Central Canada Db2 Users Group & IDUG Data Tech Summit

May 13-14, 2024

Scarborough, ON

Db2 AI for z/OS SQL Optimization Deep Dive

Tom Beavin
Db2/Db2ZAI developer

May 14th, 2024
Session, z/OS-9

Disclaimer

The information contained in this publication is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied. In addition, this information is based on IBM’s current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this publication or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM’s sole discretion based on market opportunities or other factors and are not intended to be a commitment to future product or feature availability in any way. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

Db2 for z/OS and AI

Apply AI throughout applications, transactions and operations

Business Insights

Db2 13 SQL Data Insights (SQL DI)

Uncover and monetize hidden business insights in Db2 data

Deep learning Large Language Model (LLM) embedded into Db2, leveraged by tried-and-true SQL queries

Operational AI

Db2 AI for z/OS (Db2ZAI)

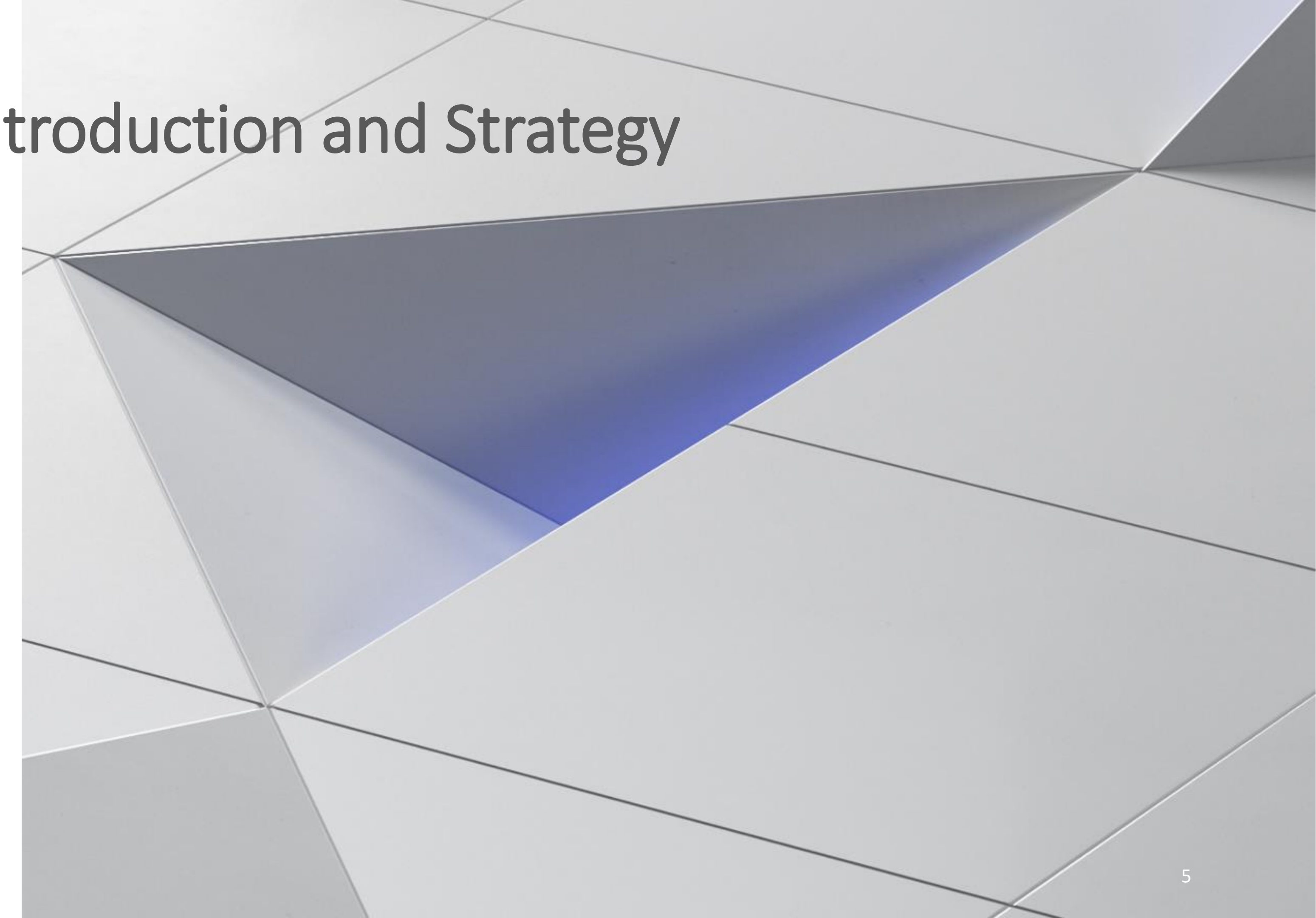
Enhance Db2 database efficiency, security, and performance

Combine machine learning and domain knowledge to enhance Db2 for z/OS operations

Agenda

- Quick introduction of Db2ZAI and strategy
- SQL Optimization Deep Dive
- SQL Optimization Future Direction
- Q&A

Db2ZAI Introduction and Strategy



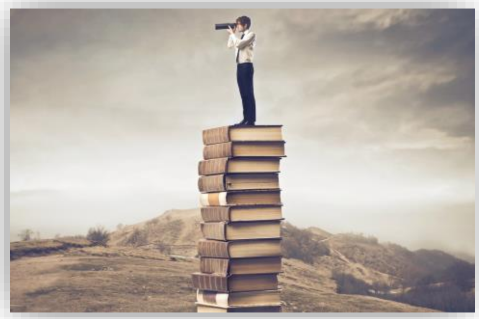
WHY – Operational AI in Db2?

Do more with less

- Increasing demands to optimize the operational cost with growing transactions

High learning curve

- Db2 has too many tuning knobs / instrumentations but no time to learn



More complex environment

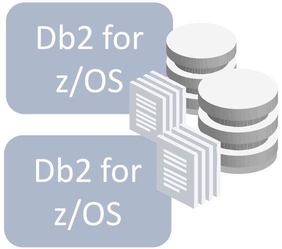
- Lack of knowledge of remote application behavior and environement

Db2 AI for z/OS

Collect the operational data, leverage AI to improve Db2 performance and resiliency

Data collection ›

Db2 for z/OS collects SQL executions, Db2 instrumentation, operational data



Learning and optimization ›

Db2ZAI learns, predicts and optimizes or recommends updates on Db2 execution behaviors

SQL Optimization

Reduce CPU consumption
CPU saving with better query access path

System Assessment

Improve Productivity
Less time spent analyzing data, more on finding solutions

Distributed Connection Control

Protect critical resources
Act quickly, uncover issues early, prevent a flood of connections

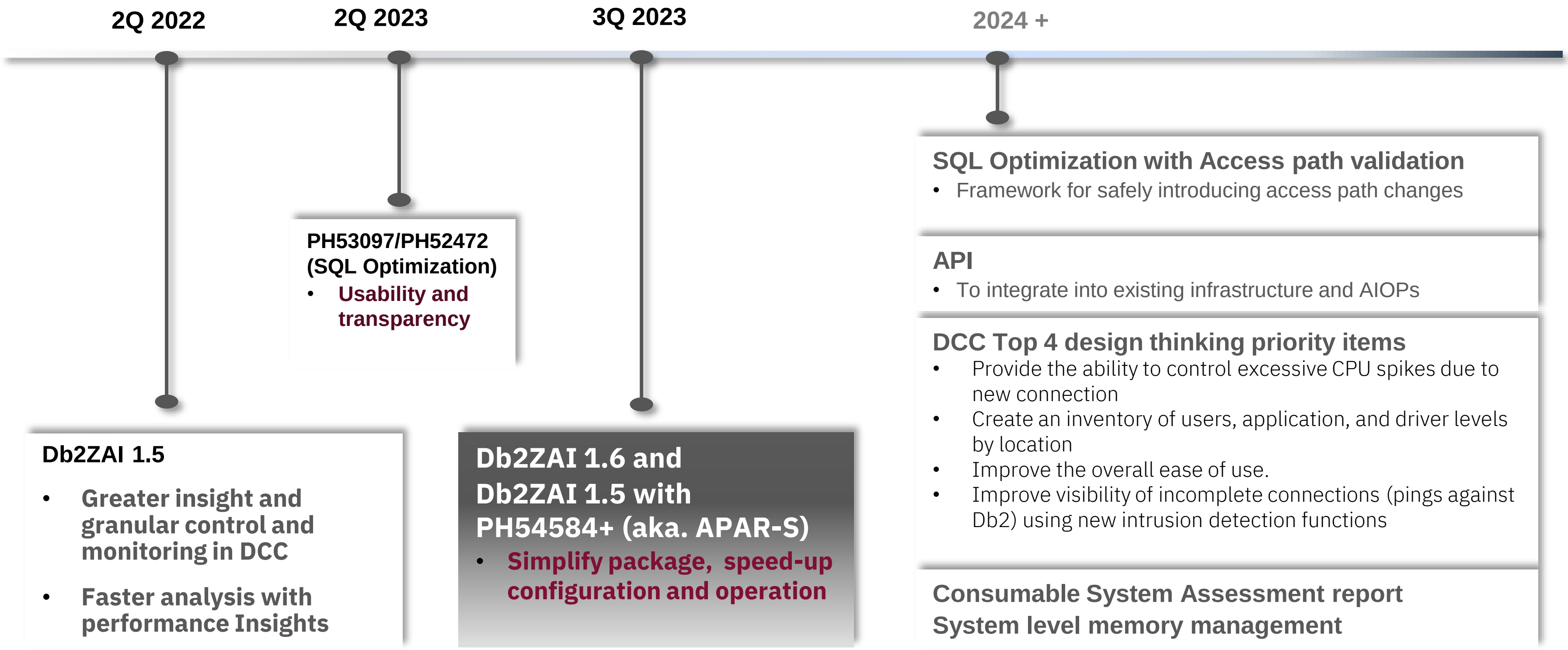
Db2 AI for z/OS Strategy

Db2ZAI targets Db2 for z/OS to be **self-managed** based on the data and patterns from the operational environment by infusing AI

The balance between AI and user control needs to be flexible to allow the user to first gain trust with AI before using it extensively.

This means the balance should initially tilt toward the user **with good defaults and with transparency**, and after trust is gained, then shift toward automated AI

Journey to Self Tuning & Managing



2Q 2022

Db2ZAI 1.5

- **Greater insight and granular control and monitoring in DCC**
- **Faster analysis with performance Insights**

2Q 2023

PH53097/PH52472 (SQL Optimization)

- **Usability and transparency**

3Q 2023

Db2ZAI 1.6 and Db2ZAI 1.5 with PH54584+ (aka. APAR-S)

- **Simplify package, speed-up configuration and operation**

2024 +

SQL Optimization with Access path validation

- Framework for safely introducing access path changes

API

- To integrate into existing infrastructure and AIOPs

DCC Top 4 design thinking priority items

- Provide the ability to control excessive CPU spikes due to new connection
- Create an inventory of users, application, and driver levels by location
- Improve the overall ease of use.
- Improve visibility of incomplete connections (pings against Db2) using new intrusion detection functions

Consumable System Assessment report
System level memory management

Notes : PH54584+ = PH54584, PH56777, PH56807, and PH56843

Db2 AI for z/OS 1.6 offers…

Too many moving parts!
Too complex to install and set up

- Same core capability but with a significantly simplified installation, configuration and ongoing maintenance
  - Updated ML learning libraries and integrated into one FMID

    From : HCOY150, HAQN240, HSPK120, HANA110 and HMDS120

    To : HCOY160
  - Reduced resource requirements, address spaces to maintain
  - Reduced pathlength by embedding the components

- Added transparency and user control in SQL Optimization

Note: WMLz 2.4 and IZoDA EOS in Feb/2024

Existing Db2ZAI 1.5 customers should apply APAR PH54584+ that provides the same capability as Db2ZAI 1.6 GA level or migrate to 1.6

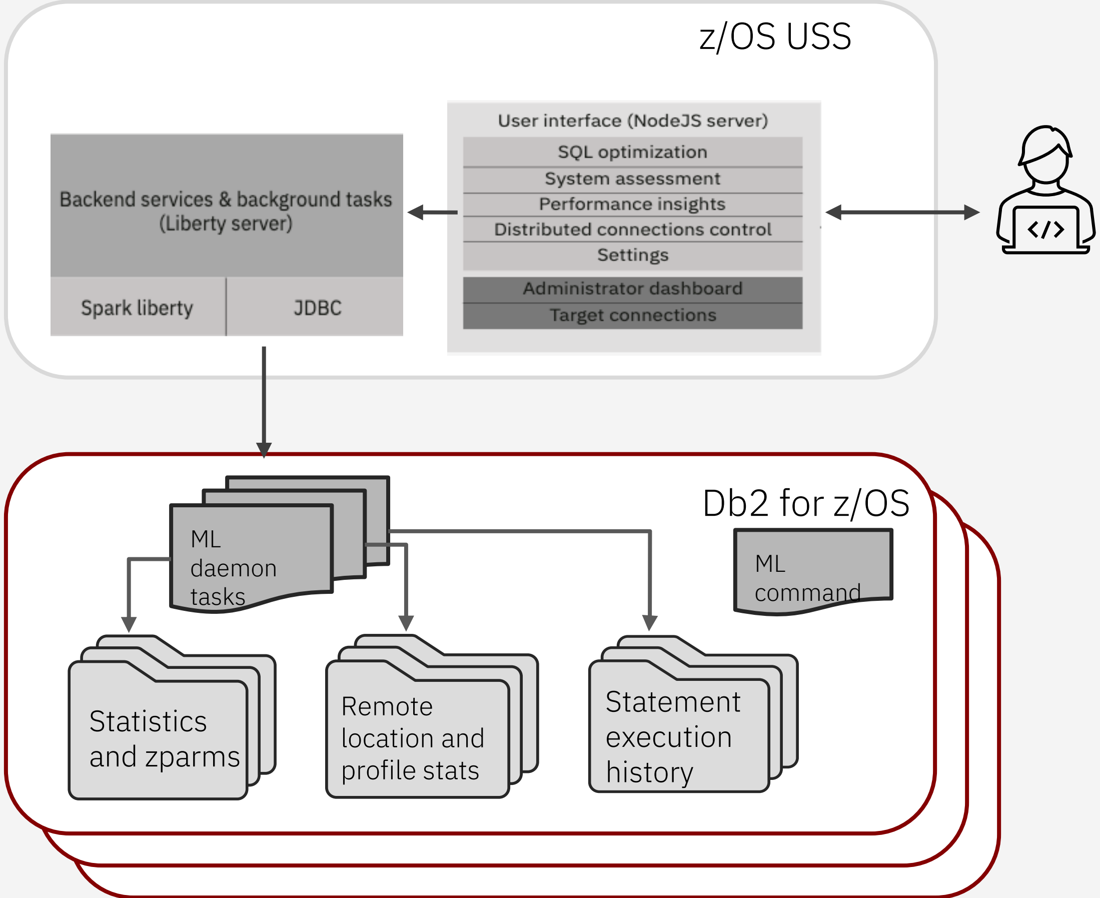
Db2ZAI 1.6 High Level Architecture

Db2ZAI instance

(metadata)

Target Db2 groups

Or subsystems



z/OS USS

Backend services & background tasks (Liberty server)

Spark liberty

JDBC

User interface (NodeJS server)

SQL optimization

System assessment

Performance insights

Distributed connections control

Settings

Administrator dashboard

Target connections

Db2 for z/OS

ML daemon tasks

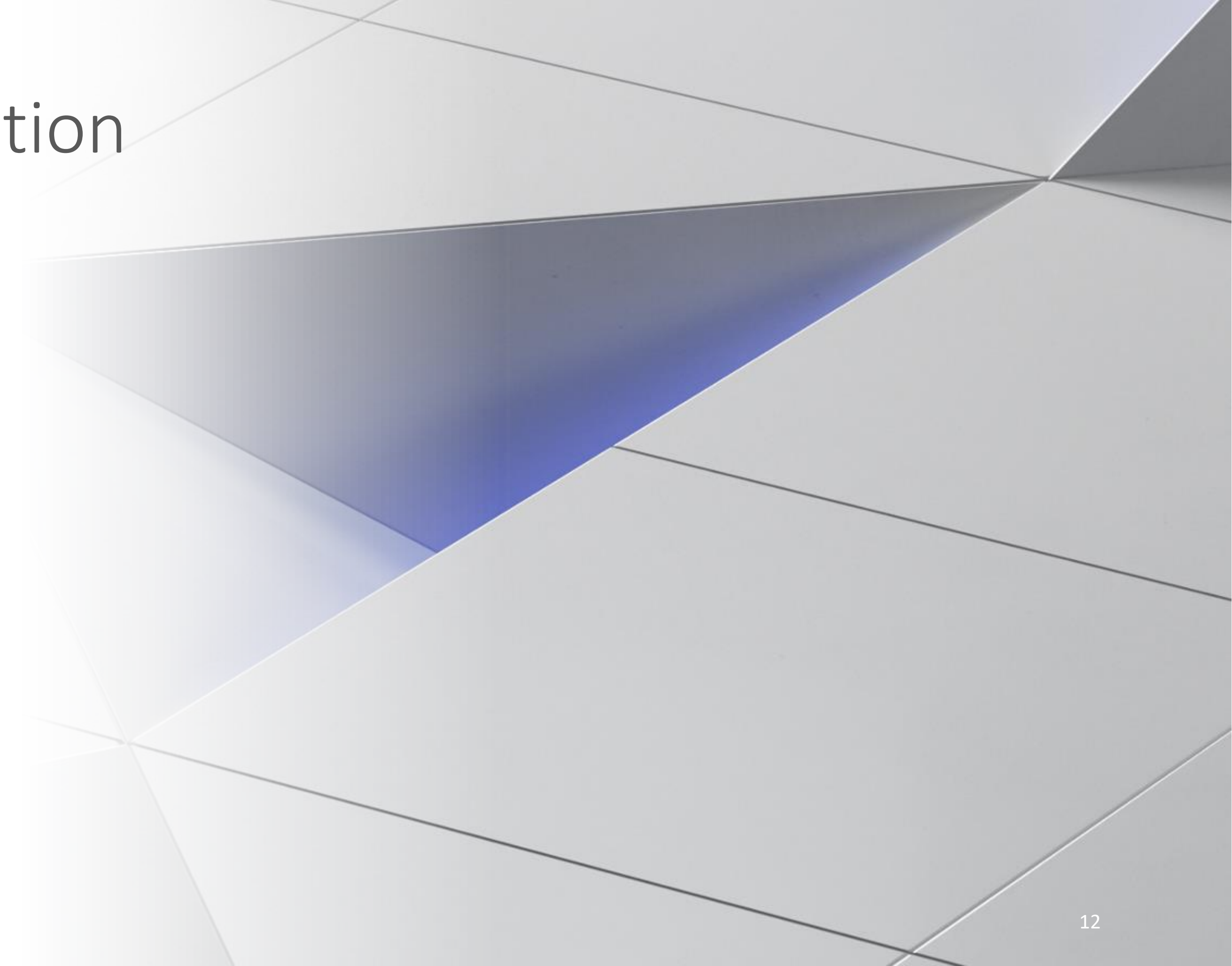
ML command

Statistics and zparms

Remote location and profile stats

Statement execution history

SQL Optimization Deep Dive



Db2ZAI
SQL Optimization

Leverage AI to improve access path selection

+

Automated access path monitoring

=

Reduced operation cost for SQL tuning

Leverage AI to improve SQL access path selection

Predict value of host variable / parameter markers

- Db2ZAI predicts likely literal values using models based on historical usage
- Helps optimizer estimate predicate selectivity (affects choice of index, join sequence, etc.)

```
WHERE C_CUSTKEY = O_CUSTKEY
AND L_ORDERKEY = O_ORDERKEY
AND C_NATIONKEY = N_NATIONKEY
AND O_ORDERDATE >= ?
AND O_ORDERDATE < ?
AND C_STATE = ?
GROUP BY … ORDER BY …
```

Actual number of rows fetched by the application

- SQL might qualify hundreds or thousands of rows
- Application only fetches a subset of the qualifying rows (one screen)
- Db2ZAI learns application behavior and adjusts access path selection appropriately
- Avoid sorting when only a few rows of the result set are needed

Parallelism exploitation

- Learns from prior execution cost and turn on the parallelism safely
- When estimated cost is low then sequential plan is chosen
- If actual cost is higher then optimizer learns from this and selects a parallelism plan (with either 2 or 4 degrees of parallelism)

Automated access path monitoring

Access path performance history

- For each SQL statement, Db2ZAI keeps track of each access path selected, and how each access path performs
    - PLAN_TABLE data (in internal format)
    - IFCID 58 stats (aggregated to the access path level)

Access path regression detection

- Db2ZAI automatically detects access path changes and compares performance metrics
- Detects access path improvement or regression

Access path regression resolution

- Resolve access path regressions by reverting to the previous access path
- Special “access path lock” interface allows single SQL statement within a package to be reverted while other statements in the package keep their current access path

SQL Optimization in Db2ZAI

SQL selection

[Static SQL]

- Manual package selection from UI by ADMIN
- Automatic package selection by a Db2ZAI daemon : every 24 hours, based on recent package activity referring to SYSIBM.SYSPACKAGE

[Dynamic SQL]

- Automatic SQL selection by a Db2ZAI daemon : top 5 SQL every 2 hours by default, based on execution frequency referring to IFCID 318

Data collection

- SQL performance statistics (equivalent to IFCID 58 record)
- Host variables / parameter markers used
- Statement text and Access path
- Collected into Db2ZAI managed Db2 tables by Db2ZAI daemons

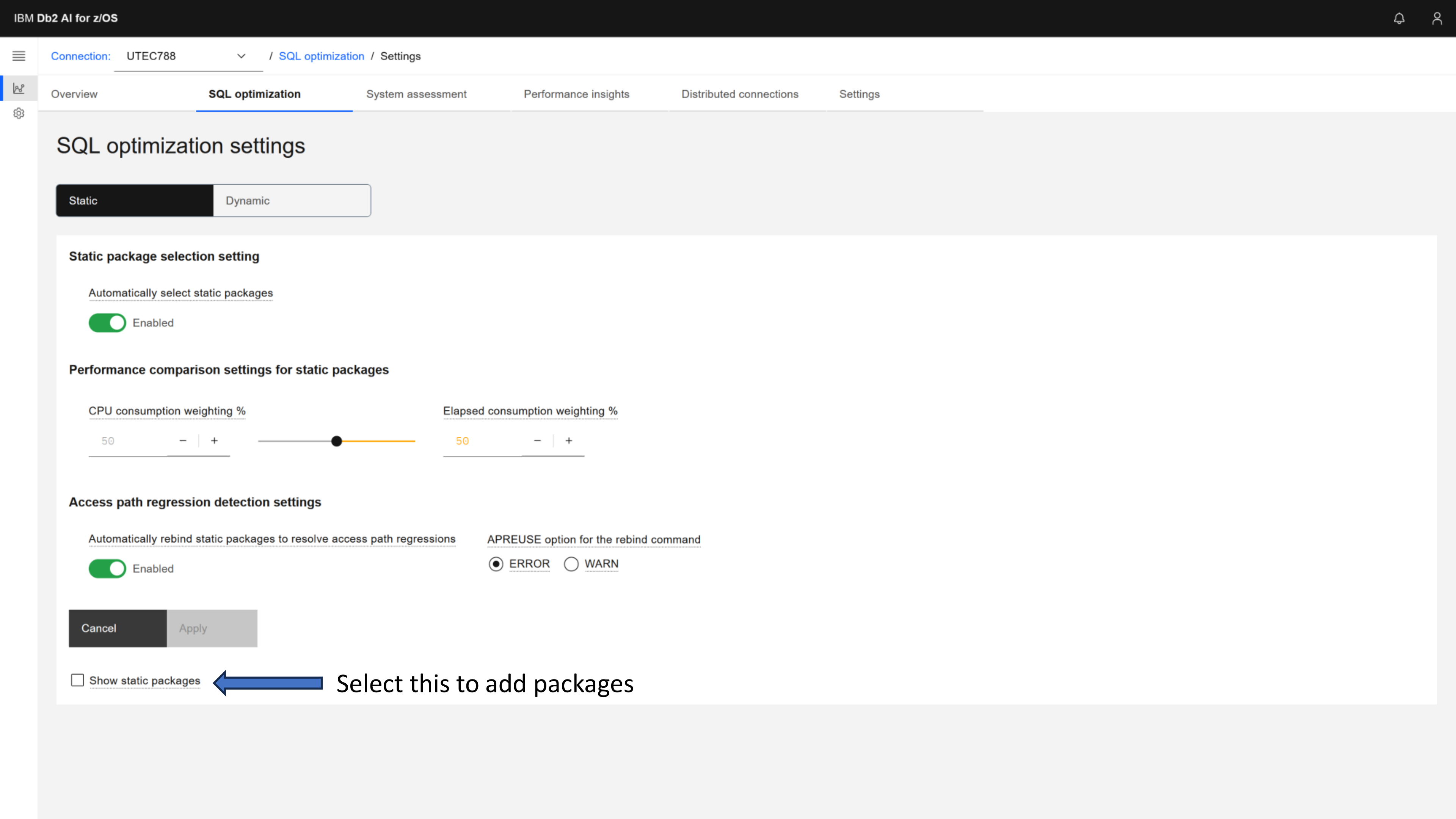
SQL Optimization in Db2ZAI

Access path performance comparison

- Customize cpu/elapsed weight (50/50 default)
- Can choose different weight for static vs dynamic

Access path regressions

- Option to automatically REBIND to revert to previous access path (static SQL)
- Choice of APREUSE(ERROR) or APREUSE(WARN)



SQL optimization settings

Static package selection setting

Automatically select static packages

Enabled

Performance comparison settings for static packages

CPU consumption weighting %: 50

Elapsed consumption weighting %: 50

Access path regression detection settings

Automatically rebind static packages to resolve access path regressions

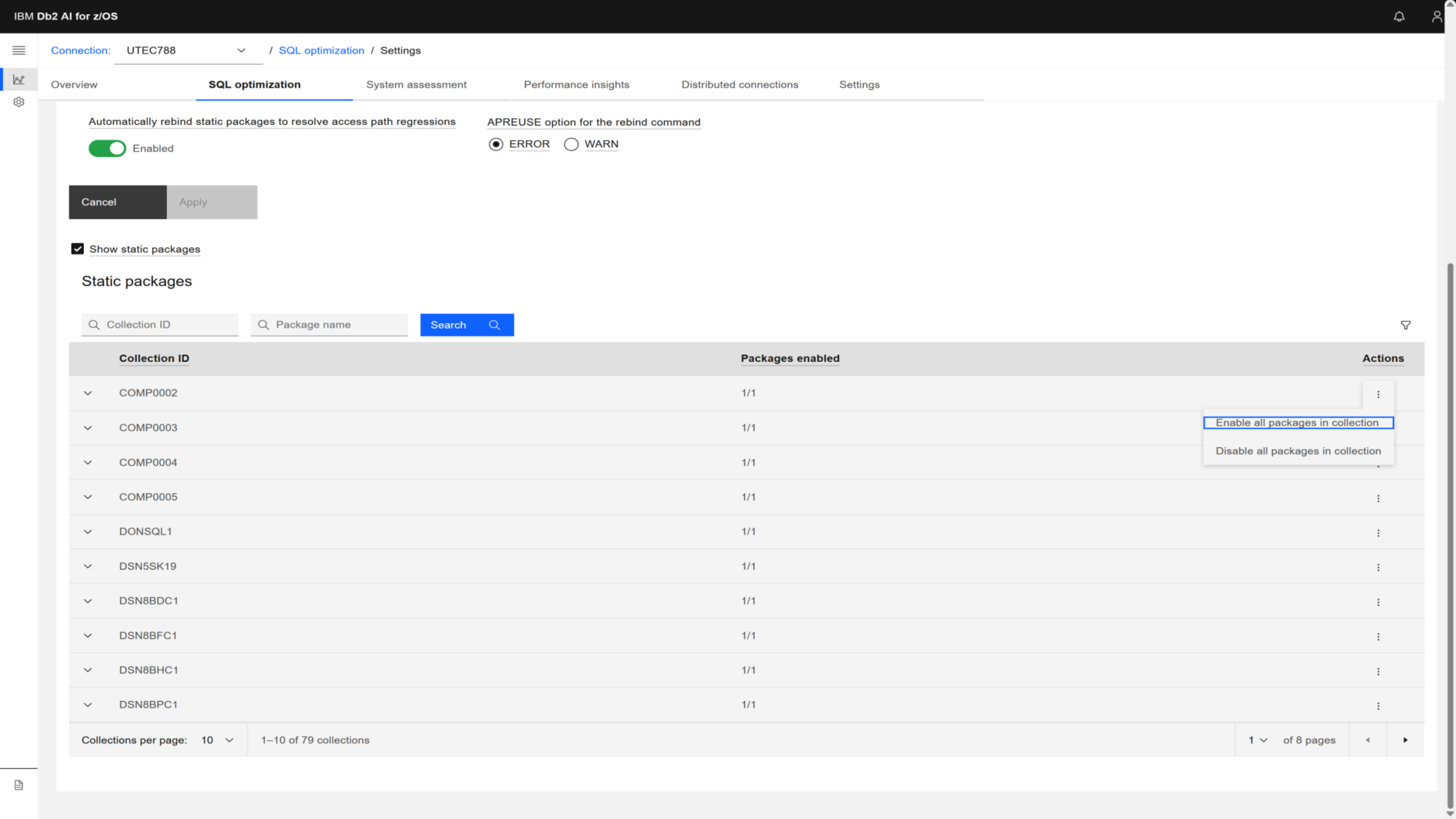
Enabled

APREUSE option for the rebind command

● ERROR ○ WARN

□ Show static packages

Select this to add packages



IBM Db2 AI for z/OS

Connection: UTEC788 / SQL optimization / Settings

Overview | **SQL optimization** | System assessment | Performance insights | Distributed connections | Settings

Automatically rebind static packages to resolve access path regressions

Enabled

APREUSE option for the rebind command

● ERROR ○ WARN

Cancel Apply

☑ Show static packages

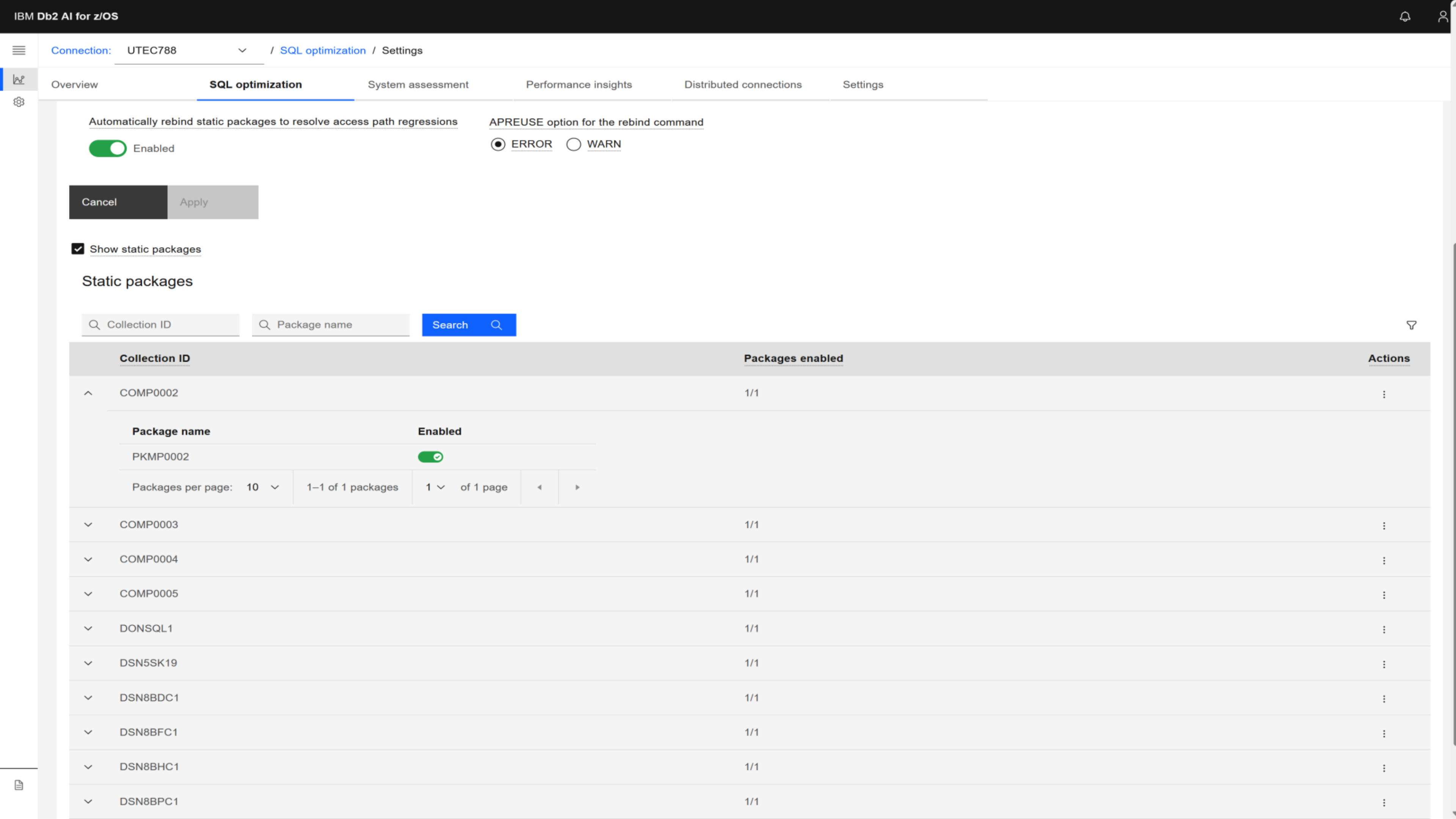
Static packages

Collection ID | Package name | Search

Collection ID	Packages enabled	Actions
COMP0002	1/1	⋮
COMP0003	1/1	⋮
COMP0004	1/1	⋮
COMP0005	1/1	⋮
DONSQL1	1/1	⋮
DSN5SK19	1/1	⋮
DSN8BDC1	1/1	⋮
DSN8BFC1	1/1	⋮
DSN8BHC1	1/1	⋮
DSN8BPC1	1/1	⋮

Enable all packages in collection
Disable all packages in collection

Collections per page: 10 | 1–10 of 79 collections | 1 of 8 pages



IBM Db2 AI for z/OS

Connection: UTEC788 / SQL optimization / Settings

Automatically rebind static packages to resolve access path regressions

Enabled

APREUSE option for the rebind command

● ERROR ○ WARN

☑ Show static packages

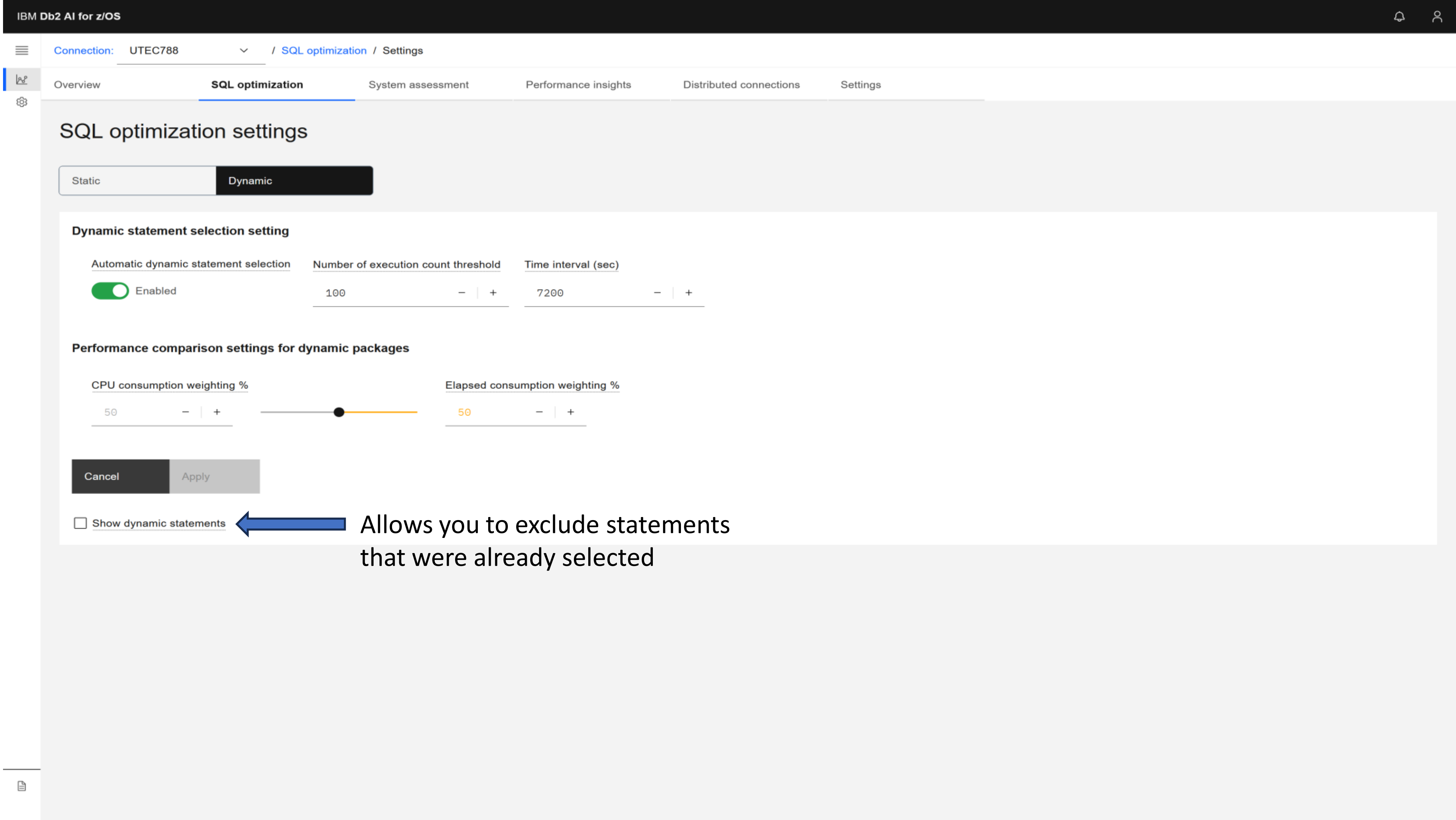
Static packages

Collection ID	Packages enabled	Actions
COMP0002	1/1	⋮
COMP0003	1/1	⋮
COMP0004	1/1	⋮
COMP0005	1/1	⋮
DONSQL1	1/1	⋮
DSN5SK19	1/1	⋮
DSN8BDC1	1/1	⋮
DSN8BFC1	1/1	⋮
DSN8BHC1	1/1	⋮
DSN8BPC1	1/1	⋮

COMP0002:

Package name	Enabled
PKMP0002	✓

Packages per page: 10 | 1–1 of 1 packages | 1 of 1 page



SQL optimization settings

Dynamic statement selection setting

Automatic dynamic statement selection: Enabled

Number of execution count threshold: 100

Time interval (sec): 7200

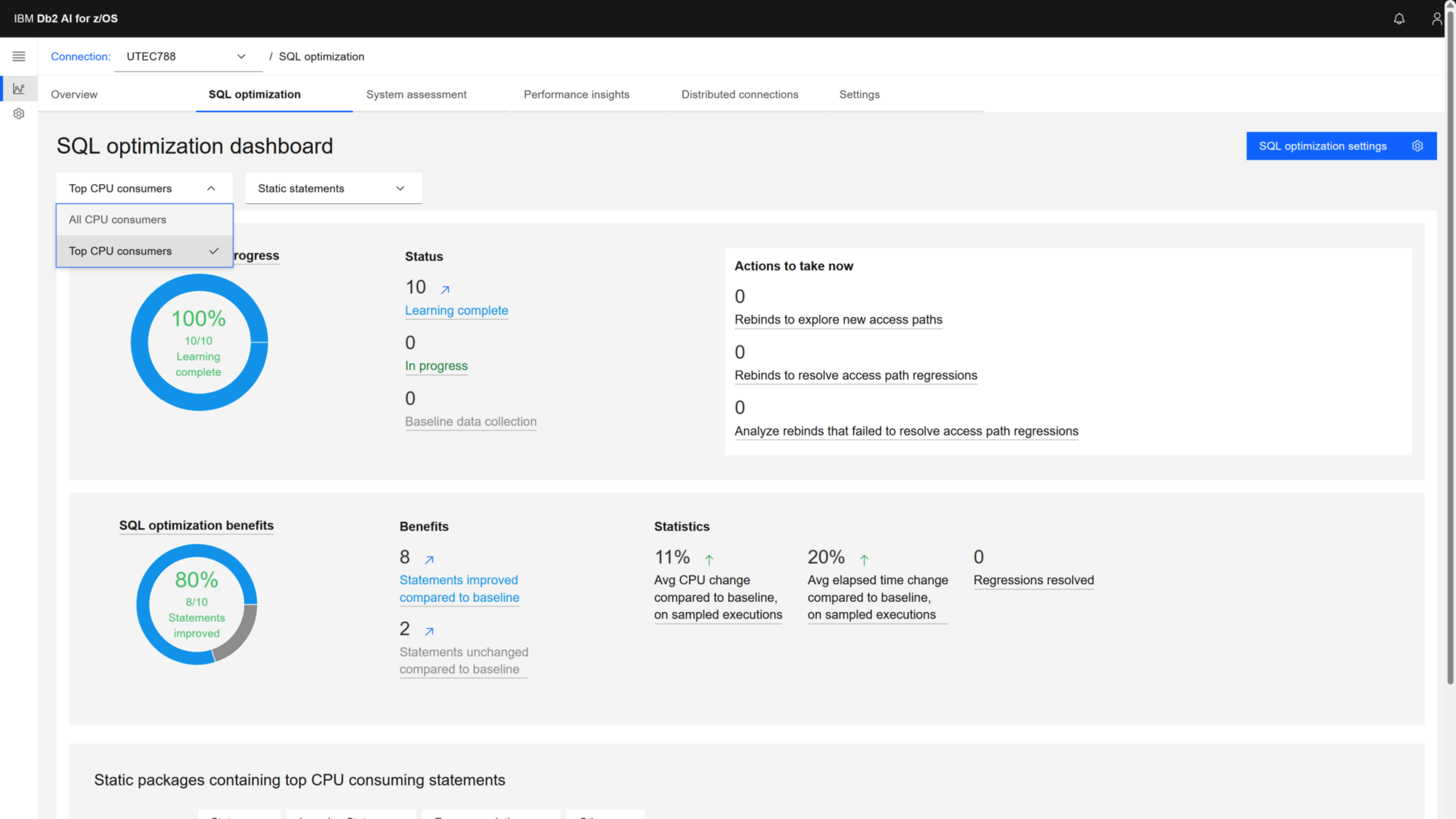
Performance comparison settings for dynamic packages

CPU consumption weighting %: 50

Elapsed consumption weighting %: 50

☐ Show dynamic statements

Allows you to exclude statements that were already selected



SQL optimization dashboard

…rogress

100%
10/10
Learning complete

Status

10 ↗
Learning complete

0
In progress

0
Baseline data collection

Actions to take now

0
Rebinds to explore new access paths

0
Rebinds to resolve access path regressions

0
Analyze rebinds that failed to resolve access path regressions

SQL optimization benefits

80%
8/10
Statements improved

Benefits

8 ↗
Statements improved compared to baseline

2 ↗
Statements unchanged compared to baseline

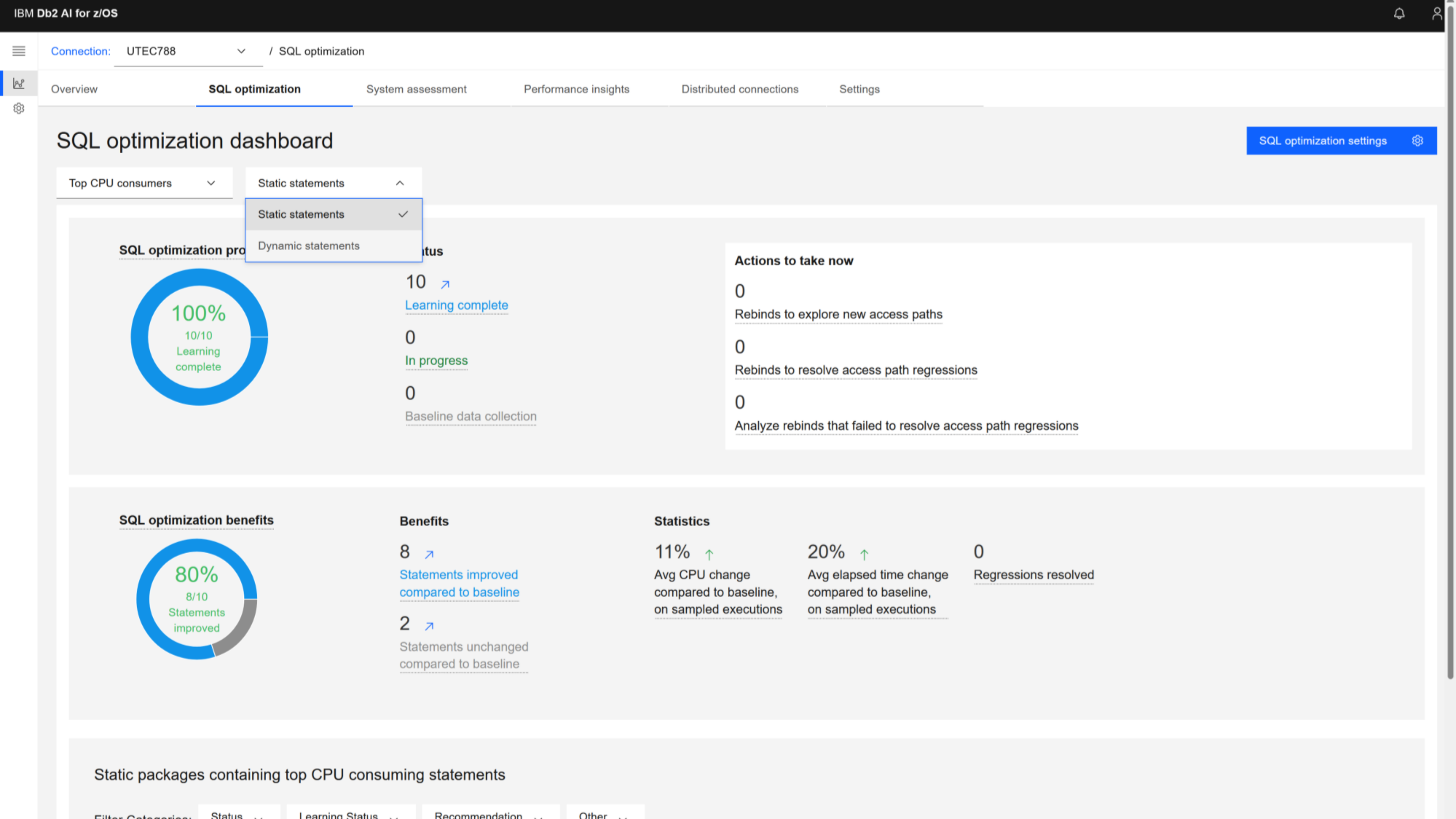
Statistics

11% ↑
Avg CPU change compared to baseline, on sampled executions

20% ↑
Avg elapsed time change compared to baseline, on sampled executions

0
Regressions resolved

Static packages containing top CPU consuming statements



SQL optimization dashboard

Connection: UTEC788

Top CPU consumers | Static statements

SQL optimization pro

100%
10/10
Learning
complete

tus

10 ↗
Learning complete

0
In progress

0
Baseline data collection

Actions to take now

0
Rebinds to explore new access paths

0
Rebinds to resolve access path regressions

0
Analyze rebinds that failed to resolve access path regressions

SQL optimization benefits

80%
8/10
Statements
improved

Benefits

8 ↗
Statements improved compared to baseline

2 ↗
Statements unchanged compared to baseline

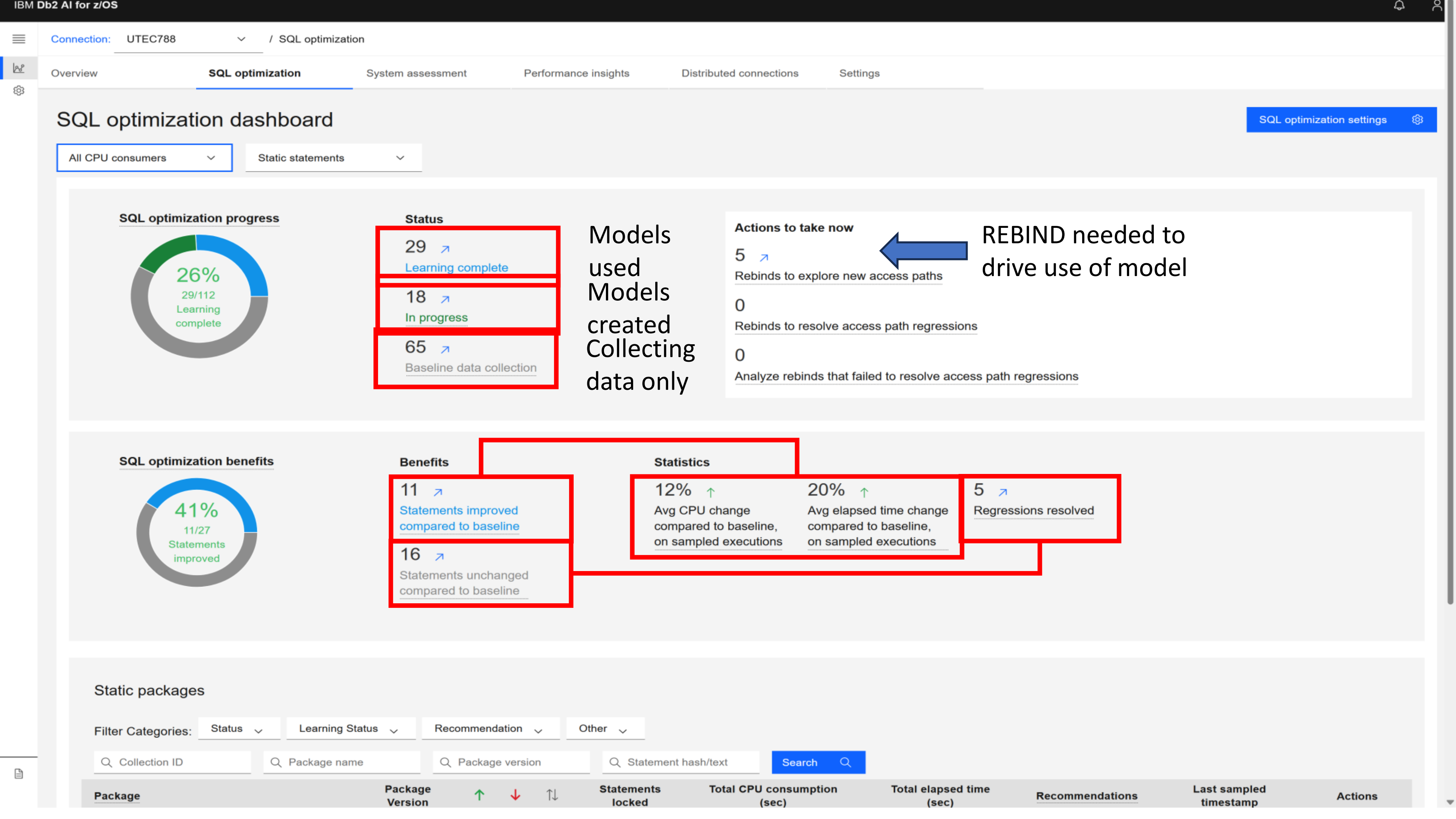
Statistics

11% ↑
Avg CPU change compared to baseline, on sampled executions

20% ↑
Avg elapsed time change compared to baseline, on sampled executions

0
Regressions resolved

Static packages containing top CPU consuming statements



IBM Db2 AI for z/OS

Connection: UTEC788 / SQL optimization

Overview | **SQL optimization** | System assessment | Performance insights | Distributed connections | Settings

SQL optimization dashboard

SQL optimization settings

All CPU consumers

Static statements

SQL optimization progress

Status

29 ↗
Learning complete

18 ↗
In progress

65 ↗
Baseline data collection

Models used

Models created

Collecting data only

Actions to take now

5 ↗
Rebinds to explore new access paths

REBIND needed to drive use of model

0
Rebinds to resolve access path regressions

0
Analyze rebinds that failed to resolve access path regressions

SQL optimization benefits

Benefits

11 ↗
Statements improved compared to baseline

16 ↗
Statements unchanged compared to baseline

Statistics

12% ↑
Avg CPU change compared to baseline, on sampled executions

20% ↑
Avg elapsed time change compared to baseline, on sampled executions

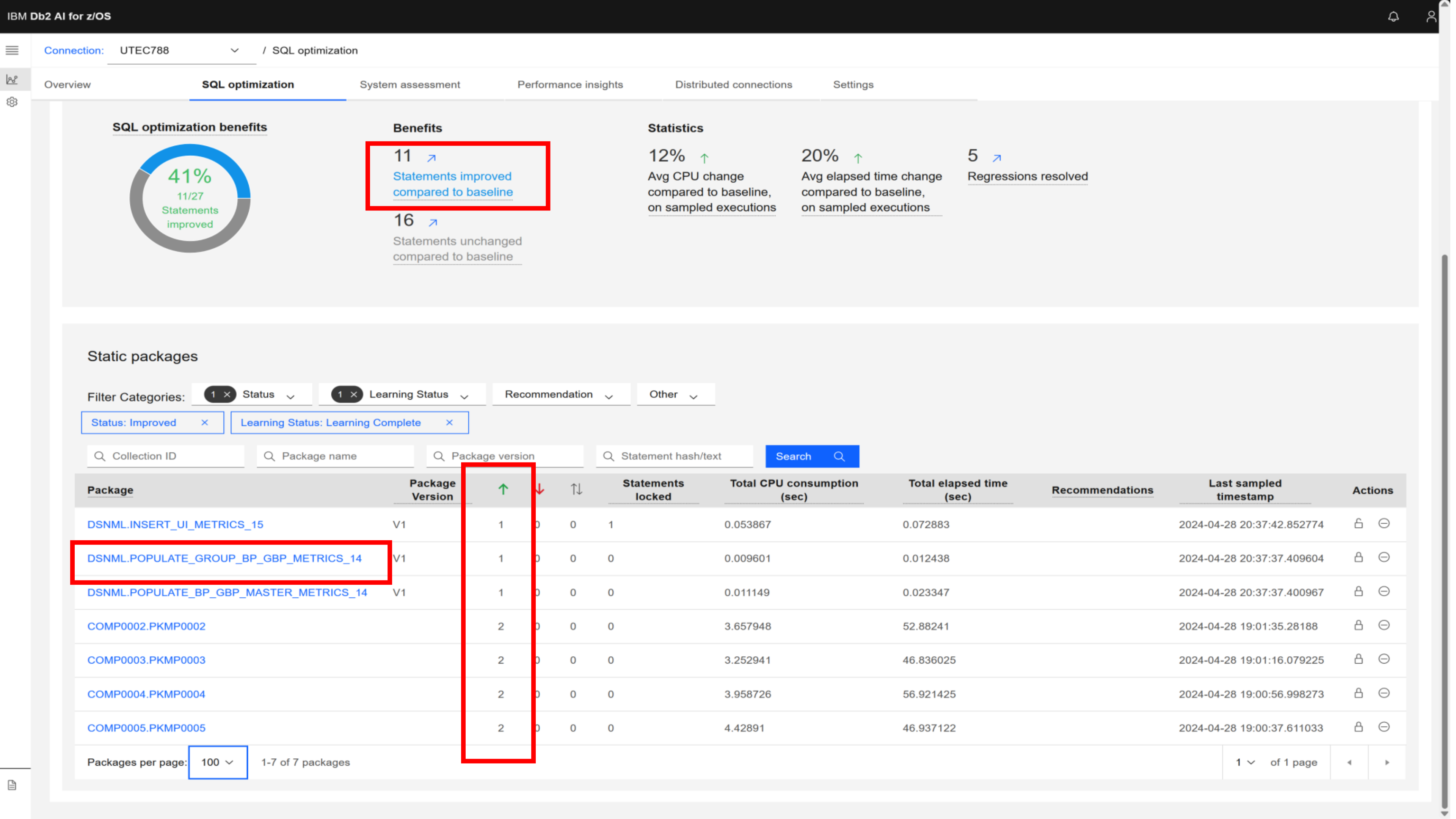
5 ↗
Regressions resolved

Static packages

Filter Categories: Status | Learning Status | Recommendation | Other

Collection ID | Package name | Package version | Statement hash/text | Search

Package	Package Version	↑ ↓ ⇅	Statements locked	Total CPU consumption (sec)	Total elapsed time (sec)	Recommendations	Last sampled timestamp	Actions



SQL optimization benefits

Benefits

11 ↗ Statements improved compared to baseline

16 ↗ Statements unchanged compared to baseline

Statistics

12% ↑ Avg CPU change compared to baseline, on sampled executions

20% ↑ Avg elapsed time change compared to baseline, on sampled executions

5 ↗ Regressions resolved

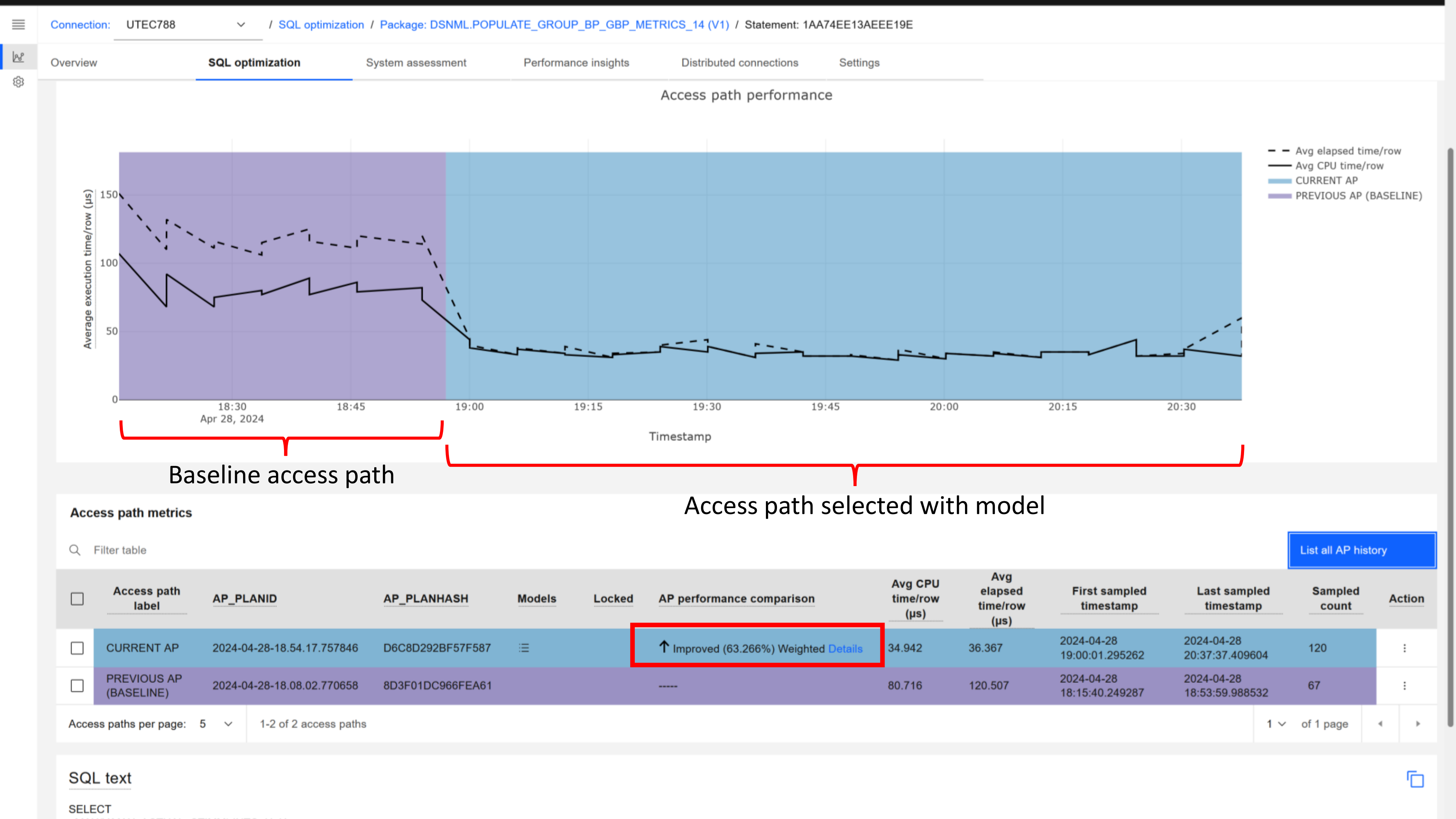
Static packages

Filter Categories: 1 × Status | 1 × Learning Status | Recommendation | Other

Status: Improved × | Learning Status: Learning Complete ×

Package	Package Version	↑	↓	⇅	Statements locked	Total CPU consumption (sec)	Total elapsed time (sec)	Recommendations	Last sampled timestamp	Actions
DSNML.INSERT_UI_METRICS_15	V1	1	0	0	1	0.053867	0.072883		2024-04-28 20:37:42.852774	🔒 ⊖
DSNML.POPULATE_GROUP_BP_GBP_METRICS_14	V1	1	0	0	0	0.009601	0.012438		2024-04-28 20:37:37.409604	🔒 ⊖
DSNML.POPULATE_BP_GBP_MASTER_METRICS_14	V1	1	0	0	0	0.011149	0.023347		2024-04-28 20:37:37.400967	🔒 ⊖
COMP0002.PKMP0002		2	0	0	0	3.657948	52.88241		2024-04-28 19:01:35.28188	🔒 ⊖
COMP0003.PKMP0003		2	0	0	0	3.252941	46.836025		2024-04-28 19:01:16.079225	🔒 ⊖
COMP0004.PKMP0004		2	0	0	0	3.958726	56.921425		2024-04-28 19:00:56.998273	🔒 ⊖
COMP0005.PKMP0005		2	0	0	0	4.42891	46.937122		2024-04-28 19:00:37.611033	🔒 ⊖

Packages per page: 100 | 1-7 of 7 packages | 1 of 1 page



Connection: UTEC788 / SQL optimization / Package: DSNML.POPULATE_GROUP_BP_GBP_METRICS_14 (V1) / Statement: 1AA74EE13AEEE19E

Baseline access path

Access path selected with model

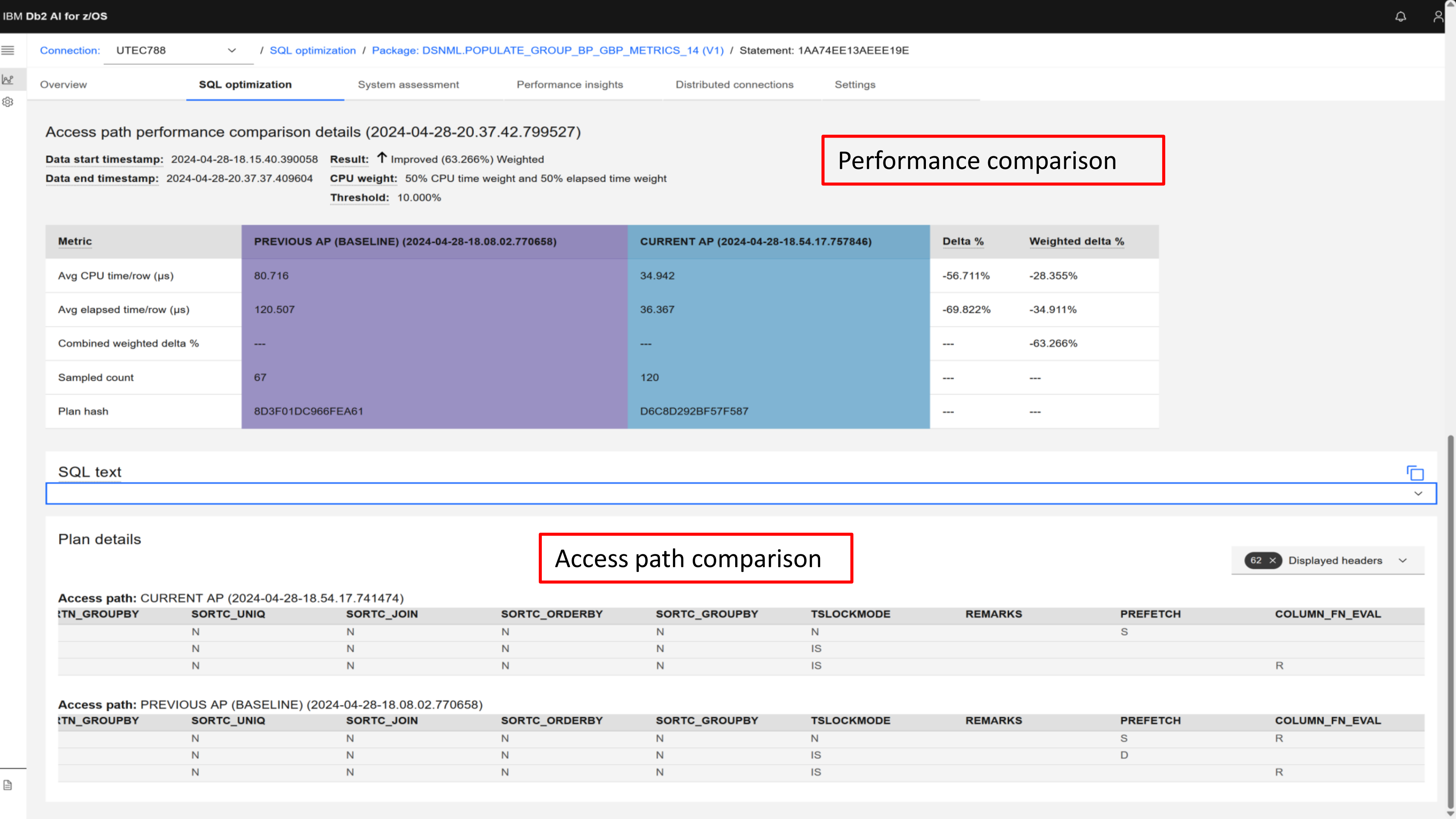
Access path metrics

☐	Access path label	AP_PLANID	AP_PLANHASH	Models	Locked	AP performance comparison	Avg CPU time/row (µs)	Avg elapsed time/row (µs)	First sampled timestamp	Last sampled timestamp	Sampled count	Action
☐	CURRENT AP	2024-04-28-18.54.17.757846	D6C8D292BF57F587	☰		↑ Improved (63.266%) Weighted Details	34.942	36.367	2024-04-28 19:00:01.295262	2024-04-28 20:37:37.409604	120	⋮
☐	PREVIOUS AP (BASELINE)	2024-04-28-18.08.02.770658	8D3F01DC966FEA61			-----	80.716	120.507	2024-04-28 18:15:40.249287	2024-04-28 18:53:59.988532	67	⋮

Access paths per page: 5 | 1-2 of 2 access paths | 1 of 1 page

SQL text

SELECT



Connection: UTEC788 / SQL optimization / Package: DSNML.POPULATE_GROUP_BP_GBP_METRICS_14 (V1) / Statement: 1AA74EE13AEEE19E

Access path performance comparison details (2024-04-28-20.37.42.799527)

Data start timestamp: 2024-04-28-18.15.40.390058

Data end timestamp: 2024-04-28-20.37.37.409604

Result: ↑ Improved (63.266%) Weighted

CPU weight: 50% CPU time weight and 50% elapsed time weight

Threshold: 10.000%

Performance comparison

Metric	PREVIOUS AP (BASELINE) (2024-04-28-18.08.02.770658)	CURRENT AP (2024-04-28-18.54.17.757846)	Delta %	Weighted delta %
Avg CPU time/row (µs)	80.716	34.942	-56.711%	-28.355%
Avg elapsed time/row (µs)	120.507	36.367	-69.822%	-34.911%
Combined weighted delta %	---	---	---	-63.266%
Sampled count	67	120	---	---
Plan hash	8D3F01DC966FEA61	D6C8D292BF57F587	---	---

SQL text

Plan details

Access path comparison

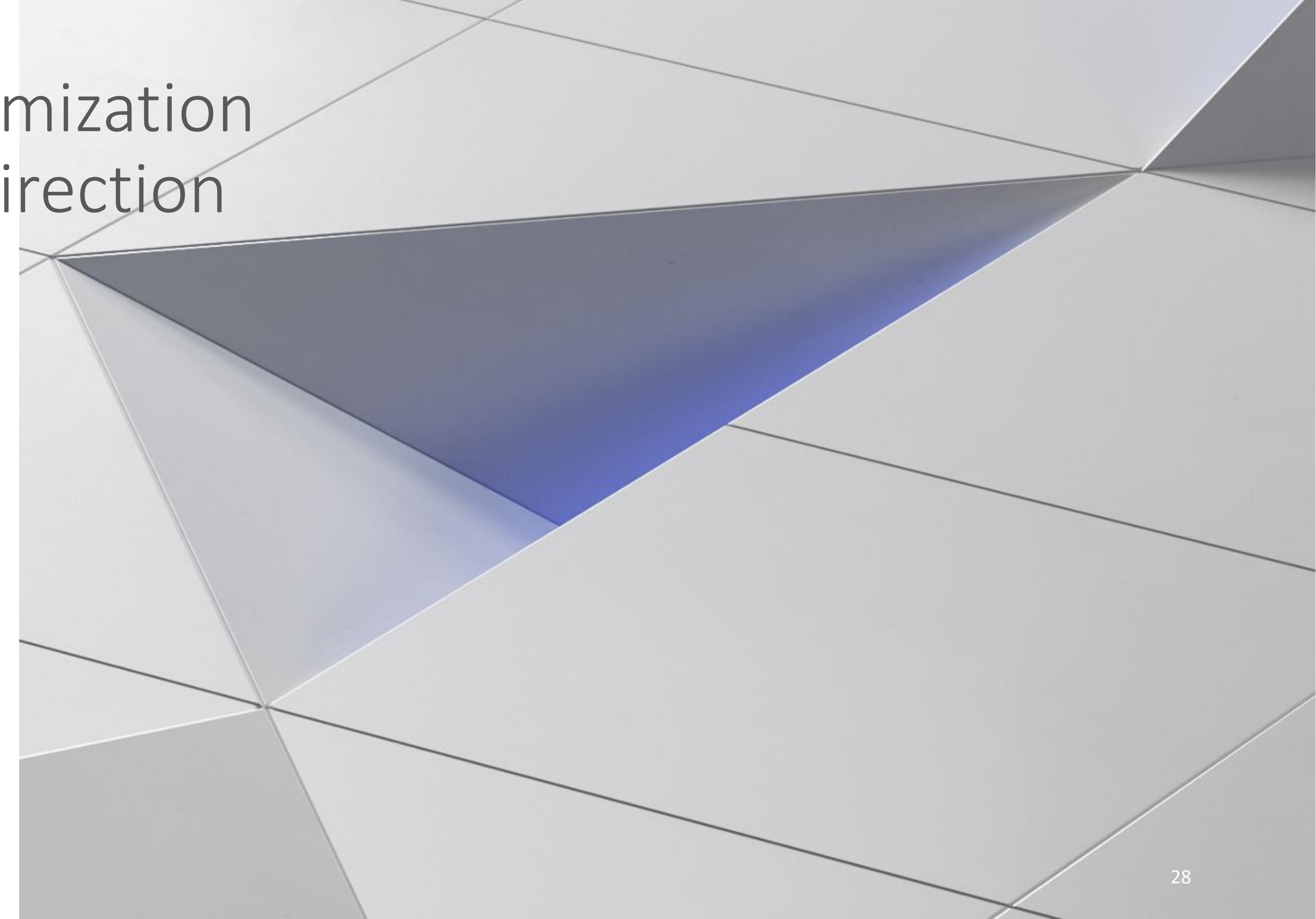
Access path: CURRENT AP (2024-04-28-18.54.17.741474)

RTN_GROUPBY	SORTC_UNIQ	SORTC_JOIN	SORTC_ORDERBY	SORTC_GROUPBY	TSLOCKMODE	REMARKS	PREFETCH	COLUMN_FN_EVAL
	N	N	N	N	N		S	
	N	N	N	N	IS			
	N	N	N	N	IS			R

Access path: PREVIOUS AP (BASELINE) (2024-04-28-18.08.02.770658)

RTN_GROUPBY	SORTC_UNIQ	SORTC_JOIN	SORTC_ORDERBY	SORTC_GROUPBY	TSLOCKMODE	REMARKS	PREFETCH	COLUMN_FN_EVAL
	N	N	N	N	N		S	R
	N	N	N	N	IS		D	
	N	N	N	N	IS			R

SQL Optimization
Future Direction



Db2ZAI SQL Optimization “Exploration”

As-is

1. Collect data
2. Create “models”
3. REBIND/FULL PREPARE
4. Collect data
5. Detect access path regression and revert to previous if needed (safety net)

Regressions happen on production queries

To-be

1. Collect data
2. “Explore”
   - Create “models”
   - EXPLAIN with VALIDATION
   - Recommend only improved access paths
3. REBIND/FULL PREPARE
4. Collect data
5. Detect access path regression and revert to previous if needed (safety net)

Regressions happen on EXPLAIN

Safety net still protects production queries

API's to allow integration/automation

Db2ZAI API	Description
/statements	Get a list of SQL statements
/explore	Get access path recommendations
/deploy	Deploy access path recommendations

EXPLAIN with VALIDATION

EXPLAIN FOR *sql-statement*

- Db2 Optimizer selects an access path for the given “*sql-statement*”

- Run the selected access path
- Performance stats from access path execution are collected

- Access path information, and collected performance stats, are written to EXPLAIN tables

Used by Db2ZAI to “validate” the optimizer’s access path choice

Requires Db2ZAI learned information

Questions?

IBM

Thank you!