BBBT Podcast Transcript

About the BBBT

The Boulder Business Intelligence Brain Trust, or BBBT, was founded in 2006 by Claudia Imhoff. Its mission is to leverage business intelligence for industry vendors, for its members, who are independent analysts and experts, and for its subscribers, who are practitioners. To accomplish this mission, the BBBT provides a variety of services, centered around vendor presentations.

For more, see: www.bbbt.us.

Vendor:	**Databricks**
Date recorded:	**April 24, 2015**
Host:	**Claudia Imhoff**, President, BBBT
Guest(s):	**Paco Nathan,** Director of Community Evangelism
Run time:	**00:17:43**
Audio link:	Podcast
Transcript:	[See next page]

Claudia Imhoff: Hello, and welcome to this edition of the Boulder BI Brain Trust, or the BBBT. We're a gathering of international consultants, analysts, and experts in business intelligence, who meet with interesting and innovative BI companies here in beautiful Boulder, Colorado. We not only get briefed on the latest news and releases, but we share our ideas with the vendor on where the BI industry is going, and help them with their technological directions and marketing messages. I'm Claudia Imhoff and the BBBT podcasts are produced by my company, Intelligent Solutions.

My guest today is Paco Nathan. Paco is the Director of Community Evangelism, I love your title, by the way, for Databricks. Welcome, Paco.

Paco Nathan: Thank you, Claudia.

CI: Interesting session today, all about Databricks, Apache Spark, all kinds of things. Let's start with an overview of Databricks itself. It was founded by the creators of Apache Spark. Is that right?

PN: That's correct. There was a lab at Berkeley, Berkeley EECS, a lab called AMP Lab. It was about algorithms, machines, and people. Out of this, there are a few different open-source projects that have emerged, but Apache Spark is the most popular one.

One of the main professors, Ion Stoica, was grad advisor for a lot of the principals doing this. They moved a couple blocks down the street and set up shop. That's Databricks.

CI: Then, the history of Spark.

PN: Spark comes out of approximately 2009. There'd been a project called Mesos, which was about getting multi-tenancy on commodity hardware.

The people working on Mesos wanted to do a perfect concept. A lot of them had been involved as interns, grad students who were working at Google or working at Cloudera, working on a lot of large-scale Hadoop deployments.

They were seeing that there were bottlenecks in Hadoop, both technically, in terms of how you work with data, but also from a business standpoint, bottlenecks in terms of staffing, because it doesn't necessarily fit how people think about analytics workflows. Also, bottlenecks in terms of not fitting use cases. In analytics, of course, we love doing real-time, these days. This is all the rage.

Hadoop was doing great off of the batch use cases, but when you start getting into more of the specialized use cases, you saw a lot of specialized systems moving away from Hadoop.

We see Millwheel at Google leading to things like Storm and S4 and others. We see things like F1 leading to Impala for SQL at scale, looking at Pregel going to Giraph for graph analytics at scale.

We see a lot of specialized systems, and there's a cost to that. Number one, you pay a lot of money to run a lot of clusters. There's also the learning-curve aspect.

CI: I was going to say confusion is a big one.

PN: I really feel that learning curve, managing learning curve, is one of the fundamental keys to ROI from data science. When I've been running teams, this is always the challenge. You're teaching up. You're managing up, but by teaching. You're teaching any time you present results and having a dialog there with the stakeholders.

But also, your people are continually learning all these different frameworks and libraries, and just managing that learning curve on a team is the hard thing.

So, a lot of Spark was how could we take and have something that was more generationally appropriate? Hadoop comes out of circa 2002, 2003 types of hardware that Google had. They created MapReduce and Hadoop was an open-source expression of this.

A decade later, you had radically different hardware: You had SSDs coming in. You had multi-core. You have large memory spaces, all these things, and Hadoop wasn't taking advantage of them.

So, the people who created Spark were seeing this in practice, in the field. They brought these learnings back into Berkeley, and they built something on top of Mesos that would show how to do what Hadoop can do, plus more, but really remove a lot of the bottlenecks, the synchronization barriers that are inherent in Hadoop, and get better pipelining across the cluster.

CI: What I'm seeing is an interesting situation where Hadoop is actually being eclipsed by Spark. Do you see it replacing Hadoop?

PN: It's interesting. Hadoop is a big word, so to speak. It's a large collection of different projects that share libraries, that share some practices.

CI: And some bizarre names.

PN: Yes, a whole zoo of animals. I see that as moving forward. That's a train that's not going to stop. The MapReduce part of this is definitely under attack from a lot of angles. Even inside of Hadoop, there are other things that are trying to get rid of some use case for MapReduce.

CI: To replace MapReduce, yeah.

PN: I see that a lot of the effort in MapReduce 2 has really been more about the infrastructure and running jobs across a cluster, having multi-tenancy, etc. The actual work on the MapReduce, as a paradigm, that's being eclipsed. I don't really have any stats to produce here for it, but what we're hearing in the field is, yes, absolutely.

CI: Heuristically, that's what we hear. What is Databricks' role, then, in the Spark environment?

PN: Most of the Spark committers are at Databricks. It's a very active project. There are a lot of people contributing, but the bulk of the commits have come out of the people at Databricks… 80% of the source code, so far.

We've really learned from the lessons of the past… Hadoop suffered… Definitely I remember Hadoop at the 0.16 stage. Everything got stalled. Maybe you had 0.18, but getting from there into 1.0 release seemed to take forever, and from there to 2.0, etc.

It just really stalled because it seemed like the committer base was fragmented, and different camps wanted different priorities, and they fought over it.

The nice about this is, number one, all the code for Spark goes back into Apache. There's no open core. In fact, a lot of the rhyme and reason of Databricks, in terms of certifying distros, is to try to prevent that, so that if anybody is bundling Spark, we are certifying that they're not doing an open core thing.

Anything that we do with Spark, we push back in open source. We're trying to not repeat the mistakes that we've seen out of Hadoop before.

CI: How do you control that, though?

PN: There is governance at Apache. Apache formally owns the software, and there's all the licensing, but there's governance in terms of how do commits come in. One of the contracts, effectively, with Apache is that you have a new point release every 90 days. We've been trying to keep to that.

There's a very formal process about votes and how do you manage features going in. And… so far, so good. It's worked out very well, I think, as far as my experience with other Apache projects that this one actually makes a lot of inroads and does keep to a very aggressive point release.

That's not necessarily the best, from a trainer's perspective, because every 90 days, I've got new things to update, but I'd rather have that.

CI: I think I would, too, than to have chaos.

PN: Exactly.

CI: Let's go back to Databricks. Databricks just launched its Cloud solution. We'll get to that in a moment, but you made some very interesting observations and things that, again, heuristically, I hear all the time, as well. That is that there are still challenges. The Cloud doesn't solve all your problems. Why don't we talk about the challenges first?

PN: Definitely. There's a lot of benefit for being in the Cloud. I've been watching. It's been very much involved since 2006, as far as Cloud architectures.

One of the things is that there are a lot of components. You go into some Cloud offering like AWS, and there are so many different things. There's EZ2, and there's RDS.

It's all these different things.

Understanding how they fit together is one problem. Understanding how to piece them together to a solution and then manage that effectively through your organization and interact with it in a way that you can really get return on investment, that's a hard problem.

In my perspective, there are really only a few companies that understand Cloud innately, and those are the providers. You've got a short list of Amazon, Google, Microsoft, IBM, I guess, arguably, would be in there as well. There are a lot of other people working with Cloud, but really, at scale, the economics and the practices are quite different than just a casual use of it, partly because the casual use is a really good way to spend money in a hurry.

If you want to leverage Cloud, there are operations practices that you have to follow. The people at Databricks have been doing this as their graduate work, understanding really from the engineering perspective, what are the economics of the Cloud?

That's a lot of what Databricks is trying to bring to this. It's not about having an enterprise version of Spark. Rather, Apache Spark is running there. How can we synchronize, coordinate all the resources underneath to make that cost-effective for you? A clear case in point is, if I want launch EC2 nodes to build up a cluster, say for Spark, each different VM is going to take minutes to launch.

If I have to launch a collection of these, if I need a thousand of them, how many are going to fail as I'm launching? They won't all launch simultaneously. I may be spending money in the Cloud for hours before I

finally get what I need, whereas what we're doing is essentially have a pool of available computer resources. We can spin off some containers.

The long and short of it is, you can get to having a cluster attached to your workflow within seconds, as opposed to waiting for hours. You don't have to have an army of systems engineers to be managing this. The actual Cloud operations are done for you.

We're really focusing on the work products. We're focusing on Cloud-based notebooks as a collaborative, a team focus for shared documents that take the place of maybe 20 Python scripts or shell scripts that we would have had in the past.

CI: Good benefits from Databricks, itself.

Let's talk about a few case studies, if you don't mind. What I'd like to hear, you've got a couple of really outstanding ones that talk about big data, how they got the big data down to the kernel that they actually did want to analyze, the sets that they thought were really the data that they wanted, and so forth.

They also had a number of business challenges that they were trying to solve. Let's start with a few of your case studies.

PN: We've had some published case studies and white papers about this that are available for download on our site, in particular, Radius in San Francisco, Automatic Labs, also San Francisco.

Radius is working with CRM data, effectively Salesforce data, and enriching that and drawing insights from it.

Automatic Labs, if you've seen the little white sensors that go into cars, if you have a rental car. They've just cut a deal with Ford, etc, a lot of telemetry coming off of cars out on the road. Another one is MyFitnessPal, which is doing exercise monitoring.

CI: I have that.

PN: They were just acquired by Under Armor. They're doing some very interesting work, in terms of diet and understanding foods. In some ways, I

would say that they're an analogy to food and nutrition as, say, maybe Factual is to business address data being cleaned up. Very interesting challenges there.

Another one that I'd point to would be Sultra and Timefold, a couple of other case studies that we have.

Let me focus, though, on Radius and Automatic, in particular. They had data, and they knew that they could produce insights that they could bring revenue off of. They could bring features and value out to customers.

The question is, where do you get all the ops team? You already have an ops team. They already have a job. In fact, they have too many jobs already. You want to add more and more analytics workflows, more features that you're building off your data. How do you avoid overburdening your ops people?

Or, how do you create an entirely new product without even having to go to ops and wait for them to spin up clusters, etc?

It's not just about spinning up clusters. The individual servers running in the Cloud, yes, those are needed. The practice behind it, though, involves people and a lot of cost. You have to monitor these things in case they fail. There are a lot of nuances to this.

What we saw with Radius, what we saw with Automatic, was they wanted to get the time reduced from raw data to insights that they can solve, and they wanted to have a broader audience of their workforce, their business people, working with analytics, jumping into the data right away without having to wait for a) the ops team, and then b) having to have programmatic access to the big data frameworks.

For both of these, what they were able to use was Databricks' Clouds so that their business people, their product managers, their analysts, their stakeholders, could go in and use SQL on the data, on the data sets that were prepared, and produce something that's useful right away.

Cutting that time down, cutting down the ops overhead, was crucial for them. What we see in both of these, and some of the other case studies as well, they had been using kind of a hodge-podge in the Cloud. They had some Tableau, they had some Redshift, they had some other things. Maybe they were using Hadoop before. In a couple of these case studies, they've repeated… they wanted to get away from having that kind of Frankenstein architecture, and just do something that's very easy to manage. So we saw Tableau and Redshift being displaced by Databricks' Cloud.

In other cases, we've seen both Redshift and Tableau being integrated, so it can go both ways.

Another thing for Radius was, again, getting back to the turnaround time, they had been using cascading workflows based on Hadoop, and had about a 12-hour batch window for their critical product. They wanted to drop that down. I believe they brought it down to about three hours. That's another benefit that we see across the board with Spark.

CI: All right, let's wrap up with my last question. You put up a wonderful slide on the analytics workflow. There has been discussion right, left, and center about artificial intelligence replacing the human being.

Are we going to have decision-making with no human interface at all into it? I think a little bit of the fear-mongering along those lines. Talk about that analytic workflow a little bit. Does the human still have a role?

PN: Absolutely. I'm going to go back to some perspectives. That slide that we showed was from a keynote at Data Day Texas earlier this year. I was doing a little bit of a retrospective, because doing work in machine learning in the '80s, the context was much broader. It had a lot more implication, I would say, across industry on optimization in general: planning, scheduling, all kinds of things.

In the late '80s, we saw a shift to focus on the algorithms: "I've got a better SVM than you do," and so there's so much focus on black box approach: "Just use my algorithm, and you'll get these results," but it really cut down the context of what you could do with analytics and machine learning.

A lot of what I was trying to show there is a generalized or an idealized workflow coming from disparate data sources, doing your ETL, doing some exploration, doing some future engineering, doing your training, your evaluation, etc, all the way out to where you're using the models in production.

In that, one of the points I was making is that, "Yeah, understanding the algorithms is important, but only a few points. You need people. You need people who are very good at exploring data, and it's a hard problem. It's not going to be automated. There's no magic bullet anytime soon."

When it comes to future engineering, this is also a hard problem. We can't just wrap this up in an API. There's no good black box yet. Deep learning is applying some nice techniques to futurize better, helping to automate high-dimensional spaces. Symbolic regression is another approach there, but this is not going to be purely automatic anytime soon.

Evaluation is also another area that I identified that you really need to know how you're using these metrics based on the learners that were used upstream. If not, an organization can really shoot themselves in the foot.

I see these three key points of: explore, futurize, and evaluate as things that won't be black boxed anytime soon.

CI: Well, thank goodness. We all have job security then.

Unfortunately, that's it for this edition of the BBBT podcast. Again, I'm Claudia Imhoff, and it's been such a pleasure to speak with Paco Nathan of Databricks today. Thank you, Paco.

PN: Thank you very much Claudia. I appreciate it.

CI: I hope you enjoyed today's podcast. You'll find more podcasts from other vendors at our web site www.bbbt.us. If you want to read more about today's session, please search for our hash tag on Twitter. That's #BBBT. And please join me again for another interview. Good bye and good business!