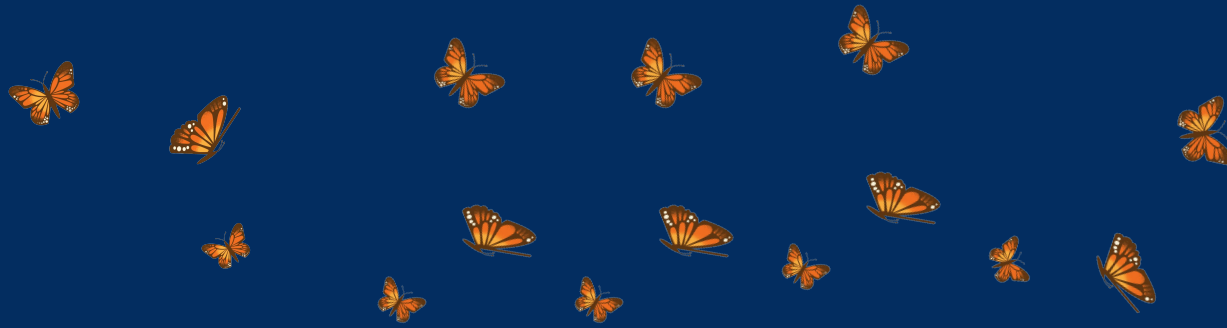


Anomaly Detection for Salesforce's Production Data Using Trino

Tuli Nivas, Geeta Shankar

Last Edited: 05/15/2023

salesforce



Tuli Nivas

Performance Engineer



Geeta Shankar

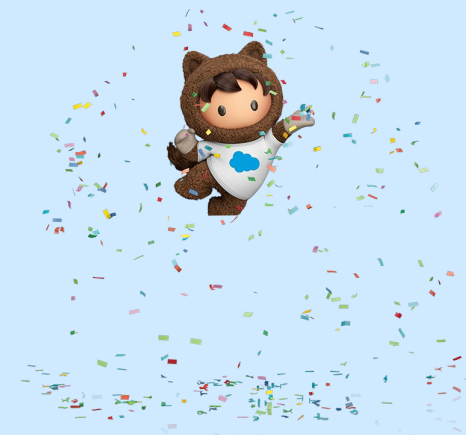
Software Engineer



Why do we need Trino?

Platform Engineering Performance Team

salesforce



Need access to copious amounts of production data in a timely manner



Able to extract relevant, actionable information from production data



Execute queries on a petabyte scale across clusters of nodes



Produce near-instant dashboard rendering and refiltering





Trino's Value



Improving our
cost-to-serve

20 min

SLA on log
latency



194%

Faster than querying
similar queries from
Splunk

>30 days of data

No longer restrained
by Splunk data
retention policies

Smaller footprint

Fewer queries to run
as compared to
Splunk

Sandbox

Ability to create tables
and views for
temporary data
storage & analytics



How are we using Trino at Salesforce internally?



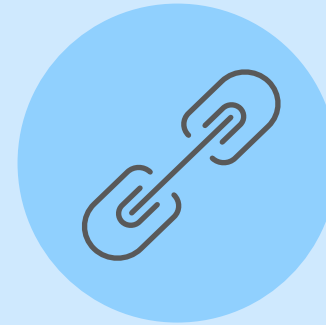
Internal Trino Platform

To analyze high-cardinality, multi-dimensional data for analytical scenarios using the Trino query engine



Storage of Production Data

Tokenized production data is stored for up to 2 years and untokenized production data for up to 30 days



Connect Via Performance Tool

JDBC connection with our performance platform provides necessary SSL certificates to access production data



Jupyter Notebook Python Analytics

Statistical analysis of query outputs and implementation of visualization libraries like Plotly



What is a Release?



Spring



Summer



Fall



Salesforce provides customers with new features and updates to our technology



Releases happen all in the cloud, so customers get access everywhere



Releases are deployed to customers on production pods



Customers share feedback and performance engineers monitor



How Performance Engineering in Production at Salesforce Works

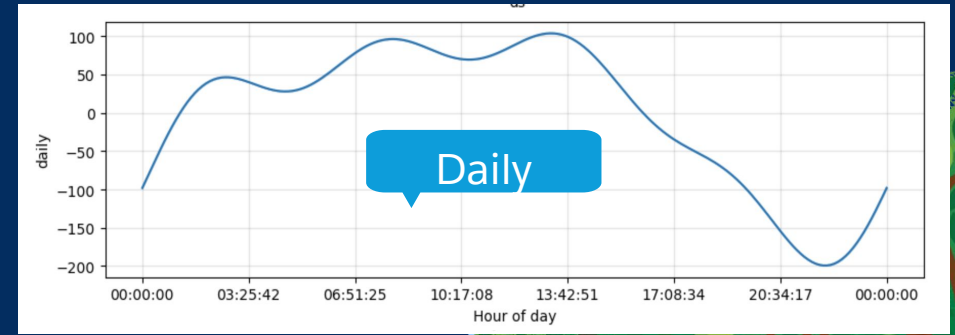
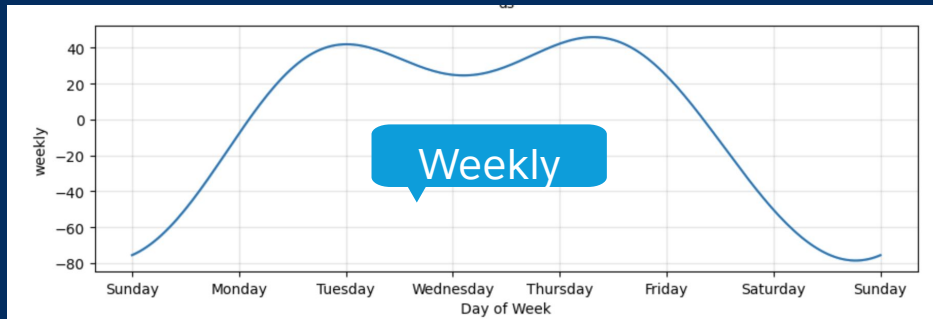
1. A new release is rolled out
2. Customers are allocated to different production pods
3. Performance engineers monitor performance on each production pod, comparing previous and current releases



A production pod refers to the complete stack of our technology infrastructure



Anomaly Detection for Response Time Metrics



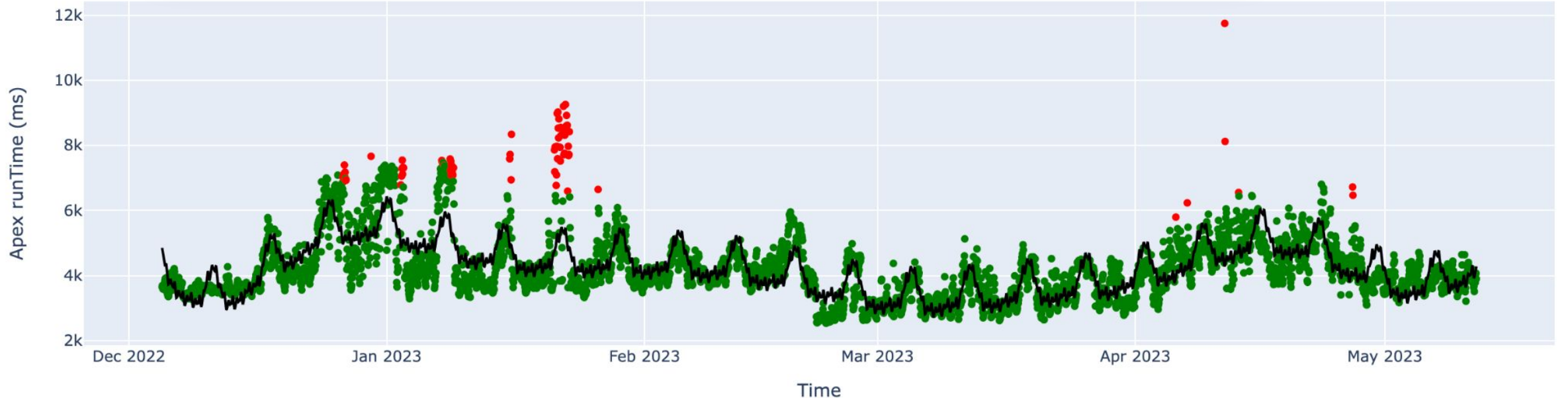
Prophet Algorithm
Meta's forecasting algorithm for non-linear time series data



Anomaly Detection for Response Time Metrics

Anomaly Detection By the Hour

From parameters analyzing daily, weekly, and seasonal usage patterns of response time metrics



Why is anomaly detection crucial for response time metrics?

1. Performance Optimization
2. Troubleshooting and Debugging
3. Capacity Planning
4. Quality Assurance

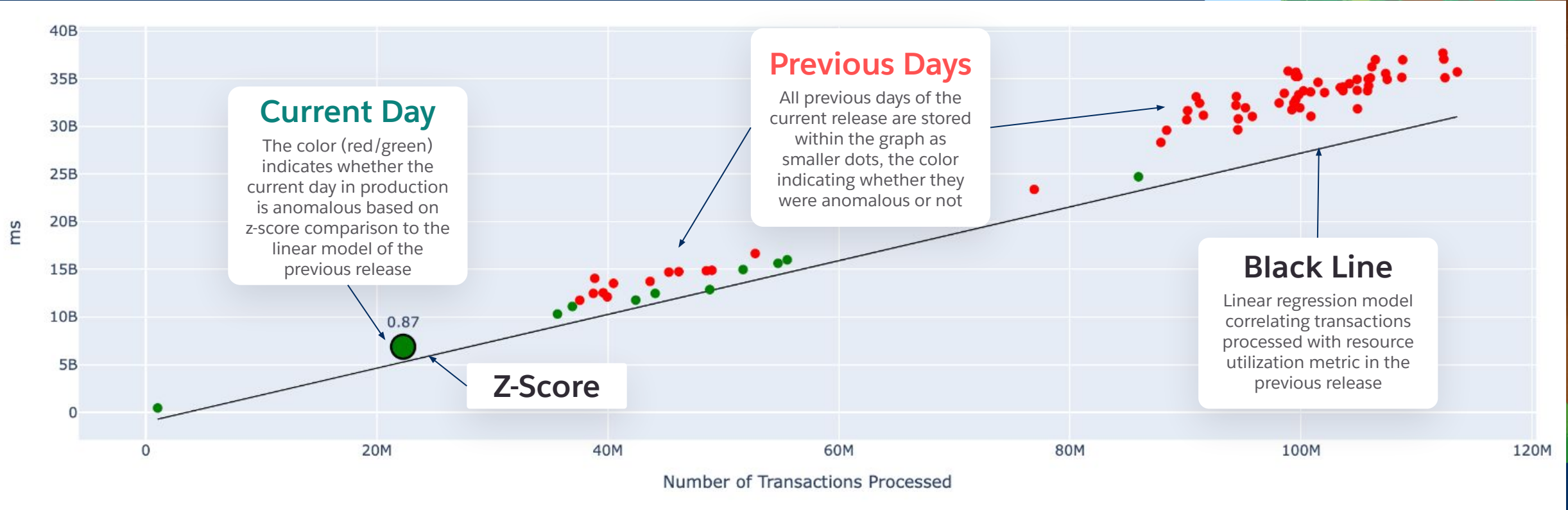
Examples of Response Time Metrics:

Runtime
Apex Execution Time
Compile Time





Anomaly Detection for Resource Utilization Metrics



Why is anomaly detection crucial for resource utilization metrics?

1. Performance Optimization
2. Resource Planning and Scalability
3. Troubleshooting and Root Cause Analysis
4. SLA Compliance and User Satisfaction
5. Performance Baselines and Trend Analysis



Examples of Resource Utilization Metrics:
Application CPU Time
Database Time



Weekly Customer Usage Patterns

Each Circle

represents a top customer in a production pod. Usually names of customers are displayed next to each circle.

Circle Size

indicates how "big" the customer is, meaning how many transactions does this customer process

Circle Color

Red indicates more than a 5% difference in cumulative db or app cpu time, yellow indicates 0-5%, and green indicates a negative percentage change



Triggers and Entrypoints at Salesforce

- Trigger: Apex code snippet
- Entrypoint: source of Apex call



We monitor triggers and entrypoints to identify specific areas that are causing performance issues





Daily Entrypoint and Trigger Behavior



Trigger Type Breakdown

“Triggers” entrypoint broken down into trigger types

Entrypoint Comparison

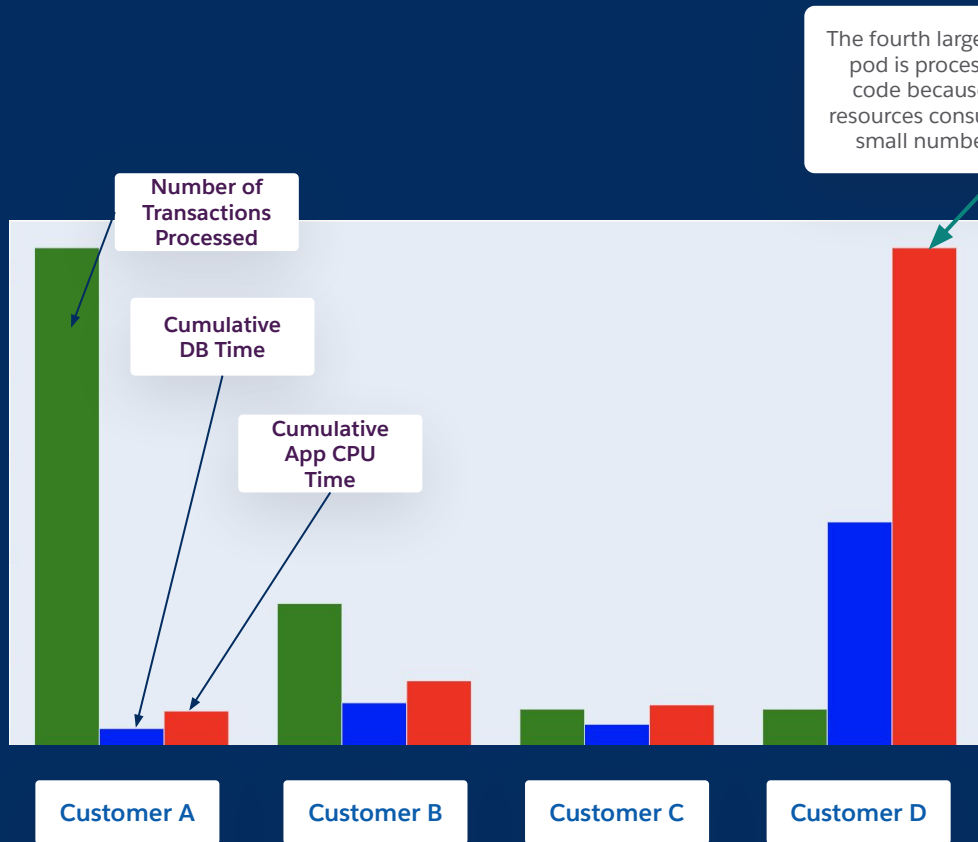
For top entrypoints in a production pod we are able to track response time improvements and degradations.

Current Release P95 Runtime	Previous Release P95 Runtime	p95_runtime	Percent Change
148.190000	142.100000		4.110000
470.140000	486.060000		-3.386000
7968.690000	8436.070000		-5.865000
382.670000	381.280000		0.363000
75.980000	28.800000		62.095000
4738.750000	4427.810000		6.562000
92.470000	95.260000		-3.017000
1679.690000	1754.040000		-4.426000
13899.940000	9268.970000		33.316000
768.270000	826.870000		-7.628000





Customer Behavior on Production Pods



The fourth largest customer on this pod is processing unoptimized code because of the very high resources consumed for a relatively small number of transactions

Which Customers' Transactions are Efficiently Processing and Optimized?

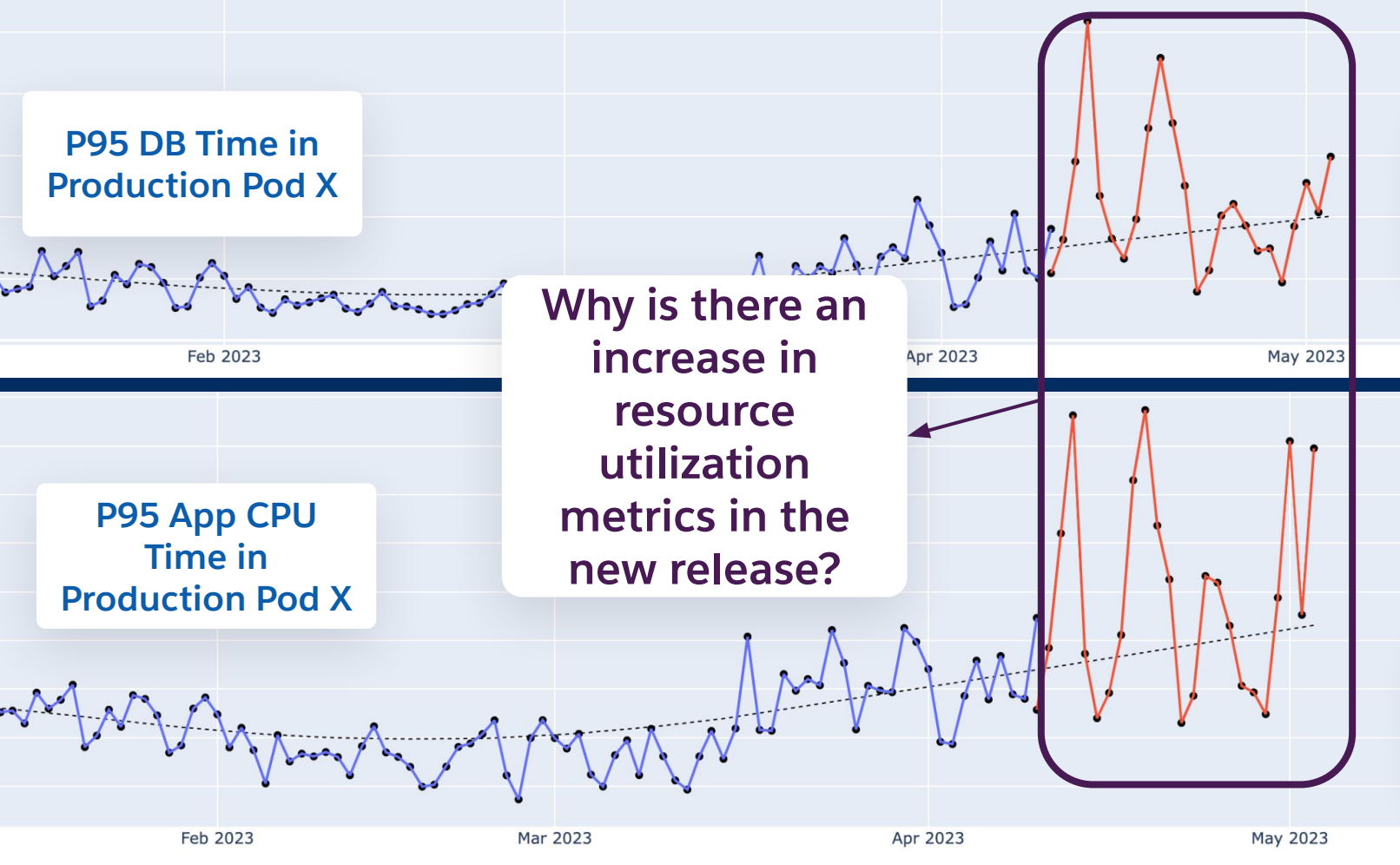
Top Customers

included in the bar chart where we can track each customer's number of transactions processed and resource utilization metrics, cumulative db time and cumulative app cpu time





New Customers on Production Pods, Introducing New Pod Behavior



Comparative Analysis

Correlation Analysis





New Customers on Production Pods, Introducing New Pod Behavior

Top DB Time Utilization Customers

Top App CPU Time Utilization Customers

Avg DB Time % Change	P95 DB Time % Change	Avg App CPU Time % Change	P95 App CPU Time % Change
-19.740000	-9.750000	-19.270000	-9.840000
nan	nan	nan	nan
18.670000	9.980000	29.690000	102.510000
-71.400000	-80.160000	-69.980000	-33.560000
nan	nan	nan	nan
nan	nan	nan	nan
nan	nan	nan	nan
nan	nan	nan	nan
nan	nan	nan	nan
-49.660000	-42.000000		-38.380000

NaNs show that many of the top customers in terms of resource utilization metrics have been added to this production pod this release

Avg DB Time % Change	P95 DB Time % Change	CPU Time % Change
-19.740000	-9.750000	-9.840000
-71.400000	-80.160000	-33.560000
nan	nan	nan
nan	nan	nan
nan	nan	nan
nan	nan	nan

New customers have an impact on performance in production pods

1. New customers bring their own set of transactions and activities
2. Resource competition emerges as new customers are added to a production pod
3. New customers may have different usage patterns compared to existing customers



What Can Trino Help Us Achieve in the Future?



Performance Assessment

Understand our performance improvements and degradations over multiple releases



Quick and Efficient Processing

Queries are able to run across more than one production pod at once



Tableau Dashboards

Presto server connection seamlessly integrates to Tableau dashboards



Real Time Anomaly Detection

With current Trino SLA and our robust algorithms, anomaly detection becomes faster than ever before



salesforce

Special thanks to
our internal Trino
teams at Salesforce

Thank you!

Q/A

