

# Case Study

## Tuning a Fast and Frequently Executed SQL

One of the top 3 heaviest SQLs in customer's environment was a fast statement (average execution of **0.013 sec**) executed **1.5 million (!)** times each day.

Obviously, due to its massive executions, even a small improvement in its execution should yield a huge performance improvement to its total resource consumption, and that is what I wanted to establish.

After I examined SQL's execution plan and used index, I found that the columns order of the index is not effective, therefore I have changed columns order, and managed to improve SQL's performance.

Average execution went down **from 0.013 sec to 0.005 sec => 2.6 times faster!**

As a result, since it was executed frequently, SQL's overall resource consumption went down dramatically.

Precise for Oracle

History

StartPoint

AdminPoint

Favorites

Settings

Actions

Precise for Oracle

Dashboard

Current

Activity

Objects

SQL

What-If

Statistics

SmartTune

Time: 16-Jan-11 07:00 - 17-Jan-11 07:14

Instance: Inst1 Running on machine1

Filter is Off...

Inst1 Running on machine1

Table View

Tree View

Inst1(machine1) [5d,19h]

SQL Statements [Top-20 sorted by In Oracle]

SQL 10452.21049.18309.54122 [05:25:39.2] Tune

SQL 42942.30505.53068.31251 [21:15:21.2] Tune

SQL 22736.00390.26692.03534 [07:51:41.7] Tune

SQL 11465.08146.41740.37109 [03:51:16.7] Tune

SQL 03762.25304.18143.39923 [03:43:02.3] Tune

SQL 65304.09433.64144.28000 [03:03:49.5] Tune

SQL 18419.01896.52078.48688 [02:55:34.3] Tune

SQL 60150.35035.41386.63714 [02:53:02.4] Tune

SQL 22491.61636.44338.30243 [02:28:49.5] Tune

SQL 59126.41953.22844.28663 [02:16:26.0] Tune

SQL 09787.49171.18947.39102 [02:09:15.6] Tune

SQL 02011.51444.11374.23843 [02:08:20.7] Tune

SQL 35831.51181.14485.55931 [02:00:26.3] Tune

SQL 18269.12027.54770.33426 [01:51:15.9] Tune

SQL 05501.46584.34383.28707 [01:45:09.9] Tune

SQL 24542.12899.24067.37489 [01:22:58.0] Tune

SQL 47123.65225.58345.26511 [01:16:30.4] Tune

SQL 28706.17632.10024.18228 [01:08:16.2] Tune

SQL 14373.26047.09240.32716 [01:07:53.5] Tune

SQL 54054.51950.53180.19989 [01:07:27.9] Tune

Programs

Objects

Execution Plans

Users

Machines

Modules

Actions

Oracle Files

Host Users

PL/SQLs

Statement: 10452.21049.18309.54122

Over Time

Overview

Bind Variables

Dictionary

In Oracle (Summed): 05:25:39.2

Executions: 1498 K

In Oracle (Avg): 00:00:00.013

Duration (Avg): 00:00:00.013

End Of Fetch Count: 1498 K

Bind Variables Captured: 8

Buffer Gets (Avg): 143.85

Rows Processed (Avg): 3.76

Parallel Servers (Min): 0

Parallel Servers (Max): 0

In Oracle (Summed)

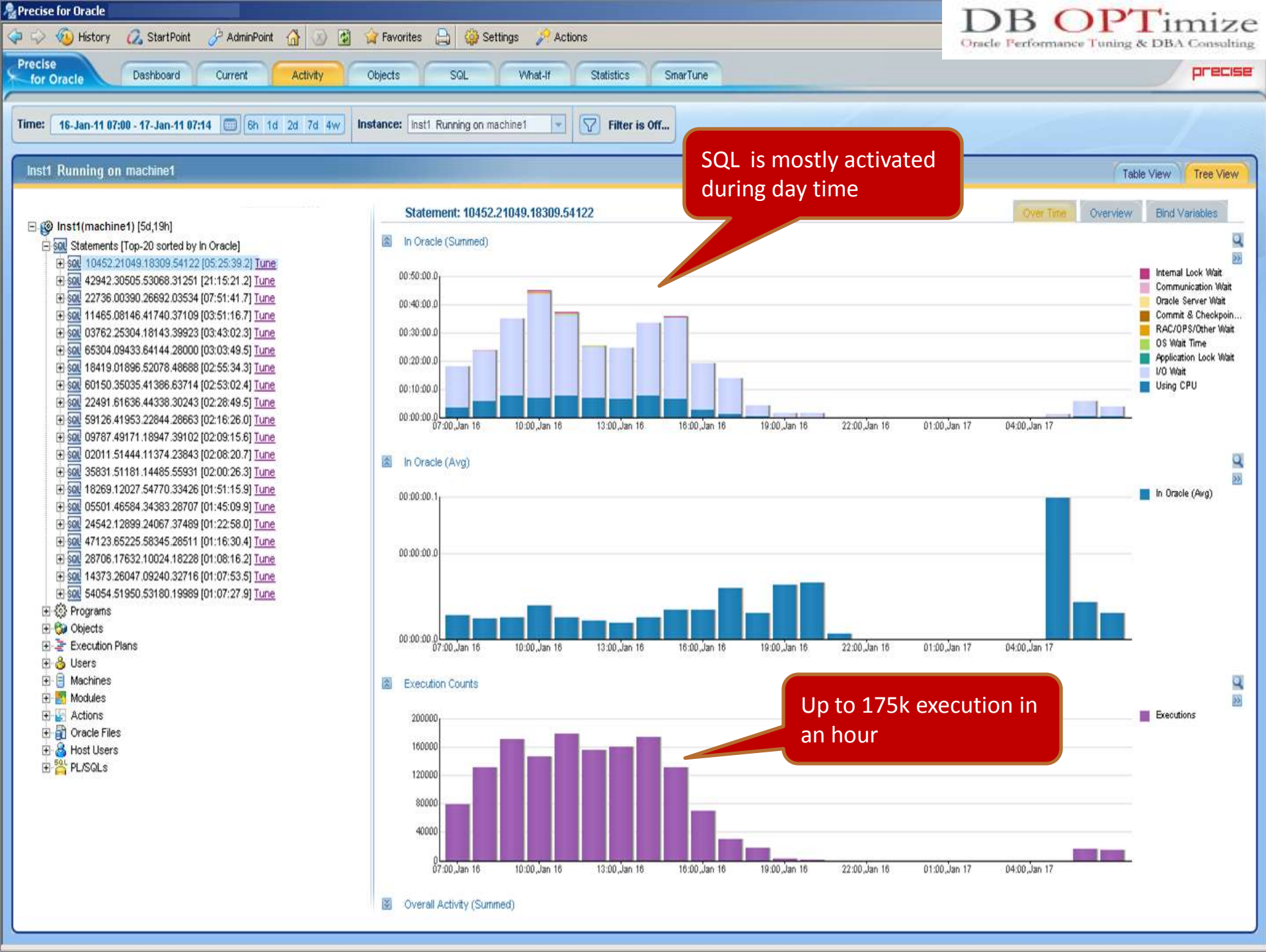
Sub-State	Time	%
I/O Wait	04:18:44.5	79.45%
Using CPU	01:06:44.7	20.49%
Oracle Client Comm Wait	00:00:03.0	0.01%
Memory Wait	00:00:03.0	0.01%
Buffer Wait	00:00:02.0	0.01%
Other Wait	00:00:01.0	0.00%
Shared Pool Wait	00:00:01.0	0.00%

Text

select \*  
from tab1  
where col\_a = :1  
order by col\_b

Frequently executed SQL, executing 1.5M times each day, with average execution of 0.013 sec.

SQL text



Precise for Oracle

History

StartPoint

AdminPoint

Favorites

Settings

Actions

Precise for Oracle

Dashboard

Current

Activity

Objects

SQL

What-If

Statistics

SmartTune

Re-Explain

Time: 16-Jan-11 07:00 - 17-Jan-11 07:14

6h 1d 2d 7d 4w

10452.21049.18309.54122 (in Inst1 Running on machine1 )

Plan

Recommend

Run Alternatives

More...

Execution Plans

Real plan: 2958899901 [05:25:39]

[3] Select Statement (Optimizer mode: CHOOSE)

[2] Table Access (By Index Rowid) TAB1

[1] Index (Skip Scan) TAB1\_INX3

Real plan: 766973358

Step 0 of 3

Real Execution Plan loaded on Jan 16, 2011 06:30

Highlights

Expanded Text

Objects

More...

Text

select \*

from tab1

where col\_a = :1

order by col\_b

Findings

Locate	Type	Object	Impact (%)	In Oracle
	Heavy Sequential I/O on Index	TAB1_INX3	46%	
The statement spent 46% of its resources waiting for <b>Sequential I/O</b> on the specified index.				
<b>Learn more</b> or proceed with the following:				
<ul style="list-style-type: none"><li>Click the <b>Locate</b> icon in order to find the relevant step in the execution plan.</li><li>Examine objects <a href="#">structure and I/O activity</a>.</li><li>Examine <a href="#">index activity</a> of the statement over time.</li></ul>				
	Heavy Sequential I/O on Table	TAB1	31%	
	Newer Execution Plan Exists			
	Bind Variables Were Collected			

Current Plan I/O Wait: 02:29:56.6

Text

Expanded Text

select \*

from tab1

where col\_a = :1

order by col\_b

46% of SQL activity comes from Index Scans on TAB1\_INX3

31% of SQL activity come from fetching the data from table TAB1 itself (select \*)



(3) Since index is build with COL\_B as the first column, and COL\_A only as the second column, Oracle chooses to scan TAB1\_INX3 index using skip scan, then it access TAB1 to fetch SELECT star. Obviously, since the SQL is asking "col\_a = :1", and since COL\_A is very selective, Oracle will benefit more from an index which has COL\_A as its first column.

(1) When examining the SQL we can see that :

1. Where clause is: col\_a = :1
2. Order By clause is on col\_b

```
select *
from tab1
where col_a = :1
order by col_b
```

(2) Index TAB1\_INX3 on TAB1(COL\_B, COL\_A). COL\_A is very selective: has 5.25M distinct values. COL\_B is not selective: has only 236 distinct values.

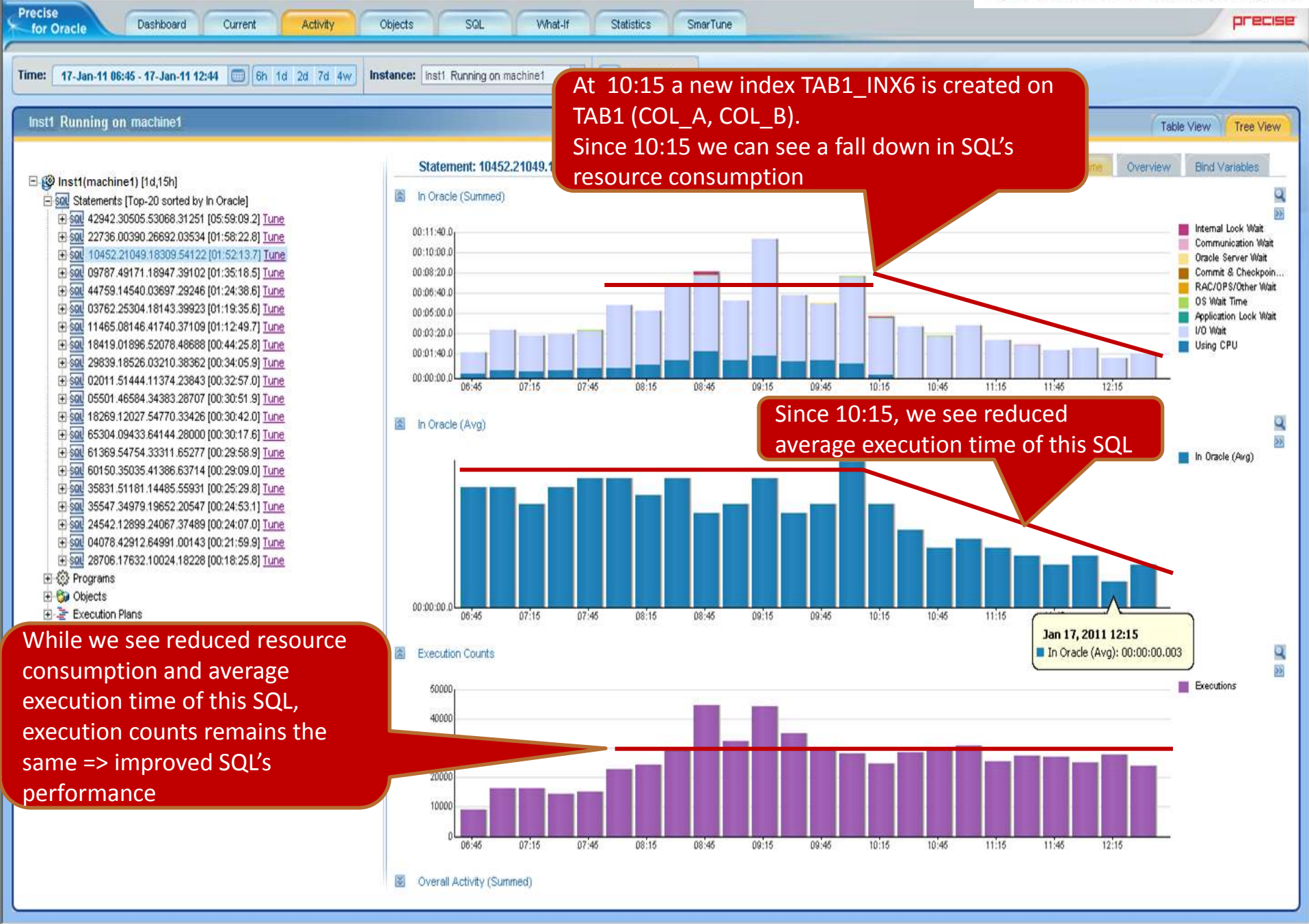
	Locate	Used	Table	I/O Wait	Rows	Blocks	Non-Empty Blocks	Last
TUNE		✓	TAB1		23264973	453120	452192	Jan

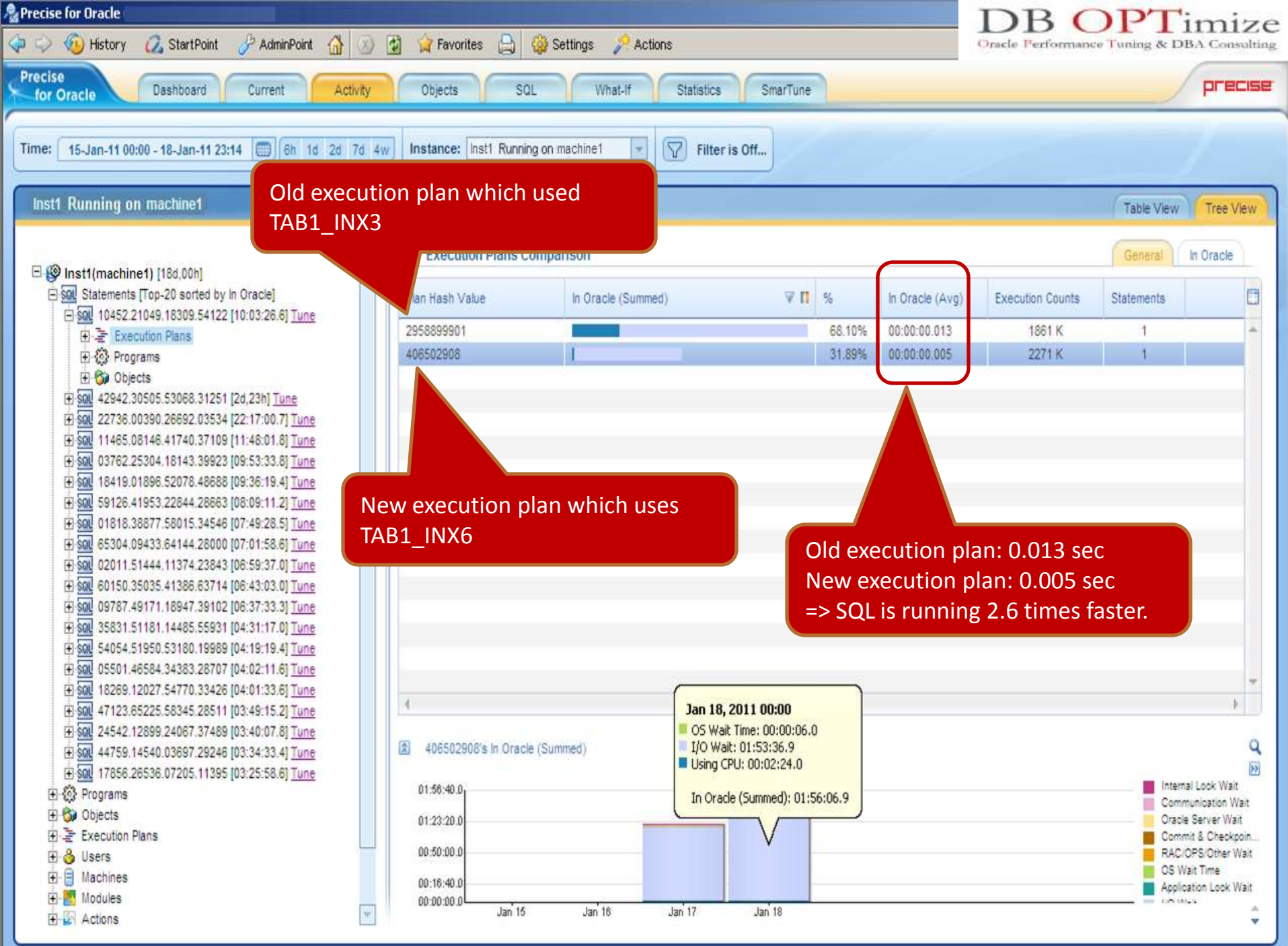
Indexes defined on TAB1

	Locate	Used	Index	I/O Wait	Unique	Type	Partitioned	Blocks
TUNE		✓	TAB1_INX3		No	Normal	No	119
TUNE			INX6		No	Normal	No	675
TUNE			TEST_1		No	Normal	No	163
TUNE			INX4		No	Normal	No	968
TUNE			INX5		No	Normal	No	104
TUNE			INX1		No	Normal	No	179
TUNE			INX2		No	Normal	No	189

Columns in table TAB1

	Column	Type	Distinct Values	Key Number	Appears In	Indexable
A1	COL_B	Number(10,0)	236	1	Select, Order by	No
A1	COL_A	Number(10,0)	5252443	2	Select, Where	Yes
	COL1	Number(10,0)	19		Select	No
	COL2	Number(10,0)	60		Select	No
	COL3	Number(10,0)	2		Select	No
	COL4	Number(10,0)	3		Select	No
	COL5	Varchar2(15)	1688		Select	No







Precise for Oracle

History

StartPoint

AdminPoint

Favorites

Settings

Actions

Precise for Oracle

Dashboard

Current

Activity

Objects

SQL

What-If

Statistics

SmartTune

Re-Explain

Time: 17-Jan-11 05:00 - 17-Jan-11 10:59

6h 1d 2d 7d 4w

10452.21049.18309.54122 (in Inst1 Running on machine1 )

Plan

Recommend

Run Alternatives

More...

Execution Plans

Real plan: 2958899901 [01:28:17]

[3] Select Statement (Optimizer mode: CHOOSE)

[2] Table Access (By Index Rowid) TAB1

[1] Index (Skip Scan) TAB1\_INX3

Real plan: 406502908 [00:10:33]

[3] Select Statement (Optimizer mode: CHOOSE)

[2] Table Access (By Index Rowid) TAB1

[1] Index (Range Scan) TAB1\_INX6

Real plan: 766973358

Real Execution Plan loaded on Jan 17, 2011 10:15

Text

Expanded Text

Objects

More...

select t

from tab1

where col\_a = :1

order by col\_b

SQL is now using INDEX RANGE SCAN on the new created index TAB1\_INX6

Findings

Locate	Type	Object	Impact(%)	In Oracle
	Heavy Sequential I/O on Index	TAB1_INX3	42%	<div></div>
<div>The statement spent 42% of its resources waiting for <b>Sequential I/O</b> on the specified index.</div> <div> <b>Learn more or proceed with the following:</b></div> <ul style="list-style-type: none"><li>Click the <b>Locate</b> icon in order to find the relevant step in the execution plan.</li><li>Examine objects <a href="#">structure and I/O activity</a>.</li><li>Examine <a href="#">index activity</a> of the statement over time.</li></ul>				
	Heavy Sequential I/O on Table	TAB1	35%	<div></div>
<div> <b>More Than One Real Plan Was Detected</b></div> <div> <b>Bind Variables Were Collected</b></div>				

Precise for Oracle

History

StartPoint

AdminPoint

Favorites

Settings

Actions

Precise for Oracle

Dashboard

Current

Activity

Objects

SQL

What-If

Statistics

SmaTune

Time: 17-Jan-11 05:00 - 17-Jan-11 10:59

6h 1d 2d 7d 4w

Re-Explain

10452.21049.18309.54122 (in Inst1 Running on machine1 )

Plan

Recommend

Run Alternatives

More...

Execution Plans

Real plan: 2958899901 [01:28:17]

[3] Select Statement (Optimizer mode: CHOOSE)

[2] Table Access (By Index Rowid) TAB1

[1] Index (Skip Scan) TAB1\_INX3

Real plan: 406502908 [00:10:33]

[3] Select Statement (Optimizer mode: CHOOSE)

[2] Table Access (By Index Rowid) TAB1

[1] Index (Range Scan) TAB1\_INX6

Real plan: 766973358

Step 1 of 3

Real Execution Plan loaded on Jan 17, 2011 10:15

Highlights

Expanded Text

Objects

More...

Tables in use

	Locate	Used	Table	I/O Wait	Rows	Blocks	Non-Empty Blocks	Last
TUNE		✓	TAB1		23264973	453120	452192	Jan

Indexes defined on TAB1

	Locate	Used	Index	I/O Wait	Unique	Type	Partitioned	Blocks
TUNE			TAB1_INX3		No	Normal	No	119
TUNE		✓	TAB1_INX6		No	Normal	No	675
TUNE			TEST_1		No	Normal	No	163
TUNE			INX4		No	Normal	No	968
TUNE			INX5		No	Normal	No	104
TUNE			INX1		No	Normal	No	179
TUNE			INX2		No	Normal	No	189

Columns in table TAB1

	Column	Type	Distinct Values	Key Number	Appears In	Indexable
A↑	COL_A	Number(10,0)	5252443	1	Select,Where	Yes
A↑	COL_B	Number(10,0)	236	2	Select,Order by	No
	COL1	Number(10,0)	19		Select	No
	COL2	Number(10,0)	60		Select	No
	COL3	Number(10,0)	2		Select	No
	COL4	Number(10,0)	3		Select	No
	COL5	Varchar2(15)	1688		Select	No

select \_  
from tab1  
where col\_a =:1  
order by col\_b

Text

Expanded Text

(2) Index TAB1\_INX6 on TAB1 (COL\_A, COL\_B)

# DB OPTimize

Oracle Performance Tuning & DBA Consulting

[www.dboptimize.co.il](http://www.dboptimize.co.il)

[merav@dboptimize.co.il](mailto:merav@dboptimize.co.il)