
ORACLE®

ORACLE
Autonomous
Database Cloud

Automatic Indexingはいかに運用/チューニングを変えるのか？

Oracle Database Technology Night

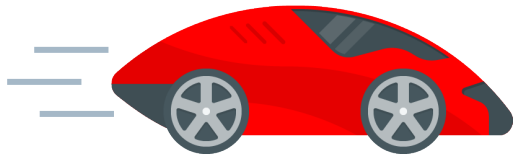
日本オラクル株式会社
データベースソリューション部
山田恭平
2019/08/20

Safe Harbor Statement

The preceding 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.

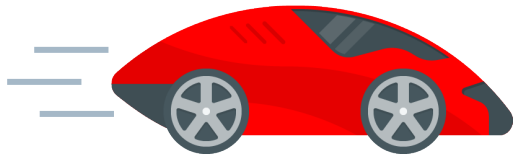
Agenda

1. 振り返り□Oracle Database 19c
2. Automatic Indexingの概要□
3. 検証内容と結果□
4. まとめ□



Agenda

1. 振り返りOracle Database 19c
2. Automatic Indexingの概要
3. 検証内容と結果
4. まとめ



Oracle Database 19cのリリースについて□

- Production on
  Exadata, Linux, Solaris,
  Windows, zLinux,
  AIX and HP-UX

New! 2019/07～□
Database Cloud Service,
Exadata Cloud Service,
Autonomous Transaction Processing Dedicated

TechNight#28-1で紹介した機能□

- **パフォーマンス**
  - **Automatic Indexing**
  - Quarantine for SQL statements　(SQL検疫)
  - オンライン統計収集の強化□
  - 自動オプティマイザ統計収集の強化□

- **可用性**
  - Active Data Guard DMLリダイレクト□
  - マルチ・インスタンスREDO適用とイン・メモリ機能の併用□

TechNight#28-1で紹介した機能□

- **テスト**
  - Real Application Testing, Database Replay機能の拡張□

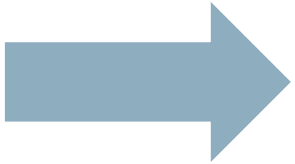
- **セキュリティ**
  - 事前定義アカウントのスキーマ限定アカウント化□
  - 権限分析の標準機能化□
  - 統合監査のトップレベルSQL限定オプション□

TechNight#28-1で紹介した機能□

- TechNight#28-1の資料はこちら□
  https://www.oracle.com/technetwork/jp/ondemand/database/db-new/db-tech-night-3508291-ja.html

TechNight#28-1のアンケートから□

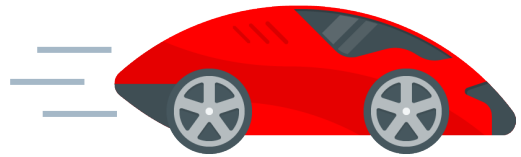
- アンケート結果(Automatic Indexingについて)
  - 自動索引について詳細が知りたい□
  - より踏み込んで話をしてほしい□
  - 削除・作成のタイミングは？□
  - 動きをみてみたい□
  - 検証工程ではどこまで検証してもらえる？□



今回はよりAutomatic Indexingに踏み込みます！□

Agenda

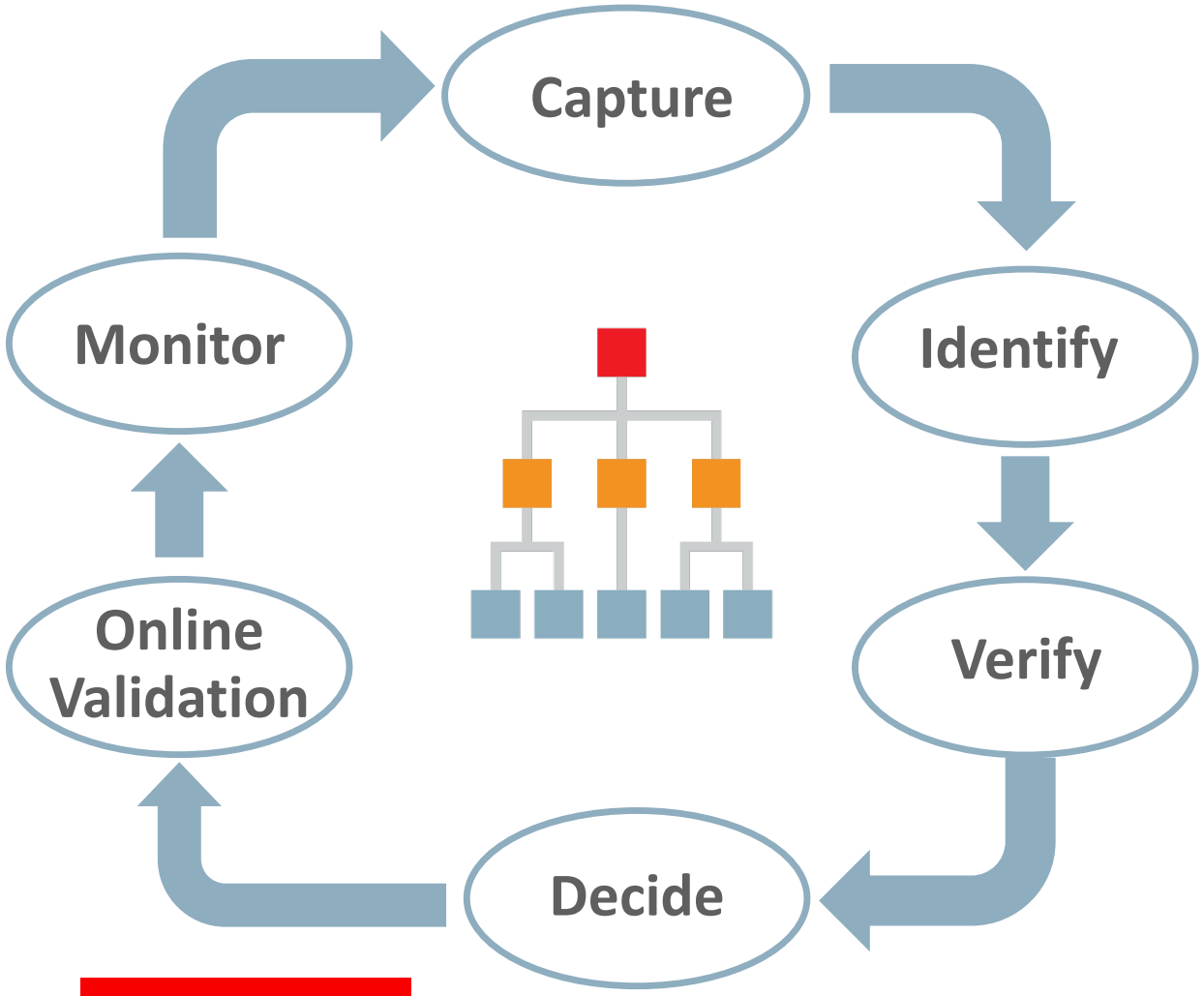
1. 振り返り□Oracle Database 19c
2. Automatic Indexingの概要□
3. 検証内容と結果□
4. まとめ□



Automatic Indexing Methodology

熟練のエンジニアが行うように絶え間なく索引チューニングを行う□

NEW IN 19c

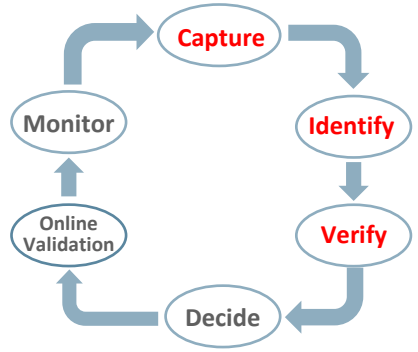


- メリット□
  - 難しく大変な索引チューニングをDBAが不在でも□可能にする□
- 機能概要□
  - アプリケーション・ワークロードを監視して、自動□的に索引の作成や削除などの管理を行う□
    - 候補索引の識別、利用前の検証などの一連のタスクはす□べて自動的に行い、そのアクティビティのレポートも可能□
      - 性能向上する索引のみVisibleに(一部SQLの性能劣化はSPMで)
      - REPORT ONLY も可能□nvisible索引のまま)
  - 15分ごとに左図の処理を実行する□
  - 以下の索引が□dvanced Low 圧縮で作成される□
    - 単一列□複数列、ファンクション・ベースのBツリー索引□
  - 実際にSQLの検証を行うため統計情報が重要□

ORACLE

Automatic Indexing

Automatic Indexing プロセスの操作□



1. Capture（取得）□
   - ASTS（Automatic SQL Tuning Set）内のアプリケーション・ワークロードから□ SQL文を定期的に取得する（SQL、実行計画、バインド変数、実行統計など）□
2. Identify Candidates（候補索引の特定）□
   - アプリケーション・ワークロードに役立つ（性能向上する）可能性のある候補索引を特定する□
   - 索引候補を□nusable および□visible 状態で作成（***メタデータの変更***のみ）□
3. Verify（検証）□
   - オプティマイザに新しく作成した索引を利用するかどうか確認□
     - 索引を作成し、SQLを実際に実行し、新しい索引によって性能が向上するかどうか評価□
   - 全ての評価は***アプリケーションの外***で行われる□

【参考】UNUSABLE,INVISIBLEの違いは？□

それぞれの違いを説明

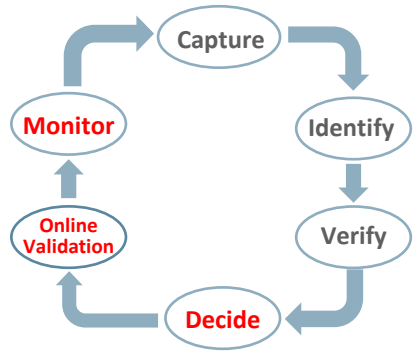
- UNUSABLEのINDEXはオプティマイザから見えない□
  なおかつDML発行時にINDEXを更新しない□
  ⇒バルクロードなどの大量データロード時にロード処理の速度が向上□
  　INDEXを再使用する際にはREBUILDが必要□

- INVISIBLEのINDEXはオプティマイザから見えない□
  DML発行時にもINDEXを更新する□
  ⇒オプティマイザのテストを行う際にINVISIBLEを使用する□

- 『Oracle Database 管理者ガイド21.2.11 使用禁止または不可視索引の使用について』□
  https://docs.oracle.com/cd/F19136_01/admin/managing-indexes.html#GUID-3A66938F-73C6-4173-844E-3938A0DBBB54

Automatic Indexing

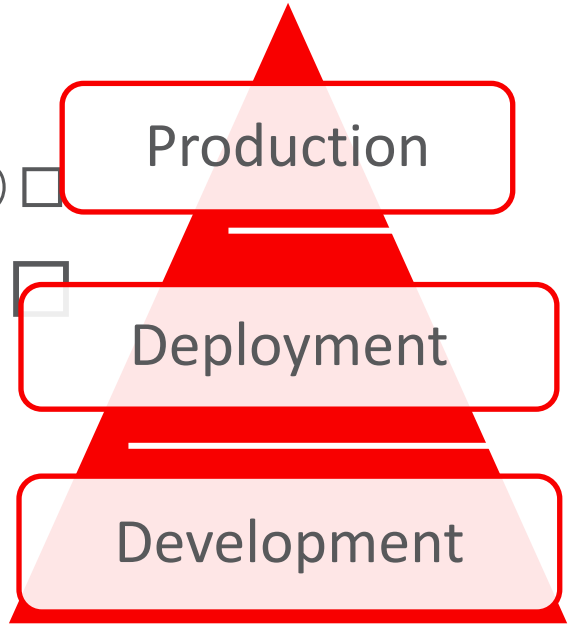
Automatic Indexing プロセスの操作□



4. Decide（索引の決定）□
   - 全てのSQLの性能が向上するのであれば、索引はVisibleに□
   - すべてのSQLの性能が劣化するのであれば、索引はInvisibleのまま□
   - 一部のSQLで性能が劣化するが、全体的に性能が向上するのであれば索引はVisibleに□
     - 性能が劣化するSQLは、SPMを使用して悪い計画を使用しないようにする□
5. Online Validation（オンライン検証）□
   - 他のSQLに対する索引の有効性検証はオンラインで継続□
   - 1セッションだけが新しい索引を利用することを許される□
6. Monitor（監視）□
   - 索引の利用状況は継続的に監視□
   - 長期間利用されていない自動作成索引はDropされる（デフォルトは373日）□

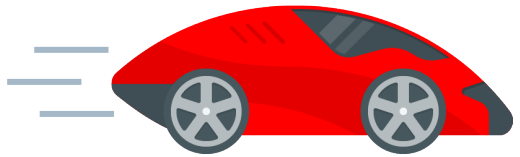
Automatic Indexing – Scope

- OLTP、DW、Mixed workloadsで有効、OLTPでは特に重要□
- チューニング済み、未チューニングの両方のアプリケーションに有効
  - チューニング済み□
    - 既存の二次索引は古すぎる、あるいは必要な索引が欠落している場合がある□
    - 一部の二次索引をDropし、新しく自動索引を追加可能□
  - 未チューニング□
    - 既存の索引は整合性制約を実現するもののみ（主キー、一意キー）□
- アプリケーション開発の全てのステージで適用可能□



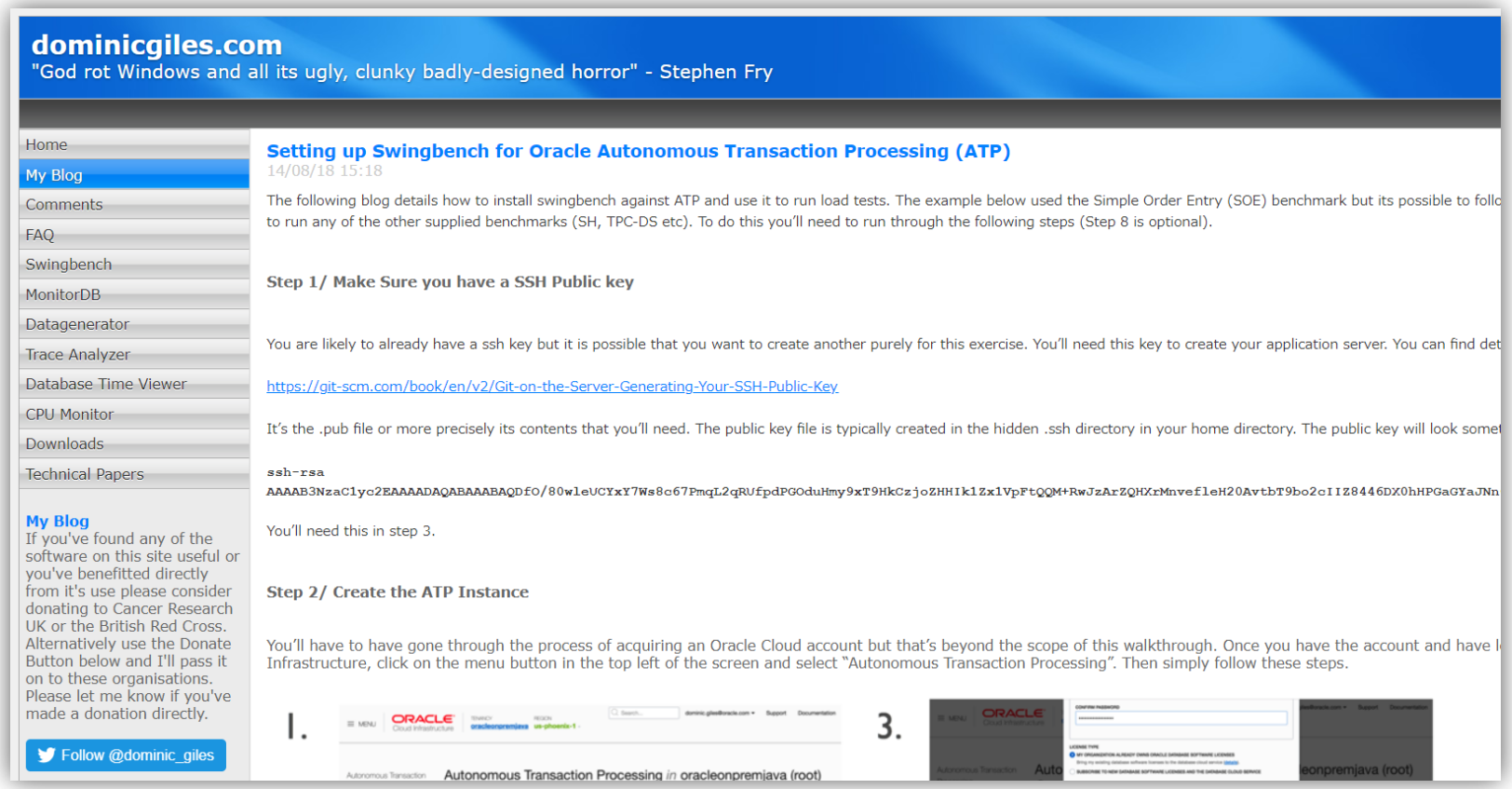
Agenda

1. 振り返り□Oracle Database 19c
2. Automatic Indexingの概要□
3. **検証内容と結果□**
4. まとめ□



検証環境について□

- 環境□
  - Exadata X2
    - Exadata 19.1
  - Oracle Database 19c
    - Database : 19.1.4
- 操作端末□
  - IA サーバ□
- 検証ツール□
  - Swing Bench 2.6
  - スキーマ：Order Entry
  - スケールファクタ：10GB

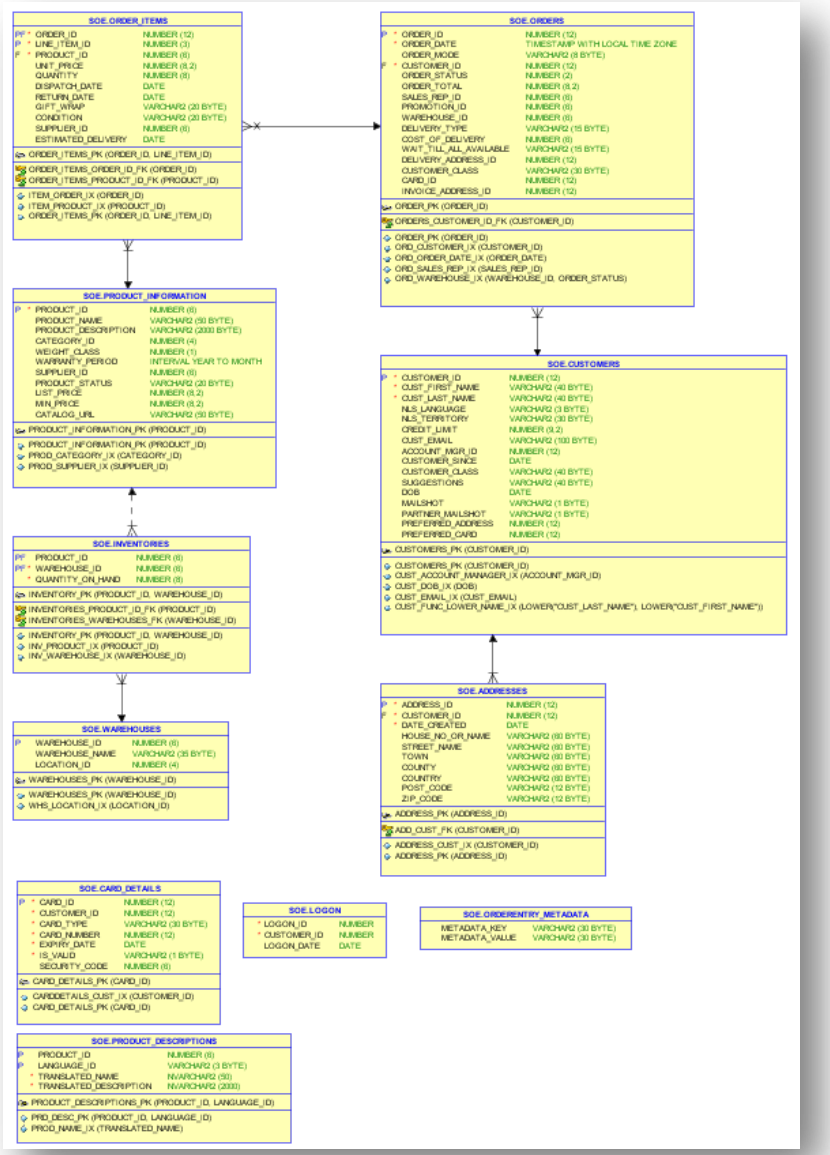


http://www.dominicgiles.com/swingbench.html

検証環境について□

- スキーマ構成（表）□

表名	件数	サイズ(MB)	Compress?	Partition?
ORDER_ITEMS	35392702	2034.4375	NO	NO
LOGON	16747143	503.5	NO	NO
ORDERS	11861064	1267.375	NO	NO
ADDRESSES	8800561	952.375	NO	NO
CARD_DETAILS	8692668	503.5	NO	NO
CUSTOMERS	6767032	1007.5	NO	NO
INVENTORIES	893672	18.640625	NO	NO
PRODUCT_DESCRIPTIONS	1000	0.2734375	NO	NO
PRODUCT_INFORMATION	1000	0.21875	NO	NO
WAREHOUSES	1000	0.0390625	NO	NO
ORDERENTRY_METADATA	4	0.0390625	NO	NO



検証環境について□

- スキーマ構成(索引)□
- PKのみ作成□

表名□	索引名□	レベル□	STATUS	Partition	Compress
ADDRESSES	ADDRESS_PK	2	VALID	NO	NO
CARD_DETAILS	CARD_DETAILS_PK	2	VALID	NO	NO
CUSTOMERS	CUSTOMERS_PK	2	VALID	NO	NO
INVENTORIES	INVENTORY_PK	2	VALID	NO	NO
ORDERS	ORDER_PK	2	VALID	NO	NO
ORDER_ITEMS	ORDER_ITEMS_PK	2	VALID	NO	NO
PRODUCT_DESCRIPTIONS	PRD_DESC_PK	1	VALID	NO	NO
PRODUCT_INFORMATION	PRODUCT_INFORMATION_PK	1	VALID	NO	NO
WAREHOUSES	WAREHOUSES_PK	1	VALID	NO	NO

今回の検証で確認したこと□

Automatic Indexingについて、今回の検証では以下を確認しました

1. 設定方法□
2. 実行結果の確認方法□
3. 自動作成されたIndexと手動作成されたIndexの違い□
4. Automatic Indexing動作時のCPU負荷□
5. Automatic Indexingの有効性□
6. ファンクションINDEX作成の方法□

1.設定方法□

DBMS_AUTO_INDEX.CONFIGUREプロシージャ

EXEC DBMS_AUTO_INDEX.CONFIGURE('<パラメータ名>', '<値>' [,<AUTO_INDEX_SCHEMAのときの値>]);

パラメータ名	概要
AUTO_INDEX_MODE	Automatic Indexing の機能を使用するかどうかを設定□ IMPLEMENT : 有効　OFF: 無効 (デフォルト) REPORT ONLY: Invisible索引を作成するが索引の使用はしない□
AUTO_INDEX_SCHEMA	スキーマ単位で有効/無効の制御□
AUTO_INDEX_RETENTION_FOR_AUTO	自動作成索引がこの期間使用されないと削除□(Default : 373日)
AUTO_INDEX_RETENTION_FOR_MANUAL	手動作成索引がこの期間使用されないと削除□(Default : 無期限)
AUTO_INDEX_REPORT_RETENTION	レポート情報の保持期間□(Default : 31日)
AUTO_INDEX_DEFAULT_TABLESPACE	Automatic Indexing が利用する表領域□(Default : NULL)
AUTO_INDEX_TEMP_TABLESPACE	Automatic Indexing が利用する一時表領域□(Default : NULL)
AUTO_INDEX_SPACE_BUDGET	Automatic Indexing が利用する領域のサイズ(%) デフォルト表領域を利用時のみ□

1.設定方法□

有効化

‘SOE’スキーマのみ有効化□

```
SQL> begin
 dbms_auto_index.configure (
 parameter_name => 'AUTO_INDEX_SCHEMA' ,
 parameter_value => 'SOE' ,
         allow => TRUE) ;
end;
/
```

AUTO_INDEX_MODEをIMPLEMENTに変更□

```
SQL> EXEC DBMS_AUTO_INDEX.CONFIGURE('AUTO_INDEX_MODE','IMPLEMENT');

PL/SQL procedure successfully completed.
```

1.設定方法□

有効化

設定を確認□

```
SQL> SELECT * FROM DBA_AUTO_INDEX_CONFIG ;

PARAMETER_NAME                 PARAMETER_VALUE LAST_MODIFIED                  MODIFIED_BY
------------------------------ --------------- ------------------------------ -----------
AUTO_INDEX_COMPRESSION         OFF
AUTO_INDEX_DEFAULT_TABLESPACE
AUTO_INDEX_MODE                IMPLEMENT       09-JUL-19 04.52.38.0 SYS
                                               00000 PM

AUTO_INDEX_REPORT_RETENTION    31
AUTO_INDEX_RETENTION_FOR_AUTO  373
AUTO_INDEX_RETENTION_FOR_MANUAL
AUTO_INDEX_SCHEMA              schema IN (SOE) 08-JUL-19 06.24.17.0 SYS
                                               00000 PM


PARAMETER_NAME                 PARAMETER_VALUE LAST_MODIFIED                  MODIFIED_BY
------------------------------ --------------- ------------------------------ -----------
AUTO_INDEX_SPACE_BUDGET        50

8 rows selected.
```

2.実行結果の確認方法□

実行履歴の確認

有効化の30分後に実行履歴を確認□

```
SQL> SELECT execution_name, TO_CHAR(execution_start,'YYYY/MM/DD HH24:MI:SS') AS execution_start, TO_CHAR(execution_end
'YYYY/MM/DD HH24:MI:SS') AS execution_end ,status FROM DBA_AUTO_INDEX_EXECUTIONS ORDER BY execution_start;

EXECUTION_NAME            EXECUTION_START     EXECUTION_END       STATUS
------------------------- ------------------- ------------------- ---------
SYS_AI_2019-07-25/18:46:09 2019/07/25 18:46:09 2019/07/25 18:46:12 COMPLETED
SYS_AI_2019-07-25/19:06:10 2019/07/25 19:06:10 2019/07/25 19:10:10 COMPLETED
```

2.実行結果の確認方法□

実行統計の確認

有効化の30分後に実行統計を確認□

```
SQL>  SELECT * FROM dba_auto_index_statistics WHERE value>0 ORDER BY 1;

EXECUTION_NAME            STAT_NAME                          VALUE
------------------------- ------------------------------ ----------
SYS_AI_2019-07-25/19:06:10 Index candidates                        8
SYS_AI_2019-07-25/19:06:10 Indexes created (visible)               4
SYS_AI_2019-07-25/19:06:10 Indexes created (invisible)             4
SYS_AI_2019-07-25/19:06:10 Indexes dropped                         0
SYS_AI_2019-07-25/19:06:10 Space used in bytes            1375731712
SYS_AI_2019-07-25/19:06:10 Space reclaimed in bytes                0
SYS_AI_2019-07-25/19:06:10 SQL statements verified                 7
SYS_AI_2019-07-25/19:06:10 SQL statements improved                 4
SYS_AI_2019-07-25/19:06:10 SQL statements managed by SPM           3
SYS_AI_2019-07-25/19:06:10 SQL plan baselines created              3
SYS_AI_2019-07-25/19:06:10 Improvement percentage              99.39
```

2.実行結果の確認方法□
アクティビティレポート

- Automatic Indexing の各タスク・アクティビティを示すレポートを生成する□
- 以下のファンクションを使用して行う□
  - DBMS_AUTO_INDEX.REPORT_ACTIVITYファンクション（特定期間のアクティビティ・レポート）□
  - DBMS_AUTO_INDEX.REPORT_LAST_ACTIVITYファンクション（最後のアクティビティ・レポート）□

```
SQL> set linesize 300 trims on pagesize 1000 long 100000
SQL> column report format a120

SQL> SELECT dbms_auto_index.report_activity(
                              sysdate-30, -- 開始日時
                              null,      -- 終了日時
                              'text',    -- レポート種別 (TEXT,HTML,XML)
                              'all',     -- 出力するセクション (ALL,SUMMARY,INDEX_DETAILS,
                                                  VERIFICATION_DETAILS,ERRORS)
                              'all',)    -- レベル (BASIC,TYPICAL,ALL)
                              report FROM dual;
```

レポート出力結果

```
REPORT
------------------------------------------------------------------------------

GENERAL INFORMATION
------------------------------------------------------------------------------
 Activity start               : 26-JUL-2019 15:43:28
 Activity end                 : 26-JUL-2019 15:43:28
 Executions completed         : 1389
 Executions interrupted       : 0
 Executions with fatal error  : 0
------------------------------------------------------------------------------

SUMMARY (AUTO INDEXES)
------------------------------------------------------------------------------
 Index candidates                                 : 8
 Indexes created (visible / invisible)            : 8 (5 / 3)
 Space used (visible / invisible)                 : 1.38 GB (1.36 GB / 17.83
MB)
 Indexes dropped                                  : 0
 SQL statements verified                          : 9
 SQL statements improved (improvement factor)     : 5 (28x)
 SQL plan baselines created                       : 0
 Overall improvement factor                       : 22.8x
------------------------------------------------------------------------------

SUMMARY (MANUAL INDEXES)
------------------------------------------------------------------------------
 Unused indexes    : 0
 Space used        : 0 B
 Unusable indexes  : 0
------------------------------------------------------------------------------
```

候補INDEXの数、□
visible/invisibleそれぞれの数、□
使用容量、□
全体の速度改善値□
などを確認可能□

```
INDEX DETAILS
------------------------------------------------------------------------------
1. The following indexes were created:
*: invisible
------------------------------------------------------------------------------
| Owner | Table               | Index                  | Key           | Type
| Properties |
------------------------------------------------------------------------------
| SOE   | ADDRESSES           | SYS_AI_4bz3nuupj3kt5   | CUSTOMER_ID   | B-TREE
| NONE       |
| SOE   | CARD_DETAILS        | SYS_AI_dt4w4vr174j9m   | CUSTOMER_ID   | B-TREE
| NONE       |
| SOE   | INVENTORIES         | * SYS_AI_2s9jaak2smbq4 | WAREHOUSE_ID  | B-TREE
| NONE       |
| SOE   | ORDERS              | SYS_AI_3zOOfrhp9vd91   | WAREHOUSE_ID  | B-TREE
| NONE       |
| SOE   | ORDERS              | SYS_AI_5p2zapcmkj174   | CUSTOMER_ID   | B-TREE
| NONE       |
| SOE   | ORDERS              | SYS_AI_gbwwy984mc1ft   | SALES_REP_ID  | B-TREE
| NONE       |
| SOE   | PRODUCT_DESCRIPTIONS | * SYS_AI_2Otjdcuwznyhx | PRODUCT_ID    | B-TREE
| NONE       |
| SOE   | PRODUCT_INFORMATION | * SYS_AI_b9k5zyqOmjwf5 | CATEGORY_ID   | B-TREE
| NONE       |
------------------------------------------------------------------------------
```

作成されたINDEX名、□
TABLE名、□
などを確認可能□

レポート出力結果

```
VERIFICATION DETAILS
--------------------------------------------------------------------------------
1. The performance of the following statements improved:

--------------------------------------------------------------------------------
 Parsing Schema Name  : SOE
 SQL ID               : 29qp10usqkqh0
 SQL Text             : SELECT TT.ORDER_TOTAL, TT.SALES_REP_ID, TT.ORDER_DATE,
                        CUSTOMERS.CUST_FIRST_NAME, CUSTOMERS.CUST_LAST_NAME FROM
                        (SELECT ORDERS.ORDER_TOTAL, ORDERS.SALES_REP_ID,
                        ORDERS.ORDER_DATE, ORDERS.CUSTOMER_ID, RANK() OVER (ORDER
                        BY ORDERS.ORDER_TOTAL DESC) SAL_RANK FROM ORDERS WHERE
                        ORDERS.SALES_REP_ID = :B1 ...

 Improvement Factor   : 12x

Execution Statistics:
-----------------------------
                       Original Plan          Auto Index Plan
                       ----------------------  ------------------------------
 Elapsed Time (s):     27750967                374744
 CPU Time (s):         26992156                332137
 Buffer Gets:          2466076                 17148
 Optimizer Cost:       82016                   45402
 Disk Reads:           203824                  48
 Direct Writes:        0                       0
 Rows Processed:       9432                    456
 Executions:           12                      1
```

Visibleになった各INDEXの□
改善数値を確認可能□

レポート出力結果

PLANS SECTION

- Original

Plan Hash Value : 3619984409

Id	Operation	Name	Rows	Bytes	Cost	Time
0	SELECT STATEMENT				92666	
1	HASH JOIN		18806	1617316	92666	00:00:04
2	NESTED LOOPS		18806	1617316	92666	00:00:04
3	NESTED LOOPS		18806	1617316	92666	00:00:04
4	STATISTICS COLLECTOR					
5	VIEW		18806	1222390	55041	00:00:03
6	WINDOW SORT PUSHED RANK		18806	639404	55041	00:00:03
7	TABLE ACCESS STORAGE FULL	ORDERS	18806	639404	55040	00:00:03
8	INDEX UNIQUE SCAN	CUSTOMERS_PK	1		1	00:00:01
9	TABLE ACCESS BY INDEX ROWID	CUSTOMERS	1	21	2	00:00:01
10	TABLE ACCESS STORAGE FULL	CUSTOMERS	1	21	2	00:00:01

Notes

- optimizer_use_stats_on_conventional_dml = yes
- This is an adaptive plan

- With Auto Indexes

Plan Hash Value : 3900469033

Id	Operation	Name	Rows	Bytes	Cost	Time
0	SELECT STATEMENT		15175	1745125	45402	00:00:02
1	NESTED LOOPS		15175	1745125	45402	00:00:02
2	NESTED LOOPS		15175	1745125	45402	00:00:02
* 3	VIEW		15175	986375	15042	00:00:01
* 4	WINDOW SORT PUSHED RANK		15175	515950	15042	00:00:01
5	TABLE ACCESS BY INDEX ROWID BATCHED	ORDERS	15175	515950	15041	00:00:01
* 6	INDEX RANGE SCAN	SYS_AI_gbwwy984mc1ft	16343		37	00:00:01
* 7	INDEX UNIQUE SCAN	CUSTOMERS_PK	1		1	00:00:01
8	TABLE ACCESS BY INDEX ROWID	CUSTOMERS	1	50	2	00:00:01

Predicate Information (identified by operation id):

```
* 3 - filter("TT"."SAL_RANK"<=10)
* 4 - filter(RANK() OVER ( ORDER BY INTERNAL_FUNCTION("ORDERS"."ORDER_TOTAL") DESC )<=10)
* 6 - access("ORDERS"."SALES_REP_ID"=:B1)
* 7 - access("CUSTOMERS"."CUSTOMER_ID"="TT"."CUSTOMER_ID")
```

Notes

- optimizer_use_stats_on_conventional_dml = yes
- Dynamic sampling used for this statement (level = 11)
- This is an adaptive plan

Visibleになった各INDEXの□
新旧実行計画を確認可能□

3.自動作成されたINDEXと手動作成されたINDEXの違い□

CREATE文

DDL文を確認□

```
SQL> SELECT statement FROM DBA_AUTO_INDEX_IND_ACTIONS ORDER BY end_time;
CREATE INDEX "SOE"."SYS_AI_0rat78jtvw85m"  ON "SOE"."CUSTOMERS"("SYS_STUEXM_RW1QTA5FN9AJCNV__#R","S
YS_STU1PEQ_9A63QSAS805132R9WU") TABLESPACE "SOE" UNUSABLE INVISIBLE AUTO  ONLINE
```

- INDEXに接頭語として"SYS_AI_"がつく□
- UNUSABLE INVISIBLEとして作成□
  - 検証後にパフォーマンスの向上するINDEXのみを有効化□
- AUTO指定□
  - *_indexesビューを確認すると新しくAUTO列が追加されており、□
    自動作成、手動作成のフラグになっている□
  - 手動でのINDEX作成時にAUTO指定すると構文エラーとなる□

3.自動作成されたINDEXと手動作成されたINDEXの違い□

INDEXに対しての変更

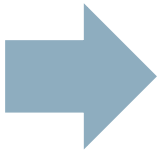
- 自動作成されたINDEXの手動削除は不可□
- DBMS_AUTO_INDEX.CONFIGUREプロシージャにて、□
  表領域を分けることで、表領域ごと削除は可能だが。。□
  ※内部情報が消えていないなどの報告があるため実運用は不可□
- 手動でのInvisible→Visibleも不可□
- 自動作成されたInvisibleのINDEXと同様の列に対して、手動でINDEXの作□
  成も不可（12.1～既存INDEXがInvisibleであれば同一列にINDEXを作成できたがAutoは不可）

➡ 完全にシステムまかせ□

3.自動作成されたINDEXと手動作成されたINDEXの違い□

ヒント句での指定

- 自動作成されたInvisibleのINDEXをSQLで使用するよう□
  ヒント句を記述しても無視□
- 初期化パラメータOPTIMIZER_USE_INVISIBLE_INDEXESをtrueに変更し、□
  INVISIBLE索引を使用するようにしても、自動作成されたINDEXには無効
  （上と同じくヒント指定しても無視）□

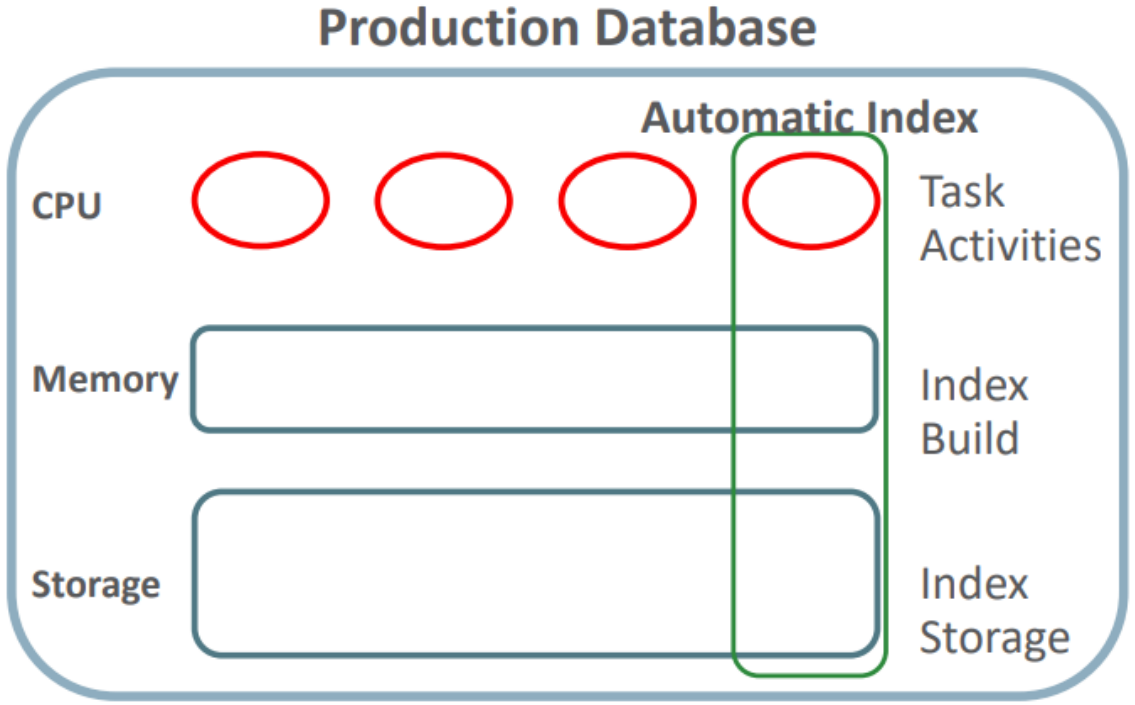


完全にシステムまかせ□

4.Automatic Indexing動作時のCPU負荷□

OOW18資料より

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



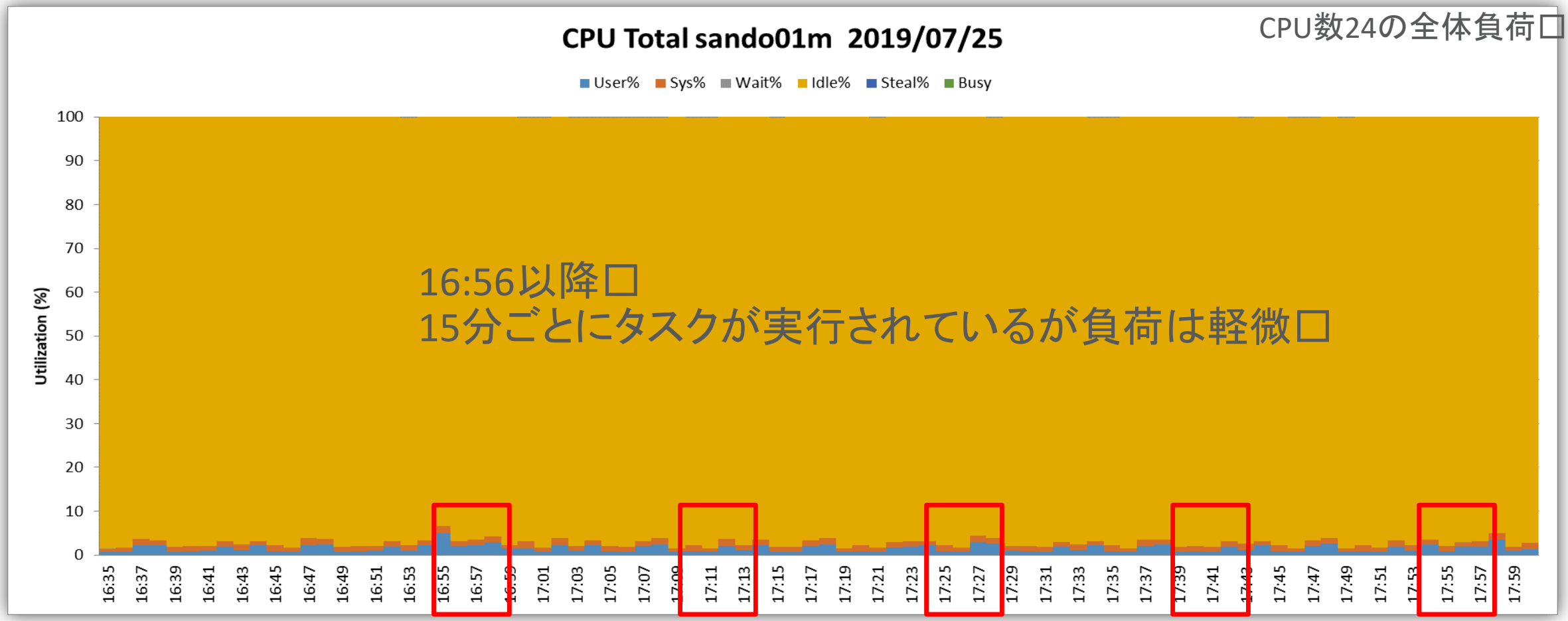
- The task does consume CPU Memory and storage
  - Resource manager plan limits the task to 1 CPU
  - Customers can control which temp tablespace is used to build indexes
  - Customers can control which tablespace and how much space can be used by Auto Indexing

Test Drive Automatic Index Creation in Oracle Autonomous Database Cloud [TRN3980]
https://oracle.rainfocus.com/widget/oracle/oow18/catalogoow18?search=TRN3980

4.Automatic Indexing動作時のCPU負荷□

INDEXが作成されなかった場合の全CPU負荷

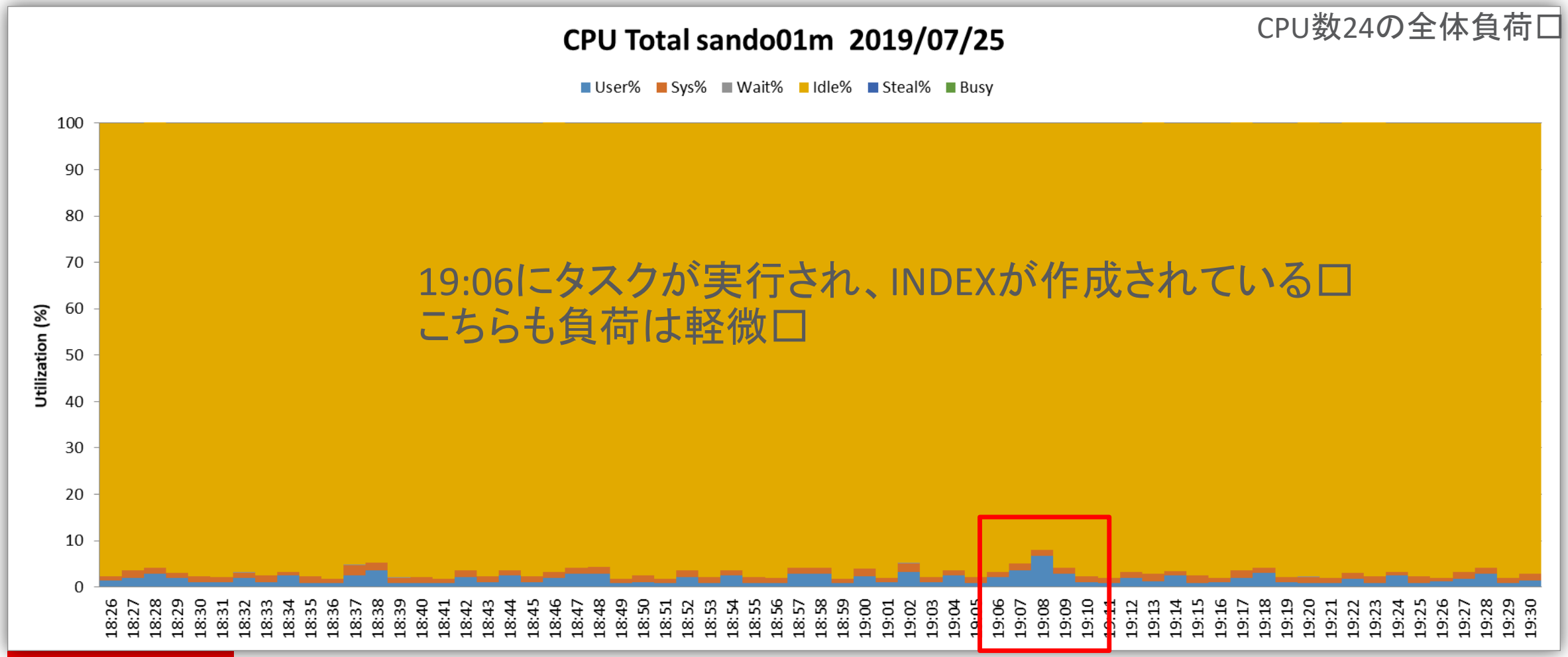
CPU数24の全体負荷□



4.Automatic Indexing動作時のCPU負荷□

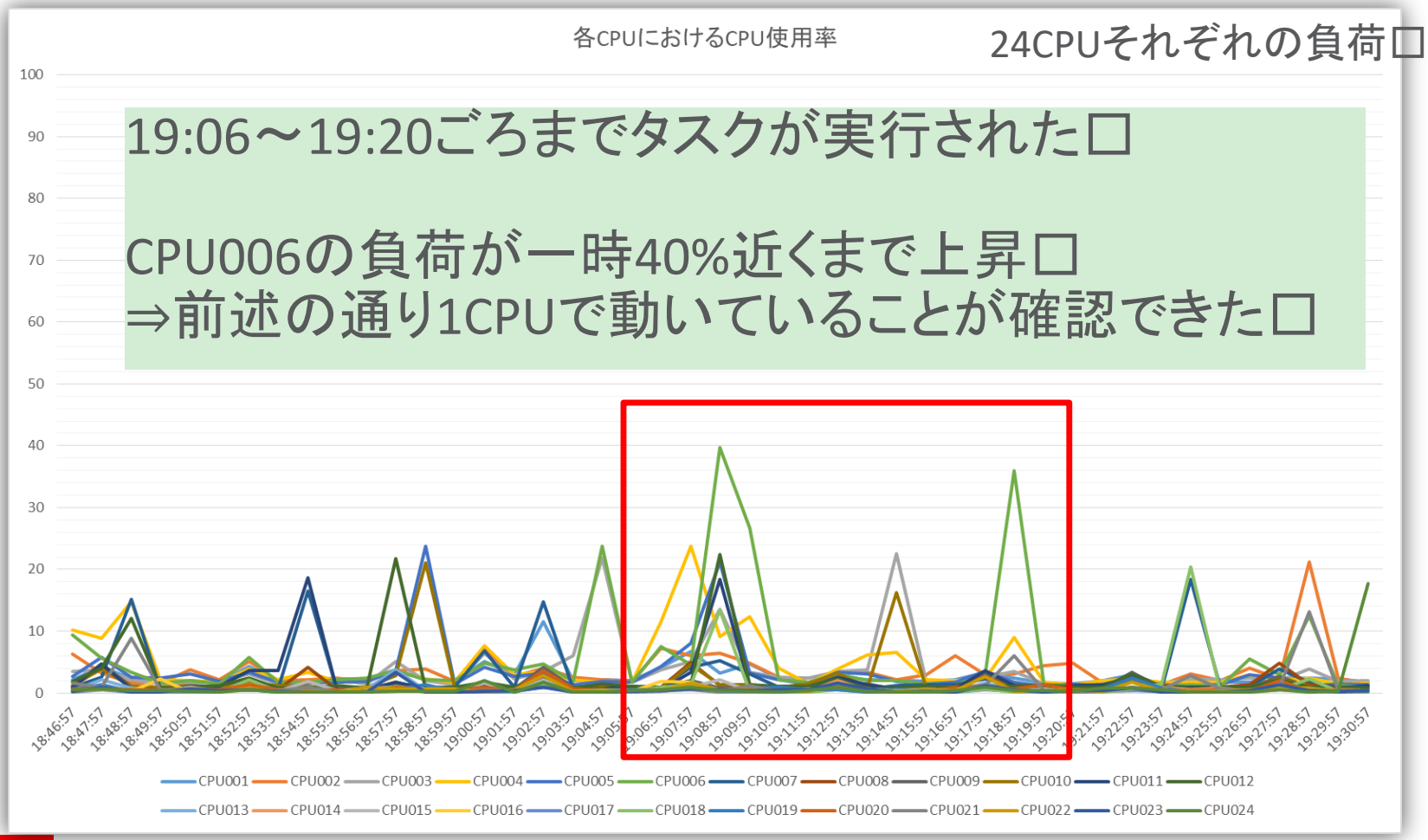
INDEXが作成された場合の全CPU負荷(8個の候補を検証し7個を有効化)

CPU数24の全体負荷□



4.Automatic Indexing動作時のCPU負荷□

INDEXが作成された場合の各CPUごとの負荷



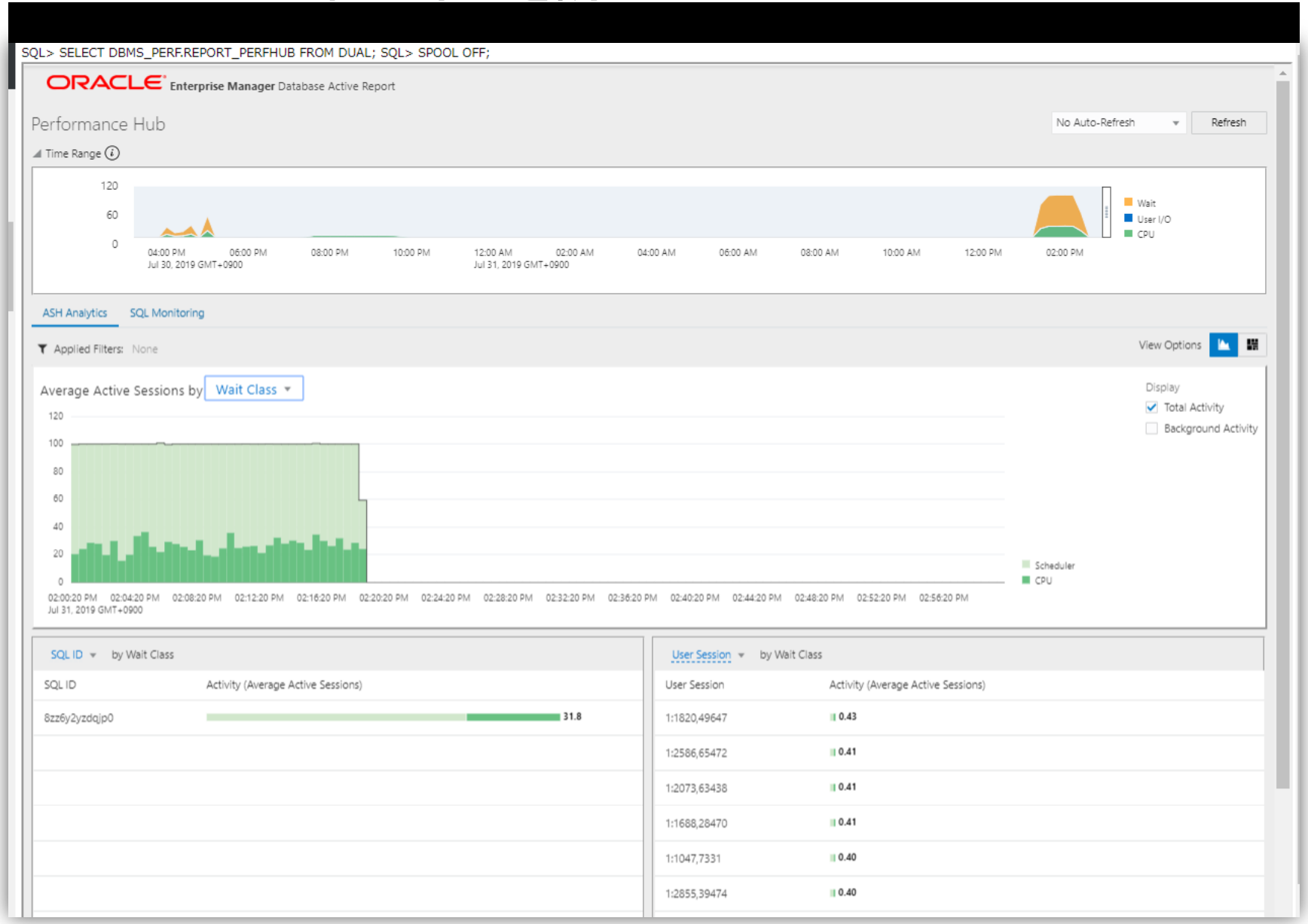
5.Automatic Indexingの有効性□

INDEX作成前にTPSを確認

INDEXなし(PKのみ)
状態でのTPS

Swing Bench
Update Customer Detailsを実行□

パフォーマンス・ハブでTPSが低い原因を調査



【参考】パフォーマンス・ハブとは□

手軽にDBのパフォーマンスを確認するツール

- 指定した期間のす□てのパフォーマンス・データの統合ビューを示すアクティビティ・レポートを確認可能□
- DBMS_PERFパッケージのREPORT_PERFHUBで、□
  Enterprise Managerのパフォーマンス・ハブ相当の画面をHTML出力□

『Oracle Database 2日でデータベース管理者□□
10.1.2 パフォーマンス・ハブを使用した□
パフォーマンスの監視□
https://docs.oracle.com/cd/F19136_01/admqs/monitoring-and-tuning-the-database.html#GUID-573F73E4-EF1C-46F2-9BAB-73DA08E7D364

5.Automatic Indexingの有効性□

パフォーマンス・ハブで確認したSQLの調査

SQL_IDをもとにSQL_TEXTを確認□

```
SQL> SELECT s.PIECE ,s.SQL_ID ,s.HASH_VALUE ,s.ADDRESS ,s.SQL_TEXT
FROM V$SQLTEXT s
WHERE s.SQL_ID LIKE '%8zz6y2y%'
ORDER BY PIECE ;

 PIECE SQL_ID         HASH_VALUE ADDRESS          SQL_TEXT
------ ------------- ---------- ---------------- ----------------------------------------------------------------
     0 8zz6y2yzdqjp0  3202041504 0000000251D51EA8 SELECT CUSTOMER_ID, CUST_FIRST_NAME, CUST_LAST_NAME, NLS_LANGUAG
     1 8zz6y2yzdqjp0  3202041504 0000000251D51EA8 E, NLS_TERRITORY, CREDIT_LIMIT, CUST_EMAIL, ACCOUNT_MGR_ID, CUST
     2 8zz6y2yzdqjp0  3202041504 0000000251D51EA8 OMER_SINCE, CUSTOMER_CLASS, SUGGESTIONS, DOB, MAILSHOT, PARTNER_
     3 8zz6y2yzdqjp0  3202041504 0000000251D51EA8 MAILSHOT, PREFERRED_ADDRESS, PREFERRED_CARD FROM CUSTOMERS WHERE
     4 8zz6y2yzdqjp0  3202041504 0000000251D51EA8  LOWER(CUST_LAST_NAME) = LOWER(:B3 ) AND LOWER(CUST_FIRST_NAME)
     5 8zz6y2yzdqjp0  3202041504 0000000251D51EA8 = LOWER(:B2 ) AND ROWNUM < :B1

6 rows selected.
```

- WHERE句でLOWER関数を使っているため、ファンクションINDEXが必要□
  ファンクションINDEX：関数の結果によるINDEX

5.Automatic Indexingの有効性□

Automatic Indexing実行後にレポートを確認

```
SQL ID              : 8zz6y2yzdqjp0
SQL Text            : SELECT CUSTOMER_ID, CUST_FIRST_NAME, CUST_LAST_NAME,
                      NLS_LANGUAGE, NLS_TERRITORY, CREDIT_LIMIT, CUST_EMAIL,
                      ACCOUNT_MGR_ID, CUSTOMER_SINCE, CUSTOMER_CLASS,
                      SUGGESTIONS, DOB, MAILSHOT, PARTNER_MAILSHOT,
                      PREFERRED_ADDRESS, PREFERRED_CARD FROM CUSTOMERS WHERE
                      LOWER(CUST_LAST_NAME) = LOWER(:B3 ) AND LOW...
Improvement Factor  : 270.5x

Execution Statistics:
-----------------------
                    Original Plan                  Auto Index Plan
                    ---------------------------    ---------------------------
Elapsed Time (s):   101907650481                   1104545
CPU Time (s):       22824994003                    137433
Buffer Gets:        1056117356                     705
Optimizer Cost:     45915                          10
Disk Reads:         0                              2
Direct Writes:      0                              0
Rows Processed:     84                             0
Executions:         6474                           1

PLANS SECTION
-------------------------------------------------------------------------------

- Original
-----------------------
Plan Hash Value  : 1005345217
-------------------------------------------------------------------------------
| Id | Operation                    | Name      | Rows | Bytes | Cost  | Time     |
-------------------------------------------------------------------------------
|  0 | SELECT STATEMENT             |           |      |       | 45915 |          |
|  1 |   COUNT STOPKEY              |           |      |       |       |          |
|  2 |    TABLE ACCESS STORAGE FULL | CUSTOMERS |   10 |  1270 | 45915 | 00:00:02 |
-------------------------------------------------------------------------------
```

```
- With Auto Indexes
- ------------------
Plan Hash Value  : 4258596255

-------------------------------------------------------------------------------------------------------
| Id  | Operation                                | Name                | Rows | Bytes | Cost | Time     |
-------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                         |                     |    4 |   508 |   10 | 00:00:01 |
| * 1 |   COUNT STOPKEY                          |                     |      |       |      |          |
|   2 |    TABLE ACCESS BY INDEX ROWID BATCHED   | CUSTOMERS           |    4 |   508 |   10 | 00:00:01 |
| * 3 |     INDEX RANGE SCAN                     | SYS_AI_0rat78jtvw85m |    7 |       |    3 | 00:00:01 |
-------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------
* 1 - filter(ROWNUM<:B1)
* 3 - access(LOWER("CUST_FIRST_NAME")=LOWER(:B2) AND LOWER("CUST_LAST_NAME")=LOWER(:B3))


Notes
-----
- Dynamic sampling used for this statement ( level = 11 )


-------------------------------------------------------------------------------
-------------------------------------------------------------------------------

ERRORS
-------------------------------------------------------------------------------
No errors found.
-------------------------------------------------------------------------------
```

5.Automatic Indexingの有効性□

実行計画の比較

INDEX作成前

```
SQL> SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY_CURSOR('8zz6y2yzdqjp0'));

PLAN_TABLE_OUTPUT
------------------------------------------------------------------------------------------
SQL_ID  8zz6y2yzdqjp0, child number 0
-------------------------------------
SELECT CUSTOMER_ID, CUST_FIRST_NAME, CUST_LAST_NAME, NLS_LANGUAGE,
NLS_TERRITORY, CREDIT_LIMIT, CUST_EMAIL, ACCOUNT_MGR_ID,
CUSTOMER_SINCE, CUSTOMER_CLASS, SUGGESTIONS, DOB, MAILSHOT,
PARTNER_MAILSHOT, PREFERRED_ADDRESS, PREFERRED_CARD FROM CUSTOMERS
WHERE LOWER(CUST_LAST_NAME) = LOWER(:B3 ) AND LOWER(CUST_FIRST_NAME) =
LOWER(:B2 ) AND ROWNUM < :B1

Plan hash value: 1005345217

------------------------------------------------------------------------------------------
| Id  | Operation                    | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT             |           |       |       | 46194 (100)|          |
|*  1 |  COUNT STOPKEY               |           |       |       |            |          |
|*  2 |   TABLE ACCESS STORAGE FULL| CUSTOMERS |     8 |  1056 | 46194   (1)| 00:00:02 |
------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter(ROWNUM<:B1)
   2 - filter((LOWER("CUST_LAST_NAME")=LOWER(:B3) AND
              LOWER("CUST_FIRST_NAME")=LOWER(:B2)))


26 rows selected.

SQL>
```

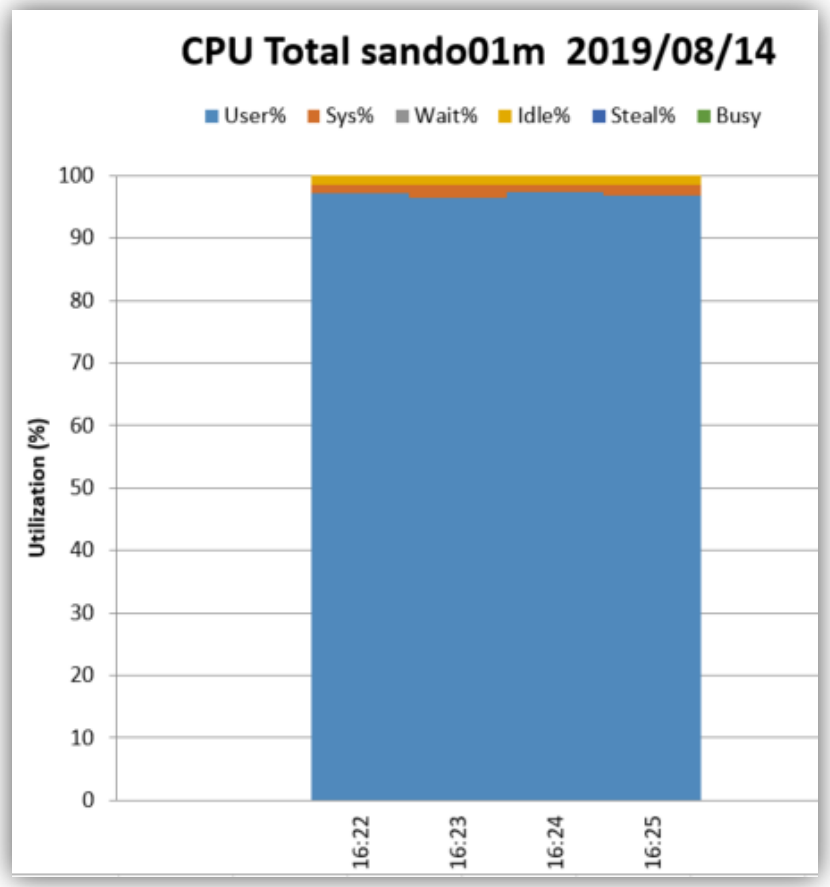
INDEX作成後

```
SQL> SELECT * FROM TABLE(DBMS_XPLAN.DISPLAY_CURSOR('8zz6y2yzdqjp0'));
SQL_ID  8zz6y2yzdqjp0, child number 0
-------------------------------------
SELECT CUSTOMER_ID, CUST_FIRST_NAME, CUST_LAST_NAME, NLS_LANGUAGE,
NLS_TERRITORY, CREDIT_LIMIT, CUST_EMAIL, ACCOUNT_MGR_ID,
CUSTOMER_SINCE, CUSTOMER_CLASS, SUGGESTIONS, DOB, MAILSHOT,
PARTNER_MAILSHOT, PREFERRED_ADDRESS, PREFERRED_CARD FROM CUSTOMERS
WHERE LOWER(CUST_LAST_NAME) = LOWER(:B3 ) AND LOWER(CUST_FIRST_NAME) =
LOWER(:B2 ) AND ROWNUM < :B1

Plan hash value: 4258596255

-----------------------------------------------------------------------------------------------------------
| Id  | Operation                           | Name                | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                    |                     |       |       |    13 (100)|          |
|*  1 |  COUNT STOPKEY                      |                     |       |       |            |          |
|   2 |   TABLE ACCESS BY INDEX ROWID BATCHED | CUSTOMERS         |    10 |  1270 |    13   (0)| 00:00:01 |
|*  3 |    INDEX RANGE SCAN                 | SYS_AI_0rat78jtvw85m |    10 |       |     3   (0)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter(ROWNUM<:B1)
   3 - access("CUSTOMERS"."SYS_STUEXM_RW1QTA5FN9AJCNV__#R"=LOWER(:B2) AND
              "CUSTOMERS"."SYS_STU1PEQ_9A63QSAS805132R9WU"=LOWER(:B3))

Note
-----
   - dynamic statistics used: statistics for conventional DML
```

5.Automatic Indexingの有効性□

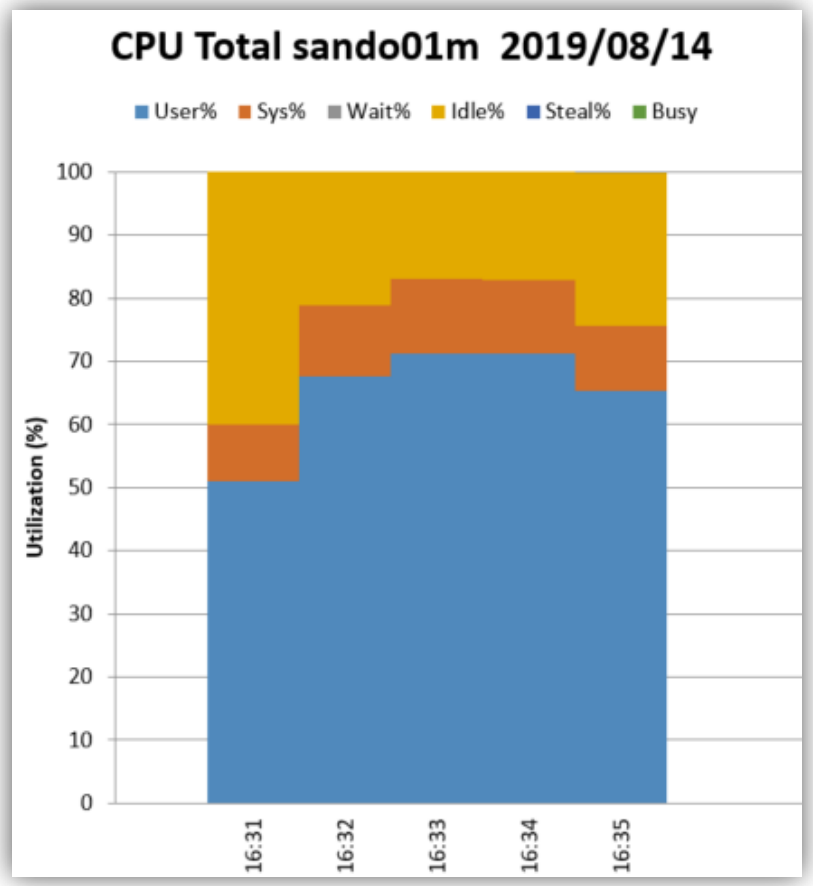
CPU使用率の比較

INDEX作成前



CPU使用率平均98%

INDEX作成後

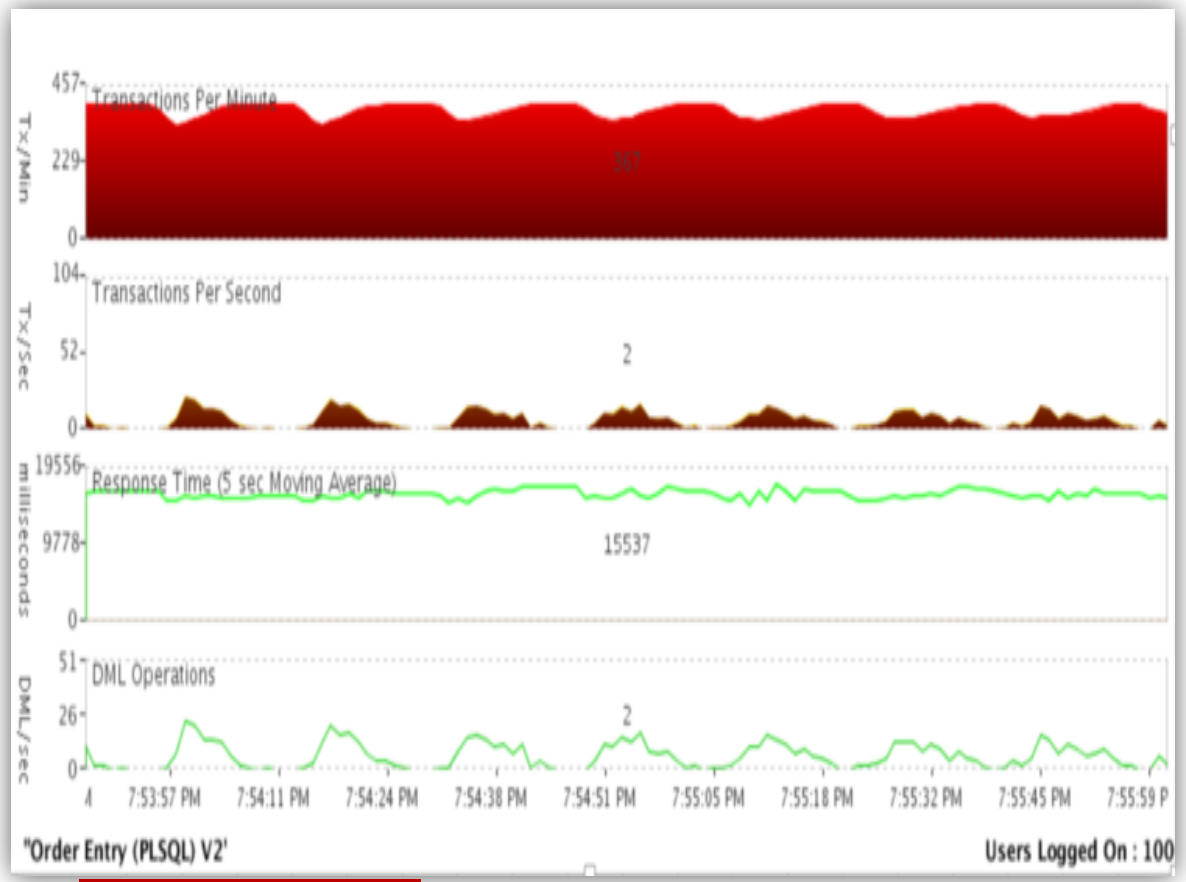


CPU使用率平均76%

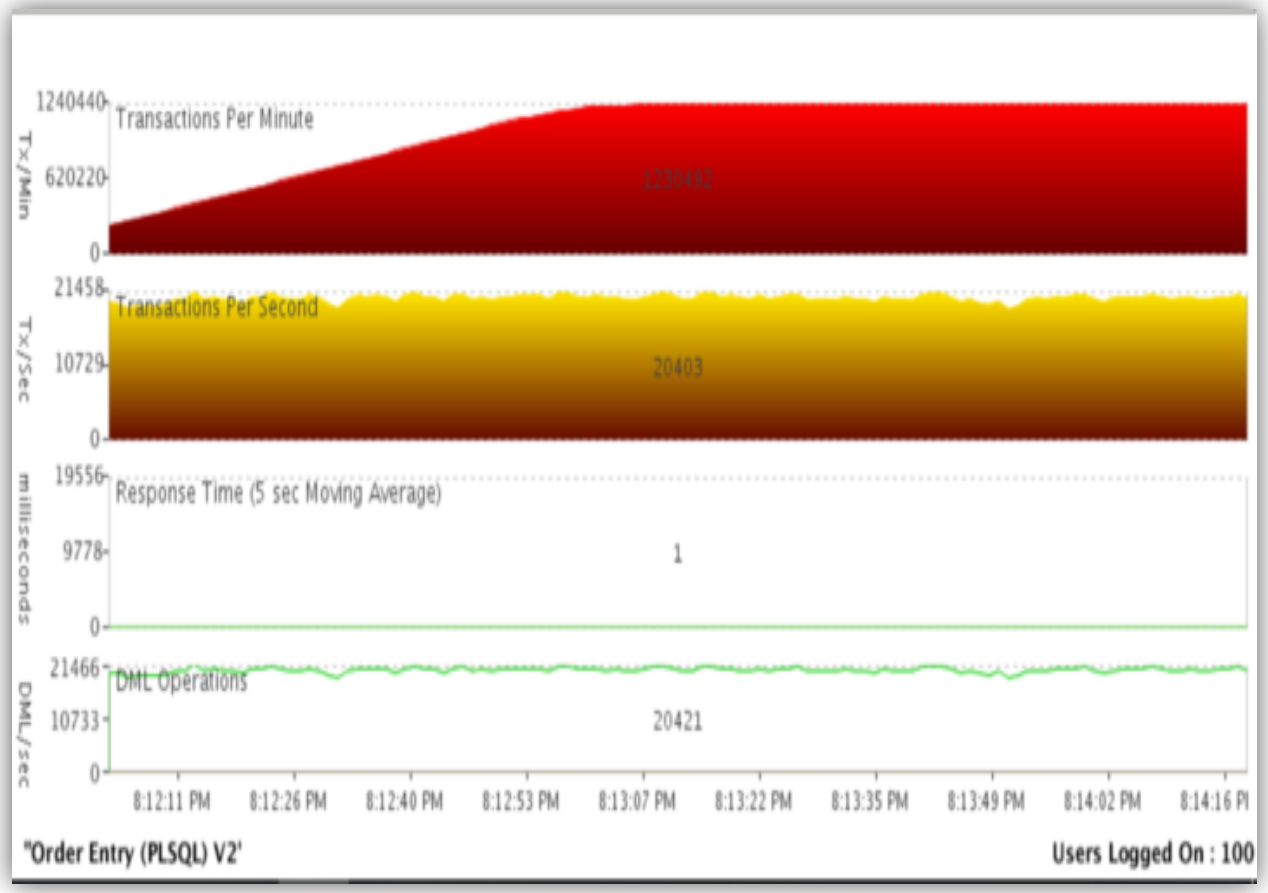
5.Automatic Indexingの有効性□

TPSの比較

INDEX作成前



INDEX作成後



TPS平均12

TPS平均20231

6.ファンクションINDEX作成の方法□

拡張統計が必要

- 5章で有効性検証をする過程でファンクションINDEXが必要になりました□
- 拡張統計の取得でファンクションINDEXが作成されました□

SQLで使用されている2つの関数に対して式統計を取得□

```
SELECT
 DBMS_STATS.CREATE_EXTENDED_STATS('SOE','customers',
'(LOWER("CUST_FIRST_NAME"))')
FROM dual
;

SYS_STUEXM_RW1QTA5FN9AJCNV__#R
```

```
SELECT
 DBMS_STATS.CREATE_EXTENDED_STATS('SOE','customers',
'(LOWER("CUST_LAST_NAME"))')
FROM dual
;

SYS_STU1PEQ_9A63QSAS805132R9WU
```

統計情報を取得□

```
SQL> EXEC dbms_stats.gather_table_stats('SOE','CUSTOMERS');

PL/SQL procedure successfully completed.
```

【参考】拡張統計とは□

- 列グループの統計（Column Groups）□
  - 同一表内の複数列に跨る統計を保持することで、列データ間の相関関係を□考慮したカーディナリティの計算を可能とする□
  - 例：顧客表の住所列と年代列など□
- 式の統計（Expression Statistics）□
  - 関数を含めた統計を保持することで、関数に組み込まれた列を持つWHERE句にお□けるカーディナリティの計算を可能する□
  - 例：Where UPPER（氏名）= :B1

6.ファンクションINDEX作成の方法□

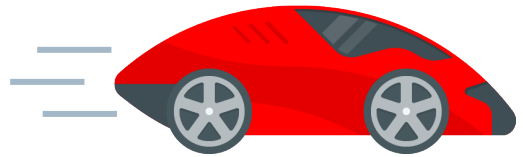
ファンクションINDEXを有　活用するために

- dba_expression_statisticsを確認し、アプリケーションが使用している関数や使用頻度を確認□
- 確認した関数の拡張統計を取得しておく□
- Automatic IndexingにINDEXの作成、有効化はおまかせ□

```
SQL> SELECT table_name, expression_id, snapshot, evaluation_count, fixed_cost, dynamic_cost, expression_text, last_modified FROM dba_expression_statistics
WHERE table_name　'CUSTOMERS' ;
TABLE_NAME         EXPRESSION_ID SNAPSHOT    EVALUATION_COUNT                    FIXED_COST  DYNAMIC_COST EXPRESSION_TEXT                                  LAST_MODIFIED
---------- -------------------- ---------- ---------------- ---------------------------------- ------------ ---------------------------------------- ------------
CUSTOMERS    4127909656998409021 LATEST               368511           .000000040181559571619900            0 "DOB"                                    31-JUL-19
CUSTOMERS    2587915600935826487 LATEST               368391           .000000040181559571619900            0 "CUSTOMER_CLASS"                         31-JUL-19
CUSTOMERS    1595178779089034508 LATEST               368494           .000000040181559571619900            0 "PARTNER_MAILSHOT"                       31-JUL-19
CUSTOMERS    4419066030080625822 LATEST               368482           .000000040181559571619900            0 "PREFERRED_CARD"                         31-JUL-19
CUSTOMERS    2133043334745669795 LATEST               367562           .000000040181559571619900            0 "NLS_TERRITORY"                          31-JUL-19
CUSTOMERS     181965016711O723394 LATEST               374162           .000000040181559571619900            0 "CREDIT_LIMIT"                           31-JUL-19
CUSTOMERS   14579729145343669634 LATEST               368273           .000000040181559571619900            0 "NLS_LANGUAGE"                           31-JUL-19
CUSTOMERS    8671557367734123024 LATEST               368511           .000000040181559571619900            0 "PREFERRED_ADDRESS"                      31-JUL-19
CUSTOMERS    7063228421985036449 LATEST              1901292           .000000040181559571619900            0 "CUST_FIRST_NAME"                        31-JUL-19
CUSTOMERS    2374889430419176534 LATEST               368443           .000000040181559571619900            0 "CUSTOMER_SINCE"                         31-JUL-19
CUSTOMERS     331425284612099477 LATEST              1901298           .000000040181559571619900            0 "CUST_LAST_NAME"                         31-JUL-19
CUSTOMERS    8227810443360716654 LATEST               367436           .000000040181559571619900            0 "MAILSHOT"                               31-JUL-19
CUSTOMERS   10746548067322721712 LATEST               368422           .000000040181559571619900            0 "SUGGESTIONS"                            31-JUL-19
CUSTOMERS   14173907062347677879 LATEST               368431           .000000040181559571619900            0 "ACCOUNT_MGR_ID"                         31-JUL-19
CUSTOMERS   17978643062471019485 LATEST               373158           .000000040181559571619900            0 "CUST_EMAIL"                             31-JUL-19
CUSTOMERS    3526214334234987549 LATEST               374154           .000000040181559571619900            0 "CUSTOMER_ID"                            31-JUL-19
CUSTOMERS   13148729305893542004 LATEST             10266774           .000010045389892905000O                REVERSE("CUSTOMER_ID")                     08-JUL-19
CUSTOMERS    6425135516690749604 LATEST                3563           .000004018155957161990O              0 LOWER("CUST_FIRST_NAME")                   31-JUL-19
CUSTOMERS   13342158372955978992 LATEST                3564           .000004018155957161990O              0 LOWER("CUST_LAST_NAME")                    31-JUL-19
CUSTOMERS   12930902945258007826 LATEST               16037           .000010045389892905000O                SYS_OP_BLOOM_FILTER(:BF0000,"CUSTOMER_ID")  30-JUL-19
```

Agenda

1. 振り返りOracle Database 19c
2. Automatic Indexingの概要
3. 検証内容と結果
4. まとめ



まとめ□

- 従来のアドバイザ機能と異なる機能□
  (アドバイザの場合、実装の判断はDBAに委ねられる)
  DBAのいないシステムでもINDEX管理を自動化できます□
  - DBMS_AUTO_INDEX.CONFIGUREで各種設定を行う□
  - 有効化されたINDEXはその有効性や実行計画の差異をレポートで確認□
  - 自動作成されたINDEXは手動での変更は不可□
  - 自動タスクのCPU負荷は軽微□1コアで稼働□
  - 拡張統計の取得でファンクションINDEXをつくってくれる□

最後に□

Automatic Indexingとの付き合いかた

- 有効化することでDBAの負荷を軽減することができる□
  他のクリエイティブな業務に時間を回しましょう□
- まずは試してください□
  開発環境、いくつかあるDBのうちの1つから動作を確認してみましょう□
- 今後の機能強化にも期待□

是非皆さんも触っていただき、分かったこと、所感など展開してください！□

テック・ナイトアーカイブ資料と お役立ち情報

各回テック・ナイトセッション資料 ダウンロードサイト

oracle technight

しばちょう先生の
試して納得！
DBAへの道

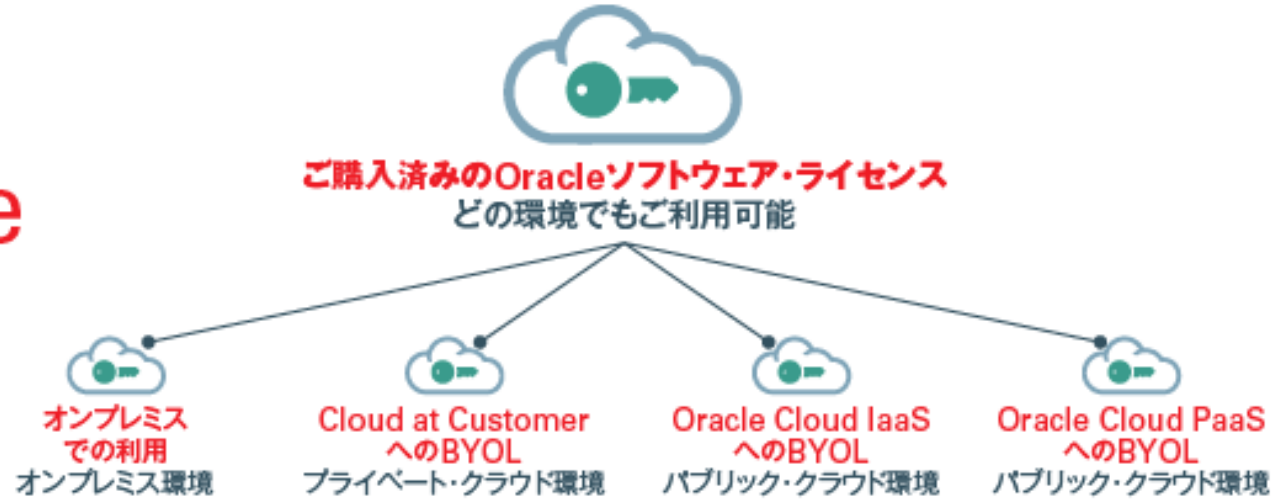
津島博士の
パフォーマンス講座

もしも
みなみんが
DBをクラウドで
動かしてみたら

基本からわかる！
高性能×高可用性
データベースシステム
の作り方

Bring Your Own License

既存のオラクル・ライセンスを
柔軟にクラウド環境で活用



300ドル分の無料トライアルでOracle Cloudを体験!

https://cloud.oracle.com/ja_JP/tryit

Oracle Cloudでは各種クラウドサービスを300ドル分無料でお試しいただけるトライアルサービスをご提供しております。無料トライアルのお申込み方法の詳細は、左のQRコード、またはURLにアクセスしてください。

Oracle Cloudのユースケース、導入事例、資料、価格などの詳細情報は、下記URLにアクセスしてください。
http://www.oracle.com/jp/cloud/platform/overview/index.html

ORACLE

こんな時、かけこむ会社が増えています。

Oracle Digitalは、オラクル製品の導入をご検討いただく際の総合窓口。
電話とインターネットによるダイレクトなコニュニケーションで、どんなお問い合わせにもすばやく対応します。
もちろん、無償。どんなことでも、ご相談ください。

Oracle Digital

お問い合わせは電話またはWebフォーム

☎ 0120-155-096
受付時間　月～金　9:00-12:00 / 13:00-17:00
（祝日および年末年始休業日を除きます）

http://www.oracle.com/jp/contact-us

ORACLE

Integrated Cloud

Applications & Platform Services


ORACLE®