

Oracle In-Memory – and all that

Paolo Kreth, DBA, Head of team Datamanagement, paolo.kreth@mobiliar.ch

Andreas Wyssenbach, Senior DBA andreas.wyssenbach@mobiliar.ch

July 2017

Agenda

1. Swiss Mobiliar in a nut shell

2. Use case

3. Results

4. Q & A

No.1 in Switzerland

We are Switzerland's number one for household contents, business and pure risk life insurance.

160 locations

79 general agencies and their roughly 80 agency offices ensure that we are close to customers throughout the country.



Nine out of ten claims are settled locally and directly by our general agencies.

CHF 3.6 bn

We managed to raise the premium volume by 3.2% in the year under review.

CHF 155 m

This is the amount we voluntarily remit to customers from our surplus fund.

5,259 employees

327 of which are trainees

24 / 7

Swiss Mobiliar's 24h Assistance service is available for customers around the clock, 365 days a year.

> CHF 30 m

We have been supporting public prevention of natural hazards throughout Switzerland since 2005.

CHF 1.7 m

private individuals and corporate customers place their trust in us.

Founded in 1826

Swiss Mobiliar is the oldest private insurance company in Switzerland.

CHF 439.5 m

Swiss Mobiliar once again posts an outstanding annual profit.

die Mobiliar - our IT department

460 Empl.
430 FTE



160 Sites
connected by IT-Network
throughout Switzerland
incl. VoIP

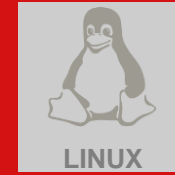
6'000 Clients



1'800 Smartphones



Budget 2016
about 125 Mio. CHF



Systems

Programming languages



Data
Bases



Standard Software



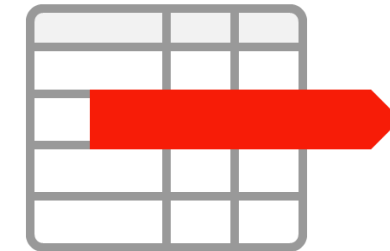
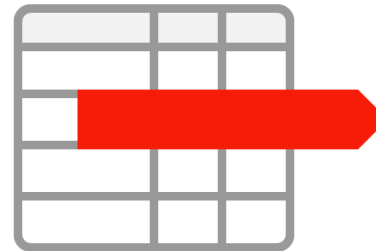
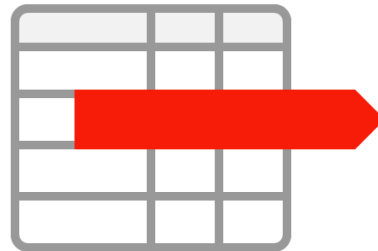
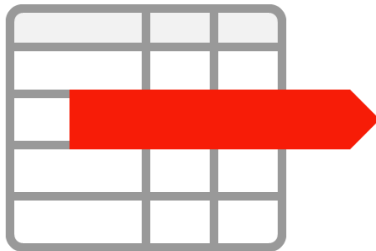
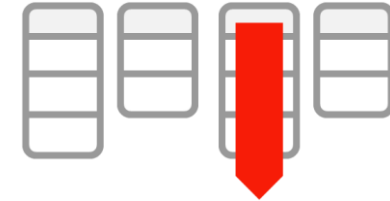
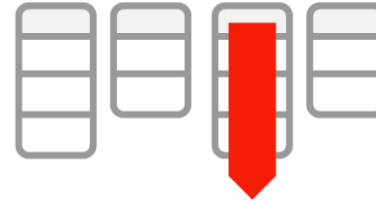
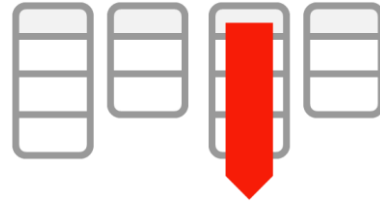
Our In-Memory journey



2014

2015

2016



Oracle 11g R2

- rows only
- rely on indexes
- rely on mat-views
- rely on partitioning
- rely on parallel execution

Oracle 12c β

- added In-Memory
- tested compression
- tested robustness
- tested usability
- tested performance

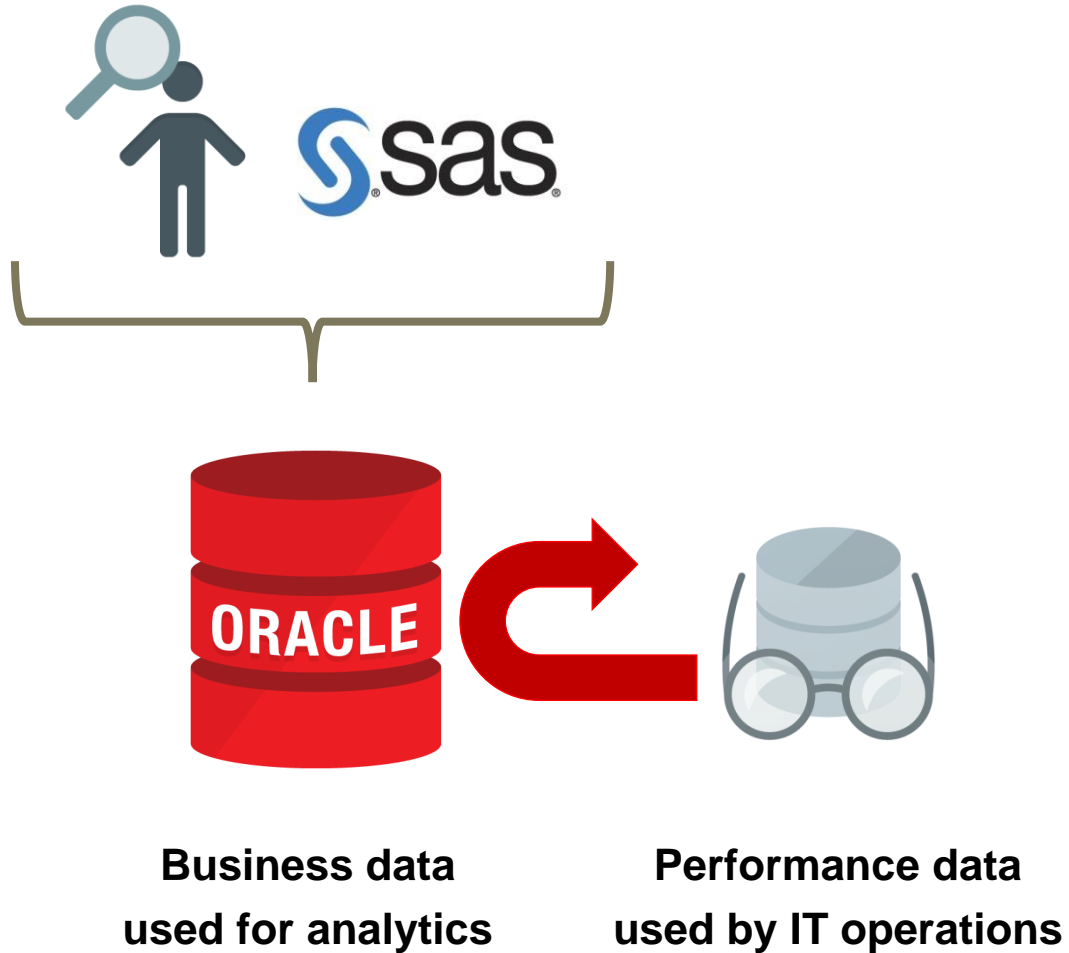
Oracle 12c

- gained practical experience
- gained performance with bundle patches

Oracle 12.2 β

- tested again
- remarkable improvements
- remarkable new features

Use case – why is it relevant to us?



- Insurance is about offering products and managing risk. Compliance reporting to authorities and auditors is critical.
- We've created *our own application RiCo «Risk Controlling»* to keep track of information.
- Our teams in actuary, product management and risk management use it.
- We separated data platform (Oracle) and analytics(SAS). This system used to be the first SAS customer in Switzerland.
- The data platform has always been demanding:
 - Simple, yet flexible data model
 - Increasing data volume
 - Increasing complexity of analytic queries
- We track resource consumption using *our Performance DWH*.

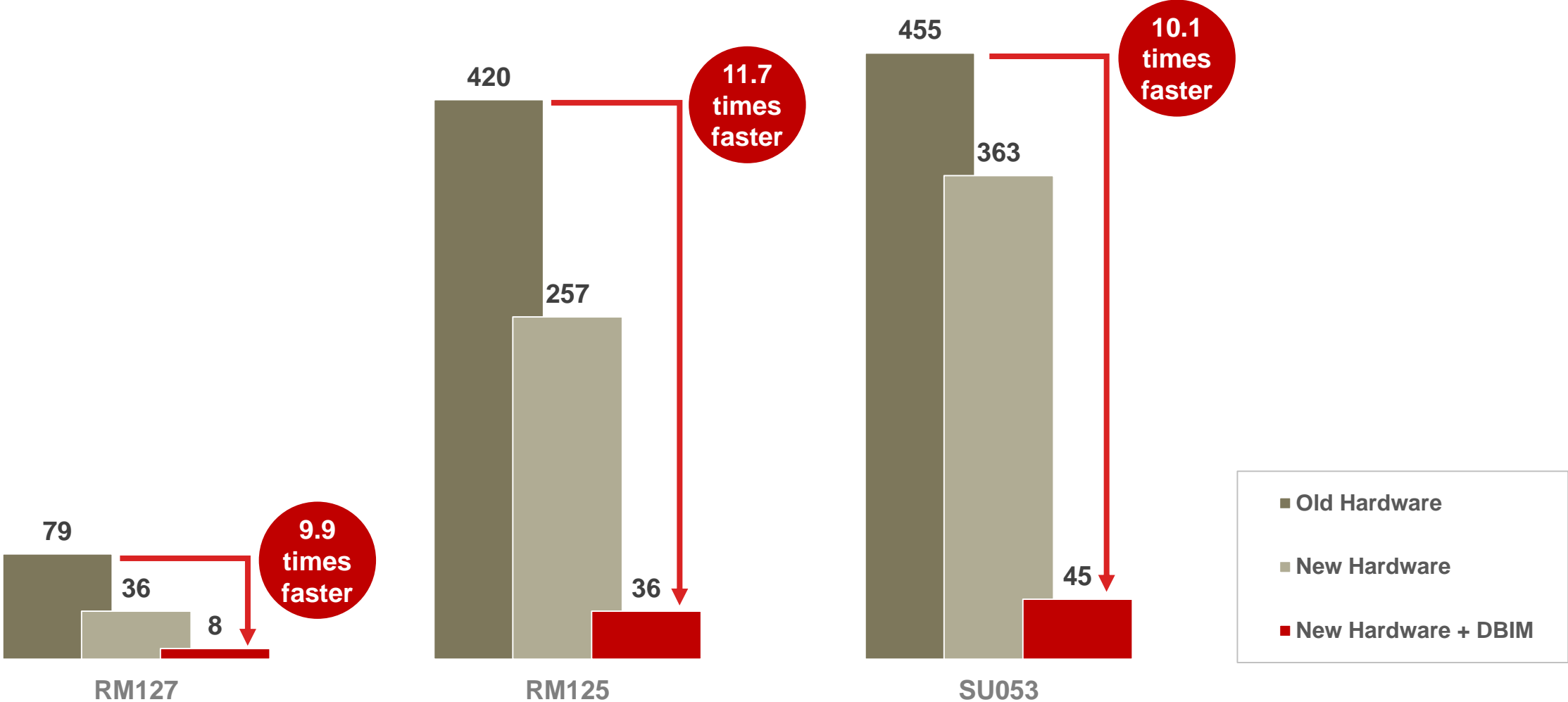
Use case – the human factor



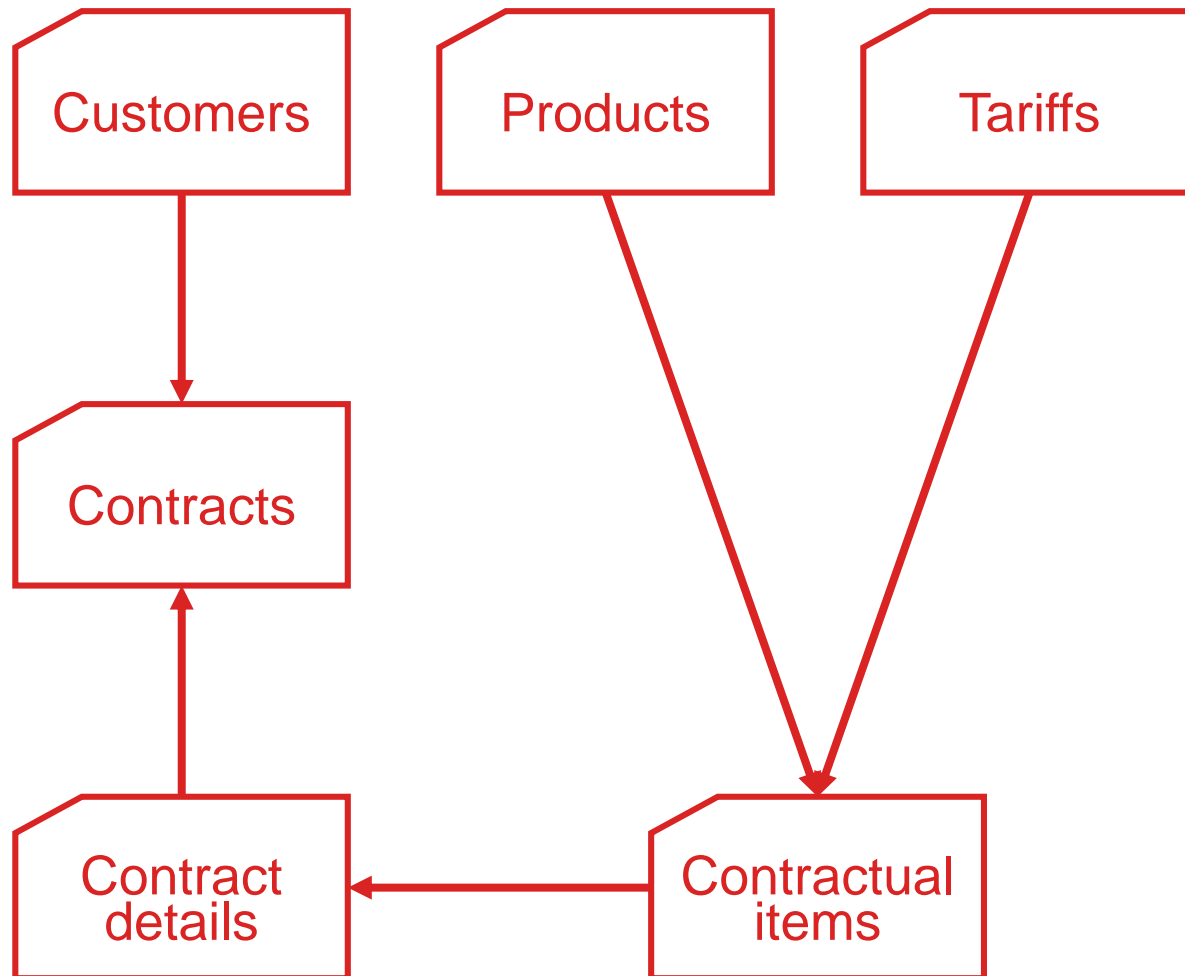
- A team of 5+ people works for actuary exclusively.
- Analytics is done using SAS because it is flexible and because we have a vast experience (25+ years in place).
- IT implementation of the overall RiCo application is robust and subject to business driven changes.
- Insights are needed but the actual analysis is the result of several steps that need to be determined.
- Response time is always a key concern, because analysis work is mainly at high stress level. Waiting for more than 2 minutes is not an option.

Rico tests on December 2nd 2016

Job duration in minutes

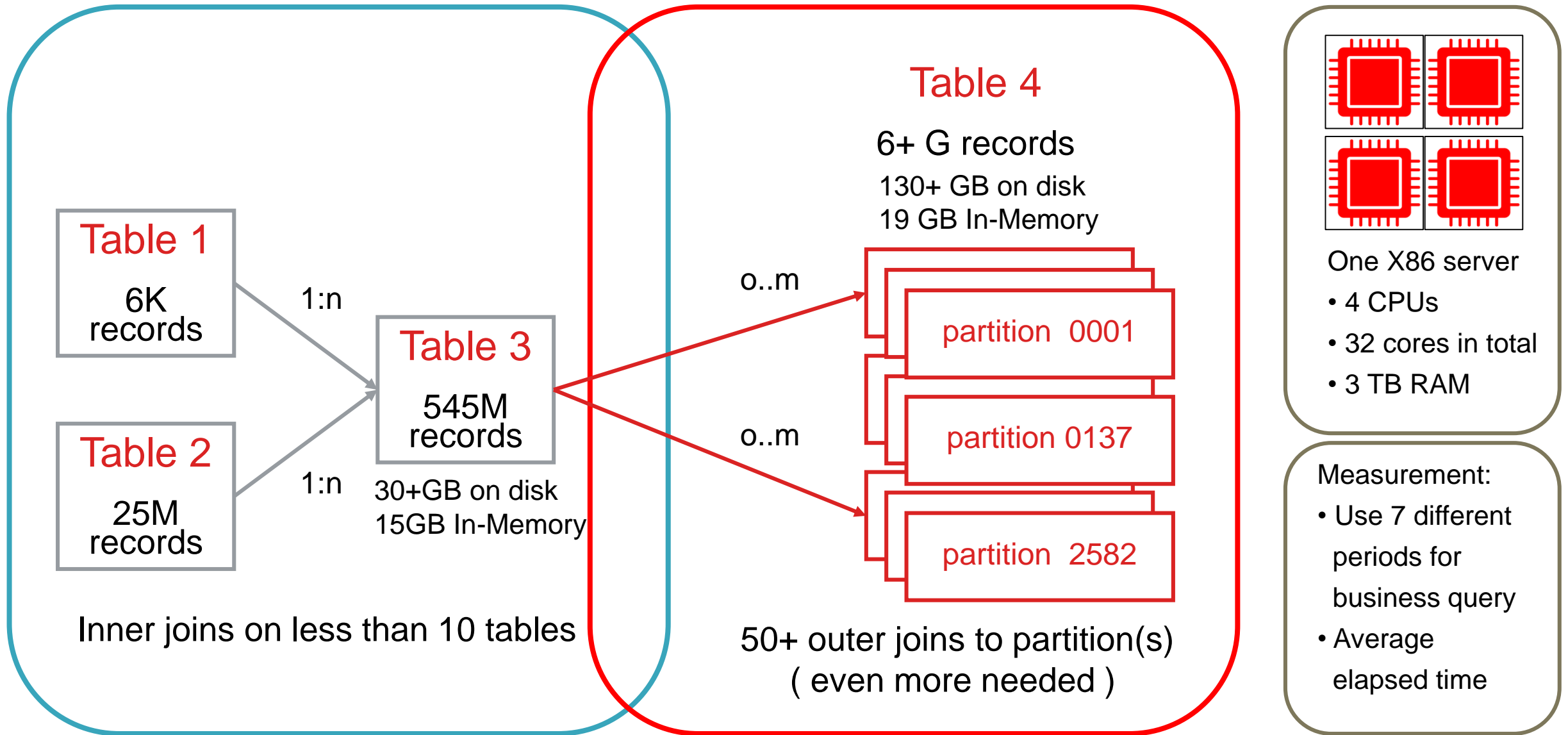


Use case – logical view



- Everybody needs information about their:
 - customers, contracts
 - products, tariffs, contractual items
 - contract details including their entire history (10+ years).
- Everybody wants flexibility when adding / changing offerings (insurance products, services)
- Over time the complexity of analytic queries rises:
 - more and increasingly complex filtering
 - comparison of data sets on the fly
 - more data sources to be joined
 - Links and hierarchies need to be traversed
- Typical questions:
 - *Which contractual items have been applicable in a given period of time?*
 - *What is the benefit / impact when changing a tariff?*

Use case – technical view – at the core of most analysis steps



Analysing the risk factor of no less than 150 tariff components leads to an execution plan having 800+ lines

Id	Operation	Name	Starts	E-Rows	A-Rows	A-Time	Buffers
0	SELECT STATEMENT		1		0	00:02:35.00	3233
1	PX COORDINATOR		1		0	00:02:35.00	3233
2	PX SEND QC (ORDER)	:TQ10148	0	505K	0	00:00:00.01	0
3	SORT ORDER BY		0	505K	0	00:00:00.01	0
4	PX RECEIVE		0	505K	0	00:00:00.01	0
5	PX SEND RANGE	:TQ10147	0	505K	0	00:00:00.01	0
* 6	HASH JOIN OUTER BUFFERED		0	505K	0	00:00:00.01	0
7	JOIN FILTER CREATE	:BF0000	0	482K	0	00:00:00.01	0
* 8	HASH JOIN OUTER		0	482K	0	00:00:00.01	0
9	JOIN FILTER CREATE	:BF0001	0	477K	0	00:00:00.01	0
* 10	HASH JOIN OUTER		0	477K	0	00:00:00.01	0
11	JOIN FILTER CREATE	:BF0002	0	473K	0	00:00:00.01	0
* 12	HASH JOIN OUTER		0	473K	0	00:00:00.01	0
13	JOIN FILTER CREATE	:BF0003	0	471K	0	00:00:00.01	0
* 14	HASH JOIN OUTER		0	471K	0	00:00:00.01	0
15	JOIN FILTER CREATE	:BF0004	0	466K	0	00:00:00.01	0
* 16	HASH JOIN OUTER		0	466K	0	00:00:00.01	0
17	JOIN FILTER CREATE	:BF0005	0	462K	0	00:00:00.01	0
* 18	HASH JOIN OUTER		0	462K	0	00:00:00.01	0
19	JOIN FILTER CREATE	:BF0006	0	457K	0	00:00:00.01	0
* 20	HASH JOIN OUTER		0	457K	0	00:00:00.01	0
21	JOIN FILTER CREATE	:BF0007	0	450K	0	00:00:00.01	0
* 22	HASH JOIN OUTER		0	450K	0	00:00:00.01	0
23	JOIN FILTER CREATE	:BF0008	0	447K	0	00:00:00.01	0
* 24	HASH JOIN OUTER		0	447K	0	00:00:00.01	0
25	JOIN FILTER CREATE	:BF0009	0	446K	0	00:00:00.01	0
* 26	HASH JOIN OUTER		0	446K	0	00:00:00.01	0
27	JOIN FILTER CREATE	:BF0010	0	440K	0	00:00:00.01	0
* 28	HASH JOIN OUTER		0	440K	0	00:00:00.01	0
29	JOIN FILTER CREATE	:BF0011	0	440K	0	00:00:00.01	0
* 30	HASH JOIN OUTER		0	440K	0	00:00:00.01	0
31	JOIN FILTER CREATE	:BF0012	0	440K	0	00:00:00.01	0
* 32	HASH JOIN OUTER		0	440K	0	00:00:00.01	0
33	JOIN FILTER CREATE	:BF0013	0	440K	0	00:00:00.01	0
* 34	HASH JOIN OUTER		0	440K	0	00:00:00.01	0
35	JOIN FILTER CREATE	:BF0014	0	434K	0	00:00:00.01	0
* 36	HASH JOIN OUTER		0	434K	0	00:00:00.01	0
37	JOIN FILTER CREATE	:BF0015	0	434K	0	00:00:00.01	0
* 38	HASH JOIN OUTER		0	434K	0	00:00:00.01	0
39	JOIN FILTER CREATE	:BF0016	0	434K	0	00:00:00.01	0
* 40	HASH JOIN OUTER		0	434K	0	00:00:00.01	0
41	JOIN FILTER CREATE	:BF0017	0	433K	0	00:00:00.01	0

Example: inner join (3 tables) and outer joins (55 tables) 150k result set

ORACLE Enterprise Manager Database Active Report

Monitored SQL Execution Details: 3wq4d93n4gpd9

Overview

General

SQL Text: `select /*+ parallel(16) */ * from (s ...`

Execution Plan: 16

Execution Started: Tue Sep 6, 2016 10:01:13 AM

Last Refresh Time: Tue Sep 6, 2016 10:01:39 AM

Execution ID: 16777216

User:

Fetch Calls: 0

Time & Wait Statistics

Duration: 26.0s

Database Time: 7.1m

PL/SQL & Java: 0s

Activity %: 100

DOP=16:26sec elapsed & 425 sec DB time

IO Statistics

Buffer Gets: 18K

IO Requests: 423K

IO Bytes: 23GB

I/O 23 GB & idle buffer cache

Details

Plan Statistics | Plan | Parallel | Activity | Metrics

Display: Tabular | Plan Hash Value: 2282879099 | Plan Note

Operation	Object	Line ID	Predicate	Pruning	Operation Cost	Estimated Rows	Estimated Bytes
└─ PX BLOCK ITERATOR		260				195M	2,735M
└─ TABLE ACCESS INMEMORY FULL	T	261		883	664	195M	2,735M
└─ PX RECEIVE		262				180M	2,881M
└─ PX SEND HASH	:TQ10046	263				180M	2,881M
└─ JOIN FILTER USE	:BF0001	264				180M	2,881M
└─ PX BLOCK ITERATOR		265				180M	2,881M
└─ TABLE ACCESS INMEMORY FULL		266		585	653	180M	2,881M
└─ PX RECEIVE		267				22M	865M
└─ PX SEND HASH	:TQ10050	268				22M	865M
└─ JOIN FILTER USE	:BF0000	269				22M	865M
└─ PX BLOCK ITERATOR		270				22M	865M
└─ TABLE ACCESS INMEMORY FULL	TP	271			197	22M	865M

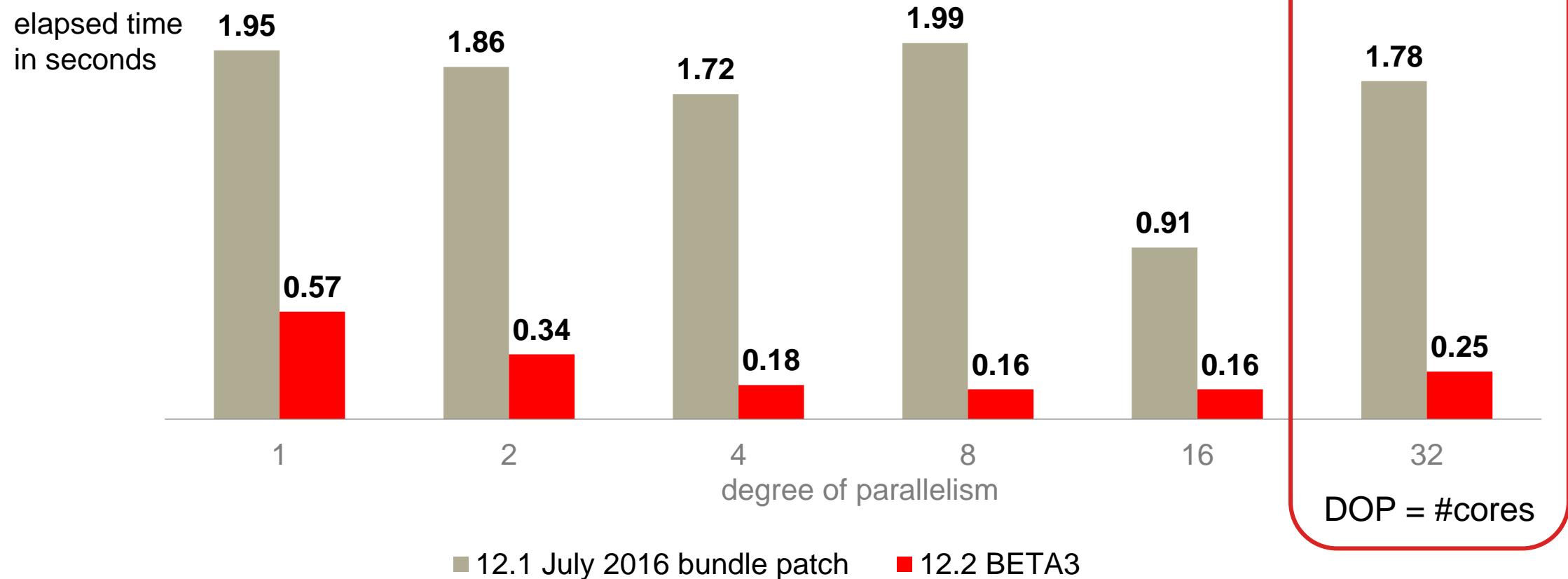
Partition pruning in action

Execution plan: 271 lines; no visible parsing effort

Copyright © 1996, 2016, Oracle and/or its affiliates. All rights reserved. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Advances since 12.1 – push-down of COUNT() and other functions

Oracle In-Memory - COUNT(*) for varying DOP
inner joins only

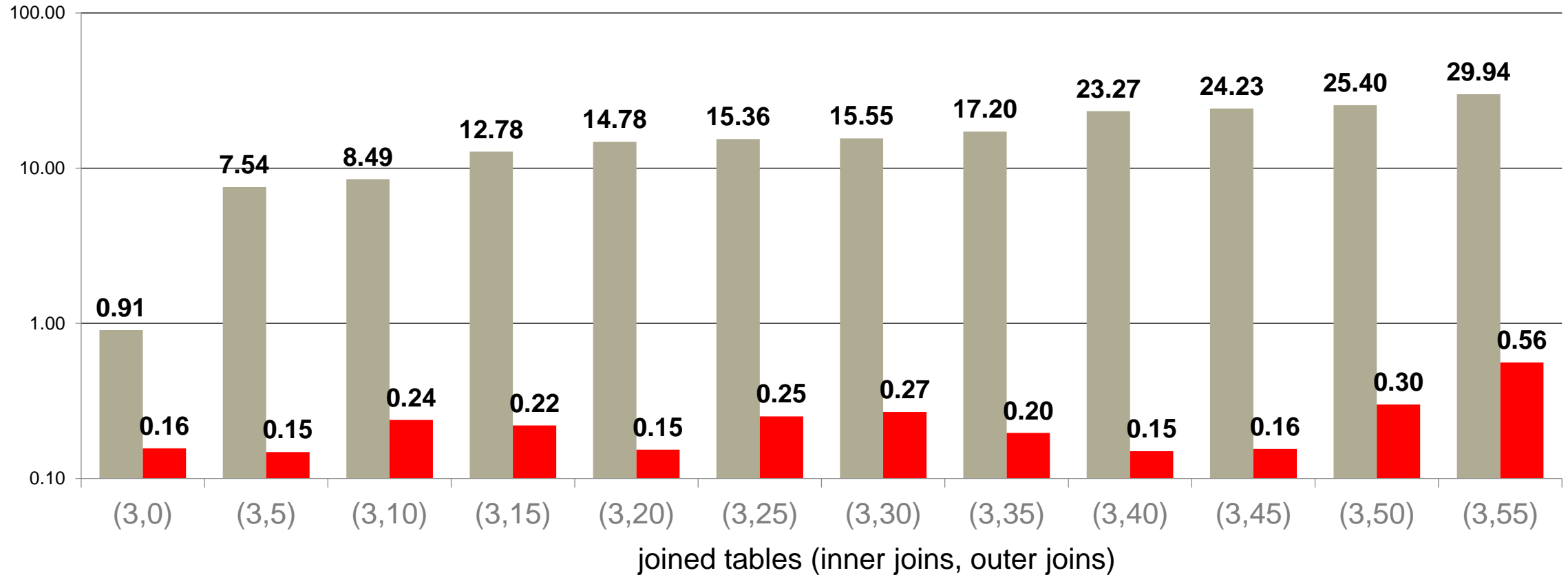


Advances since 12.1 – push-down of COUNT() and other functions

elapsed time
in seconds

Oracle In-Memory - COUNT(*), DOP = 16

inner joins AND increasing number of outer joins to 100k to 140M records

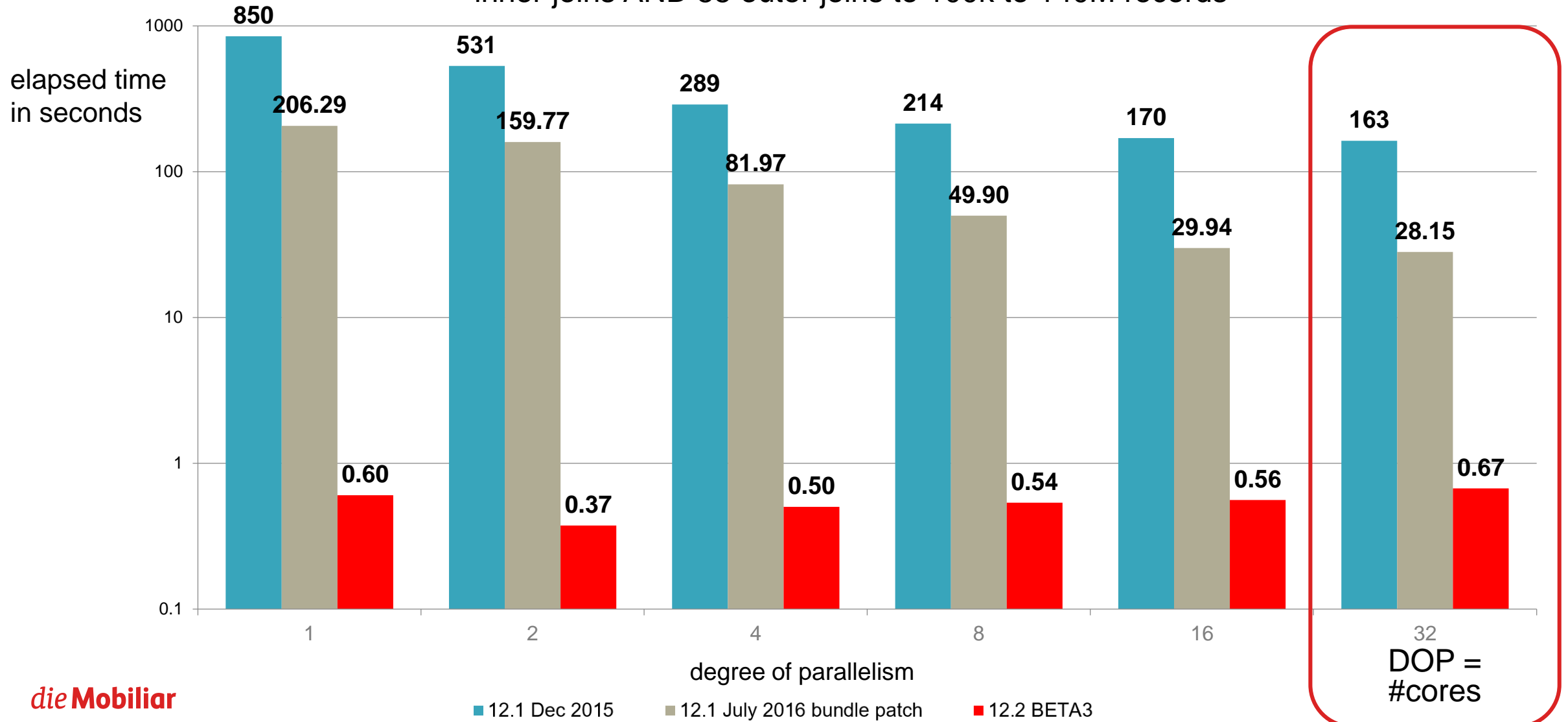


■ 12.1 July 2016 bundle patch ■ 12.2 BETA3

Advances since 12.1 – push-down of COUNT() and other functions

Oracle In-Memory - COUNT(*) for varying DOP

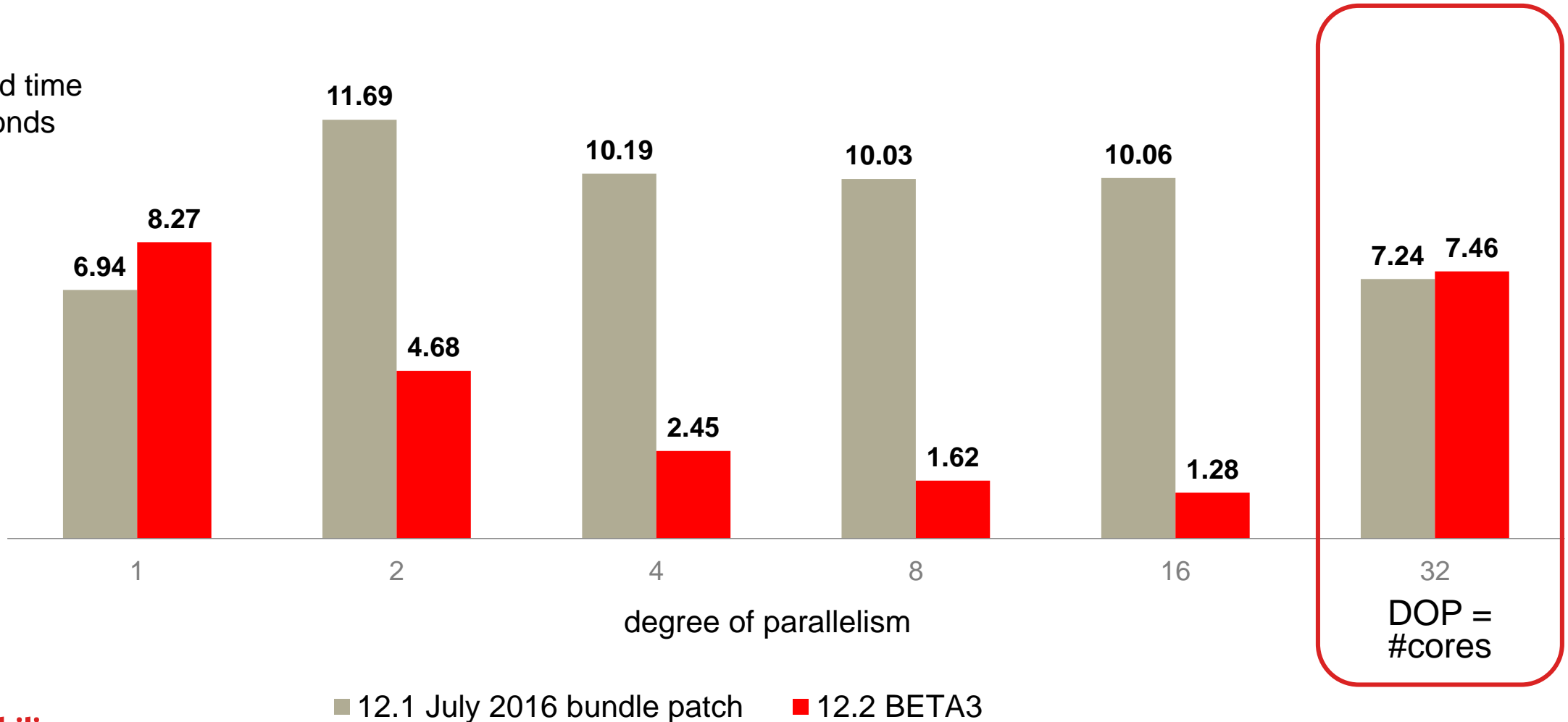
inner joins AND 55 outer joins to 100k to 140M records



Advances since 12.1 – join-efficiency

Oracle In-Memory – FULL SCAN for varying DOP inner joins only

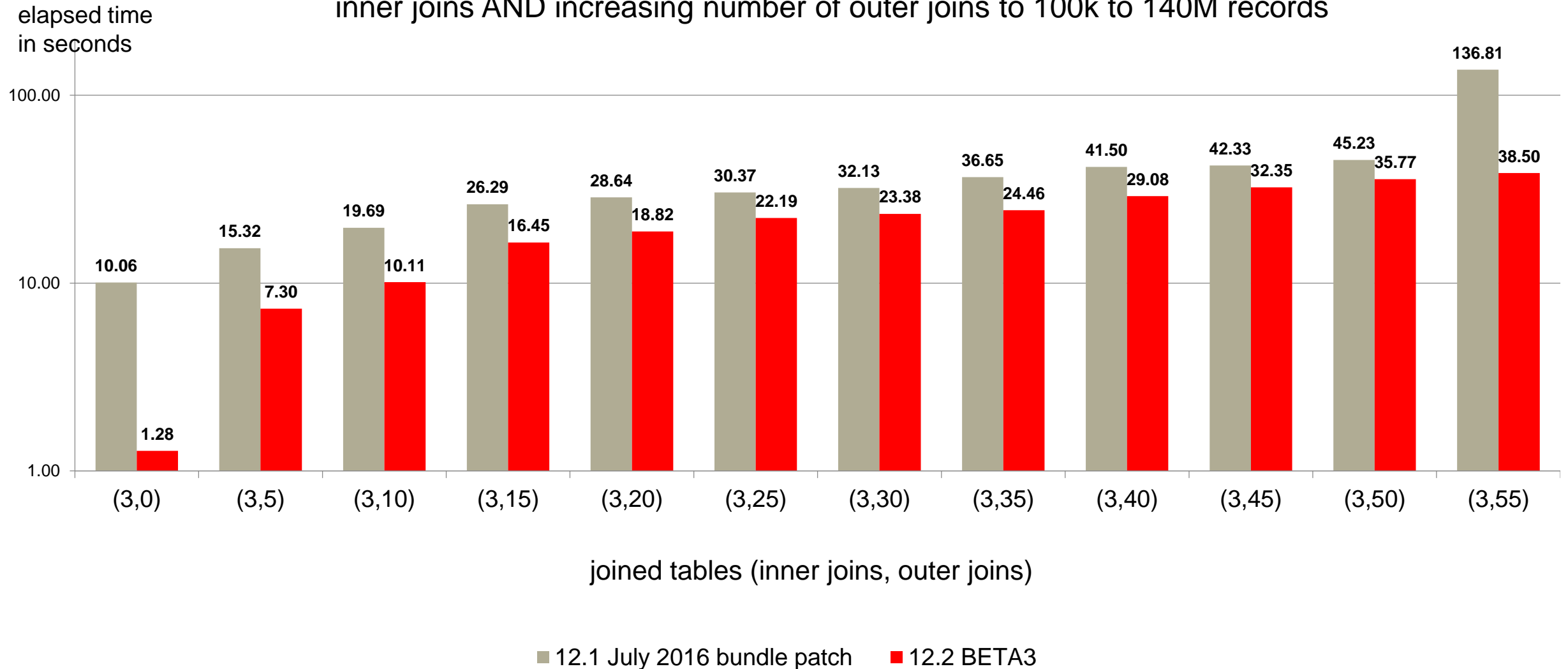
elapsed time
in seconds



Advances since 12.1 – join-efficiency

Oracle In-Memory – FULL SCAN, DOP = 16

inner joins AND increasing number of outer joins to 100k to 140M records



Our next stop on the Oracle In-Memory journey

Remarkable performance improvements observed

- PUSH-DOWN ist great for COUNT(), SUM(), MIN(), MAX()
- Vector transformation in 12.1 used to stress TEMP. In 12.2 BETA we had no such issues.



Remarkable features that fit our architecture needs

- In-Memory within a PDB
- AWR per PDB
- In-Memory on a remote database



Remarkable features, that we still haven't embraced in beta testing

- Join groups to gain even higher join performance
- Detection of expressions by In-memory and reduction of CPU load
- Information life cycle management for In-Memory



Summary – statistic team leaders perspective

Komplexe Datenabfragen sollten nicht länger als zwei Minuten dauern. So kann ich bei explorativen Datenanalysen effizient arbeiten. In-Memory ermöglicht dies in vielen Fällen.

Very complex queries should not take more than two minutes of runtime. That allows me to perform explorative analysis efficiently. With In-Memory I can achieve this in most cases.

Summary –application team leaders perspective

Wir wollen dem Business immer mehr ermöglichen und probieren anspruchsvolle Fragestellungen aus.
Mit In-Memory haben wir viele neue Möglichkeiten bei minimalem Aufwand für alle Teams.

We aim at a better service for the business.
We test increasingly demanding queries.
In-Memory gives us new opportunities at the lowest possible effort for all teams.

Summary – the DBA perspective

I gloub's ned,
dass das
scho fertig isch.

I can't believe it's
already finished.

Business's appetite grows with eating, therefore....



Next Episode on: www.mobiliar.ch/db-blog

Customer

Project

Constraints

Preparation

Upgrade

Success

Remarks

Oracle Multitenant 12.2 – time to rethink the schema based consolidation?

Today database consolidation and resource saving get even more important. Resource saving means license saving – and in the end you could save a lot of...

May 30, 2017 | 0 Comments | [Read article](#)

Oracle Multitenant 12.2 – the big challenge of PDB hot cloning in MAA environments

One of Oracle's key features introduced with Multitenant 12.2 is the possibility to clone pluggable databases while they are open for read / write operations. This...

May 30, 2017 | 0 Comments | [Read article](#)

Mobiliar Performance Warehouse – Part 1: How it all began

This is the first of a collection of articles describing the implementation of our "homegrown" Oracle Performance Warehouse, and the "aha-effects" we triggered during our journey...

April 27, 2017 | 0 Comments | [Read article](#)

SQL Server WMI development – or how not to write code for the future

Starting point Our management and reporting server was to be replaced. It was running on Windows Server 2008 R2 with SQL Server 2012 and collecting data...

August 10, 2016 0 Comments | [Read article](#)

Thomas Baumann of Swiss Mobiliar to Examine Machine Learning at IDUG's 2017 North American DB2 Technical Conference

Bern, Switzerland (25 April 2017) — Thomas Baumann will discuss the application of machine learning techniques in order to tune databases at IDUG's 2017 North American...

April 25, 2017 | 1 Comment | [Read article](#)

DB BLOG TEAM



Thomas Baumann



Hans Eichenberger



Alain Fuhrer



Christian Hählen



Paolo Kreth