EPISODE 1151

[INTRODUCTION]

[00:00:00] JM: Predicting the spread of COVID-19 is not easy. The best methods we have available require us to extrapolate trends from a large volume of data, and this requires the construction of large scale models. Because of the expertise needed for developing these models, Silicon Valley engineers were brought in to help develop a maintainable model. Two of these engineers are Josh Wills and Sam Shaw, and they join the show to talk about the engineering behind the COVID model and their work to help build it.

[INTERVIEW]

[00:00:33] JM: Guys, welcome to the show.

[00:00:36] JW: Thank you so much. It’s great to be here.

[00:00:38] SS: Great to be here as well.

[00:00:39] JM: So we have two of you, Josh, you’ve been on the show before. So we’re familiar with each other, what have you been doing since COVID?

[00:00:46] JW: Very little, like broadly speaking. I tried a job doing self-driving car stuff and failed. It turns out they did not need my help for anything. And then aside from that, I’ve been investing and advising, which is just the things you say when you’re an unemployed person in San Francisco. So yeah.

[00:01:04] JM: And Sam?

[00:01:06] SS: Yeah, likewise, investing and advising the trifecta of the Silicon Valley hierarchy of needs. Basically saying that I’m not really doing anything much right now, but I had recently sold a company. So kind of figuring out the next thing.

[00:01:21] JM: Well, let’s talk about something totally unrelated to investing and advising. What do you guys see as the primary problems faced by epidemiologists today?

[00:01:31] JW: Oh, it's a great question. I mean, I think the answer is that public policy officials don't listen to them as much as they should.

[00:01:41] JM: What about more technical, like realistic problems?

[00:01:46] JW: Sam, what do you think?

[00:01:47] SS: I think there are a couple of things, right? One, which is like the quality of the data. So actually, if you look at pulling things out of public health systems, like the data is quite messy and quite difficult. I mean, it's like a hundred lines of epic just to get any reasonable like kind of information on anything, which is very challenging. And then you have this massive kind of data integration problem to pull all those kind of data sources together.

I think the other piece around epidemiology and kind of software is that like – I mean, they're fantastic in epidemiology, and it's really good that the software stack has gotten a lot better for them to do things. We have R, we have Python, we have Docker, and they can like do all these – And a lot of open source and a lot of kind of tools that are there. I think there's still a gap between like the language that they use and the language of software. So they can't like exactly map what they want to do on to software, right? So even running things like in the cloud or in AWS, like what services should you use to connect things together?

A software person really gets that, but an epidemiologist, it's still like, "Okay, I know how to do things in R, but how do I go beyond that?" And there's still that big gap of like how do you do that for either dealing with more data or doing things, which are more difficult around, like more memory than you have on your single machine, right? And that's kind of a classic problem that a lot of data scientists face. But I think epidemiologists are like adjacent to data scientists and like they also don't speak the language and have the binocular to connect and really understand what they want to do. I think there's like a bridging of the gap that needs to happen between software and epidemiologists to like be able to be I think a lot more effective. And that gap has closed quite a bit, right? Like people use GitHub, Docker and things like that. But really using all the powerful tools that you want I think would be very challenging. [inaudible 00:03:40] epidemiologists used Spark, for example, would be an incredible Herculean effort for them. And what they probably want to do is relatively simplistic. I mean, in big data sense, but they just can't get that, can get that done right.

[00:03:55] JM: How can the technology world help epidemiologists?

[00:04:00] JW: It's a great question, like giving that some thought right now. How do we make – The good news is that, as Sam said, like an epidemiologist and a data scientist are pretty substantial. It wouldn't shock me if there were quite a few epidemiologists out there who became professional data scientists for one reason or another in the same way that like astronomers, or physicists, or sociologists, or whatever all become data scientists.

I think in the context of COVID, I think the thing that was missing and the sort of the gap that Sam and I and our colleagues had to fill was being kind of like cloud experts at the ready. So generally speaking, if you're an epidemiologist and you're working in academia, you file a grant, grant request. You get your grant fulfilled. Part of that grant, you'll get some software. You'll get some hardware. But generally speaking, at least as far as I know, getting access to like cloud resources is not generally part of something you do as part of like filing a grant. And so because of that, they don't have a ton of experience using all of the great resources in the cloud and getting access to this stuff to do their work.

But Sam said it's a knowledge problem. It's a resource problem. I think having the various cloud providers provide that expertise, or at least have it at the ready really for like epidemiologists and all kinds of scientists would be incredibly valuable. Yeah, Sam?

[00:05:26] SS: I think that's right. Even if they had access to the cloud, I'm not sure what they would know what to do with it and what is possible, right? So it's almost like a failure to imagine, because they don't understand the capabilities, right? And so you need to be able to have – I think there's a translation that needs to happen between these kind of software people and infrastructure people and epidemiologists, right? And like once that tight coupling becomes I think a lot much more tighter, I think they could like really understand what it means to run something on like a thousand computers, right? What could you actually do if you could do that?

[00:06:01] JW: Right, and how to do that exactly. Yeah.

[00:06:03] SS: Right.

[00:06:04] JM: Tell me a little bit more about how you guys have gotten involved with helping epidemiologists?

[00:06:08] JW: Yeah, Sam. Why don't you get us started?

[00:06:11] SS: Yeah. So like I think like we mentioned earlier, Josh and I were both kind of investing and advising, take a little bit of traveling and then like everyone else, COVID hits. We both knew a bunch of people in the California government, specifically people that were working on modeling and understanding the epidemic. They had some basic kind of model. They had a difficulty kind of getting at the scale and run in that sufficient in that amount of time so they could iterate on it, right? The model for California for example would take like 6 to 8 hours to run a model. For the entire United States, it would take 4 to 5 days to run. And then in a fast-moving epidemic, that is like not sufficient to deal with any sort of real question coming down the pipe from our leaders.

And so we kind of came in and helped with the California Department of Public Health to kind of go and figure out how could we do this a lot more effectively. And basically what happened was there was – Part of the California Department of Public Health, CDPH, they were working with people like Johns Hopkins and a way to kind of model this epidemic. They had no real software distributed systems people.

And so John and I and a bunch of other people actually came together to bridge the gap there and provide the ability to run the model much more efficiently, much more faster. I mean, really what happened was is in the early parts of the epidemic, I think like March 16th of this year, do you remember kind of – Put things in context, the state of the world was California was trying to understand the epidemic. I think California at the time had like 18 deaths. The entire United States has less than 500 deaths. But everyone had seen things that have been going in Wuhan. Also, the stuff that we saw things in Northern Italy where like hospitals were being overrun and then you started to hear things kind of in New York City, hospitals are starting to scramble to kind of get bed capacity, right?

So really what we wanted to do is was get a picture of this epidemic and how it would affect California, right? There was no wide-scale testing. The CDC hadn't approved any tests yet. And so really what we want to do was build a model, the coronavirus task virus, Governor Newsom could use to understand how this epidemic work, right? There was some open source or public rejections, but they were missing a few kind of key ingredients, right? One which was everything was done at like an aggregate state level, but nobody lived in the aggregate or really understand things like a country or a ZIP code kind of level.

The other piece is we needed to model and understand critical resources, like ICU beds and ventilators. And the third piece is we need to understand and like forecast or scenario plan how mitigation steps could affect the epidemic, right? So what if we had a masking in rural counties and we shutdown schools in urban counties? It's just an example, right? If you could try different kind of scenarios and see like how does it going to effect the epidemic, you can now plan and more effectively understand things. And again, understand things at the county level so you could things at a per county level.

So really what we ended up helping out with was rag tag group of like CDPH officials, John Hopkins epidemiologists, and then Silicon Valley kind of engineers to retrofit and kind of operationalize a model of the epidemic, right? And this was basically – I would put it akin to like a startup experience, where like – And this is actually pretty close to what it was like. It was like Saturday, you interview for the job. Sunday, you're hired. Monday is the first day of work, but there is a critical business issue in an unknown domain that you need to solve right away in a couple of days. You need to show something to your CEO or board immediately. You need to boot up like a new whole sense of knowledge and new software staff working with a whole new set of colleagues.

And we're able to get a deep sense of mission and kind of grind it out. There was couple of all-nighters to really get this kind of model together. And really what we ended up doing was showing to the governor stuff kind of different kind of projections of like how the epidemic would spread. One, which is if you did kind of nothing, the unmitigated scenario. Another one would be like if you did a Wuhan-like kind of lockdown, which is not possible in democracy, but that gives you like the lower bound, right? And then the unmitigated gives you the upper bound. And then we said, “Okay. Could we do the middle-of-the-road solution?” which was more like San Francisco begin to shut down. Could be used as San Francisco approach to like the rest of the state, and there wasn’t a lot of data there. So the epidemiologist kind of modeled this, like that Kansas City from 1918. And they did a really good stuff.

And what had ended up being is like this kind of project where I think someone described it as the epidemiology Olympics where you’re kind of doing like basic epidemiology, and we were doing basic distributed systems. But together, the amalgamation of that was able to develop like a pretty sophisticated model of like how ventilators, ICU capacity, infections, deaths, etc. And we actually device a report that ended up helping convince the governor to issue like the first shelter-in-place order in the country.

And then there is a recent study that came out from Berkeley in nature I think maybe two months ago or so that said that this kind of early shelter-in-place saved about 1.7 million infections by early April. So that was like amazing that like we were able to help in providing a relatively, not sophisticated, but like just kind of straightforward computer science, straightforward distributed systems work. But the joining of all these kind of groups of people together to like create something really valuable was like an immense and amazing experience that we’re able to kind of have that kind of outcome.

[00:12:23] JM: Tell me about some of the challenges you faced in building the model.

[00:12:29] JW: I mean, I think like Sam said, the onboarding experience was pretty intense. And I think we've all had the experience of you’re the new company, you’re doing startup, and job one, obviously, is to get the product built locally and like up and running and stuff like that, right? And especially when you're the first, like you’re early engineer, you’re the first to show up. A lot of the work around like how to get this thing running? What dependencies you need in place? How does the build system work? How does the tooling work? Is often tribal knowledge and it’s not documented. I mean, that was very much the case here. So our first job, like first and foremost, was just to get to a point where we could run things and sort of suss out all these implicit dependencies that Python and R could – That the epidemiologists were used to run these models. So they had built these models a long time ago.

And like Sam said, it's really like epidemiology 101 gets science. But things - is it reproducible? Was like the first immediate challenge. We spent most of our initial time on. The second thing funny enough was just getting resources along with like getting dependencies in place and figuring out how to run the code reproducibility. We were like frantically calling friends at AWS to get like our resource limits lifted on our accounts. So we can like ever larger machines with more and more cores and stuff like that. It could run more scenarios.

I think what else was really hard? And then I think just the inevitable, like when we came in and [inaudible 00:13:58] running these models, we created like the first like DevOps split this team had encountered. To that point, the epidemiologist ran the models themselves. They ran them on the same hardware where they developed the models. And so they would do like a simple simulation or a simple test run with like a few dozen simulations.

For context here, when we run these models, we don't just run like the one scenario. We run thousands of scenarios, because we want to see all these different possibilities and perturbations of how things could go so we can have a kind of a fully informed perspective on the epidemic. And one thing we found, sometimes when would run models in the “production environment in AWS”, instead of running like, we went from running for like 10 of scenarios locally on a developer machine to thousands of scenarios, we would uncover like sort of subtle little bugs, like things that didn't quite work in production the way they worked in dev, but you’d have a fire drill trying to figure out a change. What was the source of the problem? All that kind of good classic release engineering stuff we were more or less inventing on the fly in the course of a pandemic. I think those are like three of my favorites just couple of days, yeah.

[00:15:10] JM: What kind of interface did you want to provide to scientists that wanted to query the information of the model?

[00:15:18] JW: So we really did – The original kind of way the model was run was we would generate all these thousands of simulations, and then there was separate code. Some are markdown code in particular that would take the output of these simulations, aggregate them and construct a report. That's actually in our COVID scenario pipeline repository in there.

Amusingly, we have the problem of – The output of these simulations would be – I mean, Sam, do you remember? IT was like hundreds of gigabytes of data. We would generate an enormous amount of data doing this, right? Yeah. But the initial challenge was just how do we move this data back from the cloud to the local environment so they can run their reporting on it? And the honest answer was really just the combination of good old pbzip2 and like SCP more or less, right? That was the easy sort of stuff.

We did some more interesting stuff later involving like running larger simulations, sort of truth forecasting tools involving thousands of machines and AWS batch. And that was a different kind of interface challenge for the epidemiologists to figure how they could run that stuff themselves. From a reporting perspective, yeah, the was the initial challenge at lease right away.

[00:16:29] SS: And I like to add that I think that the kind of ethos that we had was we didn’t want like add a bunch of process and like change the way that epidemiologists work, because they obviously needed to be able to do things with a very, very tight timeline. So really what it was is kind of building a rapport around the ways that they kind of already worked and just scaling that up as necessary. So you could write your thing in R, Python, and all you would do is have a way to scale that out across a bunch of machines and you wouldn't necessarily – You could run like a small set of simulations on your local machine to see whether it worked. And then we could just kick it off on the cloud and like run it in a massively parallel way to do much more sophisticated things and across much more scenarios and across much more simulations and do a true like kind of MCMC type approach. And that was kind of the way that we could bridge the gap between people that didn't necessarily understand or even had the time to like learn how to do distributed systems work. And to kind of get them to be productive and kind of create a lot more value there.

[00:17:35] JW: Yeah. It was more like they were conceptually familiar with the idea of running things on thousands of machines. They understood we’re going to take what we’re doing on one machine and we’re going to do it like a thousand times simultaneously on different machines. It was more just like abstracting away the mechanics, the plumbing of how exactly that works so they didn't have to worry about it.

[00:17:54] SS: Exactly.

[00:17:55] JM: So just to be clear, these scientists are making actual decisions based on the model that you help them refurbish.

[00:18:02] JW: Yeah. I mean, yes, very much so. I mean, this is their model. It’s a model they built. Like we didn't help them – We didn’t change it. First of all to be clear, for the first couple of days, I didn't even look at the code. And I wasn't going to change any of their code, even in situations when they asked me to go like change on, like, performance critical part of the code just because I didn't really understand it. And it was not my focus. And I didn't want to do anything that would change their understanding of how things worked. It was really a very pure lift and shift kind of thing.

That changed over time. Like over time they would say, “Hey, we have this piece of the code and it's really slow. We don't know why. Can you help us figure out why?” Can you like help figure out what’s going on and see if there's a way to optimize it?” And for that stuff, it was generally speaking – Like, Sam, like pretty straightforward software performance profiling stuff we used a lot of [inaudible 00:18:53] tooling. Sam, [inaudible 00:18:56] Python performance tooling.

[00:18:59] SS: C profile, I think for most of the things.

[00:19:02] JW: Yeah. Just going through and like identifying hotspots, and once we found them, like talking about what they were doing and how they could do it faster and that kind of thing. But yeah, it’s very much their model. It's was not our model. We are not epidemiologists. And I guess I think in particular for me, since I was working – We were working with actual experts in the field, I never felt like any obligation to pretend that I was an expert in epidemiology or even like understood this stuff. I was just there to provide whatever software and cloud support they needed to get their job done. That was my only role. Yeah.

[00:19:37] SS: Yeah. I would say that there may be like three phases of involvement that we kind of had which was the first one was like kind of panic at the disco, which is like the first part of mid-March, they’re like March 16th through the 20th, which was like a kind of a crisis, rescue kind of effort. Let's build a model of the epidemic to help understand for California what's going on right now. There's a bunch of all-nighters that were there. The second kind of phase of involvement after those initial 48 hours or a couple days were how do we take a model and scale it and make it iterate quickly, right? Some of the model that took like six or eight hours for single state, or 45 days for the entire country, how do you get it down to running in minutes, right? And that was basically doing a lot of kind of elbow grease and basically optimization. A lot of profiling, understanding hotspots, understanding memory usage, figuring out the resources that were being constrained, and then working around it, right? And I think there are a bunch of kind of core principles. We can talk about like how we kind of got there.

And then the third piece was kind of taking the cloud out for a spin, where kind of expand what's possible in epidemiology by doing state-of-the-art stuff. We could run and we got to do a lift and shift in the cloud. Can we run a massive stimulation across thousands of machines? And what we ended up doing together was kind of building out like an MCMC type approach using Metropolis Hastings to do that and be able to do much more sophisticated modeling and stimulate kind the epidemic curve and just to set context, like that piece would require – The number I have written down here is 100 million computer hours and may select. It's a pretty massive kind of effort for that, that third piece. And each one of those kinds of pieces had like different kind of modes of operating and different kind of software challenges an in each kind of those three pieces.

[00:21:33] JW: It was funny. We were working with some great folks from AWS. AWS was absolutely wonderful, by the way. Money, resources, in particular. Pierre [inaudible 00:21:41] and Greg Thurman, sorry, Greg [inaudible 00:21:46], were just absolutely. They gave us like our own solution architects and stuff to help out with the stuff. It was fantastic. And what's funny, it was like after we're kind of finishing what Sam described as phase 2, we were optimizing the single node runs. We kept challenging the folk at Hopkins like what would you do if you could have a thousand computers doing this stuff? And that was really where they came up with the MCMC approach, which is really much more closer to a true like forecasting scenario, which is really what everyone wants. And then yeah, then we're on the hook to kind of build it with them, because we said it could be done. And they were like, "Okay. Well, prove it." And we had to go do it. It was great.

[00:22:29] JM: How does this project compare to other data engineering projects you've worked on in the past?

[00:22:32] JW: That's a great question. Sam, why don't you take that one first while I think about that?

[00:22:35] SS: I mean, the stakes were much higher, for sure, right?

[00:22:38] JW: Yeah.

[00:22:39] SS: The different story of the site is down and that they, hey, people are going to die. So there's certainly a much deeper sense of stakes. I think the other one was different kind of cultures of working, right? So you have a completely different styles of working with the public health officials to epidemiologists to like Silicon Valley engineers, right? And like how do you weave those cultures together so that you can kind of work together in a way that you've never worked together before and that hasn't worked before? And so that was I think like an interesting challenging around kind of building trust. And then just understanding that people have different ways of working, different styles, and they required a lot bridging of the gap from everyone involved to like learn a little bit about the other person's field and the way that they kind of – Their style of working and the kind of things that they developed on and kind of being able to connect all those pieces together.

[00:23:37] JW: Yeah, absolutely. I mean, this is by far like the most fun I've ever had. By far, the most rewarding working experience of my life. I mean, it was fantastic. Stakes, as Sam said, the mission, the people, the tools. Yeah, it was absolutely a privilege to get to do this. I really enjoyed it. Yeah, more so than any job I've ever had. Yeah.

[00:23:59] JM: How big was the overall developer team working on the project?

[00:24:03] JW: Yeah, it changed overtime. It involved kind of as things went. At first, it was Sam and myself and the folks at Hopkins. There's a developer there named Joshua Kaminsky who was kind of the overall like architect person, but the PI for the group, Justin Lessler, like still writes code and still writes code that's pretty – Like writes like good code, right? We'll see, Elizabeth Lee, another one of the PIs in the department. There were different epidemiologists who were handling runs for different states. Elizabeth was really in charge of California. So that was like 5 people. And then let's see, overtime, we expanded it with a couple of few folks from AWS, Pierre and Greg I mentioned. Karthik Raman was fantastic for performance profiling. And then Josh's wife, Catherine, is also a software developer and she joined the project when we were doing the large scale cloud batch jobs and stuff like that.

So really like fewer than 10 people sort of altogether doing most of the heavy lifting. But there were other efforts as well, folks working on dashboard and integrating lots of different predictions together. So I think the whole sort of – Obviously, first thing you do when you're setting up a new project, you create a Slack team. So we created a Slack team. I think at our peak there were something like 80 people in there in various capacities, yeah.

[00:25:24] JM: Let's talk more about the data itself. So where was the data coming from and what data are we talking about?

[00:25:32] SS: Yeah, the data is there are a couple of different pieces of data, right? One which is the historical and faction counts, that was coming in from the Johns Hopkins pipeline. That was developed. There was also a bunch of data around hospital and ICU capacity, and that was actually really challenging, because there is no real database of what that actually is in the state of California.

So what ended up happening was people were just calling various hospitals then saying, "Okay, how much ICU capacity do you have? How many beds do you have? What's your search capacity? Etc.," and just kind of writing that stuff down and putting it into a spreadsheet and just doing that continually to basically build that dataset. There are other datasets around understanding mobility patterns of like how people were moving, and that kind of understands how you can see seeding and spread of the epidemic across county borders and things like that, right?

So there's definitely a lot of different pieces of data, census data around like how many people are around, things like that. There's definitely like a large data integration problem of numerous datasets all over the place to kind of bring this stuff to bear.

[00:26:50] JW: Yeah. You have to imagine, Jeff, that with these infections, when these pandemics start they start at the ports, right? The people landed airports. And then the question is, statistically speaking, where airports travel to after that? And then the next question is like, “Okay, where do people who live in certain counties, what is the commuting relationship between different counties?” How is the infection going to spread from county A to county B based on historical commuting patterns and all the kind of good stuff?

And then for the hospitalization side, there’s options around obviously how rapidly does the spread? How many people who get infected are going to get sick enough to be hospitalized? How many people who get hospitalized are need to going to be in an ICU? These kinds of things. Epidemiologists are working to figure out as best they can with limited information in real-time based on what people had seen in Wuhan, Italy and that kind of thing.

The importance I think of the kind of modeling we were doing was we were running all these different simulations, because there was just so much stuff that was unknown, that we know relatively little of course about coronavirus now than a lot more than we did back in March. And so understanding like the spectrum of possibilities from, like, outrageously bad to the relatively benign is absolutely important in like making decisions and understanding because the data isn’t perfect. Because our knowledge isn’t perfect. We have to run a lot of simulations to have a real understanding of what the possibilities are.

[00:28:17] JM: How reliable is the data?

[00:28:19] JW: I mean, it was generally speaking good data. I mean, the census makes good data. The nice thing about like sort of case counts from the – I mean, I would say like the weak link obviously was like the case counts themselves, like Hopkins, New York Times, the COVID tracking project. Everyone is making their best effort to collect this data as fast as they can. But of course, mistakes are going to get made. And of courser because there are limitations around testing and stuff like that, you’re getting a very heavily censored sample. It was only the sickest people could even get a test, right? And so you have to compensate for those limitations in your modeling.

I mean, generally speaking, given the cadence we were operating under of doing like a couple of these runs a week for various states and for the country and stuff like that, it was generally fine. We were generally in pretty good shape, yeah.

[00:29:10] JM: Tell me more about the architecture of the data pipeline.

[00:29:14] JW: So really the interesting pipeline to work on was doing like the large scale, the large scale batch pipeline by far. So the way the algorithm worked was we're basically going to create a situation where we would generate parameters for our model, create a projection like a sort of forecast basically based on those parameters to what the future would look like. Compare the projection to what actually happened, what we knew actually happened over the sort of several weeks. And then update the parameters based on the error in those projections and run things again. We're going to do this thousands with sort of different divisions across thousands of machines.

I think one sort of principle we had for this project was that as much as possible we did not want to be in the business of running servers or running clusters or anything like that. And so working with the folks at AWS, we settled pretty quickly on using AWS batch to do these simulations to these runs. In order to kind of keep cost down given like the millions and millions of compute hours we were doing, we were using spot optimized instances. And so because as you all know, AWS spot, the machine can disappear basically any timeout from underneath you. So we wanted to do runs with like pretty frequent check pointing so that we would be able to restart if we like lost a machine or for whatever reason.

And so the way things would work is we would kickoff these enormous, what are called array jobs in AWS batch. So an array job is basically a hundred copies of a job that are virtually identical to each other. They just have one environment variable that indicates which job in the array this is; job 1, job 2, job 3, so on and so forth. And then we would chain these array jobs together. So we'd have a sequence of like – We wanted to do, say, I don't know, a hundred simulation, or a hundred runs of the MCMC algorithm. We would break it up into 10 runs of ten simulations a piece. And then those 10 runs would happen. It would write to S3. The next sequence would then start itself off. It would know where to look basically based on its job identifier for the output of the previous run. And then it would kick itself off and then go do another 10 runs and so on and so forth.

And then we had some code that would basically gather all these outputs from S3 and restructure them into a way the epidemiologists were expecting to work with from a point purposes and then notify everybody when they were done. So it was just my first time using AWS batch in kind of a professional capacity. I was not familiar with it before, and I generally learned – I think this is always the case, like a lot about the ins and outs of how batch worked. It's built on top of ECS, the Elastic Container Service. And it's funny, it's like we're doing these batch jobs, and generally speaking when you're doing a batch job, you're optimizing for throughput, not latency. You want like the whole job to finish, but you don't really care when an individual job finishes.

But we occasionally had like deadlines where some very important person was waiting on the output of these runs. And so we had to do various things. We kind of had to learn how to sort of goose batch, I don't know how to describe it. Basically wanted to, I guess. We would force it to provision more machines. I think like another one of our principles on working on this project along with like never run a server is throw money at problems wherever possible. Don't worry about like optimizing the code. Don't spend an hour optimizing the code. Just like fire up a bigger instance or whatever. So we made liberal use of Amazon's money to run things as we could without necessarily being super concerned about efficiency in a number of cases.

But yeah, it's large scale pipeline stuff. Weird things happen. Weird stuff fails. S3 is fantastic, but sometimes the data doesn't show up exactly when you expect it to. So you have to have a little like hacks and place for that. But yeah, and the sense that it was very much a classic massively parallel data pipelining problem. Like I said, it was great fun. Yeah, to be able to use the skills that we had develop in other places at LinkedIn, at Google, Cloudera, in our Slack and all that kind of stuff in service of science during a pandemic was amazing.

[00:33:33] SS: And the thing I would add to it, Josh, I think is that this was obviously a massive data pipeline, but we're not just running it for California at that point. We were running it for numerous other states and other countries and other kind of partners. And so we also need to make sure that like each kind of region or partner was going to separate and being able to manage across the different types of data that was used for each different partner in an effective and meaningful way. That's something that was actually a very, very challenge at the beginning of the project where just like managing all the different datasets and the different data across different kind of states and partners. Basically in a much more streamlined system where you could kind of have one region run kind of end-to-end in a kind of contained fashion. So we made sure there was no data leakage, which should cause problems and things like that.

[00:34:20] JW: Yeah. Making it so that it was easy to plug in new data sources as we brought on new partners in new states.

[00:34:28] SS: Yeah, exactly. We could being on new states on.

[00:34:29] JW: Huge quality of lifeline. Yeah. And we did them all in sort of the early part of phase 2 to make that possible, and that was great.

[00:34:37] SS: Yeah. Just to add a little context. I think the beginning part of – You have a lot of values that are hardcoded and things like that to code and like there's a lot of assumptions about the data, and that moved to a much more configuration-based system where everything is like basically a giant YAML file for every kind of partner or thing that you want to run. And then each YAML file, the different scenarios. Like a YAML or one of these configuration files is self-contained. I'll give the system that configuration file and it just goes and takes everything, slurps up all the data and runs in. So an epidemiologist, all they have to do is generate the configuration file. And if they generate the configuration file, then they can just run it massively at scale and they could pull in all the data that they need and they actually don't have to write any code in almost a lot of the basic parts of the simulation, right?

[00:35:23] JW: That's right. Along with that, before we kicked off one of these gigantic pipelines, I think something in the [inaudible 00:35:29] data person would be familiar with is you're getting to kick-off some massive long-running jobs. It's going to spend thousands of dollars of compute, and there's a tiny bug, or the configuration file is not formatted or something like that, right? So as part of the kick-off process, we do have a step where we actually run locally like a single simulation basically on the kind of machine to make sure that the thing basically worked and everything is going to be reasonable and sane and all that kind of good stuff before we kick-off the giant job, yeah.

[00:36:02] JM: So did the model get slower every day as more and more data was being added?

[00:36:09] JW: I'm sure that it did. I think that was sort of so minor relative to the performance optimization to the code. Everything from like switching out – We switched out the CSV files that we used for passing data, the different modules to use Feather, which is the file format behind Apache Arrow. Obviously, since we're using kind of this hybrid system where there's like a lot of Python code and a lot of R code all kind of like mixed together, that was a massive performance win. So I just think like the – It was kind of a round-off error compared to what we were doing to optimize things fortunately.

[00:36:44] SS: Yeah, just another context, like it would take about 2 minutes for a decent run for a single state, and 20 minutes for like the entire country after optimization. So then even if you're like double it from like 2 minutes to 4 minutes, or 20 minutes to 40 minutes, like in the span of things, it doesn't really matter.

[00:37:04] JW: Yeah. We were running for like hours before back in March when we got started –

[00:37:09] SS: Yeah, hours or days.

[00:37:10] JW: Yeah.

[00:37:11] JM: Tell me about the hardest parts of building this pipeline.

[00:37:15] JW: Oh, let's see.

[00:37:17] SS: I would say I think the hardest part for me was like almost like you had to – Everyone also had to be their own kind of developer epidemiologist, but they also had to be their own kind of PM and like understand what needed to be done next and like how things would connect together, because everything was just so fast-moving and we'd have new data – New data, but also like new insights and new things that people would want us to do coming all the time. And we'd have deadlines that would show up like it would be, "Hey, guess what? We need this like 5:00," and it's already 1:00.

So you had to understand and know what you need to do next, but then you also know when to stop optimizing. You need to know like how much is it optimized enough? And in the context of like all these different kind of functions of epidemiology to public health and like understand kind of what needed to be done and be able to like deliver something of value by the deadline, and I think that was for me the most challenging aspect of just like basically everyone had to be their PM, right?

[00:38:23] JW: I think Sam is right, and I think it's funny. When I think of like especially that first week, we were working crazy hours just totally unsustainably. And I feel like we fell into a lot of the classic anti-patterns of like DevOps work for lack of like structure and time – In fact, this was not a company. This was not any one organization. It was a collaboration of a bunch of different people. And so, yeah, we didn't really have that product management capacity. We didn't have time to invest in like good tooling and good processes especially right away.

And so while it was exhilarating, it was absolutely exhausting like early on to do that work. To be fair, I think Sam and I got like the best end of this. I mean, the epidemiologists have been working on no sleep, as you can imagine, for weeks before we even got there. We were relief pitching in some sense from them just so they didn't have to run some of these stuff themselves.

But yeah, I guess the part of it is always kind of – When I think of lessons learned from this that's always kind of strange to me is like without that overarching structure of a company or sort of a single organization, it's hard to correct a lot of like the kind of bad habits and anti-patterns that you know you're not supposed to do, but you do anyway, because it's expedient. And that's like sort of like everything that is good and everything that's bad about this experience were all kind of wrapped up together in that, yeah.

[00:39:54] SS: Yeah. If you work at a company like Google or Facebook or any Silicon Valley company, or any tech company, you know how to build a product. There is like a formula in some sense. I mean, you know how to kind of get it done. In this case, there was no formula followed, because it hadn't been done before, right? How do you create like a public-private partnership to be able to kind of go and deliver a product? Delivery results in a way which is completely foreign to like everybody involved, right? Eventually got a process that was there, but that was a lot of kind of trial and error and two steps back – Two steps forward, one step back.

[00:40:36] JM: All right, a semi-political question.

[00:40:41] JW: We got to be careful with these, Jeff. We got to be careful.

[00:40:43] JM: Is the lockdown actually necessary? What are we doing here?

[00:40:49] JW: Well, I mean, we're not in lockdown anymore.

[00:40:53] SS: And we're happy to talk about software. I think like talking about like epidemiology and public health I think is like outside kind of our scope of expertise. I'm happy to double click anything about the software and like the team and things like that. But like I'm just like a software engineer giving my opinion, and I don't think that would be – Add a ton of value at this point.

[00:41:14] JW: Yeah, I got to agree. We're super not experts here, and I don't want to pretend like we are, yeah.

[00:41:20] JM: Not even some speculation? Come on. You guys are unemployed.

[00:41:25] SS: I think the challenge also in these kind of projects and working on it for a few months was like ignoring all the armchair epidemiologists that were on Twitter and that were on your inbox that were telling you things. And so like I think like a big part of it is knowing what you know and being able to be effective is to be able to like bring your A game on things that you know and really try to deeply to understand like kind of what's going on, but like I think you have to understand also like where the boundaries of your knowledge are, because I think – Otherwise, it causes a lot of problems and a lot of unnecessary churn. And you can see that all over the place.

[00:42:06] JW: I mean, it's funny. I think in Silicon Valley we have this kind of love-hate relationship with experts, right? On classic, and we're very smart and we think figure anything out. I personally – And I think like I am unemployed, but I would like to employed at some point in the future. I have a reverence for experts. I listen. I like science. I like scientists. Have their own set of incentives? Well, of course they do. Everybody. But I think one of the reasons that CISCO and the Bay Area have done so well with the pandemic is that we take science very seriously here. We believe scientists when – I am not a scientist, but I do listen to them and I like follow their advice. Even if they're not perfect, because no one is, what they think of as the best thing to do information they have.

[00:42:55] JM: What norms have changed most for you guys after the lockup, lockdown?

[00:43:00] JW: I think like everybody – Not everybody else, but like a lot of other people, I was doing this work early on from home, from like our tiny little apartment in San Francisco with my wife and my 4-year-old son, and that was obviously very challenging as it would be for every other parent like during this lockdown and stuff like that. So my all heart and my empathy and everything like that goes out to every other parent out there. It's rough.

I was very, very fortunate to be able to do it under the best possible really of – I spent a lot of time thinking about my mask. There was a great tweet the other day, which is like I go outside and I forget my mask and I'm like, "Oh! I forgot my mask," like I'm Spider-Man or something like that. That's how I kind of feel in some ways. I feel like Spider-Man like, "Oh! Better not forget my mask." I don't know. Is it weird that I've kind of like grown accustomed to wearing masks now and I kind of like it? I don't know what that's about. I maybe the only person that feels that way, but that's the strangest thing for me. Yeah, Sam?

[00:44:00] SS: I'm someone who likes to meet people. So I think given the fact that serendipity is kind of gone to zero is a little challenging. And so you have to figure out ways to kind of go create serendipity, and there aren't really great avenues to go do that. So that's something that I kind of like struggle with. So I think you just need to like kind of be cognizant about that just make sure you're doing it for you. And if you're meeting people virtually, to meet new people and like kind of exercise those, especially for me, like who really enjoys that. Just learning from other people, right? Figuring out venues and avenues to go do that.

[00:44:34] JW: I'm an introvert, and so I realize a lot of the stuff has not been as hard for me as it is for other people. But it's funny, I did get together last week for a socially distance walk with a friend of mine, and I was like so excited to see – I was so excited to see him. Your old college friend or whatever. This is just like an acquaintance basically. This is not a – I'm like so absolutely thrilled. Thinking you want to give him a hug and all that kind of good stuff. Yeah, that.

[00:44:59] JM: All right you guys, anything else you want to add?

[00:45:03] JW: I don't know what to say. It was great. I love doing it. It was an absolute privilege. I would recommend it anyone. If you ever have the chance to use like things you know how to do, like use your special talents in service to believe in. It's just an absolutely rewarding, tremendous experience. I just cannot recommend it highly enough, yeah.

[00:45:24] SS: Yeah, for me I think it was, is that being able to bring my skills and have the opportunity to kind of make a meaningful impact, and to something that is kind of outside of my domain to provide that kind of expertise and really help move the needle was immense and an incredibly rewarding experience. I learned a lot from how to integrate across different kind of modalities to like even learning a little bit epidemiology. So it was a blast and I'm really glad to have the opportunity to do it, and I highly recommend people if you have the chance to go try and volunteer for some of these projects, because fundamentally, like they need software people to show them what's possible. And you'd be amazed that the kind of impact and value that you can add, just telling them what you see and things that they could do that they can't do right now with computers. We're still really much at the infancy of what's possible across many, many different domains.

[00:46:24] JM: Okay, guys. Well, thanks for coming on the show.

[00:46:27] SS: Thanks, Jeff.

[00:46:27] JW: Jeff, thank you so much.

[END]