Oracle Database Performance Tuning with AWR, ASH and ADDM

Real-World Diagnostics and Troubleshooting

Yuan Yao

Apress®

Oracle Database Performance Tuning with AWR, ASH and ADDM: Real-World Diagnostics and Troubleshooting

Yuan Yao
Guangzhou, Guangdong, China

ISBN-13 (pbk): 979-8-8688-3263-5 ISBN-13 (electronic): 979-8-8688-3264-2
https://doi.org/10.1007/979-8-8688-3264-2

Copyright © 2026 by Yuan Yao

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Shaul Elson
Editorial Assistant: Gryffin Winkler

Cover designed by eStudioCalamar

Cover image designed by Pexels

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a Delaware LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub. For more detailed information, please visit https://www.apress.com/gp/services/source-code.

If disposing of this product, please recycle the paper

To James Wang, the founder of Scutech,
whose trust, encouragement, and unwavering
support made this book possible.

Table of Contents

About the Author xv

About the Technical Reviewer xvii

Acknowledgments xix

Foreword xxi

Introduction xxiii

Part I: ASH 1

Chapter 1: ASH Fundamentals 3

The ASH Mechanism 3

ASH Architecture 3

The ASH Buffer 4

Persisting ASH Data 6

Advantages of ASH 6

Capturing Only Active Sessions for Enhanced Manageability 6

Lightweight Nature 7

Always Active with History Retention 8

Fine-Grained Sampled Data 8

Hidden Parameters That Control ASH Behavior 9

Licensing for AWR, ASH, and ADDM 11

The Impact of CONTROL_MANAGEMENT_PACK_ACCESS on the "Trinity" 12

Avoiding Unintentional Use of Trinity 14

Auditing AWR and ADDM Usage 15

Key Takeaways 15

Summary 16

Chapter 2: ASH Report 17

Generating an ASH Report 17

Using Oracle Enterprise Manager Cloud Control 18

Using SQL Developer 19

Using Command-Line SQL Scripts 21

An Example of Interpreting an ASH Report 22

The Header of the Report 22

Top Event 24

Top SQL 25

Top Sessions 26

Top Objects/Files/Latches 27

Activity Over Time 28

Conclusion 29

Summary 29

Chapter 3: Querying Raw ASH Data with SQL 31

ASH Data Capture Mechanism 31

Real-Time Sampling with V$ACTIVE_SESSION_HISTORY 32

Historical Persistence in DBA_HIST_ACTIVE_SESS_HISTORY 36

Architectural Comparison of ASH Views 37

Identifying Top Entities 39

Top SQL Statements 39

Top Wait Events 40

Top Accessed Objects 42

Top PGA or Temp Space Consumers 43

General Query Formula 44

Summarizing Session Activity Across Any Dimensions 46

Session State Breakdown by Category 46

Session Origin Analysis 47

Session Activity Summary 49

Creating Timelines of Anything with Oracle ASH 50

Analyzing Wait Class by Hour 50
Analyzing SQL Execution Patterns by Day 51
Answering Ad Hoc Questions 53
Analyzing Workload Distribution in RAC 53
Identifying SQL During an I/O Spike 56
Quantifying the Impact of Slow Commits 57
Identifying Hot Objects in I/O Wait Events 57
SQL Workload Breakdown by Wait Class 58
Analyzing Transaction Enqueue Waits 59
Summary 61

Chapter 4: ASH in Action: Practical Use Cases 63

Pinpointing the Slowest Step in a SQL Execution Plan 63
Monitoring SQL Execution Plan Changes 67
Estimating SQL Runtime and Detecting Outliers 70
The Methodology of Runtime Estimation 70
Case Study: Identifying Culprit for Transient Slowdowns 74
Case Study: Pinpointing Outlier Executions of a Known SQL 76
Determining Whether the Database Is Responsible for Slow Response 77
The Problem: Diagnostic Invisibility 78
Evidence via Standard ASH 79
Capturing Concrete Evidence 79
Tracing the Session Blocking Chain 81
ASH Blocking Identifiers 81
Case Study: The Midnight Contention Mystery 82
Identifying the Blocker Sessions 83
Finding the Root Blocker 85
The Blocking Chain Breakdown 88
Diagnosing an RMAN Job Slowdown 89
Why ASH Is Effective for RMAN Diagnosis 89
Case Study: Sudden RMAN Backup Slowdown 90
Summary 92

Part II: AWR 93

Chapter 5: AWR Fundamentals 95

The AWR Mechanism 95

AWR Snapshots 97

Snapshot Creation Process 98

Default Snapshot Settings 99

Modifying Snapshot Settings 102

Storage Considerations 104

Snapshot Baselines 105

Managing AWR in Multitenant Environments 107

Enabling Automatic AWR Snapshots at the PDB Level 107

Creating a PDB-Specific AWR Report 109

Generating AWR Reports 110

Generating AWR Reports Using Oracle Enterprise Manager Cloud Control 110

Generating AWR Reports Using SQL Developer 112

Generating AWR Reports Using Scripts 114

Generating Targeted AWR Reports 115

Automated Script for Generating Peak Time AWR Reports 117

Available AWR Report Scripts 119

DB Time 120

Understanding DB Time 121

Wait Events and Wait Classes 123

Average Active Sessions 124

Understanding Average Active Sessions 124

Relationship Between ASH, DB Time, and AAS 126

Summary 129

Chapter 6: How to Read an AWR Report 131

Report Header 132

Database, Instance, Container, and Host Information 133

Snapshot Interval and Workload Context 134

Load Profile 136
How Oracle Derives the Load Profile Metrics 137
Key Load Profile Metrics 145
A DSS Workload Example 149
Instance Efficiency Percentages 151
Top 10 Foreground Events 155
How Oracle Derives the Top Events 155
Which Events Should Be Investigated 160
Average Wait Time 161
Frequently Observed Wait Events 161
Correlating with Other Sections 165
Wait Classes by Total Wait Time 166
How Oracle Derives the Wait Class Totals 167
IO Profile 172
Comparing OLTP and DSS IO Profiles 172
Load Profile and IO Profile 176
Database I/O vs. Total I/O 177
Time Model Statistics 179
Interpreting the Columns 180
Time Model Statistic Hierarchy 181
SQL Execute Elapsed Time and DB Time 182
SQL Statistics 183
How Representative Are Captured SQL Statements? 184
The “SQL ordered by Elapsed Time” list 187
The “SQL Ordered by Physical Reads (UnOptimized)” List 190
The “SQL Ordered by Rows Processed” List 191
Complete List of SQL Text 192
Key Takeaways 193
Summary 194

Chapter 7: Querying Raw AWR Data with SQL 195

Understanding Raw AWR Data 196

Categorization by Function 196

Categorization by Temporal Nature 198

Monitoring Database Activity with System Statistics 200

Overview of System Statistics 200

Monitoring Instance Activity with DBA_HIST_SYSSTAT 201

Extending the Analysis: Cross-View Correlation 206

Analyzing Service-Level Statistics 210

Reproducing the AWR Service Statistics Section 211

Tracking Service Activity Across Snapshots 214

Gauging System Performance with Metrics 217

Understanding System Metrics 217

Querying Snapshot-Level System Metrics 217

Creating Timelines of Metric Fluctuations 221

Querying Granular Metric History 224

Analyzing SQL Performance across Snapshots 227

Understanding the Structure of DBA_HIST_SQLSTAT 227

Identifying Top SQL 228

Tracing Execution Fluctuations of a Specific SQL Statement 230

Comparing Baseline vs. Current Performance 233

Monitoring I/O Activity 235

Understanding AWR I/O Views 236

Monitoring I/O by Database Function 237

Monitoring I/O by File Type 239

Combining Function and File Type 242

Monitoring I/O by Tablespace 245

Tracking Tablespace Usage and Growth 247

Tracking Space Growth over Time 247

Identifying Segment-Level Space Growth 249

Summary 251

Part III: ADDM 253

Chapter 8: Automating Diagnostics with ADDM 255

Working with AWR and ASH 256

Diagnosing Oracle Performance Issues with the Triad 256

Merging ADDM Reports into AWR Reports 257

ADDM Reports 259

Generating ADDM Report 259

Interpreting ADDM Report 267

Comparing ADDM Periods 270

Mining Hidden Treasures from ADDM Views 275

Core Views and Their Relationships 275

Default Extended Retention 277

Retrieving Most Valuable ADDM Reports 277

Summarizing Top Findings and Actions 279

Cooperation with SQL Tuning Advisor 280

Summary 285

Part IV: Using the Triad in Action 287

Chapter 9: How Oracle Uses CPU 289

Most IT Pros Read CPU Metrics Wrong 289

Misinterpretation of CPU Utilization 290

The Probability of Waiting (The Queue) 291

Multicore Considerations 293

The CPU Run Queue 295

System Load 296

When Is CPU Utilization Too High? 297

Why Does CPU Utilization Lie to You? 298

The Hierarchy: Sockets, Cores, and Logical CPUs 300

Workload Distribution on Logical CPUs 301

The Source of Distortion 304

How Oracle Calculates CPU Metrics 305
Determining the Total Elapsed Time (The Denominator) 307
Calculating Host CPU Metrics 307
Calculating Instance CPU Metrics 308
Clarifying Load Averages 308
Detecting CPU Noisy Neighbors 309
How to Detect CPU Starvation and Evaluate Its Severity 314
Why Can the Sum of "% DB Time" Fall Far Below 100%? 314
Roughly Quantifying CPU Scarcity 317
Other Symptoms of CPU Scarcity 319
Minor Instrumentation Inconsistencies 321
Summary 321

Chapter 10: What's the AWR Report Trying to Tell Us? 323

Assessing Overall I/O Latency with Little's Law 324
Estimating Storage Response Time with Synchronous Single-Block Reads Latency 326
Monitoring System-Wide Latency Trends 326
Identifying Latency by File Type 328
Are Your Online Redo Logs Adequately Sized? 331
Signs of Undersized Redo Logs 331
Sizing Redo Logs Properly 333
Does Your Application Properly Use Bind Variables? 334
Symptoms from AWR 334
Digging Deeper with ASH 343
What Does ADDM Say? 346
Security Vulnerabilities of Literal Values 348
Solutions 348
Is There Excessive Rollback Activity in the Database? 350
A High Rollbacks Value in the Load Profile Can Be a False Alarm 350
Metrics That Reflect Real Rollback Work 351
An Inexplicable Gap in the Time Model 352
Additional Symptoms of Heavy Rollback Activity 353

Is Your PGA Size Set Correctly? 354
Identifying the Symptoms of an Undersized PGA 355
Determining the Optimal PGA Size Using the PGA Advisor 359
Do You Really Need RAC? 361
Data Affinity Determines RAC Success 361
Poor Data Affinity Symptoms in AWR Reports 362
Definitive Indicators: Global Cache Wait Events 364
Workload Partitioning Analysis 368
Can a Single Instance Suffice? 370
Is Your Connection Pool Properly Sized? 375
Dynamic Connection Pools Can Hurt the Database at Peak Load 375
How to Determine the Right Connection Pool Size 376
Are There Excessive Logons to Your Database? 385
A Stateless Application Without Connection Pooling Is Disastrous 385
Symptoms of Excessive Logons 386
Solutions on the Database Side 392
Summary 392

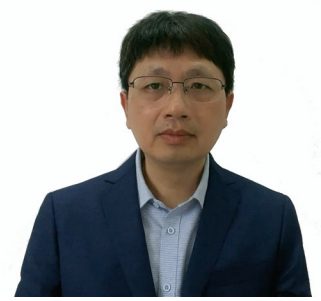
Chapter 11: Real-World Case Studies 395

Slow Standby Database Dragging Down the Primary 395
Symptoms in the Top Wait Events 396
Background Waits: The Stronger Clue 397
RAC Statistics: The Chain Reaction 398
Parameter Check: Data Safety over Performance 399
Root Cause: A Slow Virtualized Standby Database 400
Solution: Trade Data Protection for Performance 400
Case Summary 401
Finding the Contended Rows Behind enq: TX - row lock contention 402
Symptoms in the AWR Report 402
Pinpointing the Locked Rows with ASH 404

Zeroing In: Tracing the Exact Moment of Performance Regression 406
The Standout Troublemaker 406
Building a Timeline for the Plan Change 409
Zooming In Further 411
Finding the Cause of Severe “Buffer Busy Wait” 413
AWR Analysis 413
ASH Analysis 415
Solution 417
When the Average Lies: Detecting Skewed I/O Latency 417
Slow I/O with Contradictory Clues 418
Revealing Skewed Latency with the Wait Histogram 419
Locating the Affected Storage 420
Why 90% Idle CPU Became a Bottleneck 421
Symptoms: A Busy Database on an Idle Host 421
Tracing latch free to Resource Manager 425
Verifying Resource Manager Throttling 427
Contention Was Concentrated in the First 20 Minutes 430
Root Cause and Prevention 432
Summary 433

Index 435

About the Author



Yuan Yao has been working with Oracle databases since the 1990s. He has spent most of his career as a database consultant, helping clients overcome complex database challenges, including troubleshooting critical issues, optimizing performance, and architecting large-scale database systems for enterprise and data center environments.

He is a frequent speaker at database conferences in China and internationally and regularly shares practical knowledge and insights through his blog (https://byte-way.com), WeChat, and Bilibili.

Yao holds 27 Oracle certifications, including Oracle Certified Master (OCM) certifications covering Oracle Database 10g through 19c. He has been honored as an Oracle ACE and recognized as a Huawei Cloud Most Valuable Professional, an honor he received despite never having been a Huawei employee. He is also the inventor of two computer patents and the recipient of two ministerial-level awards in computer science.

He is the author of *MySQL 8.0 Operations and Optimization* (Tsinghua University Press, ISBN: 9787302602682), a widely referenced guide among Chinese database professionals.

Currently, Yao serves as Chief DBA at Scutech, a leading data management company in China with a client base exceeding 20,000.

About the Technical Reviewer



Viba Renganathan is an enterprise technology transformation leader with 13+ years of experience driving large-scale data center modernization, hybrid-cloud migration, and cross-border IT integration programs at Munich Re America Services. She has led the migration of 100+ mission-critical applications and compliance remediation across 400+ systems, delivering measurable reductions in operational effort and infrastructure footprint. A bylined contributor to *Compliance Week* and *CIO.com*, she serves as a juror for international awards programs including the Stevie® Awards, Stratus Awards for Cloud Computing, and Entrepreneurship World Cup. She holds CSM and CSPO certifications.

Acknowledgments

I am deeply indebted to many individuals throughout the journey of writing this book.

First and foremost, I would like to express my sincere gratitude to, James Wang. He gave me the opportunity to join Scutech, a position that not only supports me financially but also continually challenges me to refine my technical skills. Through this role, I have gained hands-on experience solving complex database problems for more than 20,000 clients. This experience made it possible for me to complete my first book, *MySQL 8.0 Operations and Optimization*, and this book as well.

I feel very fortunate to work alongside a team of exceptionally capable professionals at Scutech who are committed to building a great international company. I have benefited greatly from the knowledge, experience, and support of every team member. Without their collaboration and expertise, I would not have come this far.

Most importantly, I thank my family—my unwavering anchor through all the ups and downs in my life. To my always-there-for-me parents, my caring and beautiful wife, my capable and selfless sister, and the two adorable children who light up my life, Lucy and Cen—your presence is a blessing I must have earned in another lifetime.

I also wish to express my gratitude to the Oracle community, both in China and internationally. The insights I have gained from blog posts, forum discussions, and technical articles shared by Oracle ACEs and other experts have been a valuable source of knowledge and inspiration during the writing of this book.

Finally, to the countless people I've met along my life's journey, I am grateful for the predestined connections and shared moments that have shaped the person I am today. Thank you all for being part of this story.

Foreword

I've seen investigations where people immediately blamed a full table scan and others where CPU usage became the obvious suspect. Sometimes people assumed that an execution plan change was the cause as soon as they discovered it. Sometimes those assumptions were right. Often, they weren't. Good troubleshooting starts with facts, but it doesn't stop there. It requires understanding what users experience, validating assumptions against evidence, and thinking clearly about cause and effect.

I've followed Yuan Yao's work and appreciate his approach. His articles are not built around opinions or tuning recipes. They start with observations, use the evidence Oracle provides, and work toward an explanation. That sounds simple, but it is surprisingly rare. In an industry that sometimes rewards confidence more than accuracy, Yuan consistently encourages readers to observe carefully, question assumptions, and investigate thoroughly. Those contributions have earned him recognition as an Oracle ACE and, more importantly, the respect of many practitioners.

It's no surprise that he has compiled his experience into a book. Published by Apress, whose catalog features influential works by Tom Kyte, Jonathan Lewis, and Chris Antognini that shaped my view of Oracle performance, this book upholds a tradition I appreciate: guiding readers to understand the underlying mechanisms instead of simply memorizing advice.

Oracle has accumulated performance instrumentation for more than thirty years. We started with cumulative statistics. Wait events added an essential notion of time. Statspack, and later AWR, preserved history. ASH brought sampling. ADDM added automated analysis. Each generation brought something useful. Together, they provide an extraordinary amount of information.

Today's challenge isn't collecting information. Oracle already does that remarkably well. The challenge is deciding what matters and what doesn't.

That is why I particularly like the way this book approaches the subject. While many users encounter AWR or ADDM first, Yuan gives ASH the importance it deserves. In my view, ASH changed performance troubleshooting more than any other diagnostic feature introduced in the last twenty years.

The reason is simple: performance is about time. When users complain that the system is slow, the first question should be: Where is the time being spent? What is active? What is consuming CPU? What is waiting, and what is it waiting for? ASH was designed to answer exactly those questions.

By following active sessions, ASH shifts the focus from statistics to activity. Instead of looking only at averages and totals, we can see what was actually happening when the problem occurred. That is often the shortest path from symptoms to understanding.

This does not diminish the value of AWR or ADDM. On the contrary, they are essential tools. ADDM, in particular, is often one of the first things users examine when facing a performance problem. Understanding how these tools work, the assumptions they make, and their limitations remains important. Automated analysis can provide useful guidance, provided that we remember that recommendations are the start of an investigation, not the end.

Throughout the book, readers will find practical examples, rules of thumb, and recommendations. Those are useful. Experience matters. But what I appreciate most is that the book repeatedly comes back to evidence.

AWR findings, ASH samples, wait events, execution plans, and performance statistics are valuable because they record facts. Facts keep us honest. They challenge assumptions and help us understand what is really happening.

If this book helps readers develop the habit of starting with evidence and thinking carefully before jumping to conclusions, it will have achieved something valuable.

Franck Pachot
Database Performance Specialist and Author

Introduction

Why I Write This Book

An Underutilized Diagnostic Triad

Since their introduction in Oracle Database 10g, ASH (Active Session History), AWR (Automatic Workload Repository), and ADDM (Automatic Database Diagnostic Monitor) have been my favorite performance-tuning tools for more than 20 years. They have made the work of a DBA much easier by preserving the evidence needed to investigate performance problems after they occur. With this treasure trove of historical diagnostic evidence, a DBA can often determine the root cause of a performance issue without needing direct access to the affected database.

Yet, despite their importance in Oracle performance tuning for more than two decades, it is surprising that, to the best of my knowledge, no book-length treatment devoted specifically to any of these essential tools is currently available. While many excellent Oracle performance-tuning books out there discuss ASH, AWR, and ADDM, they usually cover them only in a single chapter or scattered sections. The result is often descriptive coverage that provides a high-level overview rather than the technical depth required for real-world troubleshooting. This lack of focused literature has contributed to the widespread underutilization of this powerful diagnostic suite.

Over the past 20 years, I have accumulated a wealth of knowledge, practical experience, troubleshooting cases, and SQL scripts for using these tools. I have long felt uncomfortable that, despite their power, these three tools remain underutilized. I therefore feel compelled to write this book to fill this gap in the field. Its aim is to teach you how to read and use these reports effectively, making your performance-tuning work easier by taking full advantage of this diagnostic triad.

Developing Your Ability to Interpret These Reports

Many database professionals feel deterred from reading these reports because the sheer volume of statistics can be daunting, and the meaning of individual statistics and metrics, as well as the relationships among them, can be obscure. Actually, you do not

need exhaustive knowledge before exploring these reports. Understanding a few key concepts and the underlying mechanisms is sufficient to start applying them in real-world scenarios. A learn-as-you-go approach allows your knowledge, experience, and reliable intuition to develop progressively along the way, and this book is structured around that philosophy.

One practical approach when investigating a performance issue is to read the AWR report selectively. You do not need to read an AWR report from beginning to end. Begin at the report header and read through the Time Model Statistics section. These opening sections account for only a small portion of the entire report, yet provide enough high-level evidence to determine where user sessions spend time and how the database consumes resources. Once you know where sessions spend time and how the database consumes resources, you can read the remaining sections selectively to investigate the suspected source of the problem.

Understanding Individual Statistics and Metrics

To interpret these reports precisely, you should first understand where Oracle retrieves each statistic and how it calculates each metric. To illustrate these details, I wrote custom SQL queries that reproduce key sections of the AWR reports from the underlying repository views. You may regard this as a form of reverse engineering, but it is entirely my own research work: it does not use or reproduce Oracle source code.

By reading these SQL scripts, you can trace each statistic back to its source and understand the mathematical logic behind each metric. This enables you to look beyond the surface of a report and interpret its data with significantly greater depth.

Reverse engineering the reports also helps us understand the temporal nature of the statistics and metrics they present:

- Many statistics stored in the underlying performance views are cumulative counters. To obtain the value for a reporting interval, Oracle subtracts the value at the beginning of the interval from the value at the end.
- Many reported metrics are rates, averages, or percentages calculated for the reporting interval.
- Some values are point-in-time values, such as the session counts displayed at the beginning of the AWR report header.

- Some values require special interpretation. For example, `Load Average Begin` and `Load Average End` in the Host CPU section are snapshots of `V$OSSTAT.LOAD`. On Linux, this corresponds to the operating system's one-minute load average. They therefore represent snapshots of a rolling one-minute average, rather than interval deltas or averages calculated across the entire AWR reporting interval.

Understanding the temporal nature of each reported value is essential for interpreting it correctly.

Correlating Statistics and Metrics to Answer Key Questions

Interpreting statistics in isolation is not enough to understand these reports deeply. You must correlate evidence from different report sections, and this book provides many examples of how that correlation can answer important performance questions.

The I/O Profile section provides comprehensive information about database I/O activity, but it does not directly answer a critical question: how fast are the I/O requests being completed? This section alone cannot answer that question. However, when it is correlated with the Wait Classes by Total Wait Time section, the answer can be derived.

The I/O Profile provides the I/O request rate, while the User I/O and System I/O wait classes provide the average number of active sessions waiting for I/O. By applying Little's Law, a fundamental relationship in queueing theory expressed as $L = \lambda W$, we can derive the average I/O response time: divide the average number of sessions waiting for I/O by the I/O request rate. This enables AWR statistics from two different sections to reveal the average I/O response time that neither single section reports directly.

Another important question is how much time Oracle processes spend waiting in the operating system CPU run queue. No standard AWR section directly reports this value on Linux or Windows. On some traditional UNIX platforms, Oracle exposes `OS_CPU_WAIT_TIME` in the Operating System Statistics section. It stands for the total time that processes were in the ready state, waiting to be selected by the operating system scheduler to run.

Another way to assess this CPU wait time is by comparing AWR and ASH CPU statistics. AWR derives `DB CPU` from the operating system CPU time actually consumed by Oracle processes. In contrast, the ASH event `CPU + Wait for CPU` includes both time spent executing on CPU and time spent waiting in the operating system run queue, because Oracle cannot distinguish the two states when a process is involuntarily removed from the CPU by the operating system scheduler. The difference between the ASH CPU figure and the AWR `DB CPU` figure therefore reflects the time Oracle processes spent waiting for CPU.

Many performance abnormalities leave footprints throughout the reports. The same underlying problem can appear as a dominant wait event, an unusual time-model statistic, a resource-utilization imbalance, or anomalous SQL activity. When these symptoms are correlated, they can reinforce or challenge a hypothesis and provide a clearer picture of the underlying cause.

Learning from Real Reports and Controlled Experiments

Understanding individual statistics and their correlations is only the beginning. You must also read many real-world reports to develop practical diagnostic ability and an intuitive feel for the otherwise lifeless figures. Different reports are generated under different workloads, configurations, application characteristics, and environments. Studying them helps you develop the ability to interpret the same statistics in different contexts.

Reports from controlled experiments are equally valuable. By applying a specific pressure to a database and examining the resulting statistical changes in AWR and ASH, you can develop a reliable intuition for how performance problems manifest in the reports. You can also use experiments to verify or refute your hypotheses, bridging the gap between theory and observation.

Looking Behind the Reports

Another driving factor for writing this book is the observation that many DBAs rely solely on formatted reports without querying the underlying dictionary views from which the reports are built. In reality, these views contain far richer and more detailed information than the formatted reports present. I refer to this underlying information as raw diagnostic data, including

- **ASH data:** Current sampled session activity in `V$ACTIVE_SESSION_HISTORY` and historical sampled session activity in `DBA_HIST_ACTIVE_SESS_HISTORY`
- **AWR data:** Historical workload and performance data in the `DBA_HIST_*` views
- **ADDM data:** Diagnostic findings, recommendations, tasks, and related details in the `DBA_ADDM_*` and `DBA_ADVISOR_*` views

Querying these views with SQL not only aligns with the way DBAs naturally approach investigation but also provides far more flexibility in addressing specific performance issues.

In addition, writing SQL makes it possible to perform analyses that a single report cannot support, such as examining performance fluctuations across multiple intervals and joining data from different views to produce new insights.

Home-grown SQL scripts can also reveal valuable information that predefined reports do not provide. For example, `DBA_HIST_SYSMETRIC_SUMMARY` retains approximately 160 system metrics. Most of these metrics are valuable for diagnosis but are not displayed in the standard AWR report; `SQL Service Response Time` is one example. They can be examined only by querying the underlying views.

This book provides many practical SQL scripts for querying these views. Most can be applied in your own environment with little or no modification.

Clarifying Common Misconceptions

Another reason for writing this book is to clarify several deep-rooted and widespread misconceptions about Oracle performance.

CPU is the brain of a computer, so IT professionals should be able to interpret CPU usage correctly. Yet many IT professionals, novice and experienced alike, misinterpret CPU utilization.

For example, many people interpret 40% CPU utilization on a single logical CPU as meaning that 60% of CPU capacity is sitting idle and immediately available. This is incorrect. At the level of an individual CPU cycle, CPU utilization is effectively binary: a logical CPU is either busy or idle; it is not 40% busy during that cycle. The reported 40% utilization is an average over a measurement interval. For a single logical CPU, a runnable process therefore has, on average, a 40% chance of waiting for CPU when it arrives at a random time. This book also extends this single-CPU probability model to multicore systems.

Another common misconception about CPU arises from simultaneous multithreading technologies, such as Intel Hyper-Threading and IBM SMT, which can greatly distort CPU usage readings and lead to false comfort. This book explains why CPU usage reported by the operating system should always be translated from logical CPUs to physical CPU cores.

Another common misconception concerns RAC: migrating an application from a single-instance database to a multi-instance RAC architecture does not necessarily improve performance or scalability. This book explains the key factors required for a successful RAC implementation and shows how to analyze RAC AWR reports to determine whether a single instance can support the workload and, in some cases, deliver better performance.

This book also tries to clarify other misconceptions. For example, it explains the differences in the load metric between Linux and traditional UNIX systems, and why an SQL statement that processes only one row at a time may appear efficient at the micro level but be inefficient at the macro level.

How This Book Is Structured

This book consists of 11 chapters organized into four parts. Parts I through III cover ASH, AWR, and ADDM, while Part IV applies this diagnostic triad to practical performance investigations.

Part I: ASH

Chapter 1: ASH Fundamentals

This chapter explains the core concepts of Active Session History (ASH), including its capture mechanism, key advantages, parameters that control its behavior, and licensing requirements.

Chapter 2: ASH Report

This chapter shows how to generate a standard ASH report using graphical tools or Oracle-supplied scripts. It then demonstrates how to interpret the report through a practical example.

Chapter 3: Query Raw ASH Data with SQL

Standard ASH reports present ASH data in a fixed format, which may not suit every performance investigation. This chapter teaches you how to query raw ASH data directly using SQL, offering greater flexibility for different diagnostic needs. It explains how to understand ASH data, identify top entities, summarize session activity across any dimensions, create timelines of session activity, and answer ad hoc questions. The chapter also includes practical scripts that can be adapted to specific diagnostic requirements with minimal modification.

Chapter 4: ASH in Action: Practical Use Cases

This chapter presents practical cases that demonstrate how ASH can be used for performance diagnosis and troubleshooting. It shows how to pinpoint the slowest step in an SQL execution plan, monitor execution-plan changes, estimate the runtime of individual SQL executions and detect outliers, determine whether the database is responsible for a slow response, trace a session blocking chain, and diagnose a sudden RMAN job slowdown. These cases provide practical ideas that readers can adapt to their own performance investigations.

Part II: AWR

Chapter 5: AWR Fundamentals

This chapter introduces the Automatic Workload Repository (AWR), explaining its mechanism, AWR snapshots, management in multitenant environments, and the generation of AWR reports. It also explains DB Time and Average Active Sessions (AAS), two fundamental measures for assessing database workload and performance.

Chapter 6: How to Read an AWR Report

Interpreting an AWR report can be intimidating because it may span dozens of pages and contain thousands of statistics. This chapter presents a targeted approach to reading the report, covering the report header, Load Profile, Instance Efficiency Percentages, Top 10 Foreground Events, Wait Classes by Total Wait Time, I/O Profile, Time Model Statistics, and SQL Statistics. Rather than examining every metric with equal attention, you will learn how to identify the sections that provide the most relevant evidence for a particular performance investigation.

Chapter 7: Querying Raw AWR Data with SQL

Oracle generates AWR reports from historical repository views, but querying these views directly with SQL provides greater flexibility for customized analysis. This chapter explains how to understand raw AWR data, monitor database activity through system statistics, analyze service-level statistics, evaluate system performance with metrics, analyze SQL performance across snapshots, monitor I/O activity, and track tablespace usage and growth. It includes practical SQL techniques that can be adapted to investigate specific historical performance issues.

Part III: ADDM

Chapter 8: Automating Diagnostics with ADDM

This chapter explains how the Automatic Database Diagnostic Monitor (ADDM) automates performance diagnosis and works together with AWR and ASH. It shows how to generate and interpret ADDM reports, including a case that uses a Comparing ADDM Periods report to resolve the mystery of why a lightweight incremental backup job degraded business performance while a much larger full backup job caused no noticeable impact. It also shows how to query ADDM-related views to uncover diagnostic findings, recommendations, and other valuable details that may not be apparent from the reports alone. Together, these topics provide both a conceptual understanding of ADDM and practical techniques for using it effectively.

Part IV: Using the Triad in Action

This part brings together the three diagnostic tools introduced in the earlier parts of this book: ASH, AWR, and ADDM. It applies them in combination to investigate and resolve real-world database performance problems.

In the report format used in this part, ASH and ADDM output are appended to the AWR report. Accordingly, references to an AWR report in this part refer to the integrated use of AWR, ASH, and ADDM evidence.

Chapter 9: How Oracle Uses CPU

CPU is the most fundamental resource affecting Oracle Database performance, yet CPU statistics are frequently misinterpreted. This chapter explains why CPU utilization can be misleading, how multicore processors and simultaneous multithreading complicate its interpretation, and how Oracle calculates host-level and instance-level CPU metrics. It also shows how to identify CPU-noisy neighbors that compete with the database for CPU cycles, recognize the symptoms of CPU starvation in an AWR report, and assess their severity using a scarcity ratio. These techniques are particularly useful in cloud environments where access to operating-system-level data may be restricted.

Chapter 10: What's the AWR Report Trying to Tell Us?

An AWR report can answer more than general questions about database performance. By correlating information from different report sections and applying targeted calculations, this chapter shows how to answer strategic questions about the health of your database: assess overall I/O latency using Little's Law, estimate storage response time from synchronous single-block reads, determine whether online redo

logs are adequately sized, and identify improper use of bind variables. It also explains how to investigate excessive rollback activity, validate PGA sizing, decide whether RAC is necessary, and assess whether the connection pool is properly sized.

Chapter 11: Real-World Case Studies

This chapter brings the techniques introduced earlier in the book into practice through six real-world performance investigations.

The cases cover a slow standby database that drags down the primary, the rows behind an `enq: TX - row lock contention` wait, the exact moment of a performance regression, the cause of a severe `buffer busy wait`, skewed I/O latency hidden by averages, and a database that became a bottleneck despite 90% idle CPU. Together, these investigations demonstrate how the diagnostic triad can be applied to distinguish visible symptoms from underlying causes and support practical remedies.

Disclaimer

An Oracle Database Enterprise Edition license alone does not automatically entitle you to use AWR, ASH, or ADDM. Accessing their functionality, whether through Oracle Enterprise Manager, command-line APIs, or direct queries against the underlying data, such as `V$ACTIVE_SESSION_HISTORY` and the `DBA_HIST_*` views, requires a valid Diagnostics Pack license. Readers are responsible for complying with the Oracle licensing terms applicable to their database edition and deployment.

For environments without a Diagnostics Pack license, Oracle offers Statspack—the predecessor to AWR—which is included with all Oracle editions at no additional cost. Although Statspack does not provide all the advanced capabilities found in AWR, the core performance analysis ideas and techniques discussed in this book regarding AWR still apply to Statspack data.