October 22–25, 2018
SAN FRANCISCO, CA

#OOW18

ORACLE OPEN WORLD

Automatic Indexing in Oracle Database 19c

Dev: Sunil, Chris, Zhan
Presenters: Mohamed, Maria

oracle.com/openworld

ORACLE

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. 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, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

Agenda

1. **Original Approach to Index Tuning**
2. Automatic Indexing
   - How it work
   - What to expect
   - Deployments
   - Controls
   - Auditing and Report
3. Validation

Our Original Approach To Performance Features

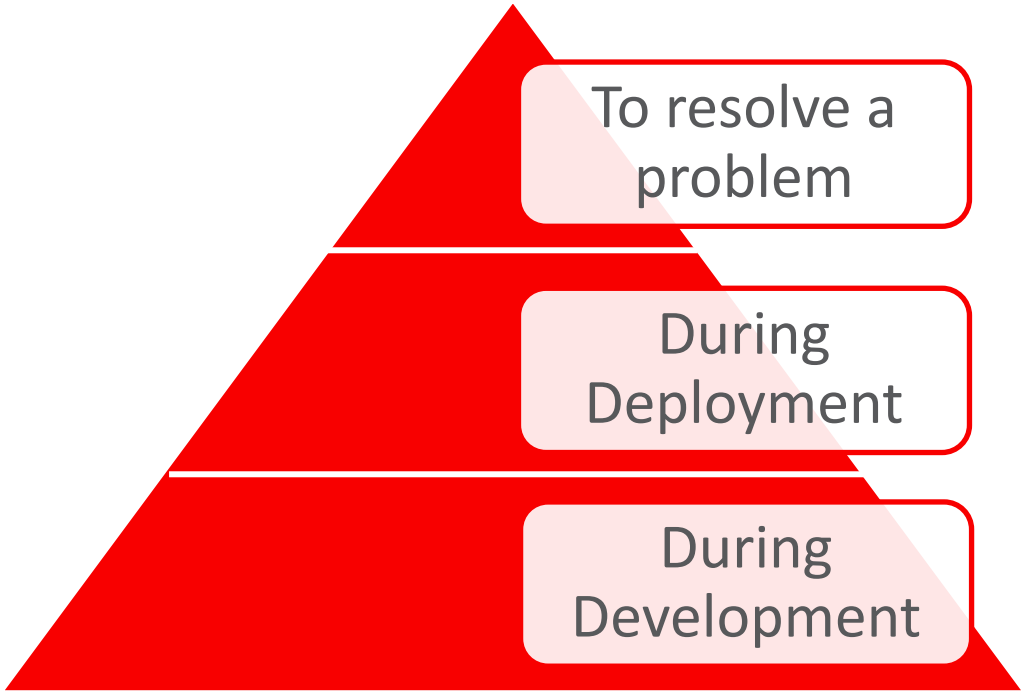
- Up until now we have developed performance features assuming there was an expert driver behind the wheel
- Some features were enabled by default but came with an emergency shut off switch
- We relied on the expert drivers (DBAs & Developers) to know when to use these switches to control a feature

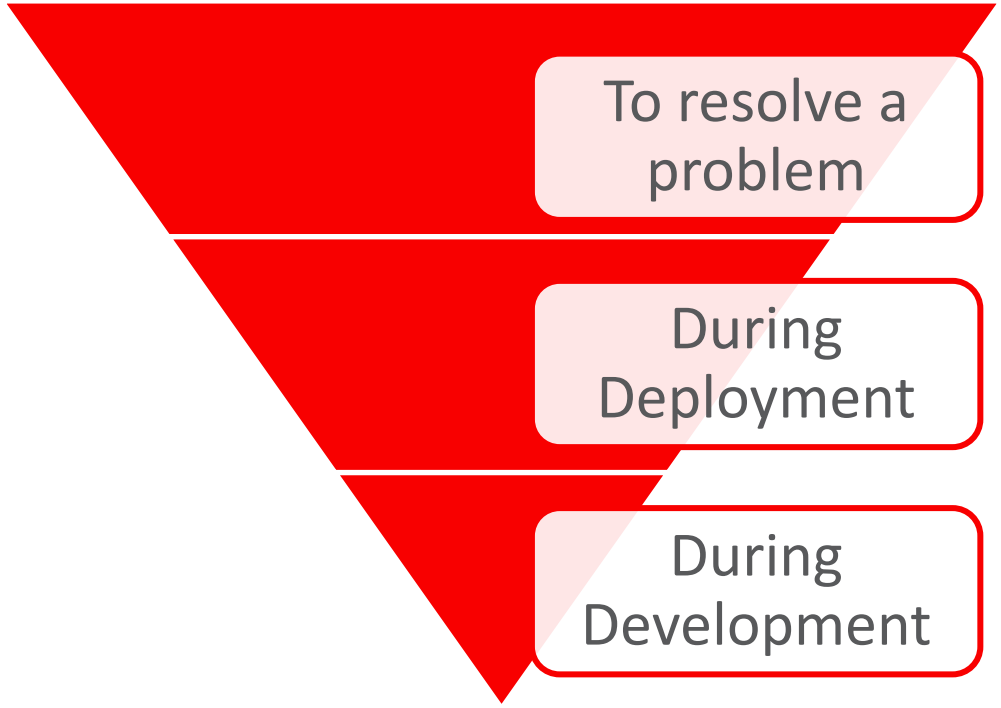
Our Original Approach To Oracle Performance Advisors

- We also provided advisors to assist DBAs in determining what performance structures maybe needed
  - Index Advisor, Partition Advisor, In-Memory Advisor, etc.
- Each advisor generated recommendations on what performance structures **might be** useful for a **given workload**
- The onus was on the DBA to:
  - Validate whether the performance structures actually helps
  - Manage the case of negative side effects, e.g. slower DML
  - Check whether the performance structures are still useful
  - Decide when to repeat the tuning process, e.g. change in the data or app

Traditional Approach To Index Creation

- Indexes are added at different stages of an applications life cycle





- Reality is a more reactive approach that replies on the expertise you have

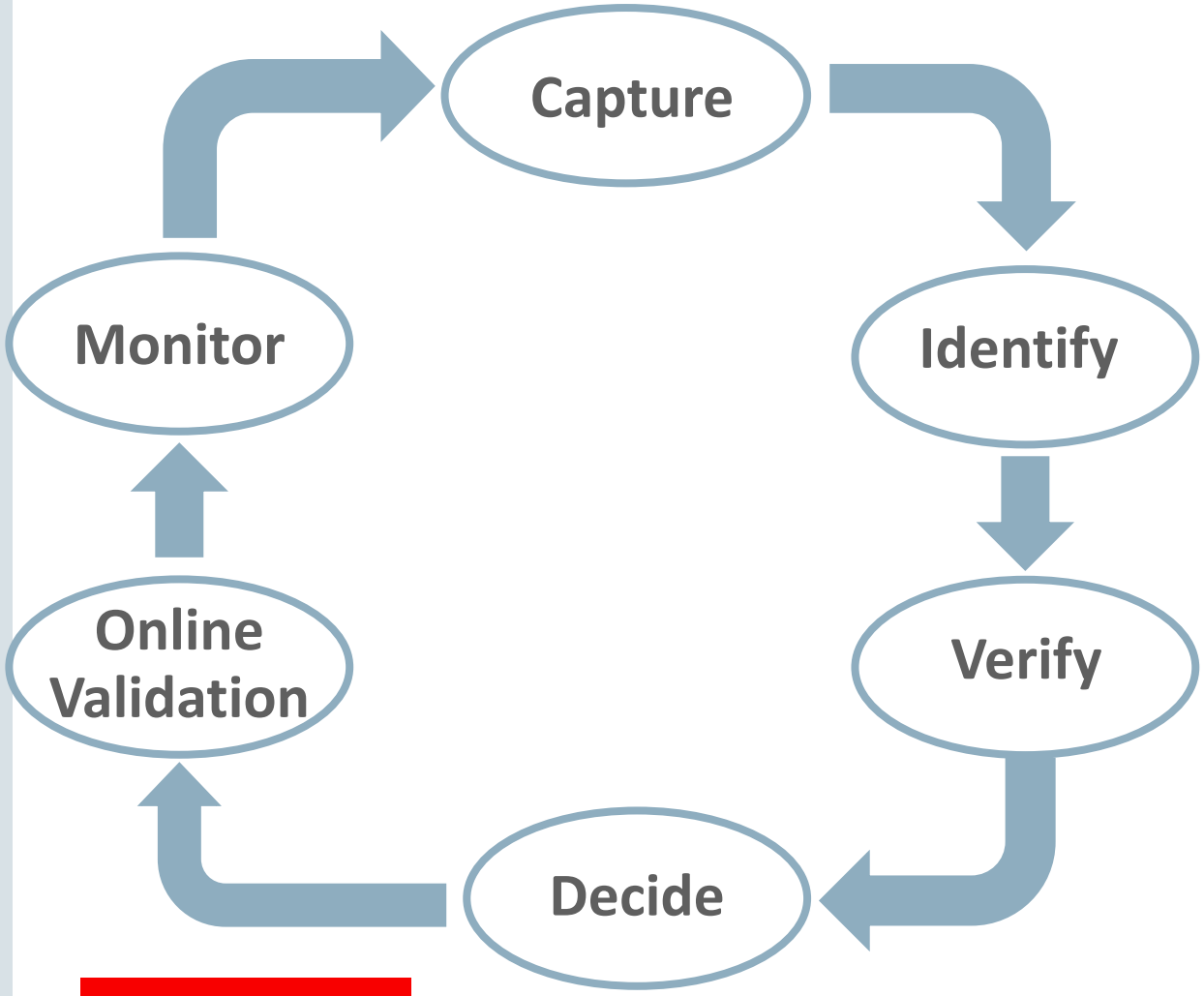
Agenda

1. **Original Approach to Index Tuning**
2. Automatic Indexing
   - How it work
   - What to expect
   - Deployments
   - Controls
   - Auditing and Report
3. Validation

Automatic Indexing – Overview

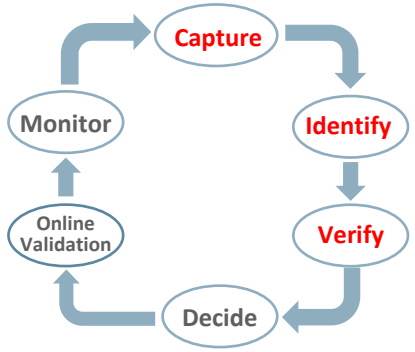
- An **expert system** that implements indexes based on what a performance engineer skilled in index tuning would do
  - But it is able to work 24 hours a day, 7 days a week, 52 weeks a year
- It is not an advisor
  - The DBA is not expected to provide any specific inputs (workload, etc.)
  - The DBA is not expected to take any action
- The interaction is limited to
  - Optional setting of **preferences**, e.g. where the indexes are stored etc.
  - Viewing a **report** on actions performed and their impact on the application
- It works in an **incremental** fashion and therefore it has to be **iterative** and **continuous**
- It takes **responsibility** for its decisions and hence its decisions must be **validated**
- It **adapts** to changes in the schema, data and application

Automatic Indexing Methodology



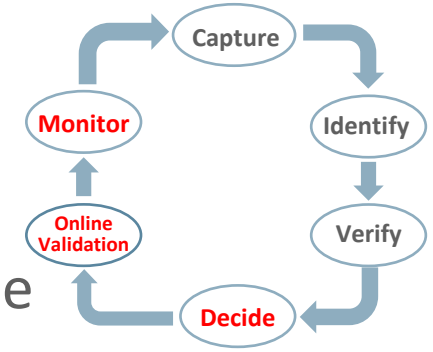
- The Automatic Indexing methodology is based on a common approach to manual SQL tuning
- It identifies candidate indexes and validates them before implementing
- The entire process is fully automatic
- Transparency is equally important as sophisticated automation
  - All tuning activities are auditable via reporting

Automatic Indexing – How It Works



1. Capture
   - Periodically capture the application SQL history into a SQL repository
   - Includes SQL, plans, bind values, execution statistics, etc.
2. Identify Candidates
   - Identify candidate indexes that may benefit the newly captured SQL statements
   - Creates index candidates as unusable, invisible indexes (***metadata*** only)
   - Drop indexes obsoleted by newly created indexes (logical merge)
3. Verify
   - Ask the optimizer if index candidates will be used for captured SQL statements
   - Materialize indexes and run SQL to validate that the indexes improve their performance
   - All verification is done outside application workflow

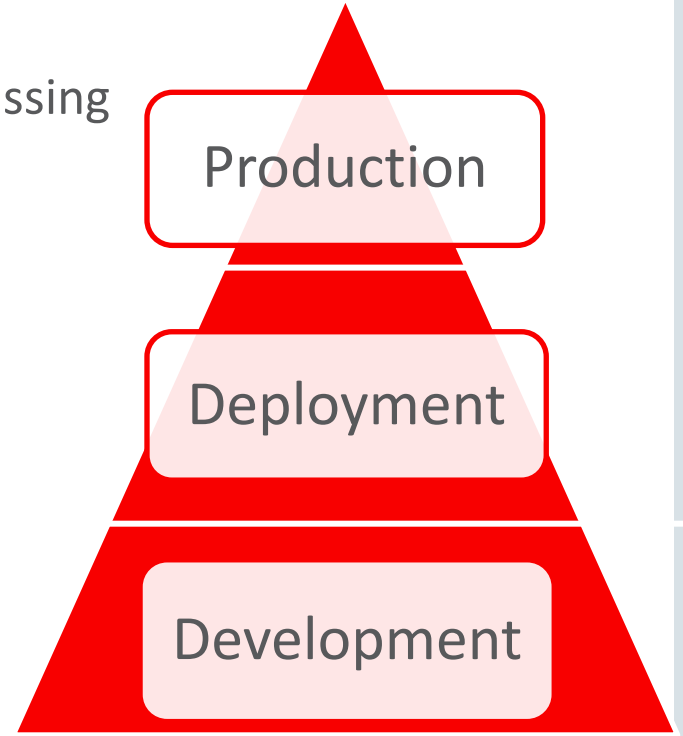
Automatic Indexing – How It Works



4. Decide
   - If performance is better for all statements, the indexes are marked visible
   - If performance is worse for all statements, the indexes remain invisible
   - If performance is worse for some, the indexes are marked visible except for the SQL statements that regressed
5. Online Validation
   - The validation of the new indexes continues for *other* statements, *online*
   - Only one of the sessions executing a SQL statement is allowed to use the new indexes
6. Monitor
   - Index usage is continuously monitored
   - Automatically created indexes that have not been used in a long time will be dropped

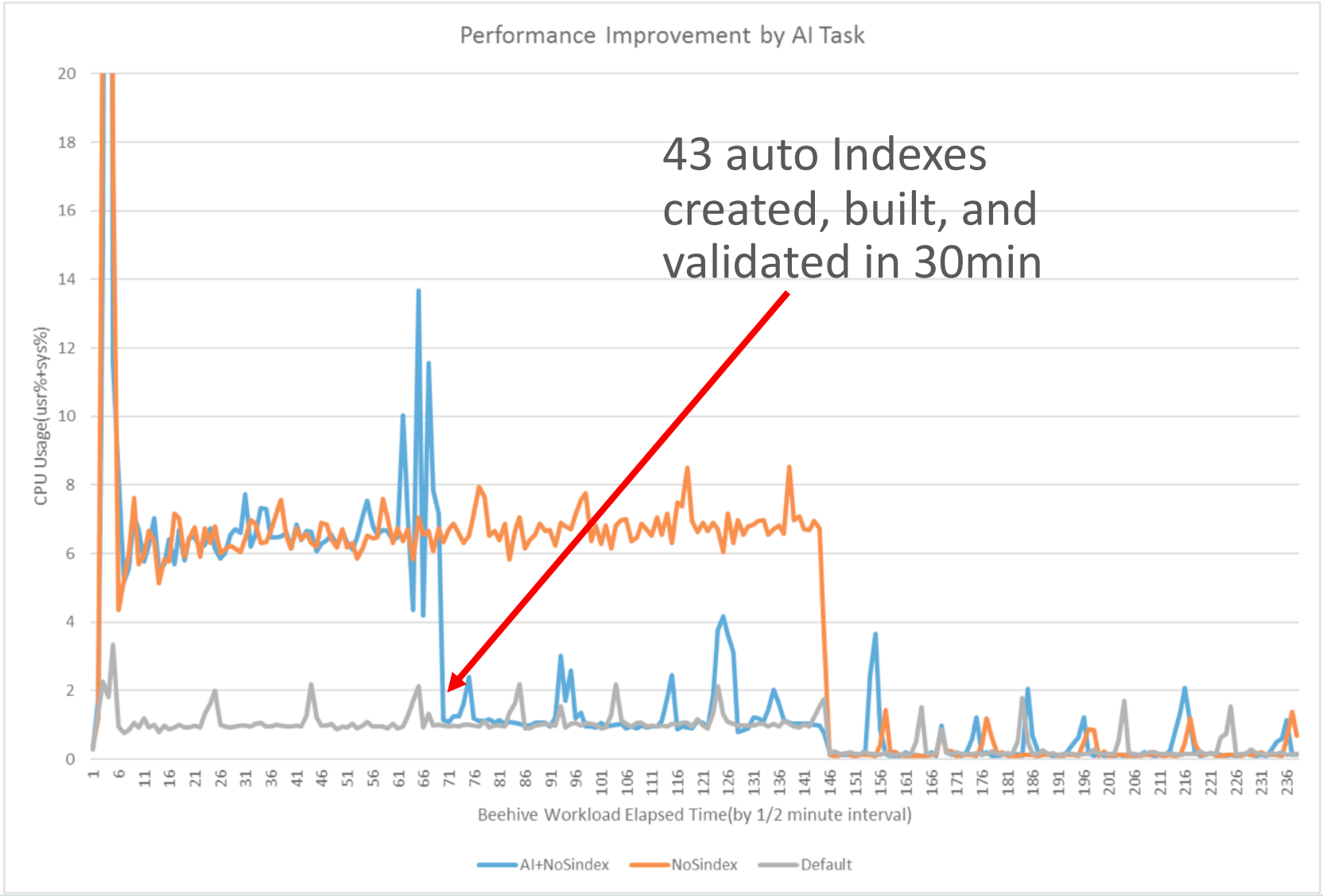
Automatic Indexing – Scope

- Useful for OLTP, DW, Mixed workloads but very ***critical*** for OLTP
- Applies to tuned and un-tuned applications
  - Tuned
    - Existing secondary indexes may be outdated or important ones can be missing
    - Some secondary indexes can be dropped and auto indexes can be added
  - Un-tuned
    - Existing indexes support primary or unique key constraints
- Applicable to all stages of an application lifecycle
- Supports
  - Single and concatenated indexes
  - Function-based indexes
  - Compression Advanced Low



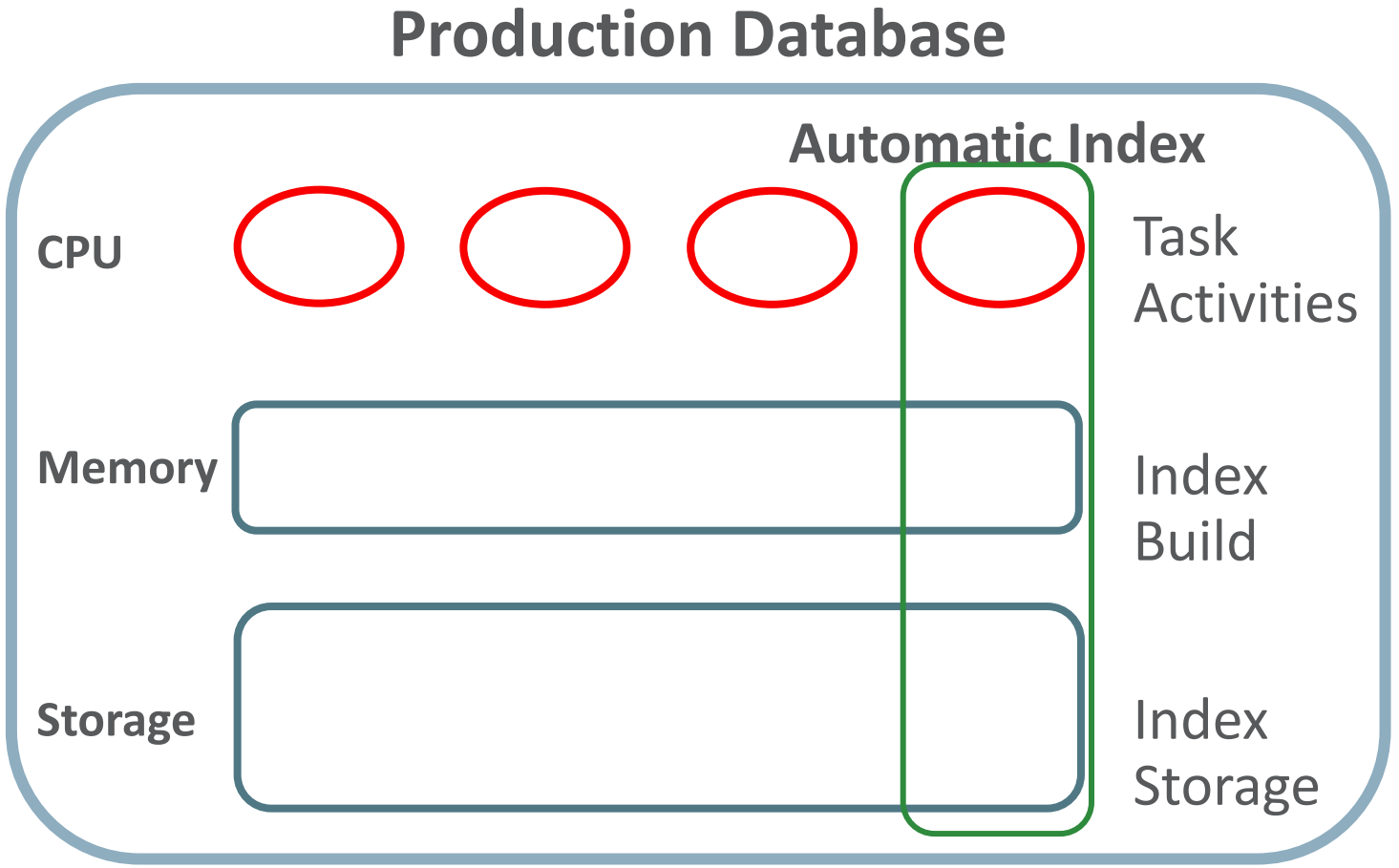
Automatic Indexing in Action (live)

Total run time 2 h
-10 min ramp-up
-60 min steady
-50 min ramp-down



Deployment – Inline Automatic Indexing

- By default Automatic Indexing runs in the same database as application



- The task does consume CPU Memory and storage
  - Resource manager plan limits the task to 1 CPU
  - Customers can control which temp tablespace is used to build indexes
  - Customers can control which tablespace and how much space can be used by Auto Indexing

New DBMS_AUTO_INDEX Package to Controls Behavior

- The following parameters control the behavior of Automatic Indexing:
  - AUTO_INDEX_EXCLUDE_SCHEMA
    specify schema name(s) to be excluded from creating indexes automatically
    By default, auto indexing consider tables in all user created schemas
  - AUTO_INDEX_RETENTION_FOR_AUTO
    Number of days the auto indexes are retained after last used date before purge (373 days)
  - AUTO_INDEX_RETENTION_FOR_MANUAL
    Number of days the manual indexes are retained after their last used date before purge (NULL)
  - AUTO_INDEX_DEFAULT_TABLESPACE
    The tablespace name where all auto indexes will be stored (user default tablespace)
  - AUTO_INDEX_MODE
    Put automatic Indexing into reporting mode (IMPLEMENT)
    Indexes are created, tested, and a report shows the impact without affecting the app

Auditing Automatic Indexing

- All activities are logged and can be viewed via new DBA views:
  - `DBA_AUTO_INDEX_EXECUTIONS` – Shows history automatic indexing tasks executions
  - `DBA_AUTO_INDEX_STATISTICS` - Shows statistics related to automatic indexes
  - `DBA_AUTO_INDEX_IND_ACTIONS` -Shows actions performed on automatic indexes
  - `DBA_AUTO_INDEX_SQL_ACTIONS` – Shows actions performed on SQL to verify automatic indexes
  - `DBA_AUTO_INDEX_CONFIG` - Shows history of configuration settings related to automatic indexes

Reporting on Automatic Indexing Activity

- Each Auto Index Task generates a report showing the task activities
- Reports can be generated via `DBMS_AUTO_INDEX.REPORT_ACTIVITY` function
  - Date/Time range
  - Format (XML, HTML, Text)
  - Level (basic, typical, all)
  - Section (Summary, Index Details, Verification Details, Errors, All)

Sample Report

```
------------------------------------------------------------------------------
GENERAL INFORMATION
------------------------------------------------------------------------------
Activity start                : 29-AUG-2018 12.20.40
Activity end                  : 30-AUG-2018 12.20.40
Executions completed          : 13
Executions interrupted        : 3
Executions with fatal error   : 1
------------------------------------------------------------------------------
SUMMARY (AUTO INDEXES)
------------------------------------------------------------------------------
Index candidates                                 : 53
Indexes created (visible / invisible)            : 12 (12 / 0)
Space used (visible / invisible)                 : 3.48 MB (3.48 MB / 0 B)
Indexes dropped                                  : 0
SQL statements verified                          : 16
SQL statements improved (improvement factor)     : 16 (3x)
SQL statements disallowed from auto indexes      : 0
Overall improvement factor                       : 3x
------------------------------------------------------------------------------
SUMMARY (MANUAL INDEXES)
------------------------------------------------------------------------------
Unused indexes (visible / invisible)  : 10 (8 / 2)
Space used (visible / invisible)      : 100 MB (76 MB / 24 MB)
Unusable indexes                      : 0
------------------------------------------------------------------------------
```

INDEX DETAILS

1. The following indexes were created:*: invisible

Owner	Table	Index	Key	Type	Properties
OPT	T_10K_CP1	SYS_AI_3cpm0ahgt469g	ROWID_UNIQUE	B-TREE	NONE
OPT	T_10K_CP1	SYS_AI_3rk4h2m9d49b5	CHAR_UNIQUE	B-TREE	NONE
OPT	T_10K_CP1	SYS_AI_5cq2h6jhmznc9	DATE_UNIQUE	B-TREE	NONE
OPT	T_10K_CP1	SYS_AI_6vg5wr5nwcqxs	THOUSAND	B-TREE	NONE
OPT	T_10K_CP1	SYS_AI_agnvzczmz4z0a	TEN,UNIQUE1,UNIQUE2	B-TREE	NONE
OPT	T_10K_CP1	SYS_AI_bcms9qy98nq1c	VCHAR_UNIQUE	B-TREE	NONE
OPT	T_5K_CP	SYS_AI_0urcv8chmxu20	VCHAR_UNIQUE	B-TREE	NONE
OPT	T_5K_CP	SYS_AI_2pvk34mqdh7pa	TEN,UNIQUE1,UNIQUE2	B-TREE	NONE
OPT	T_5K_CP	SYS_AI_428hqd6qu531y	THOUSAND	B-TREE	NONE
OPT	T_5K_CP	SYS_AI_5d2cukrm2gju2	DATE_UNIQUE	B-TREE	NONE
OPT	T_5K_CP	SYS_AI_97zrtmcmn5tz6	CHAR_UNIQUE	B-TREE	NONE
OPT	T_5K_CP	SYS_AI_cn9fsv12paxcb	ROWID_UNIQUE	B-TREE	NONE

```
-------------------------------------------------------------------------------
VERIFICATION DETAILS
-------------------------------------------------------------------------------
1. The performance of the following statements improved:-------------------------------------------------------------------
-------------------------------------------------------------------------------------
Schema Name         : OPT
SQL ID              : 2vy3tr5kyg88z
SQL Text            : select count(*) from t_5k_cp where vchar_unique ='MAN'
Improvement Factor  : 2x

PLANS SECTION
-------------------------------------------------------------------------------
Original
------------------------------
Plan Hash Value  : 3944640934
-----------------------------------------------------------------
| Id | Operation            | Name    | Rows | Bytes | Cost | Time |
-----------------------------------------------------------------
|  0 | SELECT STATEMENT     |         |      |       |      |      |
|  1 |   SORT AGGREGATE     |         |      |       |      |      |
|  2 |    TABLE ACCESS FULL | T_5K_CP |      |       |      |      |
-----------------------------------------------------------------
With Auto Indexes
------------------------------
Plan Hash Value  : 2541075899
-------------------------------------------------------------------------------
| Id  | Operation           | Name                 | Rows | Bytes | Cost | Time|
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT    |                      |      |       |      |     |
|   1 |   SORT AGGREGATE    |                      |      |       |      |     |
| * 2 |    INDEX RANGE SCAN | SYS_AI_0urcv8chmxu20 |      |       |      |     |
-------------------------------------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------
* 2 - access("VCHAR_UNIQUE"='MAN')
```

Using Automatic Indexing Hints

- You can use hints to control if auto indexes will be used for a SQL statements
- The `USE_AUTO_INDEXES` hint instructs the optimizer to use auto indexes

```
SELECT /*+ USE_AUTO_INDEXES */ emp_id, emp_name, dept_id
FROM employees
WHERE dept_id > 50;
```

- The `NO_USE_AUTO_INDEXES` hint instructs the optimizer not to use auto indexes

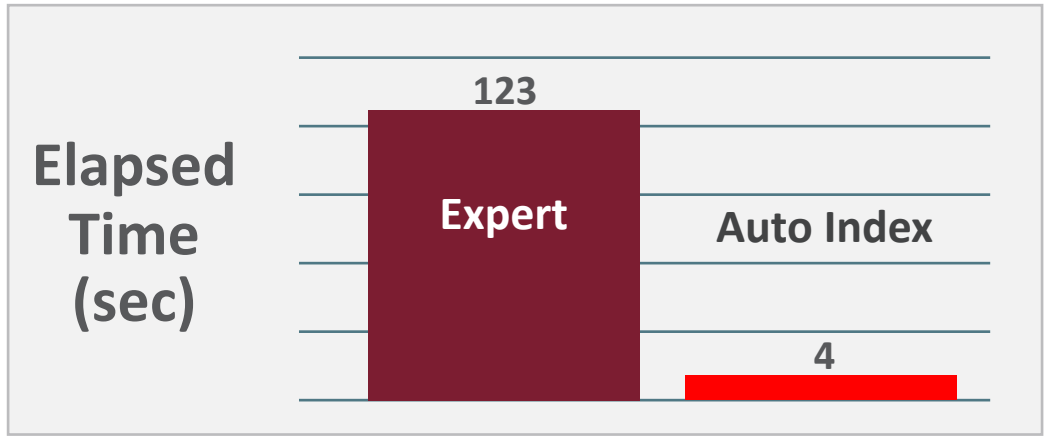
```
SELECT /*+ NO_USE_AUTO_INDEXES */ emp_id, emp_name, dept_id
FROM employees
WHERE dept_id > 50;
```

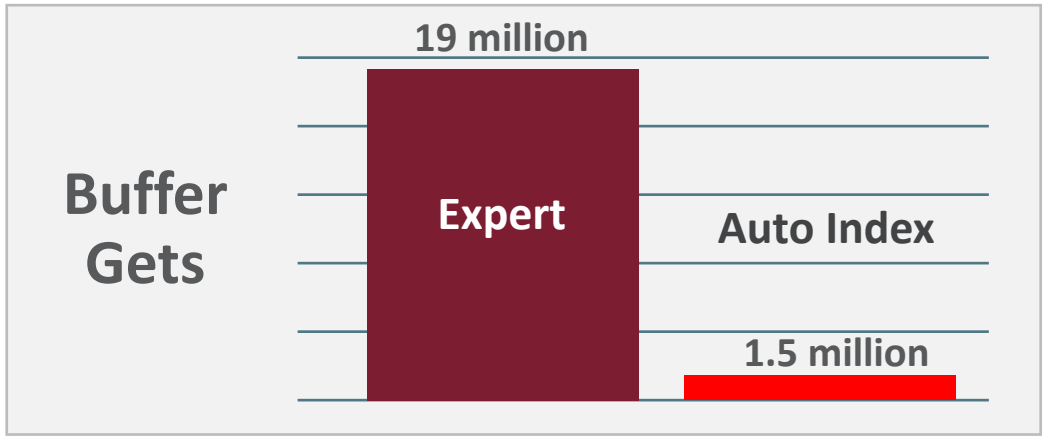
Agenda

1. **Original Approach to Index Tuning**
2. Automatic Indexing
   - How it work
   - What to expect
   - Deployments
   - Controls
   - Auditing and Report
3. Validation

Validation – **Accounts Receivable**

- Workload: 4,889 SQL statements
- Indexes:
  - Experts created 49 indexes of which 17 were used
  - Automatic indexing created 5 indexes, ***all*** of which were used





Summary

- Automatic Indexing will provide
  - Continuous optimization of the workload
  - Stable performance
  - Minimal human interaction required
  - Fully manageable if desired
- Additional automations in Oracle Database 19c
  - Statistics Management
  - Run-away SQL statements