EPISODE 1027

[INTRODUCTION]

[00:00:00] JM: A high volume of data can contain a high volume of useful information. That fact is well-understood by the software world. After all, we have already had the big data revolution, and the big data revolution had plenty of people finding lots of value in their large volumes of data. Unfortunately, it's not a simple process to surface all that useful information from this high volume of data. There are certainly plenty of insights that we are not discovering from our mountains of data, and the typical way that we are still surfacing this data is with the help of a human analyst. A human analyst needs to understand the business, formulate a question and determine what metrics could reveal the answer to such a question, and this is the case is any domain where we're looking at large volumes of data.

Sisu is a system for automatically surfacing insights from large datasets within companies. The user of Sisu can select a database column that they're interested in learning more about and Sisu will automatically analyze the records in the database to look for trends and relationships between that column and the other columns. For example, if I have a database of user purchases, including how much money those users spent on each purchase, I can ask Sisu to analyze the purchase price column and find what kinds of attributes correlate with a high purchase price. Perhaps there will be correlations such as age and city that I can use to understand where my customers live and how old they are if they're going to buy expensive purchases. Sisu can automatically surface these correlations and display them to me and help me make better business decisions.

Peter Bailis is the CEO of Sisu Data and an assistant professor at Stanford. Peter returns to the show to give his perspective on the development of Sisu, which came out of his research on data-intensive systems including MacroBase, an analytic monitoring engine that prioritizes human attention. Sisu is an ambitious project and it's a great example of the kinds of systems that can be built if you embed machine learning deep inside of a system.

[SPONSOR MESSAGE]

[00:02:29] JM: DigitalOcean makes infrastructure simple. I continue to use DigitalOcean because of the low friction and attention to user experience. DigitalOcean has kept the experience simple and I can spin up a server in less than a minute and get high quality performance for a low price. For an application that needs to scale, DigitalOcean has CPU optimized droplets, memory optimized droplets, managed databases, managed Kubernetes and many more products. DigitalOcean has the flexibility to choose the right instance for the right workload and he could mix-and-match different configurations of CPU and RAM.

If you get stuck, DigitalOcean has thousands of high-quality tutorials, responsive Q&A forums and a customer team who treats customers respectfully. DigitalOcean lets developers focus on what they are building. Visit do.co/sedaily and receive $100 in credit over 60 days. That $100 can be put towards hosting or infrastructure and that includes managed databases, a managed Kubernetes service and more.

If you want to get started with Kubernetes, DigitalOcean is a great place to go. You can use your $100 to start building your distributed system and you can get that $100 in credit for free at do.co/sedaily.

Thank you to DigitalOcean for being a sponsor of Software Engineering Daily.

[INTERVIEW]

[00:04:06] JM: Sisu is the company that you're building. It's a system for surfacing metrics that are emerging from large datasets. Give an example of a metric that Sisu would surface.

[00:04:18] PB: Yeah. One public example I can talk about, Samsung, which is a customer, tracks device upgrades. They release a number of new phones every year. It's highly important to figure out who's adapting these new phones and what campaigns, carriers and so on are underperforming and over-performing. What Sisu can do is we sit on top of structured data about the new device upgrades, make model carrier, that sort of thing. There're a huge number of columns present in this data, but there's one metric of interest, which is conversion. Did this customer convert and what's the conversion volume? Sisu sits on top of the structured data and

helps explain changes to the conversion rate as customer behavior changes and as new marketing and promo materials are rolled out.

[00:05:06] JM: Sisu was built around MacroBase originally. This is the system that you've been working on for several years. What was your thesis around MacroBase and has that thesis changed overtime?

[00:05:19] PB: Yeah, totally. I started this MacroBase project back in 2015. I just accepted a project or a system professorship at Stanford and I knew it had like 7 years as a system professor to do something before they kicked me out. I've been doing transaction for [inaudible 00:05:34] at the time building really fast, scalable rewrite engines that make sure only one person gets the last object from the shelf at Amazon, for example. But quickly realized at the end of my PhD that not only the systems we were building, but a bunch of [inaudible 00:05:48] commercializing systems were building systems that were so fast you could do like one transaction per person on the planet every minute with like half a million dollars of hardware.

It has very quickly becoming the case that like if data volumes were to keep going up, it wasn't just going to come from people reading and writing to transactional data stores, but it's going to come from recording more information about every event in a business or in a given business process. The kind of core thesis was looking around what was going on in cloud. We said, "Imagine that storage is free or near free and imagine you have really good distributed compute engines like Spark because there's a lot of them and they all do pretty good job of parallelizing compute. Would you build on top of this?"

Just spending time with some friends, initially, some folks at MIT who had a startup looking at like analyzing driver behavior, but then later on looking at larger internet scale use cases with some collaborators and sponsors, places like Microsoft, and Facebook, and Google. The core question was what do you do with massive amounts of data that's large structured or at least is event-based so you have a bunch of columns, but you don't have enough time to dig in and actually look at every individual event. How would you prioritize people to tension in these massive streams of data? When a change occurs in the system, how do you know that the change occurred and why it occurred and what to do about it? That was kind of a big project, but some of the core ideas that we found that essentially landed were, one, people have

massive amounts of structured data. Surprisingly not just at the large internet scale, but talking to people in like retail and fast food. Like all of the fast food credit card receipts were not just like in a point of sales systems, but they were being uploaded in the database like Red Shift and Snowflake, which is kind of cool to think about all that data coming together.

Two, database were just increasing because you had more people buying tacos. It was that every taco you buy, you have more information on who's at the register and what was the fryer temperature and what loyalty program were you in? So, basically, that growth in both the availability of structured data and the width of structured data were surprising that it wasn't just the internet companies having this problem. It was kind of everyone.

[00:08:01] JM: The thesis of Sisu as you said is you might have a database or a table with a specific column that you want to focus on and find correlations with. For example, the number of sales or the amount of dollars per sale. You want to maximize the amount of dollars per sale and you want to find correlations between the other columns in a particular dataset. You may have these really elaborate databases with lots and lots of columns and you just want to find correlations among those datasets. I think the idea is that you would have these correlations be served to an operational analyst. Maybe you have a bunch of insights that get automatically generated by this database and they say the churn rate of this subscriber base increases 22% if you tend to send them an email at this certain time. Then the operational analyst can look at that insight and can choose to do something with it. Am I understanding that thesis correct?

[00:09:16] PB: Yeah. I think that's an amazing summary. In a nutshell, like I think the things that have changed in addition to just having this data that's available is, as you touched on, there's an increasing number of people who have access to this type of data inside of their businesses. We're at a bank in New York last month and they got 80,000 employees and 100,000 BI licenses because someone buys or does a migration to typically a cloud warehouse today like Snowflake or Red Shift of BigQuery. Then the next step, once you've aggregated and cleaned all of these data is you just buy a bunch of BI licenses, Tableau, Looker, MicroStrategy, probably a bunch of these actually at one. Then suddenly people built dashboards on top of this. It's not just like a data analytics team that has access to this data, but it's the marketing operations team, it's financial planning and analysis, it's store level operations that are all becoming more

data-informed. But at the same time, you haven't scaled the number of analysts who can truly go in and dig deep into these metrics and actually figure out what's driving a change.

I'd say historically, especially without aggregating this data, you have data in different silos. You might have had your sales data in one database and your marketing data in another database and your transactional level data in a third database. When you bring all these together, the probability that any individual business operator, so someone who's otherwise just staring at dashboards, the problem that they have an attribute or a set of columns in the data that they can actually take action on. So you brought up some good examples, like a marketing operator can change who their targeting campaign, or the copy in the campaign. In operations, we can change pricing and discounting and coupon codes. There are a lot of changes I can actually make in the business, which we'd already be making today, but there's actually data sometimes for the first time in these large organizations that are decades old that can actually inform those decisions.

The challenge is kind of twofold. One is computationally it's very expensive to go and run these types of correlations and different hypothesis tests on top of these huge, like really massive databases. The second part is just knowing what to surface to a user who's got time to look at three to five recommendations on an any given day out of this space of like hundreds of millions of things you could go and show them on top of hundreds of millions of records. What matters most to the marketing operations person, versus the performance marketer, versus the product manager, versus the store manager?

The cool part, and the reason why I'm super jazzed about this angle on analytics is that it's not like data analysis as a specialized function. There's this thing that Andreessen Horowitz likes to say, and I think they have a really cool article on this, which is like everyone's becoming an analyst. The question is, given the data that's available to them, our thesis is that the toolkit for making that data useful has got to change. Not just from a scalability perspective, but from a user experience and expectations perspective.

[00:12:16] JM: Let's say I have a large dataset for Sisu. Let's say it's the Software Engineering Daily listener base. Let's say I've got 10 million listeners. That's way more listeners than I actually have, but let's I have 10 million listeners. So I've got 10 million rows and let's say I have

a ton of data on each of these people. I've got like 80 or 100 columns and I've got the age. I've the gender. I've got their favorite programming language. I've got their favorite database. I've got the number of episodes that they listen to of Software Engineering Daily. Do they like structured data? Do they like unstructured data? I just want to know what correlates with people who have listened to lots of episodes. I want to put all these data into Sisu and I want to be able to just have Sisu spit back out insights about what leads to lots of listens. How does Sisu do that?

[00:13:09] PB: Yeah. Great question. What we essentially do, let's say you've got a row per listener in your database and you've got some column in there which is the number of listens, or say the number of listens in the last 90 days. You have kind of more recent data. What we can essentially do is we'd plug that data into Sisu. So we only work on structured data, so we'd connect to your cloud database or on-prem database, whatever database you have this stored in and you'd come in and configure what we call an objective or a metric that you want to track. You'd say, "My data is located in this table, or I can join together multiple tables. I've got a column here which is number of listens, and tell me everything you can tell me about increasing this metric."

Then all of the other columns, like let's say date that they came in, programming language, other interests, maybe age if you've got that. We'll throw it all in to this engine essentially by default in a graphical UI. Then under the hood what we'll do is we'll pull this data in and essentially do exactly what you do if you have a lot of time in your hands. What individual programming languages affect a propensity to listen? What are the combinations of programming languages and age ranges? We'll actually split the ranges up into different cohorts to see which ones make the biggest pop. We'll look at age on their own. We can look at location plus age for programming languages. Basically, look at this entire space of not just individual variables but combinations of variables. Then we can actually rank them by how much each of these variables and subpopulations are affecting overall viewership or listenership in the case of the podcast.

The idea here is that you'll get a rank set of factors that are most contributing to your kind of diehard viewership in not just individual variables but these often hard to diagnose cohorts. You might have a local community in Boise, Idaho that really loves listening to this, holds listening

parties every Wednesday night for the podcast, and that'd be very hard to look unless you happen to slice by day and location, but pops up in the data. It's already present. It's just hard to find.

I would guess actually given your background, you probably know who your most loyal listeners are, what topics tend to get the most hits. But one that we find really challenging from a business perspective is given that you're releasing new content, weekly, monthly topics are constantly changing. Historical content sometimes resurface on Hacker News or on Twitter. That ability to continuously keep up, it's not just what's happening right now, but once I've got that data connected, what changed over the last 7 days, what changed over the last 28 days, what changed this year versus last year? It's not cost-effective to keep up in any scenario with teams of analysts even if you're at a huge internet company, because they just have so many questions to go and answer for the business. But because we've got that data hooked up and the schema is already there, unless you change the scheme, we can just continuously rerun these analysis at different time periods and then actually notify you typically via email when a new factor appears so that if this episode comes out and it turns out people are really interested in this kind of data exploration hypothesis testing or maybe this takes off with folks who are into RUST, because that's what our backend is written in. Then you can actually be notified about that as it's happening as supposed to having to like look at the viewership graph. It might look flat, but there're actually some segments that are up and some segments that are down. It does sound like rocket science when you kind of lay it out like that, but the challenge is just doing this continuously and repeatedly and knowing what to show at any point in time.

[00:16:55] JM: Right. If I imagine those 10 million rows plotted on a Cartesian plane and you could draw all these different lines of best fit through them, I suppose, or you could do clustering. You could do all kinds of things to find correlations or find interesting insights to surface to a human analyst that's looking at this set of insights that Sisu is delivering. If you got 10 million rows and 80 columns, there are just so many potential ideas that could be surfaced. How do you prioritize which ideas to evaluate with your analytics engine?

[00:17:41] PB: Totally. No, it's a great question. The funny thing actually is this idea, people talk about these insight generation tool are spending much at companies in this space. The hard part when we talk to people in practice, our customers and prospects, is that no one really uses

this stuff because a lot of the engines just give out kind of garbage results. We had garbage results for a longtime ourselves as well. You can get results like when order value is not null, you always make a sale.

[00:18:10] JM: That's not very useful.

[00:18:12] PB: No. Not useful. There's like functional dependencies in the data that are just always true. Then even if you find things that like – I remember one of our early engines we were running with – This is back on campus. We're like users in Norway who on-boarded in the last 7 days are way more likely to be unusually active than other users and it's like a cohort of like 5 people, but they didn't exist last month and now they exist this month.

The ranking and relevance part of this is actually really hard and I think that's where, one, the ability to scale to do a lot of rows gives you in a statistical sense a larger budget to test different hypothesis. But to answer your question more directly, there're various measures of ranking and relevance that depend on the distinct user.

I'll give you an example. One of the teams we worked with at Microsoft was the Microsoft Skype team, and they have a ton of data about Skype call quality broken down by region and ISP a bunch of non-PII useful stuff to figure out how is Skype doing. For a PM that cares about call quality, they likely care about where are the outlier call quality is coming from. Where is the worst call quality coming from? Because it might be like a software protocol bug or something with an ISP. But if I'm a growth marketer, what I'm going to care about is something completely, which is whatever is going to move the needle up into the right.

Just to give like a concrete example of one metric we found very, very useful is there's a concept of counterfactual or sometimes in statistics what's called an influence function, and the idea is this; if I have a cohort of users, let's say of viewers, if I removed these viewers from the overall population, how much would that affect, say, the average 90-day rolling view count?

If I have a large cohort of users that watch a little bit more than average, then that'll affect my view count. If I have a few users that are very small but they watch, they just listen a ton, right? They listen to this episode a hundred times. They're also going to influence that average. This

weight of kind of assessing what's the impact or the influence of this group with the other rows versus without the other rows is a pretty good ranking function for anyone who cares about overall growth.

We basically end up combining a bunch of these and then ranking them on a per-user basis, but it's kind of like – I have to go deeper into this stuff. I can nerd on this stuff all day, but like the devils in the details in terms of not showing stuff that's super obvious and it's pretty contextual, but these different influence functions, different causal models and some hypothesis tests, you kind of aggregate them into a mega model and then rank.

[SPONSOR MESSAGE]

[00:20:54] JM: When I'm building a new product, G2i is the company that I call on to help me find a developer who can build the first version of my product. G2i is a hiring platform run by engineers that matches you with React, React Native, GraphQL and mobile engineers who you can trust. Whether you are a new company building your first product, like me, or an established company that wants additional engineering help, G2i has the talent that you need to accomplish your goals.

Go to softwareengineeringdaily.com/g2i to learn more about what G2i has to offer. We've also done several shows with the people who run G2i, Gabe Greenberg, and the rest of his team. These are engineers who know about the React ecosystem, about the mobile ecosystem, about GraphQL, React Native. They know their stuff and they run a great organization.

In my personal experience, G2i has linked me up with experienced engineers that can fit my budget, and the G2i staff are friendly and easy to work with. They know how product development works. They can help you find the perfect engineer for your stack, and you can go to softwareengineeringdaily.com/g2i to learn more about G2i.

Thank you to G2i for being a great supporter of Software Engineering Daily both as listeners and also as people who have contributed code that have helped me out in my projects. So if you want to get some additional help for your engineering projects, go to softwareengineeringdaily.com/g2i.

[INTERVIEW CONTINUED]

[00:22:41] JM: How much manual configuration do you expect from the operational analyst? Because I understand if you want to optimize for churn, you want to understand the metrics that contribute to churn. You want to understand the metrics that can lead to conversions or something like that. It may also help to have the analyst label certain columns like gender or age or these other things that like maybe the analysts suspects might have high-value. Do you expect any operational analyst configuration to be done in addition to selecting the one column to be optimizing?

[00:23:31] PB: Yeah. No. It's a good question. I think the implicitly answer is yes, right? Garbage-in, garbage-out in terms of features. But I'd say where we kind of fit in, and this is a very deliberate decision, we really only work with customers who are already looking at their data in some kind of BI tool. My kind of like prior for this is if you're not already looking at this data on a regular basis and aren't already slicing and dicing and trying to figure out why a metric is changing, the probability that you're going to act on that and actually get value out of a tool like Sisu is very low.

We typically come in to deployments where someone's already done the work of curating the features they care about and what we're basically acting as is kind of like the flashlight to tell them what's moving the needle and then kind of a monitoring tool to continue to track it overtime. If they haven't done any work in terms of like featurizing their data, like they just have like a bunch of flat files or they don't have any columns that naturally come to mind in terms of like, "Okay, if there was movement according to this column, like segment ID, or age, or region, then they're probably not a good user of the tool."

We'll do some data enrichment. In fact, you can do a lot of stuff with non-PII. Have you noticed some of the zip code, or coarse-grained location information you can pull in auxiliary data from both public data sources and alternative data sources to enrich that? In general, we're not really in the business of the data cleaning and data prep. It's really going in to people who are kind of scratching their heads today or may have had an analyst set up a dashboard for them with a couple of different breakdowns by region and so on and give them like a power tool that only

does one thing. It's not going to visualize as nicely as their BI tool. It's not going to do data prep like their BI tool. It's not going to do reporting like their BI tool. It's just going to sit on top of this thing and say, "Why is this moving? Why is this moving? Hey, by the way, this thing moved and here's why," moving from that reactive to proactive model.

[00:25:35] JM: The interface for the operational analyst, do you expect them to be sitting in front of this thing all day or just checking in every now and then or do I get like an email every morning from Sisu that says, "Hey, your churn was down 22% and it maybe because of these reasons." What's the end-user interface?

[00:25:57] PB: Yeah. It's a really good question. I think this is one of the reasons why I was so excited to take the work we were doing on this backend engine and bring it to kind of a commercial environment of a startup. For me, there's a bunch of hard problems in terms of like columnar database execution, distributed execution, vectorized query processing, data encoding. There's a bunch of stuff required to make stuff run fast, which is what I kind of love doing and where we did a bunch of papers and kind of where I hang my hat on the academic side. But the user interface is super critical, because in some sense, if you think about how ranking and relevance works on the internet, it's all based like what this person click on. What they didn't click on? Are they sharing or emailing this link with other people? Think about how Google is like a giant collaborative filtering model. Same thing with the Facebook newsfeed and Netflix as well.

I think a lot of ML and AI, people pitch it as like you'll get like the one true answer. Your ML model will tell you what to do. But in reality, if you think about like the early Google, you had the search bar and you had the I'm feeling lucky button. The idea that you get answer right on the first try was like a joke. It was literally tongue and cheek [inaudible 00:27:11] Oh! If you're feeling lucky, we'll give you this answer."

Similarly for us, there's no clear prior work on exactly what the interface should look like and what a great user experience should look like in terms of like actually getting someone to understand what we're talking about when we have like a given z-score and a t-test result and a bunch of information about the database and the cohort size and so on. That's like one part. Just explaining what the heck is going on.

Then the other part is actually building interfaces where you can gather feedback from users without having a very lame like, “Thumbs up. This was a good answer,” or “Thumbs down. It was not a good answer.” I’d say like one of the big innovations from a computer science perspective that I’m really excited about and excited to talk more about overtime even writing some papers is really about like what is the user interface design look like where you’re giving kind of this recommendation-oriented approach and you’re able to explain things in a way that’s accessible to people who don’t even know SQL.

Just to give you like a flavor of this, we have this design principle, and some of our early designers are awesome, like [inaudible 00:28:17] actually helped redesigned the Twitter timeline going from in order tweets to the out of order rank tweets. If you don’t like that, you can stop by the office and complain to [inaudible 00:28:25]. Aaron led design on GitHub enterprise. Really, series designers. Basically, we essentially have this threshold of low-threshold, high-ceiling. We want to make so that anyone who kind of knows a little bit about their data can get an answer, but then you can keep going deeper and deeper and deeper.

For example, if you’ve got an objective configured about increasing viewership to the show, you can subscribe to daily or weekly updates about what’s changing a viewership, and in your inbox you’ll essentially get what we call a fact, which is a statement about what has changed in your data. We might say from this month compared to the prior month, or from last month to this month, viewership increased by 2,000 views. The group where topic of interest equals machine learning and refer equals LinkedIn contributed 1,500 of those views to your total. Basically, it’s English language description using the data. It’s basically column name and column value along with associate statistics that will like just show up in your inbox in a sentence.

Then if you’re like, “What do I want to do about that? Why about just only LinkedIn or what about only machine learning?” You can click in and then we can get you the raw data behind it. We can look at, say, LinkedIn, and RUST LinkedIn, and Java LinkedIn, and C++, or you can look at Twitter and machine learning and all these different alternative facets of ways of looking at this data to figure out like – Maybe the case that LinkedIn and machine learning drove a lot of traffic, but actually what you want to do is you want to start promoting your LinkedIn posts on Twitter or on Hacker News.

We don't tell you exactly what to do with that, but we tell you that something has changed in a way where you can kind of get the high-level drivers and then go deep to formulate a recommendation, where if you're like an Uber analyst, you can annotate this thing and even export a slide deck for your boss end-to-end to make that recommendation "business friendly".

[00:30:25] JM: Let's talk a little bit about the data process before entering Sisu. Modern data workflows have become centered around this data warehouse, like Snowflake, or Red Shift, Apache Spark is kind of a data warehouse some people might say. I think we started doing this show about 4-1/2 years ago and I feel like back then the running hypothesis was that the way these big data
systems would evolve would be that you would have a lot of processing take place in the stream processing systems. You have Storm, or Spark streaming, or Flink, and these would be the systems that would generate the end results. Does it surprise you that the workflows have become centered around the data warehouse?

[00:31:19] PB: Yeah. It's a good question. It doesn't actually surprising that much, and I think there are kind of two big reasons for why I think this is what I see in terms of driving this shift. One is that you've got – Like if you think where the data is coming from, a lot of data is coming from SaaS applications or microservices, right? If you think about the data volumes that people are – Like growth of data. It's not coming from like people clicking more stuff on your website. I mean, there are some data in terms of click streams, but the vast majority of this when see inside of a data warehouse at least from a customer-facing perspective is coming from getting more and more context from different systems.

Salesforce led the latest round at Snowflake, which is kind of crazy because Salesforce is like a data silo in a sense all about the sales data. But the reality is like if I want to analyze how my marketing campaigns are affecting my sales, it's actually much more efficient to aggregate that data inside of a data warehouse rather than pumping all my marketing data Salesforce and all my Salesforce data to, say, Marketo. So it's like this many-to-one connector where if you think about the data volumes at scale coming from automated sources, like the amount of "work" to pipe data into these warehouses is basically linear in the number of connections I need to make.

You've seen even among – Like there's this cottage industry of players. One of my favorite searches, if you Google like Salesforce to Red Shift, the first 20 results are like SEO'd landing pages for a bunch of companies trying to sell you connectors as a service, and some of them are really good. We actually use one of these vendors ourselves, because it's just cheaper to outsource that to someone else and pay by volume. It's almost getting commoditized.

Rick Branson over at Segment actually made the observation not on behalf of Segment officially or anything, but was talking to me. He made a smart automation that overtime there's a push to even push that connector functionality into the SaaS products themselves, which then you have standardized schemas and it's not that complicated to get your Salesforce data out and into one of these warehouses.

It is still a lot of work. Data engineering is not going away, but it's pretty good. The flipside of this is my second reason why I think we're changing this, is like it's not bad to get this data into these warehouses, and then it's really hard to write streaming jobs. Stream processing is just such a mental – It's like concurrency, right? It's really hard to write good streaming programs just because it's – As humans, we don't think about saving state and partial materialization and all the stuff like when industry and processing.

I think that compared to the complexity of writing a streaming job, doing a bunch of on the fly ETL and then loading stuff in, especially for a lot of business processes where you only answer like on hourly basis or a daily basis, you can just write like batch jobs that do some data integration on top of your Snowflake or Red Shift cluster and it's not that bad in tools like DBT and just rolling your own transforms. It's not the end of the world. Where certainly look for high-frequency trading and finance and so on, you definitely need that scale of streaming data and time series database. It's super useful, like IoT. But if I am like a marketing operations team, I can get a lot from off-the-shelf connectors and a little bit of information possibly work up front with some of the CDP platforms out there to get a UUID I can use to join. Then I'm kind of done.

[00:34:51] JM: So your explanation for why the emphasis went to the data warehouse rather than the streaming systems is basically that the data warehouse is a familiar interface to familiar environment and the upside to moving to a streaming system is that the data is going to be just slightly more up-to-date, but in most situations it's not going to matter enough to incur the

engineering cost of trying to implement one of these complex streaming systems rather than just saying, “Ah! I’ll use a data connector. I’ll get the data connector to hook into my data warehouse and then I'll give it to the analyst, and the analyst speaks SQL, and I'm done.” That's just easier than futzing around with streaming systems.

[00:35:38] PB: That’s kind of like the good summary. I think the thing I found that's really interesting is when we started doing the stuff, this research at Stanford, we were like very in the IoT. In fact, if you look on the early drafts we post on archive, it was like MacroBase was a system for IoT data, and I think what I've seen in kind of my like reasonable confidence in this prediction is there is like two classes of data by volume, right? There is like machine data where GE used to brag about how many terabytes of data it generated during a given flight, or you can take kilohertz readings of like seismic graphic data, if you're really into that stuff. You can just generate arbitrary amounts of machine data. Then there's like human scale data, where with three something billion people on the US and less than 10 billion people worldwide, like you can fit a lot of data per person even on like a single server.

For that human-generated data or that data that pertains to humans, it's not so high-volume, you need a completely new architecture unless you’re measuring people’s like pulse per like kilohertz or something like this. There’re not a lot of use cases for that fine-grained information. Even for the machine data, like outside of a few specific applications, like I don't actually care what the rotational velocity of the jet engine at every single – I don't if jet engines rotate, but whatever those sensors are gathering. I just kind of want to know like what's the fuel economy of that flight, and I can extract higher-level structured information about these like fine-grained sensors as well.

I think there's kind of bifurcation. In a lot of the really valuable enterprise use case we've seen, the data just isn't that big. People used to make a big deal like, “Oh, you can fit all these big datasets in-memory.” Now I think like the new one for a lot of businesses outside of like Google clickstream data is like you could fit all your data about your users on like one machine.

[00:37:31] JM: The data emphasis has become on the data warehouse, and how does the data get into Sisu? Does it go from a data warehouse into Sisu?

[00:37:45] PB: Yeah. I mean, we make a strong emphasis not to reinvent the wheel, especially wheels that really don't squeak a lot and are well-polished. Our default interface is basically just ODBC, or the open database connectors that are available for every warehouse. The kind of geeky, funny back story, we wrote a blog post on this, but it turns out like reading data via ODBC is actually kind of slow. For compatibility reasons, the first step is just ODBC and we can do that with any major database. But it turns out, like for things like Red Shift and Columnar [inaudible 00:38:16], ODBC is basically row-based interface. It's actually faster to do like a parallel unload to S3. Actually, it's slow because – Sorry. [inaudible 00:38:26]. It's slow because Red Shift is a partition database. It's faster to actually unload Red Shift to S3 in parallel and then read in from S3 in parallel on the backend. We have a fast code path for that because we deal a lot with Red Shift, which is like kind of insane to me how crappy these database connectors are in terms of just throughput if you really want to suck data out.

In general, yeah, we rely on the ubiquity of SQL and the value of the ODBC connector. It's just like insane how good like – As you said earlier, the warehouse obstruction is, and like every tool speaks ODBC. Every kind of like BI vendor speaks ODBC. If you don't really speak ODBC, it's unclear if the date is even valuable, at least from a business perspective.

[00:39:11] JM: Once the data gets into MacroBase, does MacroBase a full-on m database system or are you using some database under the hood like a SQL database or something like that?

[00:39:25] PB: Yeah. What MacroBase, essentially, it's still open source. What it basically was, we haven't really developed it for a while for a couple reasons I can go into. Basically, you can think about the top half of a database. It's a query processor that runs like one type of query, which is given in aggregate. Take your view count or churn rate or some expression over a column or set of columns. Tell me what's driving it. Run these large number of hypothesis tests and then rank the results according to some ranking function of those tests.

What we've essentially built with Sisu, we end up scrapping all the MacroBase code, because I love Java. I wrote a lot of the early code in MacroBase, but like running in production was a real kind of bare especially when you have really, really large, like couple hundred million record datasets, thousands of columns. We end up rewriting everything in RUST. But same principle,

distributed data flow engine. We do ingest into a specialized columnar format and use special encoding based on if you only have two distinct values in a he column, you might want to bit vector bitmap encode it into a BitVector. If you have many columns, you might want to just encode it into a normal column representation.

Then the engine basically has this in-memory representation possibly distributed and then we'll parallelize this essentially search over the space of possible factors you might want to show to a user and then you put ranking on top of it and kind of go from there. It's like I describe as the top half of a database. Because, again, like Snowflake, Red Shift, BigQuery, these are amazing systems. They're really well-done distributed databases. Rather than trying to reinvent distributed databases just like we do on Reinvent BI, we kind of sit in between and we just say, "Look, we want to be the best and the fastest at telling you why." That requires a new query processor, because you push it down into like a kub statement in a database. It takes weeks to answer because it's a combinatorial explosion. But by pulling it out and doing it in our specialized data formats and with much pruning rules we've got on our backend, it goes orders of magnitude faster.

[00:41:33] JM: If my data warehouse is Red Shift, you're going to be executing the actual queries in Red Shift and the query planning is going to take place in Sisu?

[00:41:41] PB: What we do is we'd run as much as we can. We'd push it down into Red Shift and then will extract the result set from a base query of slightly modified base query. We'll prune out certain columns or do joins inside of the system. Then we'll pull it out into our backend environment. The back environment just kind of like DataBricks does it, we can deploy it inside of some of our customers, VPCs. For one customer, we're deployed in multiple different data centers or different AWS environments because of data compliance and governance issues preventing cross-continental transfer. We also do it on top of GCP.

Kubernetes honestly makes this whole thing amazing. I think building a company in a cloud service from scratch in 2018 on top of Kubernetes was like a dream compared to what I think the DataBricks had to do back in 2013 in terms of their deployment and VPC pairing ant stuff. In a nutshell, we suck the data out of the database. It runs in some secure processing

environment, and then we extract the facts and then we have kind of a rendering template that goes out to users once this exhibit ranked.

[SPONSOR MESSAGE]

[00:42:50] JM: Today's episode is sponsored by DataDog, a cloud scale monitoring service that provides comprehensive visibility into cloud, hybrid and multi-cloud environments with over 250 integrations. DataDog unifies your metrics, your logs and your distributed request traces in one platform so that you can investigate and troubleshoot issues across every layer of your stack. Use DataDog's rich customizable dashboards and algorithmic alert to ensure redundancy across multi-cloud deployments and monitor cloud migrations in real-time. Start a free trial today and DataDog will send you a T-shirt. You can visit softwareengineering.com/datadog for more details. That's softwareengineeringdaily.com/datadog and you will get a free T-shirt for trying out DataDog. Thanks to DataDog for being a sponsor of Software Engineering Daily.

[INTERVIEW CONTINUED]

[00:43:51] JM: The initial query, I guess there's – There's query plan that's created, and then part of the query executes against my Red Shift or my Snowflake or whatever the underlying thing is, the underlying data warehouse. Then there's some results set that just is emitted and that's sitting in-memory in the Sisu architecture and then there some final steps that take place in the Sisu in-memory system.

[00:44:18] PB: Exactly. Yeah. I think there's a trend for a while with these systems like MADlib, and I think Impala supported some this as well, like to push the SQL or to push like ML down into these engines. I think BigQuery as BigQuery ML now. The reality is programming these UDF's, or user-defined aggregates, is really painful because you're basically sitting inside of like – If you do it in Postgres, you're using Postgres as memory manager. It's really hard if you parallelize this stuff and it's actually pretty dependent on the database.

G given that we have such a wide deploy base, and I'm not even sure what like Snowflake support for user-defined aggregates is. We felt for the performance reasons and portability reasons, just doing this outside of the database. Even though we take a hit in terms of data

transfer speed, it's still faster. Because, again, like you just can't do – If you like repeat a group buys, for example, to evaluate some these different hypothesis tests, even like the ODBC set up and latency for each of these successive calls over a portion of your of your [inaudible 00:45:19] is just like super expensive.

For better or worse, it's basically requires building a new engine. I think if I'd make one ask of the database vendors, which I'm sure they never do, is they can easily get the data out. Because it's the same thing that you do if you're processing on top of Spark, like being able to do all the transformations inside of a native processing environment as supposed to inside of a SQL system that's pretty brittle and specialized for storage and has archival processing and all these sort of good stuff that why you buy a warehouse in the first place just makes a lot more sense.

[00:45:53] JM: You mentioned before that the early days of the database or of Sisu, you felt that the insights weren't super useful are this system would continuously emit insights and maybe some of them won't be useful. Some of them will be questionable. I can imagine this is the core or at least one of the fundamental challenges to getting this thing to market, because if you have enough – If you serve enough bad results to the analyst, the analyst is eventually going to stop paying attention to the emails or stop painting to the dashboard or stop paying attention to whatever insights you deliver. You really have to have a high hit rate of useful insights that you give to the end user. How do you even benchmark whether Sisu is good enough or is delivering good insights at a good enough basis? I guess I'm curious about the product development process and the go-to-market process at this point.

[00:46:53] PB: Yeah. No. Totally. I think this is like depending on who you ask the billion dollar question or hundred billion dollar question. We knew that this thing would work and that we had people actively using MacroBase and variance of it in production and still have it. We wrote papers with Microsoft, Facebook, Google. We have a pre-print or a submission with Microsoft talking about queries on the system we deployed there. It's like over 20,000 per week running in production. People were finding it valuable. But the reason why it worked really well was we working very closely with our data engineering teams to set up a service where we could tune the parameters for known workloads. Like the Skype use case, we kind of knew what like good settings for these different hyper parameters would be, and it turned that when we got them

right, we were able to reduce kind of false positives and noise enough that people were pretty actively using the system.

That kind of gave us success there that the idea of like, “Okay, maybe there is that there there and we don’t have to guess at these things and we can actually go and deliver something valuable that works across these different use cases. We knew we could do it if we tuned it. Then the question was that Sisu, like as a kind of vertical-agnostic platform, like how do we actually deliver these types of results ideally without doing a bunch of data engineering on a per customer basis?

Kind of the thing we've found is there're a couple common tricks that we apply by default to everyone. For some of the Microsoft data which is pretty cleaned up, there weren't a lot of functional dependencies where like one column was perfectly correlated with the other and there weren't a lot of nulls, because they owned the end-to-end system from like the SDK running on the client all the way to the columnar storage engine. It was a really like clean room built from scratch product analytics system.

Some of the stuff like with these nulls and so on, we just had to add basically logic to detect these cases and have a set of pretty good pruning rules by default. But then the other thing that we've done quite a bit of is overtime just invest a lot more in different signals for ranking and relevance. Specifically, like we’ll look at things like click through rate and return visit rate and we can run kind of what are called like bandit style algorithms to experiment between ranking functions because we own the end-to-end product experience, and now it's one of the things that was very attractive to me in terms of doing this as a business, where we can actually look on a per account basis and understand what types of facts and what columns and combination of columns are most valuable to which type of user. That's like been super helpful, because there's no substitute for real data and we don't have the data volumes that like Netflix has about who's watching what movies or Google has in terms of jus clicking what links, but you still get some signal about that you can then use to tune all these parameters in a more automated way on a per account and per organization basis. That's been a big part, is just like looking at all the stuff you think about in consumer but on an enterprise basis and then using similar ranking functions to take advantage of that.

The final thing I'll say which is really important is also just setting the expectation for what we show people, right? This isn't like we're going to give you the silver bullet. It's like we're going to give you five results, and if two out of five changed what you were going to do today or what you're going to this week, that's still pretty valuable for a lot of our users, because they're kind of flying blind with all these dashboards.

So a lot of companies we've seen who talk about doing anomaly detection or like finding like the outliers and the data, like we've given up on outlier detection even on the research front, because what's an outlier for you? Like finding the needle in the haystack? That's completely different than what's an outlier for me. If I wake up at like 10 AM on a Saturday, like that's an outlier for me today, but was not an outlier for me like two years ago.

We make it really clear, like we're not going to find the single points in the data that are unusual or anomalous. What we're going to do is we're going to present you a set of results that are the top factors or groups of points that are moving the metric for your KPI and they're going to be ranked based on what you've clicked on and have saved in the past. That actually closes a lot of the gap between kind of what you can actually do with data and what people want to do with the data. It's an education process. In a lot of conversation with prospective customers, we say no all the time. People say, "Oh! Can you forecast my sales next quarter?" It's like, "No." It's like, "Oh, can you clean up my data for me?" We say, "No." They've gotten so used to vendors promising the world and under delivering that actually saying no and being very deliberate. We will help you diagnose the changes on top of your structured data. We will give you a set of recommendations. We will tell you the top cohorts. We are not can it detect what's going to happen in the future. We're not going to find individual outliers. We're not going to do targeting for you. That helps build a lot of credibility and when we actually give the results to them, that coupled with the risk [inaudible 00:51:42] result quality lead to a pretty good results in practice and results we couldn't get in a lab because, again, getting access to this data and actually working on these use cases beyond tech companies is a lot harder from an academic environment.

[00:51:58] JM: You worked with both Ion Stoica and Ali the code see when you were at Berkeley and I knew you also been a close watcher of sparking Druid what you always even influenced by the people who you've encountered in the the open source projects that you've

seen get commercialized. Yeah. So for say you know working with Jan and Ali Ghodsi when you were at Berkeley, and I know you've also been a close watcher of Spark and Druid. What have you – In what ways have you been influenced by the people who you've encountered and the open source projects that you've seen get commercialized?

[00:52:21] PB: Yeah. I'll first say working with Ion and Ali was like just an amazing experience. They're two of the sharpest technical minds that I know, and Ali in particular who's continued to be a very close mentor to me, he's probably has the greatest combination of technical IQ and then business savvy and EQ that I've ever seen. I think Ben Horwitz who's on both of our boards would say the same thing. I mean, he's just a phenomenal individual. Seeing them go to market and seeing how they've iterated, they started with just kind of – They were very early in the cloud. DataBricks made at bet, a big bet that they weren't going to do service and support. They basically handed that revenue off to Cloudera.

If you look at where the companies are now, it was the right bet to make, because it let them stay focused on iterating with the early customers and staying ahead of the cloud vendors because DataBricks is an infrastructure provider and their number one competitors are the cloud. Now, like Azure even sells DataBricks first party. They've done a great job of staying focused on what they want to own, which is cloud data science, and even imply with the Druid folks, they're also offering cloud offering.

For us it's a little bit different, because we're not really – We don't have super tactical buyer in the sense that we're not selling into IT. We sell to analysts and we sell to business teams who need to become analysts. That's a little bit different. I remember early on talking to the band and saying, "Gosh! We spent so much time in security review trying to get access to the data to show this to show everyone wants a magic orb that will tell them whether metrics are changing. It's another thing to actually run it on their data and show the value. So much time early days before we had like SOC 2 and HIPAA and this stuff basically waiting to get access to data and said, "Can't we just build like a desktop agent?" Because we had some pretty cool users. I was really surprised, if you'll pick up MacroBase which had like a desktop client in a single core mode and like people found some pretty cool results that were real legitimate companies tracking real metrics, and Ben's take was like, "Don't compromise on running fast and iterating.

Stay cloud-based." For the folks who aren't in the cloud yet, including much of banks even today, they'll get there over time. I think that's been really, really interesting.

For us as well, from the open source side, it's kind of funny because like if I'm a business operator, like I'm a marketing operations person, I don't care if the thing is open source or not. I just want to get the answer. We've contributed some stuff back to open source and pretty cool statistical packages around graph coloring and we'll be releasing more over time, but like for our core user persona, they're just trying to do a better job. We've just focus on getting them the best answers the easiest way, and the cloud has proven a really effective delivery mechanism for them.

[00:55:16] JM: All right, last question. What would you be building if were not building Sisu?

[00:55:20] PB: Good question. Let me think. I'm not actually sure. I don't think I'd work on transaction processing given that transactions are pretty fast or fast enough. I think the thing that I think that's actually really exciting but also very scary, so I'd want to find a way to do this in a private way, and I have a couple graduate students working this as well. But if you're thinking about the data that we do have in these structured formats, it's all coming from like software and software services. If you look at where we're in terms of the availability of center feeds, in particular with like autonomous vehicles and the fact that almost every car will have a GPU in it soon. There's a ton of data about the physical world that we just don't have captured in digital format. That's going to be very easy and cheap to capture soon enough.

Think a Google Street View. It only updates every – I don't know, couple of months or whenever the street view car goes by it. But if you suddenly have an environment where you've got cars going and looking at everything going on every sidewalk and every urban environment, there's a bunch of scary surveillance challenges there. Figuring out how to do this in a secure and privacy preserving way, but actually getting analytics and running models on the edge and trying to understand things like what's the line like at Starbucks, and what's the average wait time for Bart, and actually answering these fundamental question about the physical world using these new sensors that are just coming online. I think that's a really interesting programming interface problem. It's a systems problem in terms of data collection, aggregation, so on. These cars generate just tons and tons of data. Then finally the privacy ones are just huge because, for

better or worse if you think about what the hardware in a Tesla or inside of any one of these autonomous cars you drive around San Francisco, they're recording everything, and understanding how do we make use of this data in a productive way without compromising individual user privacy is a huge concern that independent of the applications around you retail analytics and so on, I think we need to solve.

[00:57:20] JM: Peter Bailis, thanks for coming on the show. It's been great talking.

[00:57:23] PB: Really appreciate it. Thanks so much.

[END OF INTERVIEW]

[00:57:34] JM: As a company grows, the software infrastructure becomes a large complex distributed system. Without standardized applications or security policies, it can become difficult to oversee all the vulnerabilities that might exist across all of your physical machines, virtual machines, containers and cloud services. ExtraHop is a cloud-native security company that detects threats across your hybrid infrastructure. ExtraHop has vulnerability detection running up and down your networking stock from L2 to L7 and it helps you spot, investigate and respond to anomalous behavior using more than 100 machine learning models.

At extrahop.com/cloud, you can learn about how ExtraHop delivers cloud-native network detection and response. ExtraHop will help you find misconfigurations and blind spots in your infrastructure and stay in compliance. Understand your identity and access management payloads to look for credential harvesting and brute force attacks and automate the security settings of your cloud provider integrations. Visit extrahop.com/cloud to find out how ExtraHop can help you secure your enterprise.

Thank you to ExtraHop for being a sponsor of Software Engineering Daily, if you want to check out ExtraHop and support the show, go to extrahop.com/cloud.

[END]

