start thinking about that. Some people come at it from the perspective of the health of the infrastructure. How much CPU Am I consuming? Is my application constrained on memory? And then some people come at it from the perspective of going straight to their code, right? Let me see full code profiles. It's fairly common with very modern coding languages that kind of offer profiling tools out of the box.

Each one of these would have a different adoption pattern, right? I'd start with the tools that I'm more comfortable with. And then based on the profiles of my failures, and I try to use this word very carefully, based on where the risks are, based on where we see issues, we'll see increased adoption, right? If I tend to hit CPU constraints on my machines, I'll probably adopt more infrastructure monitoring to get alerts on that earlier. If I see that a lot of my code releases tend to get blocked and I have to roll back, I'll probably adopt more of profiler and code related tools, things like that. And if I see that I'm building a very distributed environment, I'm immediately using different microservices for different parts of my application. I'll probably be instrumenting distributed tracing.

[00:21:38] KP: And what are some common gaps you see in a lot of new customers. Are there things that people are discovering for the first time that maybe they didn't even know they could get from using the right tools?

[00:21:51] OS: I'll say this this actually brings up one of my favorite anecdotes about the release, recent release of our database monitoring product. One of our design partners that Hugo mentioned earlier that we'd like to work very, very closely with, when they rolled out our database monitoring solution on their staging environment, right? This is a very, very early beta days, they had realized that they had a gap in their observability data in the form of an application that no one knew about. So what had happened is that several years ago, before anyone that we were talking to in that company was even working there. Let's say a high number of years ago. Someone had written an application that hit one of their centralized databases. That application ended up generating the most expensive queries in terms of performance on their single most important database. They realized that application actually had no production impact. They could shut it off. And the gap there was a very large company, highly distributed environment, knowing where your applications are can sometimes be a challenge. And like learning that with a tool like database monitoring was a huge win for them. They ended up completely decommissioning that piece of code. And there was no way for them to know that they had to do that with any of our other tools that require knowing your own applications.

[00:23:18] KP: And could you talk about a rollout? What does it take for a modern software organization to adopt Datadog and roll it into their solution?

[00:23:29] HK: I think, from the ground up, we've always designed our products to be frictionless, especially at rollout. And that means the minimal amount of components to install, the minimum amount of people interactions, so that it all works in a way magically. And so specifically for APM, for instance, to run APM, you need to have two things. One thing that we call an agent running on your host or your VM, and one thing that we call a library, or an SDK that will run alongside your application. And very recently, for instance, we load application engineers to enable APM without having to set up the agent. So that means application engineers don't need to talk to SRE engineers to enable APM. And I think it illustrates the concept of what we are doing, which is you just add something, one flag. And in theory, in five minutes, you have a pretty good view of all your systems. And then for more advanced use case, you spend a couple more minutes and you have everything. That's what we are aiming for.

[00:25:02] KP: That’s a good goal. We've touched on database monitoring and live search. Could you comment on any other major innovations or standouts in product enhancements that you've seen enter the APM space?

[00:25:16] HK: Absolutely. A pretty good example is a product, part of the APM suite that we call continuous profiler. I think Omri talked a bit about it. And so it was I think a major revolution. So you may tell me, “Oh, Hugo, profilers existed before I was born,” which may be true. But legacy profilers, as we call them, are tools that, of course, as any profiler, allow you to analyze the performance of your code. When I say performance, I mean, resource consumption, whether it's time, CPU, memory, etc. And they would give you that visibility by line of code. But legacy profilers, to have them running, you need to SSH into a machine, launch this profiler. It would have a big impact on performance. You'd have to download the file and then analyze it in a separate tool. And then you'll be constantly wondering, “Did I catch the right moment or the right time?”

What we've built is a continuous profiler that can run in production. And so that means you just fire and forget. And you have constantly the performance of your code and how it evolves over time. And this was – We were honestly the first ones to do it on the market, or at least the first large company to do it on the market. And a lot of customers had surprises when they turn it on. Because they realized that there were some pieces of code that took so many resources. And they never knew. They were basically blind on that. And they thought that piece of code would take only less than 1% of the overall CPU, when in fact, it was taking 30% company’s scale. So we've had customers telling us they were able to reduce their AWS or cloud bill by millions of dollars per year. And it was amazing surprises that we've heard. And, yeah, in that, profiling opened a new way of viewing your coding production. And we were the first ones to do that.

[00:27:36] KP: Well, hearing that gives me a lot of confidence that I might want to put it into my production system. Yet, if I'm a really curmudgeonly engineer, I'm going to be worried about your library, like could putting that profiler in my production system slow it down, or constrained resources or something like that? How do you put especially paranoid engineers' minds at ease for something like that?

[00:27:59] HK: Of course, and it's a regular ask from many of our users. So depending on the language, and we now cover most of the common languages, we've built or we've reused technologies that were built specifically for production profiling. If I take Java, for instance, which of course one of the most commonly used languages, reused Java Flight Recorder, which is a technology that has been built at Oracle for the past 10 years to be a production profiler, we've hired people that know how to use it and that actually built it at Oracle. And we've added a bit of secret sauce for a specific use case.

And because I think what guarantees it are two things. One is the level of technical research and how advanced we are in every language to be able to build such a profiler. We are ahead, to be honest. And the second is every release goes into a big pipeline of testing. We test and we stress test. We have a whole environment that we call the reliability environment. That stress test is any type of application under heavy load and measures with and without the profiler. And we communicate those results for every customer that requests it. And we will also very soon publish these results of the low test. And you will see that the overhead is less than 1% or 2% CPU.

[00:29:37] KP: Well, the opportunity to reduce my cloud cost is something I think almost every major enterprise must be looking at. So there's some obvious wins there. What are some of the other advantages you see that companies gain when they start taking APM seriously?

[00:29:54] HK: So I think cloud cast is definitely a second type of use case of APM. It was surprising, because we weren't expecting it to be honest. But the largest, most common APM use case is I would say two. One is I have a lot of errors suddenly. Where does it come from? What's the impact? Two is my users are reporting high latency? Where does it come from, and why? That's by far why people install APM in the first place. And then we are building a full suite to allow customers to start from that, get alerted by these two specific common use cases, and investigate. And Omri can talk a bit more about it. But even automatically detect when such a thing happens and point to the root cause. And when I say point to the root cause, I mean, point even down to the culprit commit on GitHub of the application that you may have released.

[00:31:11] OS: I think Hugo nailed it. The only thing I'll add here is if you think about the type of insights that we keep track of in APM, right? There's a reason that these are generally called the golden signals, right? So hit rates or throughput, error rates, and latencies. These are the closest generic indicators of performance issues. That means that they can apply to almost every situations. We can talk about like the edge cases of where we might not have an error, but something still happens and how to address those. But these use cases apply to most, let's say bad situations. I don't want to say issue, or incident, or anything like that, because these metrics actually apply well beyond that. And the investigative tools that we offer apply to a whole set of issues, right? And we can talk about very specific cases and how our customers use them. But the reality is that when you go for an observability tool, and when you go for an APM suite, the idea is to have very broad protection, and not hedge against one particular type of issue.

[00:32:21] KP: Well, the way developers build software has continued to evolve. I think there's no sense in assuming it wouldn't continue to evolve into the future. Containers are a big innovation. Serverless, for example, I think Datadog was founded before serverless was even a thing. Can you comment on how API has evolved to support new use cases like that?

[00:32:44] OS: So I can cover this piece. And it goes back to something that we talked about a few moments ago with the platform. We use a platform internally that is very generic and very flexible. And when we need to make modifications to it in order to support a new type of technology or a new trend in the market, we make sure that, A, we use it internally, right? Hugo mentioned that earlier. We want to make sure that we dog food, and no pun intended here, but that we dog food our own products and that we use all of these bleeding edge services and bleeding edge capabilities. And then we provide the best possible monitoring for them. And if we realize that we need a new type of system to support them, or that our system is flexible enough to just go with it, then we build whatever it takes.

On top of all of that, we try to partner with the major cloud providers and be ahead of anything that happens in the market. And if you look at our latest releases in the Kubernetes space, for example, we offer an out of the box tooling to some of Amazon's latest releases. And the same thing applied in Azure, for example, where we were release partners and we offered out of the box monitoring on the day of new service releases.

[00:34:02] KP: Very cool. Well, with that in mind, can you both comment on where you see the future of APM going? What are some of the new exciting challenges you hope to bring solutions for?

[00:34:15] HK: As I started to say, I think there are two things. The first one is definitely the rise of machine learning in root cause detection. But at my previous company, I worked a lot on machine learning, etc. And it's a buzzword, but here at Datadog we have a different view than the mainstream view, I would say, is that we want our machine learning to point customers toward health issues for which we are sure – Well, we are never sure, but you see what I mean, 99% sure that they are true issues. Because when you're in monitoring, it's not like you're recommending a new song to listen to. You're not joking. And so we want our recommendations to be very accurate. And right now, we have something. But that's where I think to reduce what we call the MTTR, mean time to resolution. That's where machine learning will have a big impact.

The second is about broader capabilities that are part of what we call APM performance monitoring. So as Omri mentioned, we have what we call distributed tracing. But we also have profiling. We have database monitoring. And we have two or three new things that are in development and that are products that are very adjacent that help. Unfortunately, I can't talk more about them precisely. But I can say these are tools to help software engineers, specifically application software engineers, solve their outages, debug their problems faster. And that's the future of APM. It's not just one thing. It's a suite of tools for software engineers.

[00:36:17] KP: Well, that's a tantalizing solution. I look forward to seeing what develops in the future in that regard. Well, Omri and Hugo, thank you both so much for taking the time to come on Software Engineering Daily.

[00:36:28] OS: Thank you, Kyle.

[00:36:31] HK: Thank you, Kyle.

[END]