Vertica ISRAEL Newsletter
October 2020

Taboola Case Study

Overview

Taboola develops new technologies that help people find what's interesting and new wherever they are. It's the largest discovery platform in the world, with over 1,000 employees, +300TB data ingested daily, 130 Vertica nodes in +7 clusters, 6.3 million queries daily, and 1PB total compressed storage on the Vertica clusters.
In this newsletter edition we share an interview with Keren Bartal, Director, Data Engineering at Taboola, as part of "The Next Database Platform Event 2020", which took place last month.
The host: Timothy Prickett Morgan.

Increased Scale and Database System Impacts, Choices

.."And now I would like to welcome Keren Bartal to the next Database platform. Thanks for joining our event.
Keren is the Director of Data Engineering at Taboola, which is an advertising and content management firm. We are going to try to get a sense of how the backend of the system works and how it changed over time?
And the challenges that you face when you have exponential growth in your company.
Tell us a little bit about Taboola and your background as well so we have that as a foundation for our conversation.
Taboola was founded in 2007. We are a discovery platform, basically finding content you may like. I'm hoping most of you have seen us when you surf the web. We get around 1.4 billion unique users a month.
We are big, we have lots of data coming in, we ingest a lot of data, analyze a lot of data, that's us.

What's the structure of that platform?

Our stack contains a bunch of MySQL servers, we are at MySQL 5.7 right now. We have Vertica, Cassandra, ScyllaDB, Hbase, HDFS and Spark.

What do all these different pieces do?

Our backend environment ingests all the data that we have coming from our frontend. Any user that clicks anything or even gets our recommendations is a session for us. This is information that we capture, and send all the way to our backend.
We get a few hundred terabytes of data every day that we analyze and ingest into various databases. On each database we do something a little bit different with that information. We start off with taking it into Cassandra, then crunching it a little bit, organizing the data, deduplicating it and pushing it to HDFS in Parquet format. Then we analyze it again, and push it into Vertica.
Over that we do a lot of aggregations and we generate reports that our business partners and our customers can look at.
In MySQL we maintain all of our configurations, some of our fact tables. Some of the data that comes in from the frontend also goes into MySQL servers. It's a cycle, a lot of our algo works on MySQL and Vertica and HDFS. All the different technologies feed each other.
So data we crunch in Vertica then goes back into MySQL, and some of the stuff we do in MySQL goes into Vertica. It's an interesting mix of things.

“We put Vertica in the center of things, because we try to make it a sort of corporate data-lake.”

Did it start out like this? Or did you start with MySQL and then you build something else?
I don’t think you started out this way, as my guess.

Right. We started off with MySQL and Cassandra for the most part.
And we couldn’t really do the reporting that we wanted to do on MySQL. It got too big. So when we reached a database of 60 or 70TB it just didn’t work all that well.
We started testing out a few technologies to replace that with, and we ended-up with Vertica, which we were very happy with. With Vertica we were able to grow to over a petabyte of data. It ingests very very fast, and is able to crunch the data fast.

I want to remind everybody, that’s a petabyte of analytical data!
That’s not a petabyte of advertising or content data. This is telemetry, this is a lot of data! That’s the point. Yes, that’s a tremendous amount of telemetry. This is a very large operation. There’re not a lot of enterprise companies that have that much telemetry. What do the other databases do? or how did you add them? When did you move to Vertica? When did you add the other layers you have as well?

As we grow, the challenges get bigger. And then we try to do the best we can with the technology that we are using right now, that’s the simplest.
But when we reach a point where it doesn’t give us what we need anymore, and we are kind of working around too many things, we start exploring new options.
We’ve added more and more Cassandra clusters, and we grew in that sense and even the MySQL environment doesn’t look the way it did 10 years ago. Now we are working with clusters of MySQL, we are working with ProxySQL, we’ve added a lot of things. We also tested things like ScyllaDB. We’ve sharded off some of the things we do into a smaller ScyllaDB cluster, because we needed a specific latency for that.
MySQL couldn’t handle the analytics that we needed so we moved off to Vertica.
We needed something a little different then Cassandra for different applications, so we brought up an Hbase cluster and we have a few of them. We try to use the best technology for every problem that we have, trying to handle the masses of data that we have.

Do you think that’s normal for companies to do?

I hope it is. You can only use the same technology for so long, you have to keep exploring and testing out new things as they come out. I guess I can’t really stay married to the same thing over time. And not to say that we’ve been able to implement that completely. We’ve stayed with a lot of things over the years, but we are trying to improve on that.

..We ended-up with Vertica, which we were very happy with. It ingests very very fast, and is able to crunch data fast.

There's kind of a push and pull. You want to adopt new technologies but it's so disruptive to do so. You get to a breaking point where are have to do something. I find it interesting that what you've built is sort of like the human brain. There is a lot of old brain and new brain and other kinds of things going on in there.
The idea that you can have one magical database that can do it all, which is sort of the dream of Oracle and DB2 and SQL server to a certain extent. It's not even possible to think about that when you actually start building production applications, that you're going to have point products doing very nishi things. And we are all going to be struggling with a mix of technologies overtime. Do you think that's valid? There is no way to get out of this is my point.

I think that's valid.
It's actually really hard because you need infrastructure to support all these different technologies.
You need infrastructure to ingest data, and then to query it. And we know that querying these different databases, even though these databases are ANSI and are all supposedly using the same query language, it's not entirely true.
There is a lot of adaptation to be done whenever we add a new technology. We developed a lot around the technology, but we developed it to be flexible enough so that we can add new things.
So we do put Vertica in the center of things, because we do try to make it a sort of corporate data-lake.
We pull a lot of the data from different technologies into Vertica.
We've created an ETL application that allows us to push and pull data across the different platforms and do what we need with it. Basically we try to build flexible things so that we can keep growing with, and embrace new things. It's not easy. Just think about the amount of puppet manifests that we have.

One of the interesting users we spoke to for this event is Edward Sverdlin from UnitedHealth who runs up their research and development arm, and 2 years ago started implementing a graph database, as you might imagine UnitedHealth largest healthcare provider here in the United States. They got a lot of different databases and they were trying to figure out how to do a 360 view of all the customers, and they chose a graph database overlay to be their analytical platform so they could get all that information out of these databases and put it in one place. It sounds like you use Vertica for much the same purpose, that it connects to everything, and then you can pull the information you need to actually run the business out of that. Is that a good analogue?

Yes. Graph database sounds really interesting. We do try to centralize the data for analysts so that they can join and play with the data in one place. Even though we do have things that can pull from different data sources and show you the data set from bunch of different places. That's hard and when you talk about larger data sets it usually doesn't really work very well. We do try to centralize, and right now we are doing it successfully with Vertica.

What's the infrastructure that supports all that stuff? Are we talking about thousands of nodes sitting in a datacenter somewhere? How big is that backend thing?

It's actually very big. We are completely on prem. But we do use BigQuery and Kusto for analytics. We have around 7500 servers around the world. That's front-end and back-end. We are based in Israel. All of our backend datacenters are in Israel. We have 3 of them and we kind of play with it. Our frontend datacenters are in various places, US, Europe, and everywhere.

How fast does that infrastructure grow overtime? Are you doubling every year in terms of the amount of capacity you have? Even 50% is hard to carry.

We grew a lot. Over the last three years we grew almost in an order of magnitude.
In terms of capacity we tripled our Vertica and MySQL servers. It really depends, not everything grew by that much.
It changes between different technologies that we have.
We have hundreds of MySQL servers, many hundreds of Cassandra servers and around 100 Vertica nodes.

That's a very large installation of Vertica.
Yes, we have 3 separate use cases, and for each one we have a few Vertica clusters that backup each other and do different things.

If you had to do it all over again, and start today, how would you build this?
I would model data differently and would possibly use different technologies for certain things. We do have many technologies and definitely would use a lot of them but it would look differently. We grew so much overtime so the model and on its own would be different.

Would you stay on premise or would you think using one of the public clouds?

If I had unlimited funds I would be on the cloud. The cloud is more expensive than on-prem. On-prem is a little bit harder, but in terms of cost it didn't make sense for us to be on the cloud. I'd love to be there, I'd love not to handle hardware issues, or get up at night when a server crashes etc. For me it would be easier to be on the cloud but for Taboola it doesn't make sense.

We've done the math in 50 different ways from Sunday and it doesn't make any sense unless you are committed to the idea that you never want to run infrastructure.
And when you get to your scale.. Sorry you don't have a choice.
You have to do it because it's too much money. It is that simple.
The only people that get cheap infrastructure are Google, Microsoft and Amazon. Because they let us pay for their infrastructure and they get it for cheap.

Thank you very much Keren for chatting with us, I really enjoyed the conversation."

The interview is available on YouTube here: https://youtu.be/nN8Wjab69VU?t=2215

The full event on The Next Platform: https://www.nextplatform.com/2020/09/01/the-next-database-platform-full-recording-available/

Industry landing pages (Updated)

Each page has first a generic section with use cases examples for common industries

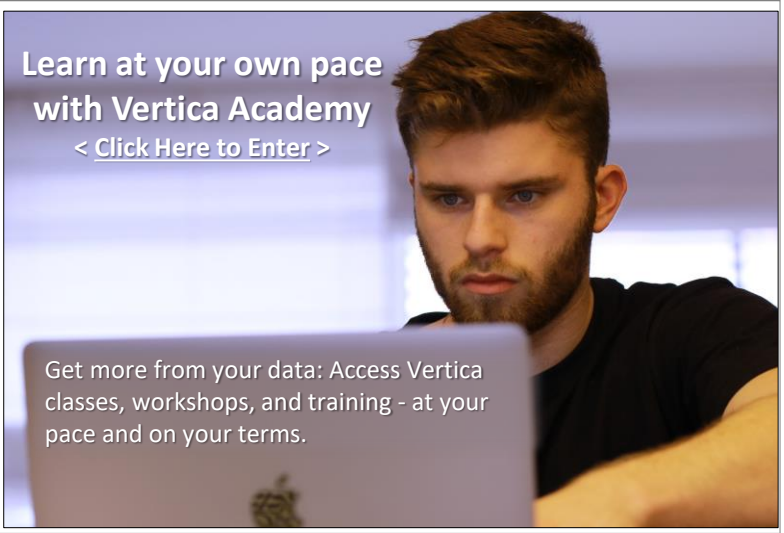
- Telco -> https://www.vertica.com/data-disruptors/telco
- CME -> https://www.vertica.com/data-disruptors/cme
- Software/ OEM -> https://www.vertica.com/data-disruptors/software
- Retail -> https://www.vertica.com/data-disruptors/retail
- Financial Ser. -> https://www.vertica.com/data-disruptors/financialservices
- General -> https://www.vertica.com/data-disruptors

Blogs, Technical Blogs and Knowledge Base

- Vertica Benefits calculator
- Vertica Integration with Microsoft Power BI: Connection Guide
- Vertica Hardware Guide
- Vertica Integration with Looker: Connection Guide
- Vertica Integration with Microsoft SQL Server Tabular Analysis Services: Connection Guide
- Case study for 14PB Vertica implementation at Trade Desk
- Vertica outperforms competing cloud analytical platforms in third-party benchmark study
- How security risk is making sense to the corner office
- What's the Deal with Vertica TCO?
- Take Nothing but Memories. Leave Nothing but Footprints
- Announcing Vertica Integration with DbSchema

Every week..

Vertica newcomers and experts are getting more from their data by engaging the many offerings at Vertica Academy. It's an online, how-to resource that can quickly get you up to speed on Vertica essentials and advanced topics, with more content being developed all the time.



Did you know?

VerticaPy is available now!

VerticaPy is a Python library that exposes scikit-like functionality to conduct data science projects on data stored in Vertica, taking advantage of its speed and built-in analytics and machine learning capabilities. It supports the entire data science life cycle, uses a 'pipeline' mechanism to sequentialize data transformation operations (called Virtual Dataframe), and offers several options for graphical rendering.
Among other features, VerticaPy is the perfect blend of the scalability of Vertica and the flexibility of Python, bringing a unique and Indispensable set of data science tools.
For example: Descriptive Statistics Data Exploration, Charts for Data Visualization, Correlation & Dependency Statistical Relationships

Useful links

- Vertica Home Page
- Vertica 10.0.x Documentation
- SP1 - 10.0.1 Release Notes
- Vertica Blogs Checklists
- Web Based Training
- Customer Case Studies
- Demos
- Webcasts
- Knowledge Base
- News
- Newsletters
- Quickstarts

VERTICA