ORACLE

HeatWave Autopilot Indexing

Machine-learning-powered, workload–aware index recommendations for OLTP workloads

Purpose statement

This document provides an overview of features and enhancements included in HeatWave. It is intended solely to help you assess the benefits of HeatWave and to plan your I.T. projects.

Disclaimer

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle. Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

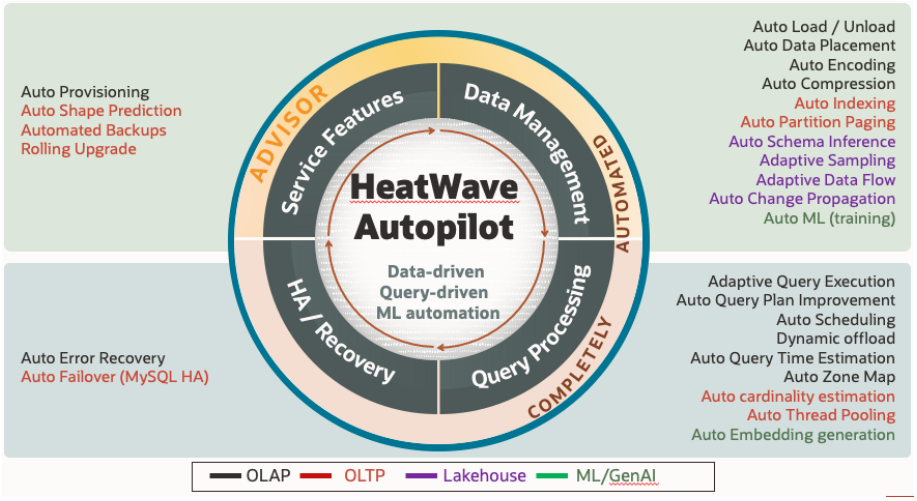
Some of the benchmark queries are derived from the TPC-C benchmark, but results are not comparable to published TPC-C benchmark results since they do not comply with the TPC-C specification.

Table of contents

Introduction

HeatWave provides automated, integrated, and secure generative AI and machine learning (ML) in one cloud service for transactions and lakehouse scale analytics—without the complexity, latency, risks, and cost of ETL duplication.

HeatWave Autopilot automates many of the most important and often challenging aspects of achieving high query performance at scale - including provisioning, data loading, query execution, and failure handling. It improves performance and scalability without requiring database tuning expertise, increases the productivity of developers and DBAs, and helps eliminate human errors. HeatWave Autopilot uses advanced techniques to sample data, collect statistics on data and queries, and build machine learning models to model memory usage, as well as network load and execution time. These machine learning models are then used by HeatWave Autopilot to execute its core capabilities.



HeatWave Autopilot features at-a-glance

HeatWave Autopilot is highly useful for OLTP, analytics, machine learning, vector processing and generative AI workloads.

For OLTP applications, HeatWave Autopilot has introduced Auto Thread Pooling which helps maintain throughput at high concurrency rates by performing smart admission control. It also includes Auto Shape Prediction which recommends the optimal buffer size and its associated shape for a given OLTP workload, helping users to always get the best price-performance.

Problems with manual performance tuning using MySQL indexes

Creating an optimal set of indexes has always been a challenging task for DBAs. On one hand, creating too few indexes results in low query performance, especially for SELECTs. On the other hand, creating too many indexes leads to excessive index maintenance during DMLs, which in turn, also hampers performance. Furthermore, the existence of too many indexes takes up storage

and can also cause the query compilation to slow down as the optimizer tries to evaluate the best index candidate for a given table/query. For DBAs to recommend indexes, not only do they need to have a deep understanding of the ways in which indexes impact various query constructs (e.g., WHERE predicates, JOINs, ORDER BY), but they also need to come up with an optimal and minimal set of indexes for a given workload that maintains a balance between performance and storage. In addition, if the user workload changes, the DBA needs to revisit the index choices all over again, making their task quite daunting.

HeatWave Autopilot Indexing

HeatWave Autopilot Indexing is an ML-based technology designed to optimize database systems for better cost and performance. With Autopilot Indexing, database administrators no longer need to manually identify which indexes are most beneficial for their workload. Autopilot Indexing automatically generates secondary index recommendations for creating and dropping indexes based on the current workload. Autopilot Indexing considers both the query performance and the cost of maintaining the indexes when generating recommendations. It provides performance and storage estimations, as well as explanations for the recommendations it generates. The Autopilot Indexing interface consists of a simple and intuitive console that customers can use to view and analyze the projected performance and storage impact of recommended index suggestions. This makes it easy to foresee the impact of changes to the database systems before applying the suggestions.

With HeatWave Autopilot Indexing, users can easily identify and resolve performance issues with their database systems. Autopilot Indexing has a comprehensive set of features that allow users to tune the performance of database systems, such as:

1. Considers both query and DML performance (index maintenance cost)
2. Recommends CREATE and DROP of indexes
3. Generates DDLs for index creation/drop
4. Provides performance prediction (per query and total workload)
5. Provides storage prediction for the recommended indexes
6. Provides an explanation for the recommendations
7. Console integration to improve user experience

Autopilot Indexing Features

ML-powered automation

In the MySQL InnoDB engine, the base tables are stored in a B-Tree structure based on the primary keys that are either user-specified or auto-generated. Hence, data lookup based on the primary keys is very efficient. For many OLTP workloads, fast lookup on primary keys alone is not enough; secondary indexes are needed to further improve performance, and they play a crucial role in performance tuning. Unfortunately, secondary indexes are hard to determine

manually as they involve complex data relationships. Autopilot Indexing uses advanced machine learning to predict secondary indexes based on the workload.

Unlike popular "what-if" approaches that create hypothetical indexes by relying on the query optimizer's cost estimation for index selection, Autopilot Indexing extracts features from existing query plans and uses machine learning models to estimate the performance impact of index candidates. By using ML models along with workload features, Autopilot Indexing:

- Takes guesswork or trial-and-error out of picking optimal indexes.
- Dynamically recommends indexes as workload changes.
- Provides explainable and quantifiable index suggestions (e.g., 2x better performance with 20% reduced storage)

Using ML models also enables Autopilot Indexing to adapt and generalize across different workloads, datasets, servers, and cloud environment characteristics. Autopilot Indexing ML models are constantly updated via a large variety of workloads, and new models are generated automatically for every release.

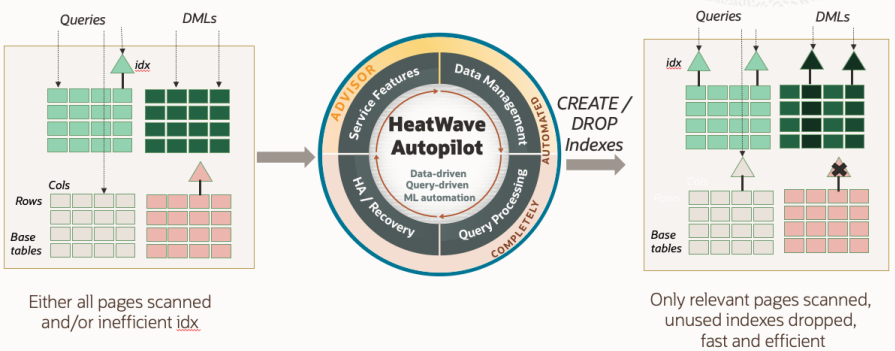
Autopilot Indexing can model desired optimization targets such as throughput, latency, and storage. As a result, Autopilot Indexing can:

- Model performance impact without creating the indexes (metadata OR the data)
- Impose minimal compute or storage overhead on HeatWave MySQL instances
- Help validate the new index candidates without executing the queries in the background

Query- and DML-aware

There are three key user benefits of Autopilot Indexing when compared to similar approaches:

1. Both query and DML performance are considered
2. Recommendations are not only provided for the creation of new indexes but also for dropping existing indexes that are no longer needed.
3. Storage and performance predictions are provided for the indexes recommended, which enable users to balance the performance and storage gains with the index maintenance costs.



High-level diagram showing a sample user workload and indexes before/after using HeatWave Autopilot Indexing.

The diagram above (left hand-side) depicts a sample user workload containing SELECT (#1, #2, #3) and DML (#4, #5) queries accessing columns of four different tables. In this scenario, the user workload already has two of the columns indexed. Autopilot Indexing suggests CREATE and DROP recommendations to optimize this workload. To do so, it not only considers the performance improvement but also the index maintenance cost, which can significantly impact DML operations, and recommends actions that strike a balance between query and DML performance. This ensures that your database remains efficient and optimized for both types of operations.

The left-hand side depicts a case where the user attempted to create appropriate indexes for their OLTP workload. Unfortunately, we observe that despite the best effort, some of the accessed columns (#1, #2, #5) do not possess an index, causing a potentially expensive scan of the base table. Furthermore, we also observe that the user-picked indexes are not necessarily used, causing an unnecessary storage overhead (i.e., indexes created for the blue base table).

The right-hand side depicts the state of the database system after the user took advantage of Autopilot indexing. Upon creating the right set of indexes, we observe that (i) queries use newly created secondary indexes (#1, #2), (ii) columns that receive DML activity use newly created indexes (#5), while for certain columns' DMLs maintenance cost prevented an index recommendation (heavy DML activity of DML #4 shown with dark color), and (iii) unused indexes are dropped to optimize storage cost and efficiency of DML operations on the affected tables (i.e., index created for the blue base table).

Generates DDLs

Autopilot Indexing generates the necessary DDL (Data Definition Language) statements for index creation and drop so it is easy for the user to apply the recommendations either via the console or SQL interface.

Performance prediction per queries and overall workload

Autopilot Indexing uses machine learning to predict the performance of queries and DML statements with different candidate indexes. This allows the system to make informed recommendations for creating and dropping indexes based on the predicted performance improvements. Autopilot Indexing has several individual models that help to predict the performance of the overall query. The machine learning models are trained on offline data and are constantly updated across versions to ensure that they are accurate and up-to-date with the latest database version. The use of machine learning models also precludes the need to create either metadata or data for indexes, saving storage and compute costs as well as time during the index recommendation process.

Storage prediction

Autopilot Indexing provides storage prediction via machine learning. This allows the system to recommend the creation and drop of indexes based on the impact on storage. The storage predictions show the potential storage impact of the recommended changes. The storage prediction models are also trained on

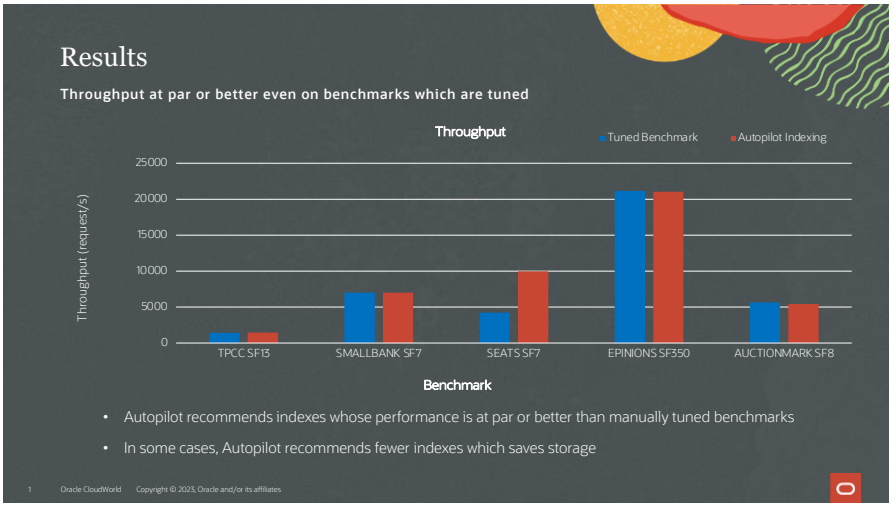
offline data and are constantly updated to ensure that they are accurate and up-to-date with the latest database version.

Explanation for the recommendations

Autopilot Indexing provides explanations for the recommended index changes, including the performance and storage benefits. This helps you understand the reasons behind the recommendations. For example, a DROP INDEX suggestion can be made because the index was *unused* or *duplicated*, while a CREATE INDEX could be suggested because it helps a query use an *index scan* instead of a *table scan*, thereby improving performance.

Autopilot Indexing Results

The diagram below presents the benefits of using Autopilot Indexing. Performance benefits of Autopilot Indexing are measured by using industry-standard OLTP benchmarks, which by default contain good, hand-tuned indexes. Our results show that Autopilot Indexing is not only able to identify indexes required for an OLTP workload, but it could also further improve the performance even when the baseline system is already tuned (e.g., Seats benchmark).



The quantitative benefits of using HeatWave Autopilot Indexing

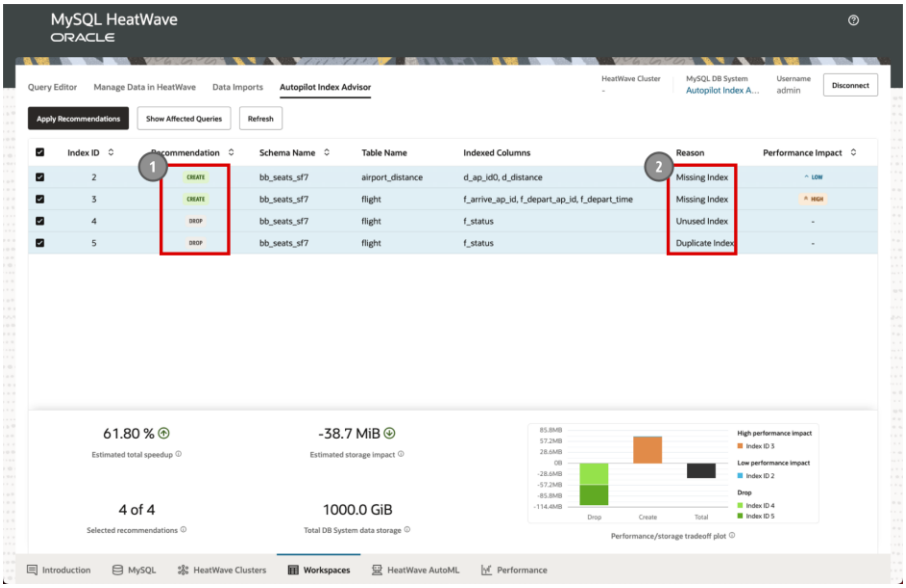
Autopilot Indexing User Interface

The Autopilot Indexing Advisor integrates with the HeatWave on AWS service console that allows users to see recommended index suggestions, navigate affected SQL statements for each recommendation, and to visualize the performance and storage benefits of the recommended index changes. This makes it easier to understand the impact of the changes and make informed decisions.

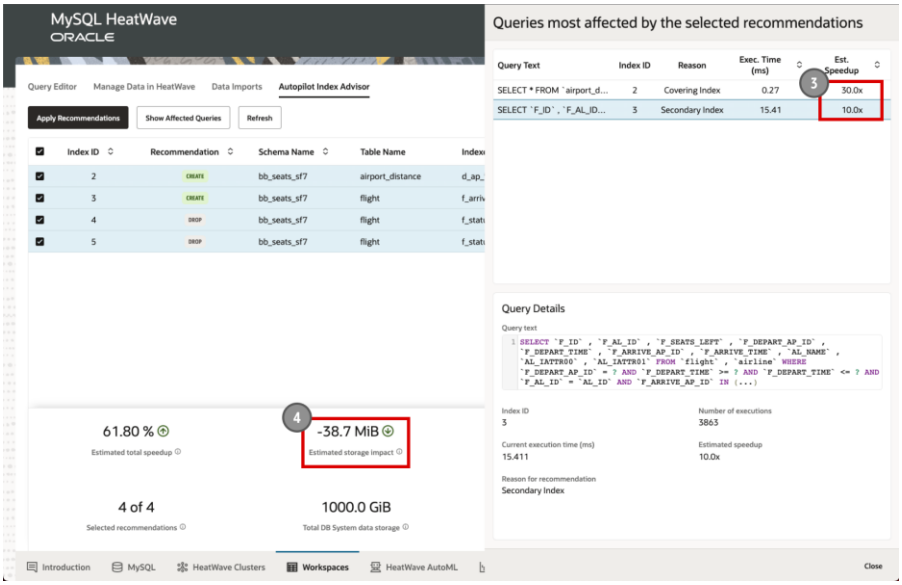
In the console, users find the Autopilot Index Advisor in the Workspaces tab. The console screenshot below depicts the landing page of the Autopilot Index Advisor. The page is populated with recommendations from a sample workload that resulted in five different secondary index-related actions. The label (1)

shows the recommendations that include CREATE and DROP indexes. Similarly, label (2) shows the exact reason why a certain index recommendation was made; in this case, the reasons vary between missing index, unused index, and duplicate index. The users are also presented with the indexed columns and performance improvement labels, shown as HIGH and LOW (for DROP suggestions no performance impact is presented given that such indexes were unused).

The second console UI figure depicts an additional window of the HeatWave Autopilot Index console that lists the affected query statements based on the selected index recommendation(s). Each affected SQL statement is fully outlined at the bottom screen with additional execution metadata such as the number of times that a particular statement was executed in the analyzed workload. The screenshot label (3) shows the performance improvement estimation per query. Similarly, the screenshot label (4) shows the total storage impact when selected index suggestions are applied. By doing so, the users can understand both the performance and storage impact of applying the indexing recommendations. To apply the suggestions, the users can simply press the “Apply Recommendations” button and proceed with creating and dropping indexes based on the listed suggestions.



Autopilot Indexing console screenshot showing (i) CREATE and DROP index suggestions, (ii) explanations



Autopilot Indexing console screenshot of (iii) estimated performance impact, and (iv) estimated storage impact

Summary

Autopilot Indexing is a powerful tool for database systems performance optimization, using machine learning to automate the creation, drop, and maintenance of indexes—helping customers eliminate the time-consuming tasks of creating optimal indexes for their OLTP workloads and maintaining those over time as workloads evolve. It provides performance and storage predictions along with an explanation for the recommendations and integrates with the console of the database system.

Connect with us

Call +**1.800.ORACLE1** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com facebook.com/oracle twitter.com/oracle