EPISODE 55

[INTRODUCTION]

[0:00:10.6] SC: Hello and welcome to another episode of TWIML Talk, the podcast where I interview interesting people doing interesting things in machine learning and artificial intelligence. I'm your host, Sam Charrington.

The show you're about to hear is part of a series of shows recorded in san Francisco at the Artificial Intelligence Conference which was hosted by our friends at O'Riley and Intel Nervana. In addition to their support for the vent itself, Intel Nervana is also our sponsor for this series of podcasts from the event. A huge thanks to them for their continued support of this show. Make sure you check out my interview with Naveen Rao, VP and GM of Intel's AI products group and Scott Apeland, director of Intel's developer network, which you can find at twimlai.com/talk/51.

At the AI conference, intel Nervana announced a dev cloud; a cloud hosted hardware and software platform for learning, sandboxing and accelerating the development of AI solutions. The dev cloud will be available to 200,000 developers, researchers, academics and startups via the Intel Nervana AI Academy this month. For more information on the dev cloud or the AI Academy, visit intelnervana.com/devcloud.

In this episode, I talk with Ian Stoica, professor of computer science and director of the RISELab at UC Berkley. Ion joined us after he gave his talk, building reinforcement learning applications with Ray. We dive in to Ray, a new distributed computing platform for RL as well as RL generally along with some of the other projects that RISELab is working on like Clipper Integra. This was a pretty interesting talk, enjoy.

[INTERVIEW]

[0:02:01.0] SC: All right everyone, I'm here at the AI Conference, presented by O'Riley and Intel Nervana and I've got the pleasure of being seated with Ion Stoica. Ion is the Executive Chairman and Cofounder of Databricks and also a professor at Berkley and director of the RISELab.

Ion, welcome to This Week in Machine Learning and AI.

[0:02:21.6] IS: Thank you, thanks for having me.

[0:02:23.9] SC: Absolutely. Why don't we get started by having you tell us a little bit about your background, how you got interested in machine learning and, you know, what your area of research interest is?

[0:02:35.4] IS: I joined Berkley in 2001 and at that time I was actually doing networking and then I moved around and did a little bit of peer-to-peer systems and then start working on big data. On big data in the context of another lab at Berkley called Amp Lab, we developed a few systems including Apache Spark, Apache Mesos and Alluxio, formerly known as Tachyon.

Basically what happens is that we develop these systems because people get more and more data but organizations in just more and more data and then when they have the data, was that it's user logs, machine logs or any other data, they try to make sense of this data, they get and try to gather some insights.

Apache Spark is an effort to process a large scale this kind of data. Now, also in the context of Apache Spark, you look at and what people do after they get some insights of the data, they want to make some decisions to take some actions based on the data, right? To improve our targeting to do medical diagnosis and things like that.

Now, the next step again in getting value out of the data, it's basically applying AI, machine learning to get intelligent, actionable decisions.

[0:04:05.3] SC: Right.

[0:04:07.0] IS: That's why I'm now doing that research in the context of systems and machine learning and AI. As part of that, we just started this new lab at Berkley called Rise where Rise stands for Real-Time Intelligence Secure Execution and this lab is about building systems and tools and algorithms to support applications which make a real-time decision on live data.

[0:04:36.3] SC: Okay.

[0:04:37.8] IS: With strong security.

[0:04:38.4] SC: okay. In fact, that's one of the areas that Spark really distinguish itself and that real-time streaming interactive decision making at least relative to Hadoop, which was kind of the incumbent system at the time.

[0:04:53.4] IS: Yeah, definitely. That's absolutely true. So Spark was compared with, at that time when we created it, compared with map reviews, provide great support for interactive queries, iterative computation like machine learning algorithms and later for streaming.

Right now, going forward, one big angle we are focusing on is the security. You know, everyone wants to take and to make personalized decisions. So the question here, how can you make personal decisions without violating the confidentiality of the users while preserving the confidentiality of the users. This is one of our big challenge.

The other aspect it's about, for instance, when you create this with Spark, you create these models using deep learning or many other approaches, machine learning approaches but this model eventually you are going to serve that model, right? So the users, you know, when the users contact you and the user makes a query, you look at the recommendation, you give back the –

[0:06:14.1] SC: So the inference part?

[0:06:15.8] IS: The inference part. They give back right answer. As a direction is focusing on building this kind of prediction serving or model serving systems.

[0:06:27.0] SC: Okay.

[0:06:29.0] IS: Another direction is that, you know, when you look at the new generation of this machine learning applications like for around reinforcement learning, they have different patterns, they do a lot of very small simulations and, you know, we build also systems for that, which are a better feat for this kind of very fine-grain computations.

[0:06:51.8] SC: Okay. Here at the conference, you were talking about Ray, which is one of the systems that you built for this particular thing.

[0:06:58.2] IS: I was talking about Ray, which is in this category. Think about Ray is the ability to execute a lot of fine grain tasks, which are dynamically connected, dynamically depend from each other. Like tasks of milliseconds takes only milliseconds you can run millions of this task per second.

It's very fine grain. Each of this task can be simulation like simulations you play a game, right? You move — Simulator moves all the pieces and then you see whether you won or you lost. One of the workloads is reinforcement learning, which reinforcement learning, you can see something like a generalization or supervised learning in which the labels of the action you make is sparse; sparse, not every action has a label as good or bad.

Like for instance when you play a game, when you make a move, you don't necessarily know whether this move is great. You need to wait until the end of the game, maybe to see whether it's good or bad. Also, the feedback is delayed in supervised learning in many times you now whether I show you some picture, you give me say this is a cat or a dog, I know instantly whether it's good or bad.

Also, it's about reinforcement learning, it's also assumes that you interact with the environment continually and you change the environment, right? Again, with playing games, self-driving car or dialogue systems. This is a dialogue, right? You need to — it's continuously interacting and building the context about our conversations and base your next actions, you know, next answers and questions based on this context, right? That's basically it is. This reinforcement learning, many one or the work load characteristics, so it requires to handle this very small fine grain simulations.

[0:08:59.5] SC: Okay, so we've actually talked about reinforcement learning on the podcast quite a bit. We've had Pieter Abbeel and Sergey Levine.

[0:09:08.3] IS: They are the expert. I'm not sure how they cannot do that.

[0:09:12.2] SC: You know, the question that comes up for me is, how does Ray compare to an environment like OpenAI Gym or OpenAI Universe that are use also in context of reinforcement learning?

[0:09:26.0] IS: It's very complimentary. The OpenAI Gym and they provide you virtual rewards and so forth. The simulators. The Ray, which provides you with the plumbing to execute those simulations and to get the results and to update the policies that is the mapping between the state of the environment or the observation of the environment and the action you are going to take at a large scale.

[0:09:53.9] SC: Is it too simple to say that Ray is like Hadoop for reinforcement learning? It sounds like you've got a bunch of data, amount to your cluster and mapping the results back?

[0:10:04.6] IS: Yeah, you can say that in the first approximation and then you take the results and new updates and policy.

[0:10:09.8] SC: Right.

[0:10:10.9] IS: Then you repeat until the policy — you are happy with the policy, the policy converges.

[0:10:15.4] SC: Right. You mentioned this attribution problem, right? You're playing this game, you've got this reinforced learner, that's involved in this environment, a game for example and you can't really know whether it's action, you know, is positive or negative and so much later on, if you don't know the ultimate benefit, how then do you gain any knowledge form these micro simulations that you've been describing?

[0:10:43.3] IS: Because a simulation, again, you take some actions. After each action, you have a state, which is a modified state of the environment and eventually you have a reward. Which sometimes you do not know whether the reward can be zero, you don't know is it good or bad right?

After you take, according to the current policy, you are going which again you feel the state, take the action, get the next state and the reward, you feed again the policy, you get the next action and so forth. You get these trajectories, which is a succession of states and rewards.

Now, eventually, at the end, you are going to see whether you achieve your task or not. Whether you won the game or you lost the game. If it's a robot, whether the task say or moving an object was successful or not, right? Or solving a puzzle, right? Then once at the end, you for instance see that eventually see whether you're successful or not, then if you are successful, there's a very high level, you are going to go back and reinforce all these actions you've taken.

[0:11:49.1] SC: Right.

[0:11:51.0] IS: Meaning, reinforcing, meaning that next time when you try, after you update the policy, it's a higher chance to take this action. If the actions led you to say losing a game, then you are going to weaken this actions, you make them less likely to be taken when you are in a similar situation.

That's basically what it is, it's like humans, right? It's like you play a game and so forth and if you won, you are – it's likely that you are going to repeat some of the kind of moves you made.

[0:12:21.9] SC: If I'm playing tic tac toe, the opening move is always in the middle.

[0:12:30.6] IS: That's very true.

[0:12:32.9] SC: I imagine a big part of what it's, or at least one part of what it's trying to do is, and this is also in a lot of ways Hadoop-like is like managing the state because you're shuffling like state for all of these little simulations that you're running, all of them?

[0:12:51.0] IS: You manage the state as well, the recent object stores. So you keep that state in the object store so tasks which run on different nodes can have access to the state. You also try to provide, you know, we provide fault all around if they node fails so you make sure that the computation is still is going to make progress and doing that correctly.

Yeah, so these are some of the – you're right, completely right, these are some of the issues you need to deal with.

[0:13:23.4] SC: Okay. You were presenting on Ray specifically, maybe what were some of the main points that you wanted to leave with the attendees at your talk?

[0:13:34.1] IS: Yeah, the high level point is that right now, over the past 10 years, there has been a tremendous progress in AI, right? Many great applications and right now as we look forward, the applications becomes more and more sophisticated and while most of the applications we developed so far were based on supervised or unsupervised learning. The new applications, we believe are going to be more and more are based on reinforcement learning and again, the reason is that, the new applications are more of about interacting with the environment around them, learning from this interaction and affecting the environment. You know, we talk about games, we talk about dialogue systems but you can think about the health applications, they constantly monitor you and then give you health advices.

[0:14:29.3] SC: How about the next level of detail though? Did you talk about the experience of using Ray or the architecture?

[0:14:35.5] IS: Yeah. Basically that's where I started but it's again, like you said, you know, for instance, if you do not have a feedback after each action, after each move, that means the space, you need to explore in terms of solution space, it's larger because I need to make a series of moves because I know it's good or bad, right? More possibilities to go wrong, right?

That's one of the characteristics, the computation requirements are increasing. That's why one way, if you cannot afford simulations, that's a very good way to do because you can do the simulation fast. Now, like for instance playing games or things like that, you do a lot of simulations, right? Even with robots, you'll try to simulate the physical world, right? Because you can learn much faster, right? That's the key.

The key is basically then the computation pattern is basically saying, “I have a policy, I’m going to evaluate the policy by running many simulations, right? And, I’m going to feed in their outputs, these trajectories, right? What happened, and update the policy and run again,” right? That’s kind of — it’s a pretty regular computation because different simulations they can dig different amounts of time. Like for instance, if you play a game of chess, I can lose in three moves or it might take 60 moves to win, right?

You want to learn as soon as some of these simulations happen, you want to learn from them and updated the policy. You don’t want to, you know, if I run 100,000 simulations to evaluate the policy, I don’t want to wait for all of them. This is kind of, a computation is more irregular. Also, the other thing is that, you know, many of these policies are implemented by neural network, neural networks, which run on GPU’s, so you want also to support more heterogenous hardware and also the kind of patterns you are talking only about the simulations. But then you want to search to be able to search the space, the solution space and things like that.

So, you end up basically with a very fine grain computation graph, you want to execute on heterogenous hardware. That’s what Ray is. How is Ray achieving that? It’s using this in memory object store to share the data. So it’s very fast on a single machine, we share memory. On the back end, the scheduling we totally distribute the scheduler. Each node can schedule locally and then when it’s overloaded or the inputs of the task are not available locally, it only then it’s going to contact a global scheduler.

We also replicate the global scheduler so we do a lot of this kind of things to try to scale and to have very high throughput. Finally, we also provide the bindings in Python. Why Python? Because you know, most of the AI community develops in Python, right? Now it seems to be the most popular language.

[0:17:46.5] SC: Whereas Spark is Java and Scala.

[0:17:49.1] IS: Park also, you know, has a binding — it was developed in Scala so obviously has API in Scala but also has in Java and Python and Python actually is very popular API and of course, SQL.

[0:18:04.1] SC: I guess early on, I had the impression that some of those early efforts with Python were kind of like second class citizens. Has it evolved to be more?

[0:18:11.4] IS: We are evolving, you know? I think that a very large percentage of users use Python, so they are improving fast.

[0:18:18.4] SC: Okay. Where is Ray on the maturity cycle?

[0:18:24.3] IS: You know, it's still an early project. We had a first release a few months ago, we are going to have soon the second release. But it's, you know, it's still early so we have just a few users. But, you know, we believe that going forward, this reinforcement learning will play a major role in the future of AI and I think that the usage adoption of Ray will grow as well.

[0:18:49.5] SC: Right. What size environments have you tested so far?

[0:18:54.1] IS: We tested this over hundreds of nodes. So we are at that level now but it's again, maturity is not only about how scalable it is. The robustness and the kind of features you need; monitoring and things like that you need really to deploy something like Ray in production.

[0:19:14.6] SC: Does Ray lend itself to being deployed in cloud environments or is it —

[0:19:20.2] IS: No, you can deploy equally like Spark and many others, you can deploy in the cloud or can deploy on-prem.

[0:19:27.3] SC: Okay, does it work with Spark in any particular way or?

[0:19:30.6] IS: Of course. What we use, I mention about this object store, actually for the object store we use Apache Arrow. It's easy for Apache Arrow to change the data between Spark and Ray. Ray has a much lower level API. It's again, it's targeted for this particular reinforcement learning and distribute AI applications.

While spark has a higher level and it's very more general and very mature API, you know? Very easy to use when it comes to processing big amounts of data.

[0:20:07.4] SC: Thinking about that API, what's the kind of fundamental unit of work or –

[0:20:13.3] IS: The unit of work is a task. You can add a decorator on a task and basically that task will be run remotely.

[0:20:19.8] SC: Okay. You know, getting from a task to a reinforcement learner is all of that in user code or does Ray provide any abstractions for reinforcement learning in particular?

[0:20:33.2] IS: For reinforcement learning right now, we are, in the next release, we are going to add a new library offerings for learning algorithms.

[0:20:41.9] SC: Okay.

[0:20:41.7] IS: Right? It will be a small library at the beginning but we are going to implement the most, it's going to offer the implementations of some of the most common reinforcement in the rings. And will take it from there.

[0:20:56.2] SC: Okay, you will be able to – and I've used that with the same decorator model, you decorate your task and then kind of tie in.

[0:21:05.0] IS: It will be sound function calls, right? It will be on media.

[0:21:07.9] SC: Okay, great. Based on your experience with Spark, how do you see Ray evolving?

[0:21:14.5] IS: See, now, it's very hard to predict that. I mean, when I started Spark we never thought it was going to become so popular and it depends on a lot of things. For us, for now, is that we want to build as with Ray, we want to build a platform which allows us to speed up the research and the application in reinforcement learning.

That's our first goal and it's again, we are talking about only us at Berkley but obviously across the board, you know? It's academia and the industry. That's kind of our goal.

[0:21:56.2] SC: Your focus is on making a tool that helps accelerate research and if, you know, the Spark-like success comes then the Spark-like success comes. But, you know, and you're focused on the user problem?

[0:22:07.6] IS: Yes.

[0:22:09.1] SC: Excellent, excellent. Are there any companies offering like commercial support for Ray?

[0:22:13.5] IS: No, not yet. It's still early.

[0:22:14.5] SC: It's too early for that.

[0:22:16.5] IS: There ere some companies that interested, they are playing with it. But yeah, like I mentioned, in order for someone to put in production, it still has a little bit to go.

[0:22:25.9] SC: Yeah, okay.

[0:22:26.8] IS: You know, we are moving fast, it will take a bit of time.

[0:22:31.8] SC: Yeah. So you mentioned some other projects around security and other things. Do the projects all kind of tie together? Like, can you think of like the RISELab is creating this platform and the security thing, talks to Ray, talks to the next thing? Or are they more or less different.

[0:22:47.9] IS: Those are a bunch of projects.

[0:22:49.0] SC: I was actually surprised like, you know, considering Rise is relatively new lab, there were huge number of projects on the website.

[0:22:56.0] IS: Yes, it started — we have quite a few people, quite a few students. But it did start only this year in January, officially at least. We have a few other project, actually there are some projects also like you mentioned, you know, in the Spark is also about real-time and of course streaming.

We have a few projects in the context of Spark. One is about accelerating and reducing the latency of streaming called Drizzle, and actually, there is a talk later today form Intel, which we collaborate with this from this platform, big deal. Which is experimenting with Drizzle and they are going to present some results today. Then there is another project called Tegra, it's about how to process time evolving graphs. You now that Spark also has graph X, which is a graph frames, which are some libraries to provide graph processing.

Tegra takes that at the next level and basically saying that most of these systems today are processing static graphs. You want to process the graphs as they change. Also, you want to be able to ask questions about past instances of the same graph. So what of your friends, you know, January 15 this year, right? How are these friends are changing over the four months period, right? Who are the new friends, who are the friends?

[0:24:28.7] SC: That sounds super hard.

[0:24:29.8] IS: Yeah, these are the kind of questions you want to answer. This is what the Tegra project is trying to answer. On the security side, I will say, it's one project called Opaque. Opaque, it actually is again, it's like taking Spark SQL and making more secure. What do I mean by that?

You know, today, there are solutions you encrypt the data addressed, maybe in motion. But still, they do not defend, for instance, if the operating system or ASR application are compromised. Opaque uses, you know, new developments of hardware enclaves, which now is used by many companies.

Also Apple as their more recent announcement for this chip, bionic chip, right? On the iPhone to run their neural network. These enclaves basically defend the application, the code you run within the enclave, even if the operating system hyper visor are compromised. Basically, we try — so there Opaque is about providing Spark SQL, taking Spark SQL and providing Spark SQL functionality but now is secured against this kind of very strong attacks.

[0:25:40.8] SC: But running in the hardware enclave?

[0:25:43.1] IS: Running part of it, right? The operators run in the hardware enclaves.

[0:25:46.8] SC: Okay. Oh wow.

[0:25:48.8] IS: The last one maybe I want to, well, there are two other ones. I want to mention one is Clipper. This is about prediction service. So there we developed this very modular architecture so you can serve models, which are developed with very different systems like of course Spark, it will be actually probably the first prediction serving system, which will provide native support for Spark models.

Develop in Azure in many Azure frameworks like TensorFlow and Azure. So that's Clipper, and the last one I want to mention is Ground. So Ground, it's about managing the meta data. This is one very important problem, but I do have data, which you generated and modified, think about an enterprise, different sources, different people.

This is about really tracking the meta data and answering, being able to answer the questions, "Who creates the data? Where the data is coming from? Who modifies the data? Who deletes the data, and then on top of that of course you can have access control and authorization."

This is kind of a few projects we are working on.

[0:27:00.1] SC: Ground makes me think a little bit of, it might be an interesting block chain application.

[0:27:05.1] IS: The block chain actually is another project, which is based on block chain which is doing authorization. It's called Wave.

[0:27:12.0] SC: Wave? Okay, interesting. Right. Well, before we finish up, is there anything else that you like to leave listeners with?

[0:27:20.7] IS: Well, I think, you are very nice. You ask all these questions, I think that I unloaded almost everything I have. So I'll save something for the next time.

[0:27:29.6] SC: Great, well, thanks so much Ion. It was a pleasure having you on the show.

[0:27:32.4] IS: You are most welcome. Thank you.

[END OF INTERVIEW]

[0:27:37.8] SC: All right everyone, that's our show for today. Thanks so much for listening and of course for your ongoing feedback and support. For more information on Ion or any of the other topics covered in this episode, head on over to twimlai.com/talk/55.

For the rest of this series, head over to twimlai.com/aisf2017, and please, please, please send us any questions or comments that you may have for us or our guests via Twitter @twimlai or @samcharrington or leave a comment on the show notes page. There are a ton of great conferences coming up to the end of year. To stay up to date on which events we'll be attending and hopefully to meet us there, check out our new events page at twimlai.com/events. Thanks again for listening and catch you next time.

[END]