Deloitte.

For Cloud Professionals
October 2020

For Cloud Professionals, part of the On Cloud Podcast

David Linthicum, Managing Director, Chief Cloud Strategy Officer, Deloitte Consulting LLP

Title: **Serverless unleashed: making it faster and more efficient**

Description: Serverless computing definitely has its advantages, such as the ability to scale quickly and manage costs effectively, but there are also latency and complexity concerns. However, there are open source projects in the works to address these issues and make serverless more attractive. In this episode of the podcast, David Linthicum and guest Vikram Sreekanti, discuss Cloudburst, Vikram's open-source project that aims to bring more applications into the serverless mold and make it faster and easier to use, while also supporting other programming patterns, application architectures, etc., that can benefit from the power of serverless computing.

Duration: **00:21:55**

Operator:
This podcast is produced by Deloitte. The views and opinions expressed by podcast speakers and guests are solely their own and do not reflect the opinions of Deloitte. This podcast provides general information only and is not intended to constitute advice or services of any kind. For additional information about Deloitte, go to Deloitte.com/about. Welcome to On Cloud, the podcast for cloud professionals, where we break down the state of cloud computing today and how you can unleash the power of cloud for your enterprise. Now here is your host David Linthicum.

David Linthicum:
Welcome back to the On Cloud podcast, your one place to find out how to make cloud computing work for your enterprise. This is an objective discussion with industry thought leaders who provide their own unique perspective around the pragmatic use of cloud-based technologies. Today on the show, we have our special guest, Vikram Sreekanti, and he's – Sreekanti, excuse me, and he's the creator of Cloudburst, which is a unique concept and open source project, and he is here today to talk to us about what it is, what it does, and how it came about and how you may be able to get involved with this technology moving forward. Vikram, how you doing?

Vikram Sreekanti:
Doing well, David. Thanks for having me on the podcast.

David Linthicum:
So, we talked a bit about you before the show. You're actually in Berkeley. Hopefully you're not too far in the hills of Berkeley because I hear it's on fire.

Vikram Sreekanti:
Yeah, we're further down. The hills are bad but we're just dealing with the smoke, which is frustrating, but thankfully completely safe.

David Linthicum:
So, you're an interesting dude because you're involved in research. You and I have that in common, so tell us about what your research is, what your day job is how you came to work with the Cloudburst technology and concept going forward and maybe some of the things you do part-time as well, hobbies, things like that.

Vikram Sreekanti:
Sure. So, I'm a grad student at Berkeley. I've been here for four years now. I did my undergrad here as well, so I've been in Berkeley for a while now. I work on serverless technology, and Cloudburst is an open source project, a research prototype that we've been building for the last few years here that's aiming to sort of push the boundaries on how we can bring more and more applications into the serverless mold, make it easier for people to use the cloud, but also support really common kinds of programming patterns, application architectures and so on that would benefit from simplicity of serverless infrastructure, but can take advantage of the systems that exist in the state-of-the-art today. So, that's been something that I've been really excited about for the last three or four years now, and we've made some really good progress on the kinds of things that we're able to support on the Cloudburst platform.

In terms of hobbies, I'm a big sports fan. I've gotten really excited about the NBA bubble and the NFL being back, so that's been a good pastime, and I also like reading a lot, so you know, lots of free time in COVID quarantine to do all those kinds of things.

David Linthicum:
So, how was Cloudburst conceived? In other words, I've been involved with open source projects over the years in one way, shape or form, and so building these things, there's always that eureka moment where we're going to in essence solve this particular problem and we're going to solve it for a number of people, and we're actually going to release this to a community, so it's not going to be a proprietary product we build within the confines of some company. It's going to have a wider release and it's going to have more interest in more people moving forward, so how do you come to the conclusion that Cloudburst had that interest and that you're going to move it toward the open source space?

Vikram Sreekanti:
Well, I think something that's really interesting is that, over the last five or ten years, we've seen that open source infrastructure has gained a lot of popularity. The traction that a system like Kubernetes, for example, has gained in open source by sort of taking advantage of that community aspect that you pointed out has really helped it evolve and meet new use cases and try to get the broadest set of users possible. And so, looking at that, we felt like there was an opportunity to layer on top of that. And we've seen a number of functions-as-a-service systems built on top of Kubernetes or as a part of the Kubernetes ecosystem, but it felt to us like they were primarily trying to bring the standard ways that FAST systems operate on top of Kubernetes.

Looking at that, we felt like there was an opportunity to take a lot of the benefits of that, the ease of use, the autoscaling, the reduced operations and so on, but to try to expand those benefits to a broader set of applications, things like data-intensive applications, applications that require communication between various actors in the system, and so on. And when we looked at the state of the art in serverless computing a few years ago, what we found was that the reason that folks were starting to migrate onto systems like Lambda was for the benefits, for the simplicity, the scalability, the ease of use and so on.

But once they started growing or if the application had certain constraints, they either weren't able to use Lambda, or they had to switch to another system or maybe roll out their own infrastructure. And we felt like Cloudburst, or what became Cloudburst would be able to fill the gap to essentially enable people to keep all those benefits but to be able to take advantage of a system that provides caches that look physically close to the compute so that you don't have to waste bandwidth and time shipping data over the network and to simplify things like function and composition so that applications with strict latency constraints would be able to operate on a FAST system without paying 50 to 100 milliseconds every time they invoke the function.

And so, we felt like those benefits were really probably the applicable, could take advantage of having the open source label on it so that you could get more users, get more interest and feedback. And also, being from Berkeley, there's a long lineage of open source projects that have been developed and released to great success here, most recently something like Apache Spark, but dating back to Postgres and DSD and Ingres and so on. So, it felt like the right place to take on that kind of project.

David Linthicum:
So, ultimately there's different ways to leverage this technology, sounds to me, so what would be a typical use case? In other words, what would be something that would be a very technical use case and what would be something that would be very business oriented?

Vikram Sreekanti:
Sure. So, the kinds of things that we've been looking at after we sort of built the initial prototype of the system have focused on developers who maybe don't have deep technical expertise, don't really have years of experience with a system like Kubernetes, for example, but might have a lot of data that they're wrangling. And those kinds of use cases are becoming increasingly common, actually. We see this on campus with the ever-growing data science curriculum, where students actually are commonly wrangling data sets of a few gigabytes, maybe 10 to 20 gig, and the VMs that they're given as a part of the course infrastructure are usually limited to something like 2 or 4 gig. The thing is, the way that the campus infrastructure's set up, they have that VM allocated to them for the whole semester, which is extensive, that means it doesn't scale very well, and the night before any assignment that’s run, that cluster inevitably falls over and everyone's scrambling to get things up and running.

What you really want is infrastructure that allows students to burst in order to make sure that they're able to deal with the data, deal with the compute that they want to do for the few hours that they're doing an assignment, but the rest of the time, you don't want to pay for it. And so, the prototype that we've built here is a serverless backend for Jupyter Notebooks. So, a data scientist, for example, would be able to show up, run some code, maybe even upload a large data set into the cloud and do some sort of computation over it with the pandas data frame that they're manipulating.

But once they're done with it, all of that data gets put into hold storage, the compute goes away, and they aren't charged for that, which provides them a lot of flexibility and allows them to actually scale beyond what they might normally be able to do if they were given, say, a single AC2 machine with a fixed number of CPUs and a fixed amount of RAM. So, it's that flexibility to both scale to zero but also scale up to meet the use case demand in terms of compute, memory, network bandwidth and so on that makes the serverless model so appealing. But the existing infrastructure doesn't allow that because, again, you'd have to ship data over the network, you'd have to worry about invocation latency and all those kinds of things.

David Linthicum:
What kind of cost savings can we expect from, say, the as-is state without Cloudburst and the to-be state with Cloudburst? And where does that cost saving come from?

Vikram Sreekanti:
Yeah, so the cost savings primarily come from the ability to scale to zero. The standard practice in industry today is to provision applications for peak load, or somewhere close to peak load, meaning that whatever the projected maximum resource requirement of the application is, that's how many machine servers we're going to allocate for that application. Which means that in the rest of the time, when they're not at peak load, you're going to be paying for a bunch of extra compute just to make sure that when you do get to peak load, you have enough resources to serve that. And so, the cost savings are basically going to be the sort of troughs in the workload graph. And I know it's not very useful to talk about graphs on a podcast, but if you imagine the empty space between a flat line and the workload usage graph, those are the kinds of savings you could get from serverless. The more empty space you have there, the more amount of time that you have resources allocated but not used, the more likely you are to see serious cost savings from serverless infrastructure.

To give a sort of example, if you imagine that you have, let's say a workload that follows usage patterns of US time based on morning and afternoon and nighttime patterns, if you have eight hours a day, let's say between 11:00 PM and 7:00 AM, that you aren't going to have very much or any usage at all, maybe your cost goes to zero or close to zero during that time. And during work hours during the day, you're going to have all of the normal resources you would have allocated. So, in that example, you would have, like, one-third cost saving.

David Linthicum:
So, moving forward, what is the user, or what is the consumer of this technology? Are you going after the developer? Are you going after the architect? Are you going after the CIO? Is this a strategic level acquisition or is this something that is going to be more tactically focused lower down in the IT work?

Vikram Sreekanti:
I think the most tangible benefits in the short term are definitely for the individual developer. The way that I've heard it put, which I think is very appealing, is that serverless infrastructure removes the barrier between the application owner and the infrastructure owner. It enables the application owner to be as close to the physical hardware as possible, which makes them more effective as workflows become more agile, as developers deploy code more and more often. I think simplifying the processes by which that happen, removing the complexities of APIs and Docker and Kubernetes, and enabling people to get applications into the cloud as quickly as possible is where people are going to see the most benefit. As those applications scale, as they get bigger and bigger and they have more interest in workload patterns and more interesting resource usage, I think there's definitely opportunities to talk about the cost savings, about the efficiency that you get from the infrastructure, but I think the first order benefit is definitely on the simplicity and the usability in showing developers that they can get applications deployed really quickly, really seamlessly without having to bend over backwards.

David Linthicum:
One thing I learned over the years as an architect is that technology like this has a tendency to want to work and integrate with other technologies, database security, governance, things like that. What do you see as the key enabling technologies that will likely be a one plus one equals three kind of scenario with Cloudburst?

Vikram Sreekanti:
I think the most interesting things are definitely, from our perspective, on the data side. The way that we built the initial version of the system is to focus on reducing data movement to the extent possible, which allows us to have really big wins around bandwidth, latency, and data access and so on, with some smart scheduling techniques built in to enable that. And I think as you sort of project forward, if you can integrate with, let's say, a transactional database, that's the system of record for a web app, or integrate an analytical database and try to surface interesting things from that database, data scientists offer their analysis, I think that's where you can more and more use cases getting unlocked. You can start imagining really unique ways in which developers can deploy new features or perform new analyses without necessarily having to, again, ship data over the network or worry about how the application performance is going to be affected by those kinds of things because all that is going to be managed and automated by the infrastructure because those are the kinds of things that we're focused on in the system that we've built.

David Linthicum:
So, there's always a question with open source projects in terms of how we're going to commercialize something or monetize something. Is that even coming up now? Are you guys at a portion of your life cycle where you're just considering the uses for this and focusing on the solution rather than focusing on how it's going to be monetized, different companies distributing the systems, things like that?

Vikram Sreekanti:
Yeah, we're definitely in the sort of earlier stages here. We are focused on interesting use cases, interesting applications, trying to figure out what the things that we can do with the system that enable new applications are and trying to go after those early on, and the other questions or concerns aren't really things that we've addressed yet.

David Linthicum:
Yeah, and it's probably a good thing. I think that where open source projects have a tendency to go sideways is when people think too quickly about the commercial applications of it and in essence try to build a business around it. Because at the end of the day, you're trying to get this out into community, getting people to adopt it and really getting to show its value to the use of the thing and putting it in the hands of the different developers who are actually

making things happen. As they move along and, in essence, be creative and innovative with the technology, suddenly the technology becomes more valuable unto itself. That's what you're trying to do?

Vikram Sreekanti:
Yeah, and I think the nice thing about being at Berkeley and having the campus community around us is that we have the opportunities to collaborate with different folks but the research all the students that allow us to understand some of these use cases better and to drive really the interesting system architecture, the interesting features that we can build based on tangible things that we see being done in the campus community, which is Berkeley's a world-class place to be for those kinds of things, so that's really neat, and those are the kinds of things that we're focused on in terms of finding those use cases, going after the campus infrastructure and thinking about interesting things that can be enabled with this sort of new cut on serverless infrastructure.

David Linthicum:
So, a lot of open source people listen to this podcast because I hear from them all the time, folks that are involved with Kubernetes and even other open source projects things that are Apache licensed, things like that. So, how can they get involved at this point? You know, how can they understand more about your technology, where you guys are looking to go, and even give it a try?

Vikram Sreekanti:
Yeah, so we have, on the idea side, we have a bunch of research papers that we published on this front that have discussed the system architectures and various consistency guarantees and performance improvements and so on that we've made. We've also started to work on some of the application level components in this space, so we've recently published a paper on a prediction serving pipeline framework that we've built on top of Cloudburst that allows users to basically compose machine learning models into releasable pipelines and deploy them on top of Cloudburst so that they get all of the sort of autoscaling and serverless benefits and also get some optimizations to improve performance under the hood.

In terms of the open source, we have a GitHub organization that's called The Hydro-Project, that's H-Y-D-R-O dash Project, where all of the systems that we've built – Cloudburst, the open source key value storage built on top of and everything are all available there for people to check out. And you know, we'd love to hear from folks as they sort of start to play around with it and have interesting feedback and thoughts on things that could be improved, interesting applications, and so on.

David Linthicum:
So, let's build a time machine, go forward three years. Where's Cloudburst in the market and where you would love to see it in the marketplace in three years, and how do you see getting themselves there? How do you see you getting there?

Vikram Sreekanti:
Yeah, I think the opportunity for a system like Cloudburst is to really unlock the potential of serverless computing, meaning that more and more developers are able to see not only that serverless is easy to use, but also really an option for them to start building their application on, whether that's data science, whether that's prediction serving like I just talked about, or even something like building a web application, building internal tooling for companies. I think these are all places where the simplicity, the speed, the agility of serverless computing can be really beneficial, but again, due to the constraints of existing systems, is unable to be unlocked. And I think as we develop the system, as we get more feedback and more interesting features in there, I think being able to help developers deploy those kinds of applications, figuring out the use cases that help them do their job more effectively without having to worry about operational concerns, and seeing that you can get new applications deployed really easily is where I think we have the opportunity to have the biggest impact.

David Linthicum:
So, what are the advantages for the cloud server providers out there moving forward? So, if I'm a CSP, cloud services provider, I'm looking at your technology, what should I be thinking at this point?

Vikram Sreekanti:
I think that we will see that existing functions of serverless systems will get better. You know, one of the sort of advantages that we have building a sort of prototype system and releasing it into open source is that we don't need to worry about necessarily building a system that achieves the operational scale that AWS Lambda does on day one. And so, a lot of the work that we've heard from folks at AWS has gone into that system is really stability, making sure that it is as reliable as possible and so on. But as they meet those milestones, I think we'll also see that they're able to start integrating some of these newer features and to get more and more maturity and breadth in their platform.

You know, we've already seen in a simple example that AWS Lambda went from having a one-minute time limit for functions when it was released to five minutes to now 15 minutes. And I think on other axes, whether that data access, network bandwidth, whatever it is, we will see those things improve, but at the end of the day, those systems are going to be sort of limited by the operational constraints that their ops teams are going to have to support, whereas an open source system like Cloudburst is able to have a little bit less burden on that front and can maybe focus more on the features that will enable new applications. So, that's the real difference that I see.

David Linthicum:
What about security models that work and play well with Cloudburst? What should I be thinking about for that?

Vikram Sreekanti:
Yeah, I think this is where the cloud providers have really done a great job of supporting the open source community. I think a tool like gVisor from Google in particular is really interesting. For folks who may not be familiar, gVisor is basically a container runtime that provides a VM level isolation guarantee but is Docker compatible, so you can run Docker containers in user space just like they run normally but get VM level isolation. So, that's actually a really interesting direction that we're looking into. We haven't done that work just yet. We're sort of running on regular Kubernetes right now, so we have weaker isolation guarantees, but I think using something like gVisor where it is tested, we sort of trust Google's open source credentials here and we trust that they're able to build these systems well, is I think a good place to start and good sort of way to get those kinds of guarantees without necessarily having to rip apart the whole system and re-architect things from scratch.

David Linthicum:
So, I hear a lot of people out there who are interested in this kind of stuff, so where can they go find out more about Cloudburst on the web, or can they download the research papers? Where can they find this on GitHub, all those sorts of things?

Vikram Sreekanti:
Yeah, so the GitHub organization is Hydro-Project, and all of the code is open source there. All of our research papers you can find on my personal web site, which is VikramS.io. They're sort of scattered all over the internet in various conference publications, but we sort of aggregated them in one place there. And you know, if folks are interested, we'd love to hear even just applications that they would be interested in sort of exploring on top of serverless infrastructure. Feel free to open a GitHub issue or shoot us an e-mail. You can find our contact information on GitHub. So, on all those fronts, we'd love to hear from folks.

David Linthicum:
This is exciting stuff. This is where the rubber meets the road with creativity and innovation as we try to figure out how to make things better moving forward. And so, Cloudburst, this type of project, like the other projects out there that are really looking to take the next level and not doing so through, in essence, investment from investors, but the ability to have a rank and file organization and developers kind of run to their keyboards and build a better mousetrap. This looks like it's moving in the right direction. So, if you enjoyed this podcast, make sure to like and subscribe on iTunes or wherever you get your podcasts. Also check out our past episodes, including the On Cloud podcast hosted by my good friend, Mike Kavis, and his show Architecting the Cloud. If you'd like to learn more about Deloitte's cloud capabilities, check out DeloitteCloudPodcast.com, and if you'd like to contact me directly, you can reach me at dlinthicum@deloitte.com. So, until next time, best of luck in building your cloud computing solutions. We'll talk to you guys real soon. You guys take good care. Bye.

Operator:
Thank you for listening to On Cloud for Cloud Professionals with David Linthicum. Connect with David on Twitter and LinkedIn and visit the Deloitte On Cloud blog at www.deloitte.com/us/deloitte-on-cloud-blog. Be sure to rate and review the show on your favorite podcast app.