

Reading Sample

Dive into portions of two chapters in this reading sample. In Chapter 2, start analyzing the performance of your hardware, database, SAP memory configuration, and SAP work processes of SAP NetWeaver Application Server ABAP using simple instructions. In Chapter 9, first master the concept of locking for SAP and database systems. Then learn how you can monitor database locks and SAP enqueues.

"Introduction"



"Analysis of Hardware, Database, and ABAP Application Server"

"Locks"



Contents



Index



The Author

Thomas Schneider

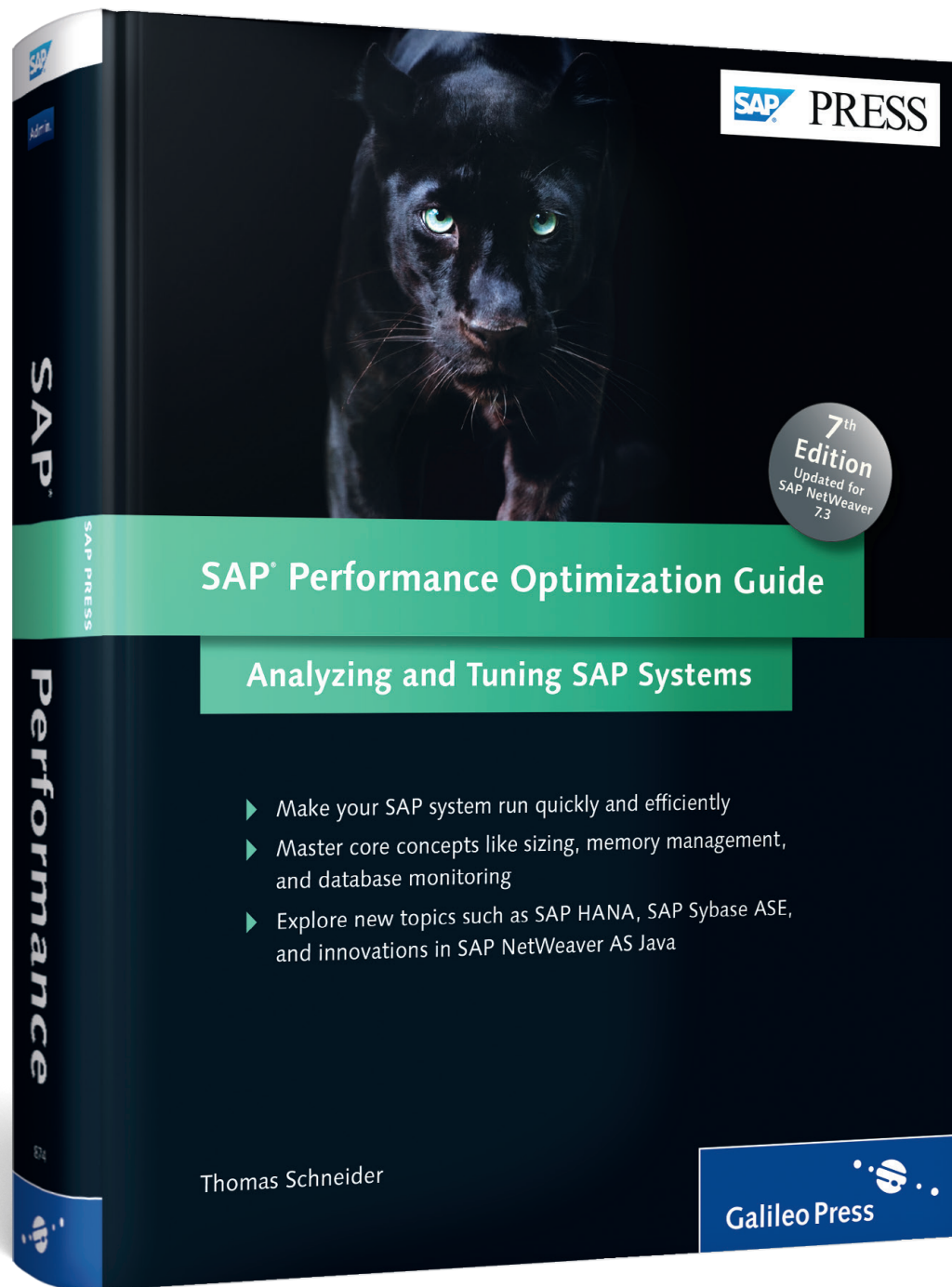
SAP Performance Optimization Guide

837 Pages, 2013, \$79.95

ISBN 978-1-59229-874-7



www.sap-press.com/3360



Contents

Preface and Acknowledgments	17
Introduction	19
1 Performance Management of an SAP Solution	33
1.1 SAP Solution Architecture	34
1.1.1 SAP Solutions and SAP Components	34
1.1.2 Client/Server Architecture	38
1.2 Monitoring and Optimization Plan for an SAP Solution	47
1.2.1 Requirements of a Monitoring and Optimization Plan	47
1.2.2 Service-Level Management	51
1.2.3 Plan for Continuous Performance Optimization	57
1.2.4 Tools and Methods for the Monitoring and Optimization Plan	63
1.2.5 SAP Solution Manager	66
1.3 Summary	71
2 Analysis of Hardware, Database, and ABAP Application Server	75
2.1 Basic Terms	76
2.2 Hardware Monitoring	77
2.2.1 Analysis of a Hardware Bottleneck (CPU and Main Memory)	79
2.2.2 Identifying Read/Write (I/O) Problems	86
2.2.3 Further Analysis at the Operating System Level	87
2.3 Database Monitoring	89
2.3.1 The Performance Monitor in the DBA Cockpit	90
2.3.2 Analyzing the Database Buffer	91
2.3.3 Identifying Expensive SQL Statements	95
2.3.4 Identifying Read/Write (I/O) Problems	102

- 2.3.5 Other Database Checks 104
- 2.4 Analyzing SAP Memory Configuration 112
 - 2.4.1 Analyzing SAP Buffers 113
 - 2.4.2 Analyzing SAP Extended Memory, SAP Heap Memory, and SAP Roll Memory 115
 - 2.4.3 Displaying Allocated Memory 117
 - 2.4.4 Other Monitors for Detailed Analysis 119
- 2.5 Analyzing SAP Work Processes 121
 - 2.5.1 Work Process Overview Fields 122
 - 2.5.2 Analyzing Work Processes 125
 - 2.5.3 System-Wide Work Process Overview 128
 - 2.5.4 Monitoring the Dispatcher Queue 130
- 2.6 Analysis of the Internet Communication Manager (ICM) 133
- 2.7 Continuous Monitoring Using CCMS 134
 - 2.7.1 Working with the Alert Monitor 135
 - 2.7.2 Arranging Monitoring Tree 138
 - 2.7.3 Arranging Automatic Alert Messaging 141
 - 2.7.4 Graphical User Interface in SAP Solution Manager 142
- 2.8 Summary 143

3 Workload Analysis 145

- 3.1 Basics of Workload Analysis and Runtime Analysis 146
- 3.2 Workload Monitor 148
 - 3.2.1 Working with the Workload Monitor 149
 - 3.2.2 Technical Settings for the Workload Monitor 152
- 3.3 Workload Analysis 152
 - 3.3.1 Transaction Step Cycle 153
 - 3.3.2 Other Time Components 155
 - 3.3.3 Interpreting Response Times 156
 - 3.3.4 Activity, Throughput, and Load 159
- 3.4 Performing Workload Analyses 161
 - 3.4.1 Analyzing General Performance Problems 162
 - 3.4.2 Analyzing Specific Performance Problems 169
- 3.5 End-to-End Workload Analysis 172
 - 3.5.1 Basics of End-to-End Workload Analysis 173
 - 3.5.2 Central Workload Monitor 174
 - 3.5.3 Working with the Central Workload Monitor ... 175

- 3.5.4 End-to-End Workload Monitor in SAP Solution Manager 178
- 3.6 Summary 179

4 Identifying Performance Problems in ABAP Programs 183

- 4.1 Single-Record Statistics 183
- 4.2 Performance Trace 188
 - 4.2.1 Activating a Performance Trace 188
 - 4.2.2 Evaluating an SQL Trace 190
 - 4.2.3 Other Functions in the SQL Trace 195
 - 4.2.4 Evaluating a Buffer Trace 197
 - 4.2.5 Evaluating an RFC Trace 198
 - 4.2.6 Evaluating an HTTP Trace 199
 - 4.2.7 Evaluating an Enqueue Trace 199
- 4.3 Performance Analysis with ABAP Trace (Runtime Analysis) 201
 - 4.3.1 Activating an ABAP Trace 201
 - 4.3.2 Evaluating an ABAP Trace 203
 - 4.3.3 Using Function Variations 205
 - 4.3.4 Outlook: Single Transaction Analysis 207
 - 4.3.5 Using Single Transaction Analysis 209
- 4.4 Analyzing Memory Usage with ABAP Debugger and in the Memory Inspector 210
- 4.5 Code Inspector 216
- 4.6 Central Single Statistics Records 218
- 4.7 End-to-End Runtime Analysis in SAP Solution Manager 220
 - 4.7.1 Activating the Runtime Analysis 221
 - 4.7.2 Displaying an End-to-End Runtime Analysis in SAP Solution Manager 223
- 4.8 Summary 225

5 Hardware Sizing and System and Load Distribution ... 229

- 5.1 Initial Hardware Sizing 231
 - 5.1.1 Overview of the Project for Initial Sizing 232
 - 5.1.2 Performing a Sizing Project in Detail 238
 - 5.1.3 SAP Standard Application Benchmarks 242

- 5.2 Sizing to Deal with Increased Workload, Change of Release, or Migration 247
 - 5.2.1 Overview of a Sizing Project in the Environment of an Installation that is Already Used 247
 - 5.2.2 Performing Sizing in the Environment of Installations Used in Production 248
- 5.3 Planning the System Landscape 256
 - 5.3.1 Distribution of SAP Application Instances 257
 - 5.3.2 Hardware Consolidation 258
 - 5.3.3 System and Database Consolidation 260
- 5.4 Services of the SAP NetWeaver Application Server 262
- 5.5 Summary 266

6 Memory Management 269

- 6.1 Memory Management Fundamentals 270
 - 6.1.1 Basic Terms 270
 - 6.1.2 SAP Roll Memory, SAP Extended Memory, and SAP Heap Memory 272
 - 6.1.3 SAP Extended Shared Memory and SAP Paging Memory 279
- 6.2 Features of Operating Systems 280
 - 6.2.1 Zero Administration Memory Management for Microsoft Windows 280
 - 6.2.2 Memory Management for Linux 281
 - 6.2.3 Memory Management for IBM i 282
- 6.3 Configuring and Monitoring Memory Areas 283
 - 6.3.1 Monitoring Swap Space 285
 - 6.3.2 Configuring and Monitoring SAP Memory Areas 287
 - 6.3.3 Assistance with Troubleshooting 291
- 6.4 Summary 298

7 Load Distribution and Remote Function Calls 301

- 7.1 Load Distribution within the ABAP Instances 302
 - 7.1.1 Distributing Message, Enqueue, and ATP Services 303

- 7.1.2 Distributing Dialog, Background, and Spool Work Processes 304
- 7.1.3 Distributing Users and Work Processes over CPU Resources 305
- 7.1.4 Dynamic User Distribution: Configuring Logon Groups 308
- 7.1.5 Limiting Resources per User 311
- 7.1.6 Operation Modes 312
- 7.1.7 Configuring Dynamic Work Processes 312
- 7.1.8 Update 313
- 7.2 Remote Function Calls (RFCs) 320
 - 7.2.1 Fundamentals and Concepts 320
 - 7.2.2 RFC Cycle 323
 - 7.2.3 Configuring and Testing RFC Destinations 326
 - 7.2.4 Monitoring Inbound and Outbound Loads 330
 - 7.2.5 Configuring Parallel Processes with Asynchronous RFCs 335
 - 7.2.6 Monitoring Data Transfer with Transactional RFCs 336
 - 7.2.7 Background RFCs 338
- 7.3 New Load Distribution Concept 338
- 7.4 SAP Virtual Machine Container 341
- 7.5 Summary 343

8 SAP GUI and Internet Connection 347

- 8.1 SAP GUI 348
 - 8.1.1 Interaction Model and Performance Measurement 348
 - 8.1.2 Analyzing and Optimizing the Performance of GUI Communication 350
- 8.2 SAP Web Applications 356
 - 8.2.1 Planning the Use of Web UI and the SAP GUI ... 357
 - 8.2.2 HTTP Trace in the Internet Communication Manager 359
- 8.3 Analyses on the Presentation Server 360
 - 8.3.1 Presentation Server Trace for Web Applications 362
 - 8.3.2 Operating System Performance Tools 364

- 8.3.3 Continuously Monitoring Web Applications 365
- 8.4 Business Server Pages (BSPs), Web Dynpro ABAP, and Integrated ITS 367
 - 8.4.1 Fundamentals of Business Server Pages and Web Dynpro ABAP 367
 - 8.4.2 Fundamentals of the Integrated ITS 370
 - 8.4.3 Fundamentals of ABAP Web Services 372
 - 8.4.4 Performing the Performance Analysis of ABAP Web Services, BSPs, Web Dynpro ABAP, and ITS Applications 373
 - 8.4.5 Monitoring Web Service Calls 376
- 8.5 Summary 376

9 Locks 381

- 9.1 Lock Concepts of Database System and SAP System 382
 - 9.1.1 Database Locks 382
 - 9.1.2 SAP Enqueues 383
- 9.2 Monitoring Locks 385
 - 9.2.1 Database Locks 385
 - 9.2.2 SAP Enqueues 389
- 9.3 Number Range Buffering 391
 - 9.3.1 Fundamentals 391
 - 9.3.2 Activating Number Range Buffering 396
 - 9.3.3 Monitoring Number Range Buffering 397
- 9.4 Locking with Quantities and ATP Server 398
 - 9.4.1 Fundamentals 399
 - 9.4.2 Configuring the ATP Server 400
 - 9.4.3 Monitoring the ATP Server 403
- 9.5 Summary 404

10 Optimizing Java Virtual Machine and Java Programs ... 407

- 10.1 Garbage Collection 409
 - 10.1.1 Construction of the Java Heap 409
 - 10.1.2 Garbage Collection Algorithms 411
 - 10.1.3 Selection of Collectors 412
 - 10.1.4 Parameterization of Java Heap Memory and Garbage Collection 413

- 10.1.5 Logging Garbage Collections 415
- 10.2 Just-in-Time Compiler 415
- 10.3 SAP Management Console 417
- 10.4 SAP Java Virtual Machine Profiler 419
 - 10.4.1 Architecture 419
 - 10.4.2 Establishing the Connection 421
 - 10.4.3 Allocation Analysis 424
 - 10.4.4 Performance HotSpot Analysis 430
 - 10.4.5 Method Parameter Analysis 434
 - 10.4.6 Synchronization Analysis 436
 - 10.4.7 File and Network I/O Analysis 439
 - 10.4.8 Garbage Collection Analysis 441
 - 10.4.9 Class Statistics and Heap Dump 444
- 10.5 Memory Analyzer 446
 - 10.5.1 Creating HPROF Files 447
 - 10.5.2 Analysis of the Java Heap with the Memory Analyzer 447
- 10.6 Local Database Monitors and SQL Trace in SAP NetWeaver AS Java 449
- 10.7 Java Workload and Runtime Analysis with CA Wily Introscope 452
 - 10.7.1 Functionality of Introscope 453
 - 10.7.2 Working with WebView and the Wily Introscope Workstation 454
 - 10.7.3 Introscope Trace 456
- 10.8 Summary 456

11 Optimizing SQL Statements 461

- 11.1 Identifying and Analyzing Expensive SQL Statements 462
 - 11.1.1 Preliminary Analysis 463
 - 11.1.2 Detailed Analysis 465
- 11.2 Optimizing SQL Statements through Secondary Indexes 469
 - 11.2.1 Database Organization Fundamentals 470
 - 11.2.2 Administration for Indexes and Table Access Statistics 480
 - 11.2.3 Rules for Creating or Changing Secondary Indexes 487

11.3	Optimizing SQL Statements in the ABAP Program	495
11.3.1	Rules for Efficient SQL Programming	496
11.3.2	Example of Optimizing an SQL Statement in an ABAP Program	501
11.3.3	Presetting Field Values in Report Transactions ...	511
11.4	Expensive SQL Statements Due to Incorrect Buffer Settings and Administration Tools	515
11.5	Summary	517
12 SAP Buffering		521
12.1	Table Buffering Fundamentals	523
12.1.1	Buffering Types	523
12.1.2	Buffer Accessing	526
12.1.3	Buffer Synchronization	528
12.1.4	Activating Buffering	531
12.1.5	Which Tables Should Be Buffered?	533
12.2	Monitoring Table Buffering on the ABAP Server	535
12.2.1	Table Access Statistics	536
12.2.2	Analyzing Buffered Tables	540
12.2.3	Analyzing Tables That Are Currently Not Buffered	544
12.2.4	Detailed Table Analysis	546
12.2.5	Monitoring Buffer Synchronization (DDLOG Entries)	549
12.2.6	SQL Statistics and SQL Trace	550
12.3	Monitoring Object-Oriented Application Buffers	551
12.4	Monitoring Table Buffering on the SAP NetWeaver AS Java	556
12.5	Summary	558
13 Optimizing Queries in SAP NetWeaver Business Warehouse		561
13.1	Fundamentals of SAP NetWeaver Business Warehouse ...	562
13.1.1	Overview of the Most Important Concepts in SAP NetWeaver BW	563
13.1.2	InfoCube and Extended Star Schema	566
13.1.3	Overview of the Options for Performance Optimization	574

13.1.4	Overview of the SAP NetWeaver BW Administration Tools	578
13.2	Analysis of Expensive BW Queries	580
13.3	Technical Optimization Options in Detail	586
13.3.1	Managing Indexing	588
13.3.2	Settings on the Data Selection	594
13.3.3	Compressing the Fact Table	595
13.3.4	Monitoring the OLAP Cache	596
13.3.5	Pre-calculation of Reports	601
13.3.6	Aggregates	603
13.4	Summary	612
14 Optimizing Search Queries Using TREX		615
14.1	Fundamentals and Architecture of TREX Considering Performance Aspects	616
14.1.1	TREX Application Scenarios and Search Functions	617
14.1.2	Architecture of TREX	620
14.1.3	Specific Characteristics of the SAP NetWeaver BW Accelerator	624
14.1.4	Overview of TREX Administration Tools	626
14.2	Implementing the Performance Analysis on TREX	628
14.2.1	Monitoring the Utilization of the TREX Servers ...	628
14.2.2	Creating and Evaluating the Python Trace	634
14.2.3	Creating and Evaluating a Performance Trace ...	635
14.3	Technical Optimization Options in Detail	636
14.3.1	Dividing Indexes	637
14.3.2	Reorganizing TREX	638
14.3.3	Using Delta Indexes	639
14.3.4	Using Master and Slave Index Servers	642
14.3.5	Configuring Queue Servers	645
14.3.6	Distributing the Preprocessing (Configuring the Preprocessor)	648
14.3.7	Increase the Number of RFC Connections between the ABAP System and TREX	650
14.4	Indexing InfoCubes in the BWA	650
14.5	Summary	656

15 Optimizing Database Queries with SAP HANA 659

- 15.1 Application Scenarios for SAP HANA 661
- 15.2 Principles of Main Memory Data in SAP HANA and
TREX 665
 - 15.2.1 Column-Based Data Storage 667
 - 15.2.2 Data Compression 668
 - 15.2.3 Data Storage in Main Memory 672
 - 15.2.4 Partitioning and Parallelization 675
 - 15.2.5 Indexing 676
 - 15.2.6 Data Modeling from a Performance View 679
- 15.3 Overview of the SAP HANA Architecture and Sizing 681
 - 15.3.1 SAP HANA Architecture 681
 - 15.3.2 Sizing 685
 - 15.3.3 SAP HANA Administration Tools 687
- 15.4 Technical Optimization Options in Detail 690
 - 15.4.1 Main Memory Analysis 690
 - 15.4.2 Identifying and Analyzing Expensive SQL
Statements 698
 - 15.4.3 Creating Indexes (Inverted Index and
Concatenated Index) 707
 - 15.4.4 Administration of Delta Indexes 710
 - 15.4.5 Load Distribution, Parallelization, and
Partitioning 715
 - 15.4.6 Optimization of InfoCubes and DataStore
Objects for SAP HANA 717
- 15.5 Summary 721

Appendices 725

- A Database Monitors 727
- B Selected Transaction Codes 777
- C Review Questions and Answers 781
- D Glossary 787
- E Sources of Information 799
- F The Author 815

- Index 817

Introduction

Why is the performance of your business IT application important? Users will be motivated and work efficiently with an application only if response times are good. A slow system leads to downtime and frustration. If the situation deteriorates further, at worst, you no longer have the throughput necessary for running business processes. The results are overtime, delays in production, and financial loss. In contrast, the systematic, proactive optimization of performance considerably increases the value of your business application.

Performance

A data processing system's performance is defined as the system's ability to fulfill requirements in terms of response time and data throughput. The system might be required to achieve, for example, a throughput of 10,000 printed invoices in one hour or a response time of less than one second for the creation of a sales order. Good performance, however, is not an absolute characteristic of a business application. Rather, it should be viewed as always relative to the demands made on the application.

Proactive Performance Management

In this book, *performance optimization* refers to a process that always includes five phases. The first two phases are *understanding the business processes* and *setting and quantifying performance goals*. These steps involve all participating parties—that is, technicians and application experts. Optimization can be successful only on the basis of these prerequisites. Phases three to five involve the *systematic monitoring, identification, and analysis* of problems, the *implementation of optimization measures*, and further analysis to *verify the success* of the introduced measures (see Figure 1). We advise against randomly tinkering with configuration parameters and similar impulsive tuning measures. The object of this book is to enable you to identify and analyze performance problems in order to deal with them effectively.

Performance
optimization

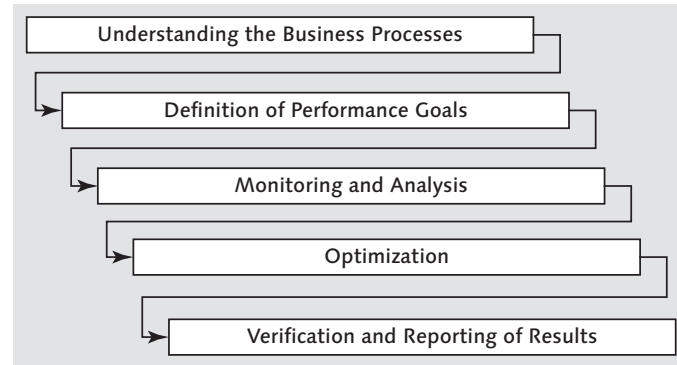


Figure 1 Performance Optimization in Five Phases

Technical optimization

From a technical point of view, a business IT application is made up of many different components. These include the logical components: processes such as services, threads, or work processes, as well as memory areas such as buffers and user contexts. There are also the physical components such as processors (CPU), main memory (RAM), hard disks, and network segments. Each of these components allows for maximum throughput and optimal response time. If the interplay among the components is not appropriately balanced or an individual component has reached its performance limit, wait situations that have a negative effect on throughput and response times can occur. In this book, *technical optimization* refers to the identification, analysis, and solution of these problems by tuning the components and distributing the system's load.

Application optimization

The second important task of performance optimization is preventing unnecessary load. Inefficient programs or their suboptimal use can weaken performance. The optimization of individual programs is referred to as *application optimization*.

The goal of optimization is to improve the system settings and applications to achieve the desired performance, based on existing hardware resources. If the existing resources are not sufficient, they must be extended according to the knowledge gained by analysis.

How much tuning is necessary?

How much effort is involved in the performance analysis and tuning of an SAP solution? The answer to this question depends largely on the size of the system. For a small or medium installation with no modifications

to the SAP standard or customer developments, it is normally sufficient to do performance optimization just before and shortly after the start of production and after large-scale changes, such as upgrades, large data transfers or client transports, or when new SAP solutions or additional users are introduced into the system. Of course, it is also necessary to intervene when acute performance problems occur. The tuning potential, along with its inherent effort in analysis and optimization, increases proportionately with the size of the system. Experience has shown that many performance bottlenecks are caused by customer developments and modifications to the standard SAP software. The most common reason for this is insufficient testing, but problems can also arise as a result of time constraints or lack of experience on the part of the developer. The extreme case would be a large, constantly developing installation with several hundred users, complicated process chains, a dozen or more developers (often from different consulting firms, working on the system at different times and in different places), and outsourced system management. In such a system environment, it is absolutely necessary for a small group of administrators and developers to have an overview of the entire system and keep an eye on performance.

SAP's remote services offer help with performance analysis and tuning—namely, SAP GoingLive™ Check, which enables your system to make a smooth transition to production operation, and SAP EarlyWatch® Check, which monitors your system and suggests additional optimizations.

How does *proactive performance management* help you to achieve the objective of successfully running a business application? You should keep two influencing factors in mind if you're going to achieve this objective: the satisfaction of users and the costs of running the business application. Operating costs arise, on the one hand, from the cost of hardware (e.g., infrastructure, CPU, main memory, hard disks, and networks) and personnel (e.g., administration, maintenance, and error analysis). However, you should not overlook the costs that arise if an application is not available or does not achieve the required performance. In these cases, losses incurred in a few hours or days can exceed an average year's investment in proactive performance optimization. You must compare these costs to the costs of proactive performance management. Table 1 shows the value of proactive performance management with two concrete examples.

Proactive performance management

Proactive Measure	Effect on System	Immediate Value, Owing to Increased User Satisfaction	Immediate Value, Owing to Lower Operating Costs	Diminished Risk of Deterioration
Optimizing SQL statements	Reduction of database load	Faster response times for certain transactions	Stretching hardware investments (e.g., database server and memory system)	Avoiding overloading the database system
Proactive data management (e.g., data avoidance, archiving, and reorganization)	Reducing database growth Shorter times for maintenance work on the database (e.g., backup/recovery, upgrade, migration, and system copy)	Faster response times for certain transactions Shorter downtime during maintenance work	Stretching hardware investments Fewer personnel requirements for maintenance work	Maintaining manageable database size

Table 1 Examples of the Value of Proactive Performance Management

Current Developments

Challenges posed by the Internet and mobile devices

With the development of the Internet, smartphones, and tablet computers, there has been a paradigm shift in the world of business software: software is no longer aimed at the highly specialized employee on his PC, but to users of the Internet (externally, or internally as an intranet) or the user of a mobile device. With SAP R/3, the traditional strategy of process automation was based on highly specialized users who accessed their SAP Enterprise Resource Planning (ERP) system from fixed work centers via installed SAP graphical user interfaces (GUIs). The role of these specialized agents, who had to be trained to use the software, is becoming unnecessary in many cases. Instead, the end user can have direct access to the enterprise's SAP ERP systems via the Internet and mobile devices. Today, for example, the employees of many enterprises can enter their work and absent times, travel expenses, and so forth into the system themselves via the intranet, whereas previously, this would have been done by central users. Increasingly, customers order products directly

via the Internet and no longer by means of letters, faxes, or telephone calls to sales centers.

User expectations concerning the usability and performance of an Internet or mobile application are disproportionately higher than the traditional employee's expectations regarding their SAP ERP system. The employee relies on his own SAP ERP system; if it normally helps to make day-to-day work easier, it is accepted, and minor errors or weak points in performance are tolerated. The Internet user is quite different: if applications offered over the Internet do not work easily and effectively, users can immediately switch to the competition and, for example, make their purchases there—the competition is only a mouse click away. In addition, the Internet does not finish work at five p.m.; an e-business solution on the Internet must be available and work efficiently 365 days a year, 24 hours a day. Users of mobile applications apply the usability and performance standards they are accustomed to with other mobile apps to an SAP application.

With SAP HANA, SAP has succeeded in launching a product on the market that analysts today refer to as the most important innovation for years in the field of business software. The core element of the innovation is a *main memory database* around which additional services, such as an additional server referred to as *XS Engine*, are grouped. In this book, we tackle the SAP HANA database platform exclusively because practical experience is still lacking in relation to the additional services.

SAP HANA

The most important argument in favor of introducing SAP HANA is performance. SAP HANA takes advantage of the availability of huge main memories and massive parallel processor architecture and consistently transforms them into performance by the most modern software architecture. You are perhaps wondering, do I even need a performance book if I have SAP HANA? Or does SAP HANA solve all performance problems? We are firmly convinced that the answer is *no* on both counts. On the one hand, even SAP HANA cannot perform miracles if your program reads complete database tables in the application server and—worse still—sends the data to the user's web browser. The lion's share of the runtime is then attributable to the application server, the network, and the browser. SAP HANA does not override the basic rules for efficient programming! A second *no* stems from the fact that every technical

innovation is confronted with "greed" or, in other words, challenges. One of these challenges is called *Big Data*.

Big Data By Big Data, we mean the phenomenon whereby more and more data of interest to companies is created, but which up to now, could only be evaluated and made available insufficiently, if at all. Examples of such data include posts in social networks; logs of web accesses; transaction data of persons (for example, via mobile positioning or posts with location data in social networks) and products (which, for example, are covered via RFID chips); data from cameras, microphones, and other sensors; financial transactions; and stock market data, as well as consumption data in the energy sector. Companies are eager to collect, link, and evaluate this data and thus gain valuable insights into their customers, markets, and products. A production engineer in the consumer goods industry can be used as an example here. In the future, he will be able to use not only past production figures and current orders, but also the current trends of social networks, to plan more effectively.

Small "errors," major impact All of these data sources have in common that their quantity exceeds that of traditional business data (so-called master and transaction data) many times over. When transferred to our performance issue, this means that a non-performance-optimized system or program has a much greater impact than a "traditional" SAP Business Suite system. Thus, we conclude that performance know-how will continue to be a valuable asset in the future.

IT services The demand for an open, flexible software architecture requires specialized, independently running software components that are linked via interfaces, which means a business process involves several software components. The constantly growing number of solutions and components presents an administrative challenge for data centers. The number of components has grown from the manageable SAP R/3 (with SAP instances, database, and hardware/operating system) to a constantly increasing range of technologies, including products that SAP does not produce but offers as a reseller.

Consequently, business process operators counteract this trend by integrating more and more service partners into the service and support processes. Outsourcing may involve only hardware (e.g., computer performance, hard disk memory, network resources, and so on), or it may

also involve the application itself (i.e., *application service providing*, or ASP); for example, the services of an Internet product catalog can be completely allocated to a service provider instead of being operated by the catalog software in the enterprise. It is thus not only necessary to monitor hardware and software components, but monitoring must also go beyond company and component boundaries.

Overall, completely new requirements arise for administration and monitoring of SAP solutions—requirements that you cannot deal with using traditional concepts.

About This Book

The methods for performance analysis and optimization presented in this book reflect those initially used by experts in the *EarlyWatch service* and *GoingLive Check* and are included in the SAP Basis training courses ADM315 *Workload Analysis* and ADM490 *Optimization of ABAP Programs*. This is the seventh edition of this book, and with each new edition, we take the opportunity to thoroughly describe current trends in product development at SAP and, wherever relevant, to consider developments in the IT world in general.

In this edition, we have almost completely rewritten the Java topics and combined them in one chapter. Another new chapter is dedicated to the most important innovation from SAP for many years, SAP HANA, which initially focuses on the SAP HANA database platform. Other SAP HANA services are not yet dealt with due to a lack of practical experience. All other chapters have been revised and updated for this edition; for example, we describe the new ABAP load distribution concept (new to SAP NetWeaver 7.40) and have included the topic of background remote function calls (RFCs). We have also added a section on the SAP Sybase ASE database to Appendix A (Database Monitors).

Figure 2 presents the chapters of this book based on the five phases of performance optimization at a glance. **Chapter 1** of this book, Performance Management of an SAP Solution, is intended for both SAP administrators and SAP consultants, as well as application developers and SAP project

Fundamentals
of this book

Structure

leads. It discusses the following fundamental questions about performance analysis at a non-technical level:

- ▶ Which preventative measures must you take to guarantee the optimal performance of an SAP solution?
- ▶ What performance tuning measures should you take into consideration?
- ▶ Who is involved in the tuning process?

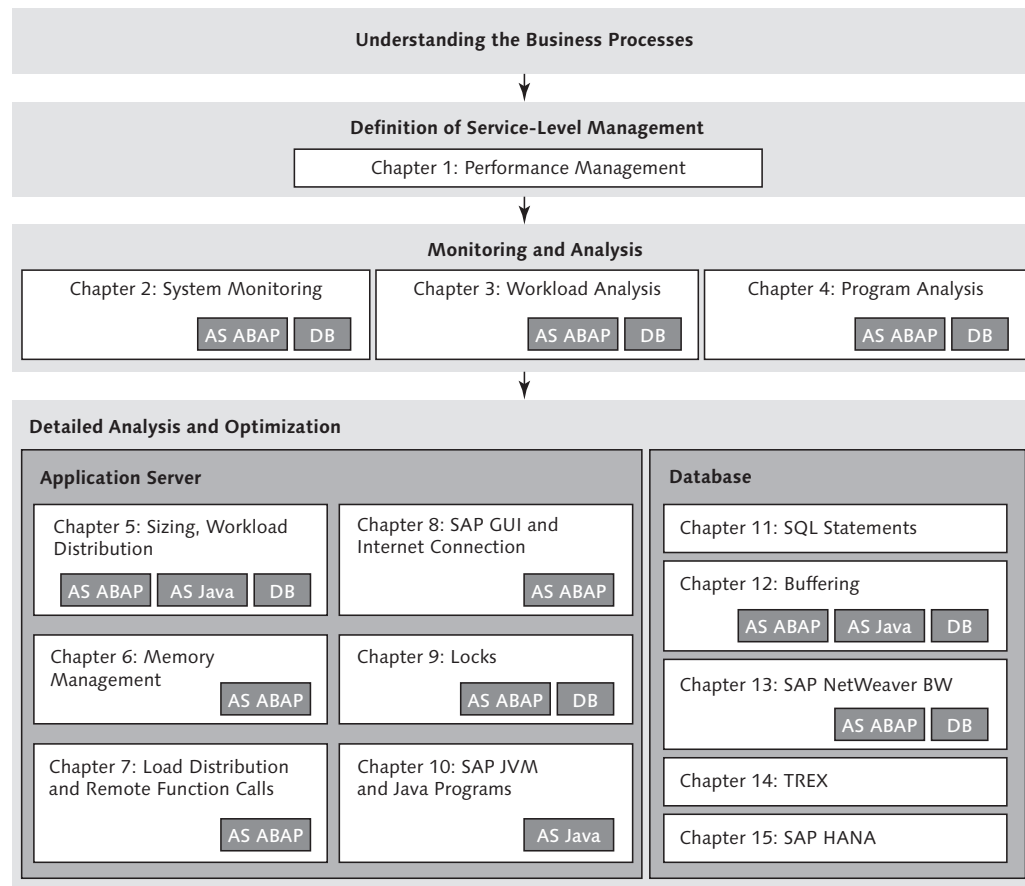


Figure 2 The Chapters of This Book

The service provided for the user frequently turns out to be a combination of a number of different services carried out by a network of partners.

Parts are provided by many different, sometimes external, service providers. To master this complexity, many service providers and customers implement *service-level management* (SLM). SLM calls for a structured, proactive method to ensure an adequate service level for the IT application users, taking into account both cost efficiency and the customer's business objectives. In this book, we'll describe the tools and methods used to implement SLM for an SAP solution.

Chapters 2-4 present performance analysis based on SAP NetWeaver Application Server (AS) ABAP. After reading this chapter, you will be able to perform a systematic performance analysis for AS ABAP, including databases and operating systems.

In this book, we initially follow the bottom-up analysis strategy, starting in **Chapter 2**, Monitoring Hardware, Databases, and ABAP Application Server, with an examination of the operating system, database, SAP memory management, and SAP work processes. At the same time, we provide solution proposals that should enable the administrator or consultant to solve the most important performance problems. For small and medium-size installations, this level of tuning is often sufficient.

Then, **Chapter 3**, Workload Analysis, discusses the more complex workload analysis as an example of top-down analysis. In **Chapter 4**, Identifying Performance Problems in ABAP Programs, you will find methods for analyzing individual programs using tools such as single-record statistics, SQL trace, and ABAP runtime analysis, among others.

The remainder of the book, Chapters 5-15, presents information necessary for a more in-depth performance analysis. These chapters are intended for SAP consultants responsible for the efficient functioning of large systems who need to reach the full tuning potential of their systems. These chapters are independent units to a large extent, and you can read them in any order once you are familiar with the content of the first four chapters. Any dependencies are shown at the beginning of each chapter.

Chapters 5-10 deal with the topics that relate to the application server and the presentation server:

- ▶ **Chapter 5**, Hardware Sizing and System and Load Distribution: This is the guide to avoiding hardware bottlenecks on the one hand, and

limiting the costs of unnecessary hardware on the other hand. Server consolidation—that is, the concentration of all services on a few powerful computers—has become an important IT market trend in recent years. We'll describe what you must take into account to use these technologies efficiently.

- ▶ **Chapter 6, Memory Management:** The configuration of the memory areas allocated by the SAP system has a considerable influence on performance.
- ▶ **Chapter 7, Workload Distribution and Remote Function Calls:** Optimal workload distribution of web, dialog, update, and background requests helps ensure efficient use of hardware and the avoidance of bottlenecks brought about by suboptimal configurations. Interface performance between software components also contributes greatly to the efficiency of the entire solution. E-business solutions that consisted solely of a monolithic R/3 system were rarely used, even in the past. Instead, open solutions that comprise several components connected to each other via interfaces are the standard.
- ▶ **Chapter 8, SAP GUI and Internet Connection:** Analysis and configuration recommendations demonstrate the optimization potential of linking GUIs (i.e., a classical SAP GUI or web browser) with the application. The chapter discusses performance aspects of SAP GUI controls, Internet Transaction Server (ITS), Business Server Pages, and Web Dynpro for ABAP in detail.
- ▶ **Chapter 9, Locks:** Database and SAP locks ensure data consistency. You can avoid bottlenecks in throughput with an optimized administration of locks (for example, with the ATP server or by buffering number ranges).
- ▶ **Chapter 10, Optimizing Java Virtual Machine and Java Programs:** This chapter contains the description of the tools with which you can perform the performance analysis of Java Virtual Machine (SAP JVM) and Java programs.

Chapter 11 introduces the series of chapters on database topics and is a prerequisite for Chapters 12-15:

- ▶ **Chapter 11, Optimizing SQL Statements:** Ineffective SQL statements make heavy demands on the database and can hamper the performance

of the entire application. In this chapter, we present a detailed analysis of "expensive" SQL statements, as well as optimization options via database indexes and program optimization (i.e., "five golden rules").

- ▶ **Chapter 12, SAP Buffering:** Buffered tables on the application servers speed up access to frequently read data and help ease the load on the database.
- ▶ **Chapter 13, Optimizing Queries to SAP NetWeaver Business Warehouse:** SAP NetWeaver Business Warehouse queries are special SQL statements that usually process large quantities of data. Special optimization options exist for this type of queries.
- ▶ **Chapter 14, Optimizing Search Queries Using TREX:** You can use TREX for optimizing text-based and attribute-based search queries and SAP NetWeaver Business Warehouse queries (SAP NetWeaver BW Accelerator) instead of traditional database indexes and aggregation tables.
- ▶ **Chapter 15, Optimization of Database Queries with SAP HANA:** A separate chapter is dedicated to the new "child prodigy" from SAP, SAP HANA, which introduces the principles, tools, and methods for performance analysis and optimization.

Knowledge of performance optimization of SAP systems and applications is highly beneficial for SAP administrators, SAP application managers, SAP developers, and SAP project leads, and these are the target groups of this book. Every chapter first provides an introduction that is followed by a short section, "When Should You Read This Chapter?," that specifies the target group of the chapter.

This book assumes theoretical and practical knowledge of the administration of SAP components in areas that involve the specific implementation of recommendations. You should be familiar with the use of the *Computer Center Management System (CCMS)*, in particular. SAP NetWeaver Application Server ABAP System Administration (see Appendix E, Sources of Information) should serve as good preparation. Parts of this book (for instance, Chapters 4, 9, 11, and 12) also assume familiarity with the ABAP programming language, the functioning of relational databases, and SQL.

The book does not cover the following topics:

- ▶ **Hardware tuning and network tuning**
Although this book helps you to identify bottlenecks in the CPU, main

Target groups

Prerequisites

Limits of this book

memory, I/O, or network, a detailed analysis would require hardware or network provider tools. In view of the enormous number of products offered, we cannot include this subject (especially the tuning of hard disks).

► **Databases**

In the CCMS, SAP offers tools that standardize most administrative and analysis tasks for different database systems. If you want to do more in-depth database tuning, however, you need to be familiar with the different database system architectures. It is not possible for this book to go into sufficient detail on the fine points of all database systems that can be used in conjunction with SAP solutions. However, this information is also unnecessary because reference material on tuning is available for all database systems. This book cannot replace these materials, nor does it endeavor to do so. Instead, the emphasis is on the SAP-specific context of database tuning and on explaining concepts common to all database systems. The specific examples used always refer to individual database systems. In Appendix A, you will find an overview of the most important monitors for analyzing database systems.

► **Application tuning**

Many problems with performance can be solved only with detailed knowledge of the application and the individual SAP system modules. A change in customized settings often solves the problem. This book does not provide the know-how for tuning individual SAP system modules. However, it does provide you with analysis strategies so you can narrow performance problems down to certain applications and then consult the appropriate developer or consultant.

Release
dependency

One question that was heatedly discussed prior to this book's publication is the extent to which release-dependent and time-dependent information, for example, menu paths, recommendations for configuration parameters, and guide values for performance counters, should be included. Factors such as a new version, patches (for the SAP system component, database, or operating system), or a new generation of computers, among others, could render previous information obsolete overnight. In the worst-case scenario, outdated recommendations could even have negative effects on performance. We are aware of this risk. Nevertheless, we have decided

to include time-dependent information and rules in this book. This is the only way you can use this book as a reference for daily work in SAP administration. On the other hand, it is clear that this is not a book of fixed rules and regulations, and anyone who views performance optimization as mechanical rule following is mistaken. This book cannot replace direct analysis of the solution, the SAP Help Portal, or up-to-date SAP Notes on the SAP Service Marketplace. It aims only to support them.

All information on menu paths, references to performance monitor screens, and guideline values for performance counters refer to SAP NetWeaver 7.30, unless otherwise noted. At some points, we give a preview of SAP NetWeaver 7.40.

SAP NetWeaver
7.30

In this book, you will find several orientation aids that are intended to facilitate your reading of its contents.

Notes on reading

Highlighted information boxes include content that is worth knowing and useful, but which is also beyond the actual explanation. To enable you to immediately classify the information in the boxes, we have marked the boxes with symbols:

- The *Tips* marked with this symbol give you specific recommendations that can make your work easier. **[«]**
- In boxes that are marked with this symbol, you will find information about *additional topics* or important content that you should remember. **[+]**
- This symbol indicates *features that you should note*. It also *warns you* about common errors or problems that may occur. **[!]**
- *Examples* identified by this symbol indicate scenarios from practical experience and demonstrate the presented functions. **[Ex]**

As for previous editions, we will provide updates and, if necessary, corrections to the book on the publisher's website (www.sap-press.com).

www.sap-press.com

With the analysis of hardware resources, the database, and the work processes and memory areas of SAP NetWeaver AS ABAP, we are going bottom-up into performance analysis. Get an initial overview of the current situation in the system.

2 Analysis of Hardware, Database, and ABAP Application Server

This chapter provides the basic information on analyzing the performance of your hardware, database, SAP memory configuration, and SAP work processes of SAP NetWeaver Application Server ABAP. Procedure roadmaps at the end of each section summarize the most important analysis paths and clarify when to use the various monitors. The last section describes the central Alert Monitor, which integrates the performance indicators from all areas.

This chapter will provide simple recommendations to help you optimize each component, except where in-depth explanations are required (these are given in subsequent chapters). Unnecessary background information is intentionally kept to a minimum so that even application consultants or system administrators with limited experience in performance analysis can use this chapter to improve the performance of their system. For example, we describe monitoring and customizing SAP extended memory without explaining SAP extended memory in detail. You can find more detailed information in Chapters 5-15. Our experience suggests that you can solve many performance problems in the operating system, database, and SAP Basis by using simple instructions, without delving into technical details.

When Should You Read This Chapter?

You should read this chapter if you want to use your SAP system to technically monitor and optimize the performance of your SAP system, database, or operating system.

2.1 Basic Terms

The terms *computer*, *server*, *application server*, *SAP instance*, *database*, *database server*, and *database instance* are used in this book as follows:

- Computer** A *computer* will always mean a physical machine with a CPU, main memory, IP address, and so on.
- SAP application instance** An *SAP application instance*, also referred to as an *SAP instance*, is an administrative unit: it consists of a set of SAP work processes that are administered by a dispatcher. It also includes a set of SAP buffers located in the host computer's shared memory and accessed by the work processes. An SAP application instance can be an ABAP application instance (*SAP NetWeaver Application Server ABAP*, referred to as *AS ABAP*), or a Java application instance (*SAP NetWeaver Application Server Java*, referred to as *AS Java*). There can be multiple SAP instances on one computer. As a result, there will be multiple dispatchers and sets of buffers. An *application server* is a computer with one or more SAP instances.
- Database** Every SAP system has only one database. The term *database* refers to a set of data that is organized into files, for example. The database can be thought of as the passive part of the database system.
- The active part of the database system is the *database instance*, an administrative unit that allows access to the database. A database instance consists of database processes with a common set of buffers in the shared memory of a computer. A *database server* is a computer with one or more database instances. A computer can be both a database server and an application server if a database instance and an SAP instance run on it.
- In the SAP environment, there is normally only one database instance for each database. Examples of database systems in which multiple database instances can access a database are DB2 and Oracle Parallel Server. This book does not cover the special features of these *parallel database systems*.
- SAP system** We refer to SAP software components that are based on SAP Basis as *SAP systems*. These are SAP ERP, SAP NetWeaver BW, SAP APO, SAP SRM, and SAP NetWeaver Portal.
- According to this terminology, an SAP ERP system can consist of one or two systems, depending on whether the Java and ABAP parts run on

a joint system with one database (for example, SAP NetWeaver double stack) or on two systems with separate databases. This terminology also applies to SAP Solution Manager.

SAP documentation and literature use the term *server* in both a hardware sense and a software sense. Therefore, the term can refer to a computer, for example, in the term *database server*, and to a logical service, such as in the terms *message server* and *ATP server*. Thus, we also use *ABAP server* or *Java server* as short forms for *SAP NetWeaver Application Server (AS) ABAP* or *Java*.

Server

2.2 Hardware Monitoring

The operating system monitor analyzes hardware bottlenecks and operating system problems. To start the operating system monitor for the application server you are currently logged on to, select the following menu:

Operating system monitor

TOOLS • ADMINISTRATION • MONITOR • PERFORMANCE • OPERATING SYSTEM • LOCAL • OPERATING SYSTEM MONITOR

Alternatively, you can use Transaction ST06. The main screen of the operating system monitor appears.

The operating system monitor was revised for SAP Basis version 7.10. Since the revision, Transactions OS06, OS07, and ST06 open a monitor that you can use to monitor both the local and remote computers. For versions prior to 7.10, the new transactions are available under Transactions OS06N, OS07N, and ST06N; with Transactions OS06, OS07, and ST06, you can still access the older transactions. All information discussed in this book is also available in the old transactions. You can view the detail analysis by clicking the **DETAIL ANALYSIS MENU BUTTON**.

Call and availability

You can also call the operating system monitor from the server overview:

TOOLS • ADMINISTRATION • MONITOR • SYSTEM MONITORING • SERVER (Transaction SM51)

Then, position the cursor on the desired application server and, in the menu, choose **GoTo • MONITORS • OS MONITOR**.

Structure The operating system monitor screen is divided into three areas (see Figure 2.1). In the top-left window, you can view the list of computers that are monitored. There, you can select a computer that you want to analyze. In the lower-left window, you select the analysis data. The window on the right contains data on the selected computer and analysis.

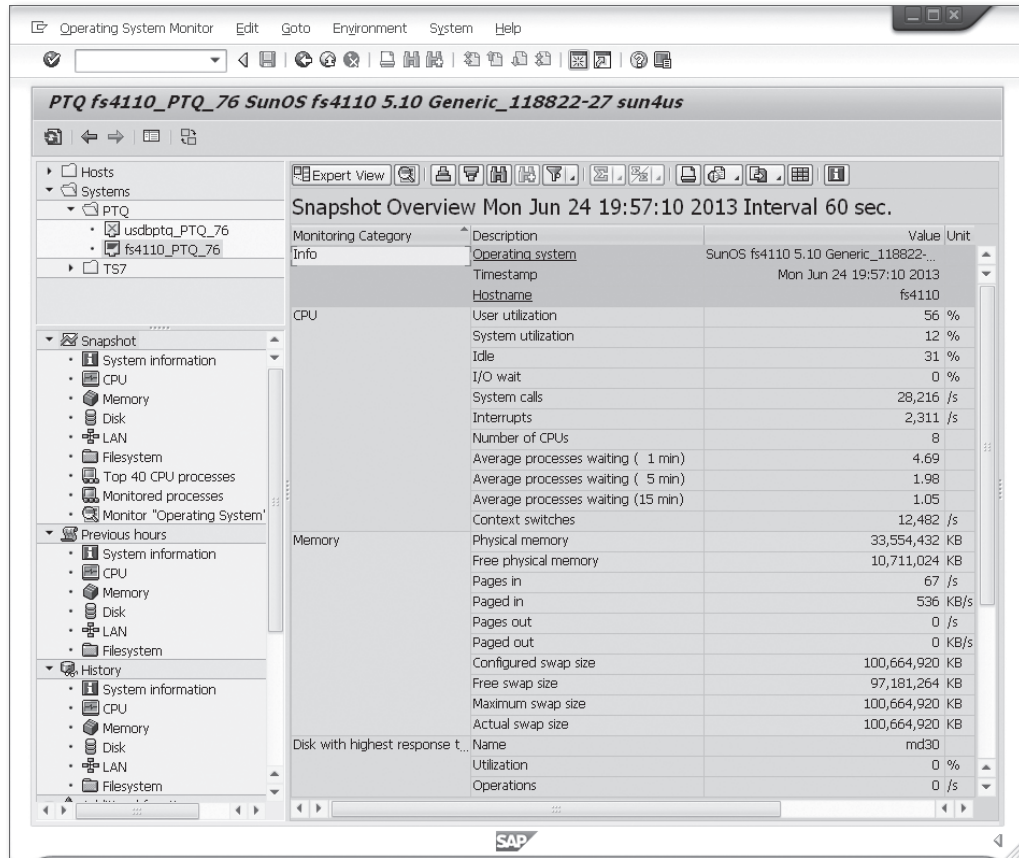


Figure 2.1 Main Screen of the Operating System Monitor

By default, the selection list shows all computers on which SAP ABAP instances have been installed. Essentially, any computer can be integrated into the remote operating system monitor, provided a monitoring agent has been installed on the relevant computer. We strongly recommend installing monitoring agents on computers that run a stand-alone database, an SAP Java instance, or a TREX.

You should install this monitor even if you use a tool from a different vendor to monitor utilization of your computers. If you need support from SAP, an SAP expert can analyze the computers only via the SAP monitor.

SAP Support

2.2.1 Analysis of a Hardware Bottleneck (CPU and Main Memory)

You will find an overview of the most critical operating system and hardware data under SNAPSHOT in the analysis selection of the operating system monitor (see Figure 2.1). All data is refreshed every 10 seconds by the auxiliary program `saposcol`. To update the data on the screen (after 10 seconds or longer), you need to click the corresponding button.

Under the header CPU, you will find the fields USER UTILIZATION, SYSTEM UTILIZATION, and IDLE. These values indicate the percentage of total CPU capacity currently being used by user processes (i.e., the SAP system, database, and other processes), the percentage being used by the operating system itself, and the percentage not being used. The NUMBER OF CPUs field indicates the number of CPU threads. AVERAGE PROCESSES WAITING is the average number of work processes waiting for a free processor. This value is indicated as averaged over 1, 5, and 15 minutes. The other values in the CPU header are of less importance for the performance analysis. Table 2.1 provides an overview of the fields of the operating system monitor.

CPU load

Field	Explanation
USER UTILIZATION	CPU workload caused by user processes (SAP system, database, etc.).
SYSTEM UTILIZATION	CPU workload caused by the operating system.
IDLE	Idle CPU capacity. This value should be at least 20%, but ideally at least 35%.
NUMBER OF CPUs	Number of CPU threads.
AVERAGE PROCESSES WAITING	Number of processes waiting for CPUs, averaged over 1, 5, or 15 minutes.
PHYSICAL MEMORY	Available physical main memory (RAM) in KB.

Table 2.1 Operating System Monitor Fields

[>>] Processors, Cores, and Threads

As a description of the computer equipment, you can find the following specifications, for example: "2 processors, 8 cores, 16 threads, processor of manufacturer X with 2.93 GHz clock speed." What does this information on the number of *processors*, *cores*, and *threads* mean for the SAP system?

The term *processor* refers to the *central processing unit* (CPU) of a computer, which is capable of executing programs. Here, a distinction is made between single-core processors and multicore processors. Multicore processors have multiple fully developed processing cores on one chip. The individual cores share only the bus; that is, they are considered full CPUs. Multithreaded CPUs have *one* CPU, but register as *multiple* CPUs on the operating system. As a result, various queues exist for these cores between which the core switches. To optimize this switching, each thread has its own register set, including a stack pointer and program counter, so you can switch among the threads without additional processor cycles. These hardware-based threads should not be confused with the threads that generate the application processes (that is, user or software threads). Within a process of the database or the ABAP, Java, or TREX server, you can generate multiple (software) threads that the operating system executes in time slices. The switch between the (software) threads is referred to as a *context switch*. Considering this, it can be assumed that additional (hardware) threads promote context switches among (software) threads and therefore support a better utilization of the existing core; however, the increase in performance doesn't fully come up to an additional core.

Main memory utilization and swap space

The MEMORY header contains information on the available physical main memory (PHYSICAL MEMORY field) and values of the operating system paging.

Under the SWAP HEADER, the amount of currently allocated *swap space* is listed. The swap space must be greater than the total of the configured memory area.

[!] Program Terminations Due to Memory Shortage

If the sum of the physical memory and swap space is smaller than the total amount of memory required by the SAP system, database, and other programs, this may cause program terminations or even operating system failure. You should therefore ensure that there is sufficient swap space.

History: CPU and main memory

To display the CPU workload over the previous 24 hours, select the analysis PREVIOUS HOURS • CPU. The PREVIOUS HOURS CPU screen is displayed. The

column headers are the same as in the fields under CPU in the operating system monitor initial screen, except that the values are for one hour. Similar overview is available for main memory usage (PREVIOUS HOURS • MEMORY) for the swap space, and so on.

When Is There a CPU or Main Memory Bottleneck?

The unused CPU capacity IDLE should normally average at least 20% per hour. This enables the system to accommodate temporary workload peaks. A reading of 35% idle CPU capacity is even better. For the paging rate, the following guideline values apply:

Guideline values

- ▶ For computers that contain a database, Java instance, or TREX, only very minor paging rates should occur; that is, they should be dimensioned in such a way that the available main memory corresponds to the configured memory areas.
- ▶ For computers that include only ABAP instances, you can tolerate moderate paging rates of up to 20% of the physical main memory per hour.

For operating systems that page continuously (for example, Microsoft Windows), the value indicated in the operating system monitor as the *paged-in rate* is the key statistic on paging performance. For other operating systems that page only when necessary (such as most UNIX derivatives), the key statistic is the *paged-out rate*. If the operating system monitor sometimes shows values that exceed these guideline values, this does not automatically mean you have a hardware bottleneck. Rather, you should use the workload monitor to check whether the high CPU workload or the paging rate is associated with poor response times. Corresponding analyses can be found in Chapter 3, Section 3.4.1, Analyzing General Performance Problems.

If you observe high paging rates on several computers, you should calculate the virtual main memory allocated by the SAP instances and the database (see Sections 2.4.3 Displaying Allocated Memory and 2.3.2 Analyzing the Database Buffer). Compare this with the available physical main memory. As a rule of thumb, there should be approximately 50% more virtual memory than physical memory.

Main memory bottleneck

In Microsoft Windows and Oracle Solaris operating systems, the analysis of the paging rate on the database server can lead to misinterpretation because in these operating systems, read/write operations (I/O) can sometimes be counted as paging. For more information on this issue, please refer to SAP Notes 124199 (Solaris) and 689818 (Windows).

Causes of Hardware Bottlenecks

If you detect a hardware bottleneck on one or more SAP system computers, it may be due to one or more of the following causes:

▶ Incorrect load distribution

In a distributed system with multiple computers, if you discover a hardware bottleneck on at least one computer while other computers have unused resources, the workload is probably not optimally distributed. To improve performance, redistribute the SAP work processes and the user logons.

It is extremely important that the database server has enough resources. A CPU or main memory bottleneck on the database server means the required data cannot be retrieved quickly from the database, which causes poor response times in the entire system.

▶ CPU load of individual programs

In the operating system monitor (Transaction ST06), select the analysis SNAPSHOT • TOP CPU PROCESSES. The overview of the operating system processes is displayed. Here, you can see all currently active processes and their demands on resources.

Figure 2.2 shows an overview of a system on which an ABAP instance and a DB2 database are installed. You can identify the following processes:

- ▶ `dw_<instance>`: SAP work process of the SAP ABAP instance on a UNIX operating system. On Windows operating systems, the name is `disp+work`.
- ▶ `db2sysc`: Database process of DB2 database. The processes of other databases normally carry the brand name (such as Oracle), which appears in the process or user name.

Operating system processes, which you can identify from the following elements of the name, are also part of SAP instances:

- ▶ `jstart ...`: Server process of SAP Java instance.
- ▶ `TREX ...`: TREX process. The server type is indicated in the process name, for instance, index server, preprocessor, and so on.
- ▶ `icman ...`: Process of the Internet Communication Manager (ICM).
- ▶ `saposcol ...`: Auxiliary program, which collects the data for the operating system monitor, for example.

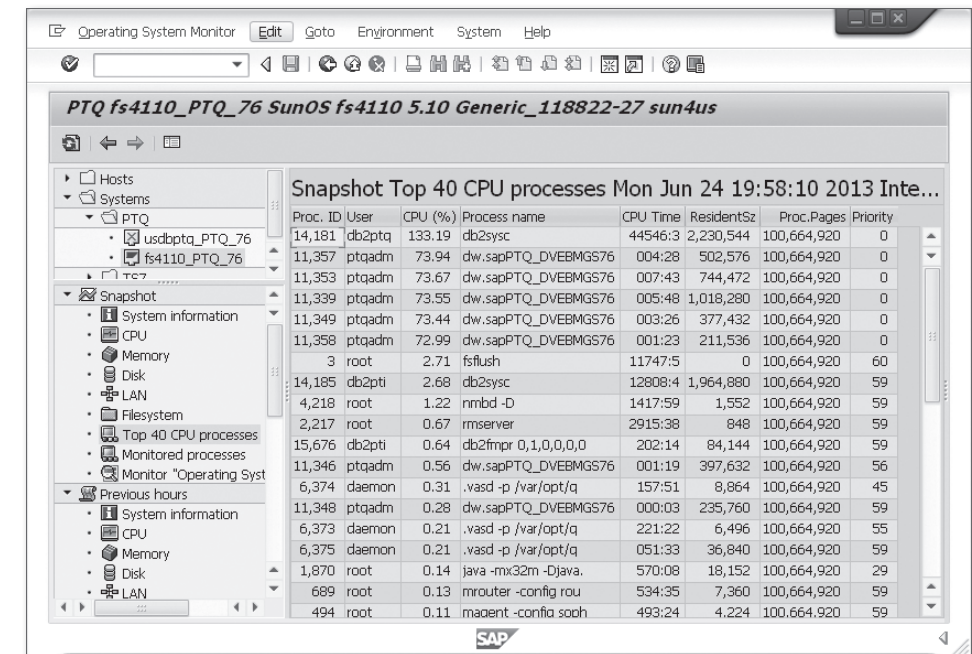


Figure 2.2 Analysis of the Top CPU Processes in the Operating System Monitor

To check whether individual processes are placing a heavy load on the CPU for long periods of time, refresh the monitor periodically and observe any changes in the value CPU (%). If the processes that place a heavy load on the CPU entail processes of SAP Basis or the database, the subsequent specified monitors provide further information on the processes' activities.

Analysis of the CPU load

Overview of operating system processes

Start the monitor in a second mode, identify the process with the heavy CPU load using the process ID, which you can also find in the corresponding basis monitors, and check the monitors to determine which program or tables, queries, and so on are being processed by the process.

► **SAP work processes of ABAP instance**

Open a new user session and call the local work process overview (see Section 2.5, Analyzing SAP Work Processes). From the work process overview, note the name of the ABAP program and the user corresponding to the process identifier (PID).

► **Server process of Java instance**

Open the SAP Management Console (see Chapter 10, Section 10.3, SAP Management Console). Use a thread dump to obtain process-internal information.

► **TREX processes**

Open the TREX administration tool (see Chapter 14, Section 14.2). You can find details on the TREX services in the SERVICES monitor.

► **ICM**

Open the ICM monitor (see Section 2.6, Analysis of the Internet Communication Manager [ICM]).

► **Database processes**

Open the database process monitor in the Database Administration (DBA) Cockpit (see Section 2.3.3, Identifying Expensive SQL Statements) to identify the SQL statements that are being processed by the database.

Using the operating system monitor in conjunction with the monitors mentioned, you can fairly easily identify programs, transactions, SQL statements, or TREX queries that cause high CPU load.

External processes A CPU bottleneck can be caused by *external processes*. In the operating system monitor, if you find external processes (that is, processes that are not part of the SAP system) with high CPU consumption that cause a CPU bottleneck, you should find out whether these processes are really necessary for your system or whether they can be switched off or moved to another computer. The following are examples of external processes:

administrative software, virus scanners, backups, external systems, screen savers (!), and so on.

Identifying a CPU bottleneck

Suppose you notice a CPU bottleneck during times of peak user activity. The process overview in the operating system monitor reveals a single SAP work process that is causing a CPU load of 30% over several minutes. At the same time, the SAP work process overview shows a long-running background program. You should try to see if the background program could be run at a time when the dialog load is lighter.

[Ex]

To identify programs with high memory requirements that may be causing a main memory bottleneck, you can use a method similar to that previously described for CPU bottlenecks. See also Chapter 6, Memory Management.

Memory requirement of individual programs

Operating systems normally administer their own *file system cache*. This cache is located in the main memory, where it competes for memory space with the SAP system and the database. If the cache is too large, it causes high paging rates, even though the physical main memory is more than large enough to accommodate both the SAP system and the database. SAP recommends reducing this cache to 7-10% of the physical memory.

Minimize file system cache

The operating system parameters for configuring the file system cache include `dbc_max_pct` for HP-UX, `ubc-maxpercent` for Digital UNIX, and `maxperm` for AIX.

UNIX

To reduce the size of the file system cache for Microsoft Windows, call the network settings (symbol: NETWORK) in the Control Panel of your Windows operating system. Select the tab SERVICES, the service SERVER, and the PROPERTIES button. In the following screen, under the screen area OPTIMIZATION, select the MAXIMIZE THROUGHPUT FOR NETWORK APPLICATIONS option, and confirm by clicking OK. You must reboot the computer to activate the file cache's new settings.

Windows

A main memory bottleneck creates excessive paging, which in turn requires more processor use and can lead to a CPU bottleneck. Removing the cause of excessive paging usually makes the CPU bottleneck disappear.

2.2.2 Identifying Read/Write (I/O) Problems

In the operating system monitor (Transaction ST06), you'll find in the analysis view `SNAPSHOT • DISK`, among other things, information on hard drive load and (if the operating system makes it available) information on the drives' wait and response times.

Hard drive monitor By double-clicking a row in the hard drive monitor, you can display an overview of the average response times over the previous 24 hours for the selected hard drive. Table 2.2 lists the displayed fields and their significance.

Field	Explanation
DISK	Operating system name of the hard disk.
UTILIZATION (%)	Load on the hard disk (in %).
QUEUE LENGTH	Number of processes waiting for I/O operations.
WAIT TIME (MS)	Wait time (in ms).
SERVICE TIME (MS)	Service time (in ms).
TRANSFER (KBYTE/S)	Transfer rate (in Kb/second).
OPERATIONS (PER SEC)	Number of I/O operations (per second).
RESPONSE TIME (MS)	Average response times of the hard disk (in ms).

Table 2.2 Fields of the Hard Disk Monitor

I/O bottleneck If you determine via these monitors that individual drives are heavily loaded (`LOAD (%) > 50 %`), a potential *I/O bottleneck* exists. However, you can gain only limited information about I/O problems from the SAP system. To perform a more detailed analysis, you need tools provided by the hardware manufacturer.

An I/O bottleneck is particularly critical if it is on the hard drive on which the operating system's paging file resides. The operating system monitor is recommended particularly for the database server. To prevent bottlenecks during read or write operations to the database, use the database performance monitor and the hard drive monitor. For further details on these problems, please see Section 2.3.4, Identifying Read/Write (I/O) Problems.

2.2.3 Further Analysis at the Operating System Level

For UNIX operating systems, the SAP system logs all operating system parameter changes. The change log can be displayed via the following path in the operating system monitor: `OTHER FUNCTIONS • PARAMETER CHANGES`. Place the cursor over the name of a server and click the `HISTORY OF FILE` button. This log lets you determine whether the start of performance problems can be linked to the time when particular parameters were changed.

Parameter changes

With the `OTHER FUNCTIONS • LAN CHECK BY PING` tool, you can carry out a quick test on the network. Select any database server, application server, or presentation server and test the network connection (for example, response times or whether there was any data loss). Although the analysis is incorrectly called LAN check, you can also address computers in WAN. You can find an example of an analysis with this tool in Chapter 8, Section 8.1.2, Analyzing and Optimizing the Performance of GUI Communication.

Network check

Summary

Performance problems may be indicated if at least one of the following items are true:

- ▶ The average idle CPU capacity is less than 20% every hour.
- ▶ More than 20% of the physical main memory is paged every hour.
- ▶ Utilization of individual hard drives is more than 50%.

Excessive utilization of the hard drives, particularly on the database server, can cause system-wide performance problems. To check whether the high CPU load or paging rate significantly damages response times in the SAP system or database, use the workload monitor (see Chapter 3, Section 3.4, Performing Workload Analyses).

Figure 2.3 and Figure 2.4 show the procedure for analyzing a hardware bottleneck: a common solution for resolving bottlenecks is to redistribute the workload (for example, move work processes). Possible causes of a CPU bottleneck include inefficient applications, which can usually be identified in the database process monitor and work process overview, and external processes that do not belong to an SAP instance or the database instance. You should always perform a complete performance

Analyzing hardware bottleneck

analysis before deciding whether the existing hardware is sufficient for SAP system demands.

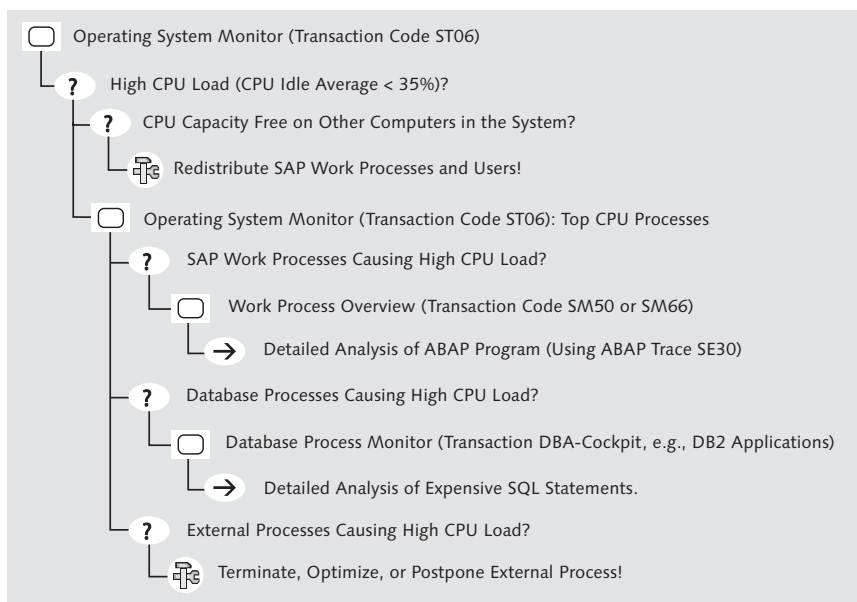


Figure 2.3 Detailed Analysis of a Hardware Bottleneck (CPU)

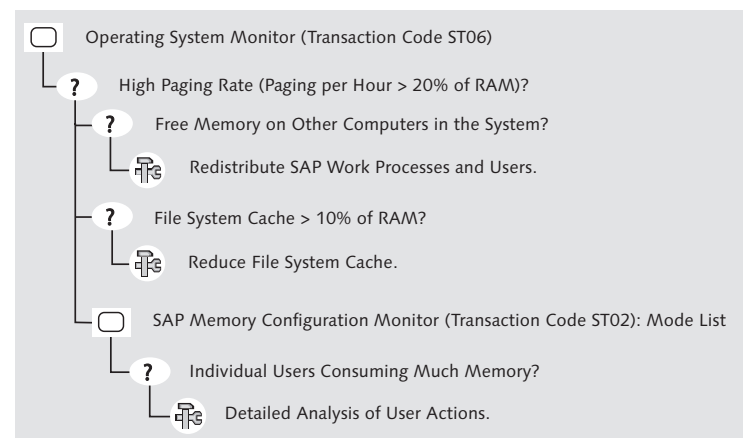


Figure 2.4 Detailed Analysis of a Hardware Bottleneck (Main Memory)

The roadmaps in Figure 2.3 and Figure 2.4 show the procedure to follow in the event of a hardware bottleneck. They refer to monitors and analyses described later in this book. You will find similar roadmaps throughout this book.

2.3 Database Monitoring

SAP NetWeaver Application Server (AS) can currently be operated with eight different relational database systems, as well as with the main memory database SAP HANA. Even if these database systems all have a different architecture, performance problems can still occur independently of the database system used. To help customers analyze and tune their databases, SAP NetWeaver AS ABAP has its own database monitor with basic functions that work independently of the database system used.

In this section, we present the steps of database performance analysis based on an Oracle database. In Appendix A, you will find information and notes on the other databases, as well as more details on Oracle databases. Chapter 15, Optimizing Database Queries with SAP HANA, deals with the SAP HANA main memory database.

The most important functions that you require for performance monitoring of the database include the following:

- ▶ Overview of the database buffers' status
- ▶ Overview of the currently running database operations, particularly the SQL statements
- ▶ Statistics on the executed SQL statements and their costs
- ▶ Overview of the distribution of read and write accesses at the logical (*tablespaces*) and physical level (for example, *container* for DB2 for Linux, UNIX, and Windows or *datafiles* for Oracle) and thus about the hard drives
- ▶ Overview of the current lock situations

Basic functions
of the database
monitor

Locks on database tables or business objects are a prerequisite for consistent data. If locks are held too long, performance problems may occur in the system because users and processes have to wait for the locks to be released. This chapter presents the lock concepts and their monitoring in detail.

9 Locks

In an SAP system, many users can simultaneously read the contents of database tables. For changes to the dataset, however, you must ensure that only one user can change a particular table's content at a time. For this purpose, table content is locked during a change operation. The first section of this chapter introduces you to the concept of locking for SAP and database systems.

If locks remain in place for a long time, wait situations can occur, limiting the throughput of the SAP system. The second section of this chapter deals with the general performance aspects of using locks.

The SAP system uses special buffering techniques for availability checking with Available to Promise (ATP) logic and for document number assignment, which minimize the lock time and maximize the throughput. These techniques will be discussed in the third and fourth sections.

When Should You Read This Chapter?

You should read this chapter to help you do the following:

- ▶ Find out more about database locks and SAP enqueues
- ▶ Analyze system problems that are caused by database locks or enqueues

This chapter does not offer instruction on programming SAP transactions. Use ABAP textbooks or SAP Help for this.

9.1 Lock Concepts of Database System and SAP System

Data consistency The consistency of the data in a database or SAP system is achieved by locks. The lock concepts of SAP and database systems have the same ultimate purpose of preserving data consistency, but they are based on different technologies and used in different situations. Locks that the database system manages are known as *database locks*, and locks that the SAP system manages are known as *SAP enqueues*.

[Ex] Ordering a Computer

When you configure and order a new computer, you must check that all necessary components are available—for example, housing, CPU, main memory, hard drive, and so on. The “all-or-nothing” principle applies: when a component is sold out, the availability of the entire computer cannot be confirmed. Since you usually check the availability of the various components in succession, you want to be certain that other users do not access the already checked and approved components until the whole order has been finally confirmed or canceled.

9.1.1 Database Locks

Database locks are managed by the lock handler of a database instance. The locked entity is typically a row in a database table (special exceptions are explained at the end of Section 9.2.1, Database Locks). Database locks are set by modifying SQL statements (`UPDATE`, `INSERT`, or `DELETE`) and by the statement `SELECT FOR UPDATE`. Locks are held until the SQL statement `COMMIT` (database commit) finalizes all changes in the database and then removes the corresponding database locks. The time interval between two commits is called a *database transaction*. A program can undo the effects of all modifying SQL statements by executing a database rollback with the SQL statement `ROLLBACK`. In this case, all database locks are also removed.

[Ex] Ordering a Computer (Continuation)

The aforementioned example of the availability check when ordering a computer using database locks is achieved in programming with the SQL statement `SELECT FOR UPDATE`: a particular item of travel data is read and locked

with this statement. When this check has been successfully performed for all components, the data is then changed (with an `UPDATE` in the relevant table rows), and then the `COMMIT` command is used to finalize changes and release all locks. Once a lock has been set, other users can still read the affected data (a simple `SELECT` is still possible), but they cannot lock it. Neither an `UPDATE` nor a `SELECT FOR UPDATE` can be performed. Such a lock is known as *exclusive*.

After a transaction step, the SAP work process automatically triggers a database commit (or a database rollback). This removes all database locks. This means a database lock is not held during multiple transaction steps (via multiple input screens in the SAP system).

9.1.2 SAP Enqueues

To hold locks during multiple steps of an SAP transaction, use SAP enqueue administration. Work processes in the enqueue table, located in the main memory, manage SAP enqueues. To retain these enqueues even when an SAP instance is shut down, save them in a local file on the enqueue server.

An SAP enqueue locks a logical object. Therefore, an enqueue can lock rows from several different database tables if these rows form the basis of a single business document. An SAP enqueue can also lock one or more complete tables. SAP enqueue objects are created and modified in the ABAP dictionary (dictionary section `LOCK OBJECTS`). They are closely related to the concepts of *SAP transaction* and *SAP logical unit of work* (SAP LUW). Both of these are described extensively in the ABAP literature for dialog programming. Therefore, this chapter will not discuss the functions and uses of these techniques as part of ABAP programs. Rather, we will focus on aspects related to performance analysis. If you discover performance problems caused by the incorrect use of SAP enqueues, consult the responsible ABAP developer.

An SAP enqueue is a logical lock that acts within the SAP system. If a row in a database table is locked by an SAP enqueue, it can still be changed by an SQL statement executed from the database or by a customer-developed ABAP program that does not conform to SAP enqueue conventions. Therefore, SAP enqueues are valid only within the SAP system. Database locks,

SAP enqueue objects

in contrast, resist all change attempts. They lock a table row “tight” for all database users and prevent changes by users outside the SAP system.

Function modules For each object that can be held by an enqueue, there are two function modules: an enqueue module and a dequeue module. An SAP enqueue is set explicitly within an ABAP program by an enqueue function and is explicitly released by a dequeue function module. As a result, SAP enqueues can be held in place through multiple transaction steps. When an SAP transaction is completed, all SAP enqueues are automatically removed.

[Ex] **Ordering a Computer (Continuation)**

Using our example of computer configuration and ordering, we will explain how SAP enqueue management works. A computer consists, for example, of the housing, CPU, main memory and hard drive. The individual components are edited on different input screens—that is, with several transaction steps—and are locked for editing by SAP enqueues. After determining the availability of each component, you can confirm the order for the entire computer. This concludes the dialog part of the transaction.

Under the protection of the enqueues, an update work process then transfers the changes to the database tables. When the update has been completed, the SAP LUW is finished, and the enqueues are unlocked.

An SAP LUW can also contain program modules that require a V2 update. An SAP enqueue is not used for this. You should not use modules that use this V2 update to process data that requires the protection of enqueues (also see Chapter 7, Section 7.1.8, Update).

Table 9.1 compares the main features of database locks and SAP enqueues.

	DB Locks	SAP Locks (Enqueues)
Locked object	Individual rows of a database table	Logical object (for example, a document) defined in the ABAP Dictionary
How object is locked	Implicitly, using modifying SQL statements (such as UPDATE and SELECT FOR UPDATE)	Explicitly, by the ABAP program calling an enqueue module

Table 9.1 Features of Database Locks and SAP Enqueues

	DB Locks	SAP Locks (Enqueues)
How lock is removed	Implicitly, with the SQL statement COMMIT or ROLLBACK Usually at the end of a transaction step	Explicitly, by calling a dequeue module Usually at the end of an SAP transaction
Maximum duration	One transaction step	Over multiple transaction steps
Result of lock conflicts	Wait situation (exclusive lock wait, also see section 9.2.1, Database Locks)	Program-specific, for example, the error message “Material X is locked”
How to monitor	Transaction DB01, Exclusive Lock Waits	Transaction SM12, Enqueue Monitor

Table 9.1 Features of Database Locks and SAP Enqueues (Cont.)

9.2 Monitoring Locks

In this section, you will find information on how to monitor database locks and SAP enqueues.

9.2.1 Database Locks

What happens in the event of a lock conflict—that is, when a work process wants to lock an object that is already locked? With database locks, the second process waits until the lock has been removed. This wait situation is known as an *exclusive lock wait*. Most databases do not place a time limit on these locks. If a program fails to remove a lock, the wait situation can continue indefinitely.

Exclusive lock waits

This could become a major problem if the program fails to release a lock on critical SAP system data, such as the number range table NRIV. There is a danger that one work process after another will be waiting for this lock. If all work processes are waiting, no work process is available to allow you to intervene from within the SAP system. If you can identify the program holding the problem lock, you can terminate it through the operating system as a last alternative.

Checking lock wait situation

To monitor current lock wait situations, call the database-lock monitor (Transaction DB01), which you can start from the DBA Cockpit (Transaction DBACOCKPIT) by selecting PERFORMANCE • WAIT SITUATIONS ON LOCKS AND DEADLOCKS or from the system-wide work process overview (Transaction SM66) by selecting GoTo • DB LOCKS.

For a description of this monitor and information on how to troubleshoot lock wait situations, see Chapter 2, Section 2.3.5, Other Database Checks. Lock wait situations increase database time and result in high database times in workload-monitor statistics. Some database systems explicitly monitor lock wait times, which you can view in the database performance monitor.

[Ex] Lock Situation in the Database

With the following sample program, you can provoke a lock situation in the database:

```
REPORT zts_lock.
DATA: lv_text type natxt.
SELECT SINGLE FOR UPDATE text FROM T100 INTO lv_text WHERE
sprsl = 'DE' AND ARBGB = '00' AND msgnr = '001' .
BREAK-POINT.
```

To do so, proceed as follows:

1. Start the program in the ABAP Workbench (for instance, via Transaction SE38). After a few seconds, the debugger opens, and the program stops at the `BREAK-POINT` command. Before that, the program has set a database lock using the `SELECT SINGLE FOR UPDATE` command. Since the program waits in the debug mode, this database lock is not undone.
2. Open a second session and restart the program. An hourglass is displayed in the second session.
3. Again, you can restart the program in a third session, and the system again displays an hourglass.
4. Open another session and start the database lock monitor as previously described. The lock situation is displayed, and you can see which work process holds the lock and which one waits. Based on the work process overview (Transaction SM50) and the database process monitor, you can now analyze what happens in the process that holds the lock. In this example, the process overview displays the `STOPPED` status and gives `DEBUG` as the reason.

5. Go to the debugger, where you continue the execution of the program in the first session. The program is terminated, and the database lock is undone due to an implicit commit or rollback of the database interface. As a result, the program can continue in the second session, which had to wait at the `SELECT SINGLE FOR UPDATE` command up to now. Within a very short period of time, it will thus reach the `BREAK-POINT` command and start the debugger.
6. Continue the program in the debugger for the second session and, if you started the program in further sessions, in these sessions to release the locks.

Basically, you should set programs to request locks as late as possible. It is preferable for a program to read and process data from the database before setting locks or making changes in the database. This is illustrated in Figure 9.1. The top part of the diagram shows how several changes are made during a database transaction and how, as a result, database locks are held for too long. The lower part of the diagram shows a more appropriate programming method: the transaction is programmed so that it collects the changes in an internal table and then transfers these changes to the database as a group at the end of the transaction. This reduces the lock time in a database.

Typical problems

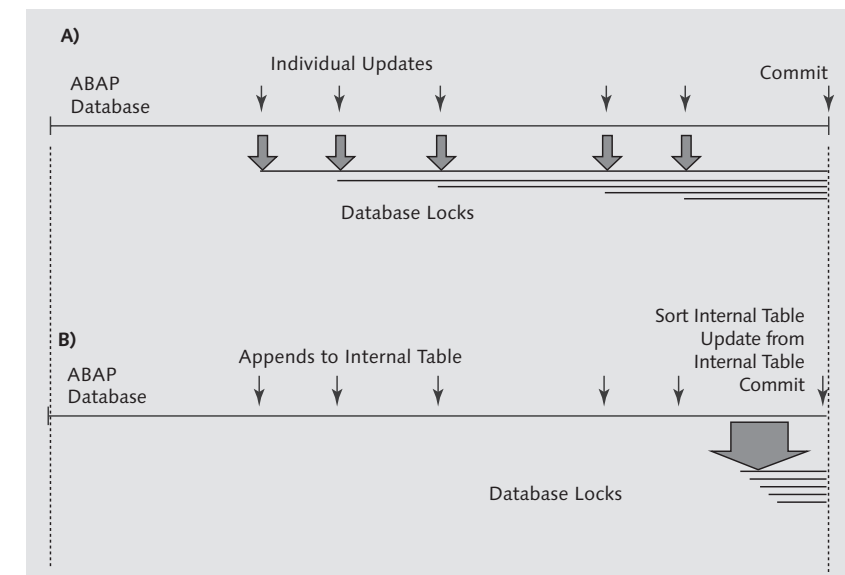


Figure 9.1 Locks Should Be Set as Late as Possible

Causes Performance problems due to delays in releasing locks frequently occur when customers modify the programming of update modules. The separation of update modules from dialog modules is an attempt to reduce the number of locks needed in the dialog part of a transaction because changes to the database and the associated locks are mainly the task of the update modules. However, sometimes the update module is modified—for example, to supply a customer-developed interface with data. This modification may cause problems if the update module has already set locks and, for example, the modification generates expensive SQL statements. The locks cannot be released until the SQL statements are fully processed, and lengthy lock waits may result.

Another source of problems with locks is background programs that set locks and then run for several hours without initiating a database commit. If dialog transactions need to process the locked objects, they will be forced to wait until the background program finishes or initiates a database commit. To solve this problem, you should ensure that the background program either initiates a database commit at regular intervals (without sacrificing data consistency) or runs only when it will not interfere with dialog processing. Similar problems may occur when background jobs are run in parallel—that is, when a program is started several times simultaneously. Parallel processing is recommended only when the selection conditions of the programs do not lock the same data.

While you are working in the ABAP debugger, database commits are generally not initiated, and all locks stay in place until you are finished. You should therefore avoid using the debugger in a production SAP system.

[Ex]

Deadlocks

We'll now present an example of a situation known as a *deadlock*. Assume that work process one and work process two both want to lock a list of materials. Work process one locks material A, and work process two locks material B. Then, work process one tries to lock material B, and work process two tries to lock material A. Neither work process is successful because the materials already have locks on them. The work processes block each other. A deadlock is identified by the database instance and solved by sending an error message to one of the work processes. The corresponding ABAP program is terminated, and the error is logged in the SAP syslog.

You can avoid deadlocks with correct programming. In our example, the program should be changed so that its internal material list is sorted before any locks are set. Then the lock on material A will always be set before the lock on material B. Therefore, programs requiring the same materials are serialized and not deadlocked.

Deadlocks should occur very rarely. Frequent deadlocks indicate incorrect programming or configuration of the database instance.

If, in some database systems (for example, DB2 and SAP MaxDB), a work process places single-row locks on more than 10% of the single rows in a table, the locks are automatically replaced by table locks. Here, the database decides that it is more efficient to lock the entire table for a work process than to maintain several locks on individual rows. Table locking has consequences for parallel processing in background jobs, where each program is intended to update a different part of the same table at the same time. It is not possible to schedule background jobs so that one updates the first half of the table and the other updates the second half because the database may decide to lock the table exclusively for one of the jobs. One program that is particularly affected by this is the period closing program in materials management.

Table locks

Table Locks

There are database parameters you can use to specify when the database should convert single-row locks to a table lock.

Sometimes, the database locks entire tables for administrative reasons. This happens when indexes are created or when particular tables and indexes are analyzed—for example, during the Oracle analysis `VALIDATE STRUCTURE`. If these actions are performed during production operation, substantial performance problems may result.

[«]

9.2.2 SAP Enqueues

SAP enqueues are managed in the enqueue table located in the global main memory of the enqueue server. The work processes in the enqueue server directly access the enqueue table; the enqueue server also carries out lock operations for work processes from other application servers, which are communicated via the message service (❶ and ❷ in Figure 9.2).

The following abbreviations are used in Figure 9.2: DIA for dialog work process, ENQ for enqueue work process, and ENQ tab for enqueue table.

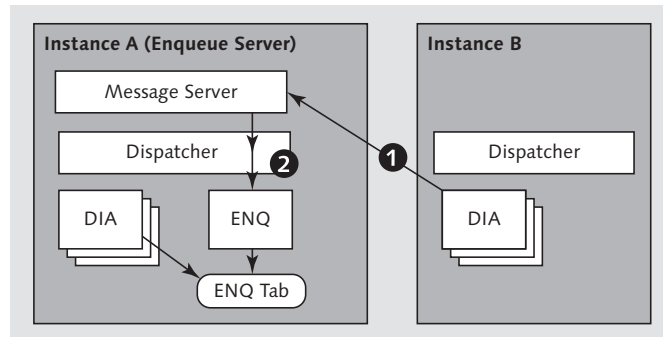


Figure 9.2 Communication for Setting and Removing SAP Enqueues

For work processes in the enqueue server, setting and releasing locks takes less than 1 millisecond; for work processes in other application servers, it takes less than 100 milliseconds.

Performance problems with SAP enqueues

If an SAP enqueue is requested but already held by another user, the attempt to set a lock is rejected and an error message is sent back to the ABAP program. The application developer has to decide how to deal with this error message with suitable programming. For programs in dialog mode, the error message is normally forwarded to the user, for example, with the message "Material X is locked by user Y." For background programs, you will normally attempt to set the lock again later. After a certain number of unsuccessful attempts, an error message is written to the program log.

If SAP enqueues are held for too long, performance problems can arise because after a failed attempt, the user will repeat the entry. Take, for example, a user who needs to process a material list, and to do, so needs to set 100 SAP enqueues. If the attempt to set lock number 99 fails, the program is interrupted with the message "Material number 99 is locked," and all of the previous system work is in vain and must be repeated. Therefore, rejected enqueue requests lead to higher system workload and restrict the throughput of transactions.

You can get an overview of all currently active SAP enqueues by using Transaction SM12, under the following menu:

TOOLS • ADMINISTRATION • MONITOR • LOCK ENTRIES

Start the test programs under the following menu to diagnose errors:

EXTRAS • DIAGNOSE OR EXTRAS • DIAGNOSE IN VB

If errors are identified, check the SAP Service Marketplace for notes, or contact SAP directly.

You can view statistics on the activity of the enqueue server with the menu option EXTRAS • STATISTICS. The first three values show the number of enqueue requests, the number of rejected requests (unsuccessful because the lock requested was already held by another), and the number of errors that occurred during the processing of enqueue requests. The number of unsuccessful requests should not be more than 1% of the total number of enqueue requests. There should be no errors.

Enqueue statistics

9.3 Number Range Buffering

With many database structures, it is necessary to be able to directly access individual database records. You do this with a unique key. *Number ranges* assign a serial number that forms the main part of this key. Examples of these numbers include order numbers and material master numbers. SAP number range management monitors the number status so that previously assigned numbers are not re-issued.

9.3.1 Fundamentals

A business object for which a partial key must be created using the number range is defined in the SAP system as a *number range object*. A number range contains a *number range interval* with a set of permitted characters. The number range interval is made up of numerical or alphanumeric characters and is limited by the FROM-NUMBER and TO-NUMBER fields. You can assign one or more intervals to a number range.

The current number level of a range, which is the number that is to be assigned next, is stored in the database table NRIV. If a program needs

Technical implementation

Index

A

- ABAP, 465, 468, 787
 - web services*, 372
- ABAP application instance, 76, 265
- ABAP Class Builder, 553
- ABAP class library, 778
- ABAP debugger, 183, 210, 226
- ABAP Dictionary, 107, 480, 481, 503, 521, 551, 787
 - display*, 778
 - maintenance*, 778
 - table*, 778
- ABAP Editor, 296, 512, 778
- ABAP/heaplimit, 169
- ABAP instance, 344
- ABAP List Viewer, 205
 - Control*, 348
- ABAP objects, 553
- ABAP program, 495, 501
 - quality analysis*, 216
 - runtime analysis*, 188
- ABAP program termination, 291
- ABAP repository information system, 778
- ABAP runtime analysis, 201, 203, 226, 496
 - runtime error*, 779
 - Web Dynpro applications*, 202, 376
- ABAP server, 155
 - statistics*, 147
- ABAP shared object, 293
- ABAP trace, 65, 183, 201
 - summary*, 208
 - single-transaction analysis*, 209
 - variants*, 205
- ABAP web service
 - performance analysis*, 373
- ABAP Workbench, 216, 386, 515, 778
 - BSP development*, 369
- Access plan, 748
- ACID principles, 787
- Action profile, 176
- Activation queue, 720
- Active data, 720
- Active session history, 100
- Activity, 169
- Adaptive Computing Controller, 258
- Address space, 270, 787
- Administration tool
 - database*, 516
- Agent, 173, 746, 748
- Agent private memory, 746
- Aggregate, 467, 603, 609, 651, 659, 681
 - compress*, 611
 - create*, 606
 - functions*, 497
 - indexing*, 610
 - maintenance*, 609, 610
 - suggestion*, 608
- Aggregation, 206
- Alert, 136
- Alert messaging
 - automatic*, 141
 - monitor*, 787
 - server*, 622, 627, 638
- Allocation analysis, 424
- Allocation rate, 429
- Allocation trace, 457
- American National Standards Institute (ANSI), 787
- Analysis linguistic, 619, 664
- Analysis method, 135
- Analytical applications, 562
- APO server, 261
- Appliance, 664
- Application analysis, 145
 - business document volume*, 779
- Application buffer, 559
 - monitor*, 279, 551
 - object-oriented*, 521
- Application error, 294
- Application level, 38
- Application Link Enabling (ALE), 787
 - Administration*, 777
- Application monitor, 779

Application optimization, 20
 Application server, 41, 76, 155, 195, 787
 Application Service Providing (ASP), 25
 Application support layer, 746
 Application tuning, 61, 30
 Archiver stuck, 108, 164
 Array fetch, 192
 ATAB, 516
 ATP logic, 398
 ATP server, 304, 398, 400, 403
 ATP service, 262, 303
 Attribute, 563, 569, 575
 display attribute, 569
 navigation attribute, 569
 Attribute index, 44
 Attribute vector, 669, 677
 Automatic workload repository, 100
 Auto-reaction method, 141
 Auxiliary storage pool, 282
 Availability, 47, 176, 303
 Availability check, 398, 553
 Average response time, 170

B

Background job
 scheduling monitor, 777
 Background load, 165
 Background processing, 787
 analysis, 778
 Background program, 230, 388
 Background service, 262, 302, 304
 Backup, 48, 49, 53
 Backup index server, 642
 Bandwidth, 674
 BAPI, 787
 Batch input, 230
 Benchmark, 242, 245, 795
 BEx analyzer, 565
 BEx web analyzer, 565
 bgRFC monitor, 338
 Big Data, 24, 663
 Binary search, 677, 730
 Blade server, 664
 Block, 92

Bottleneck analysis, 145, 306
 Browser, 787
 BSP applications
 runtime analysis, 202, 376
 B* tree, 590
 Buffer, 43, 91, 521
 access, 526
 attribute index, 44
 buffer hierarchy, 44
 catalog buffer, 746
 communication buffer, 746
 database buffer, 44
 data buffer, 92
 data model-oriented, 44
 get, 465
 management, 526
 metadata buffer, 742
 object-oriented application buffer, 44
 operating system buffer, 44
 package buffer, 746
 pool, 746
 process buffer, 741
 quality, 92, 110
 setting, 114, 515
 single-record buffer, 526
 status, 539
 storage subsystem, 44
 synchronization, 528, 558, 777
 synchronization monitor, 549
 TABL, 526
 table buffer, 44, 526, 557
 TABLP, 526
 trace, 188, 197
 types, 44, 521, 523
 Business hours, 53
 Business process, 47
 Business Server Pages (BSPs), 39, 347, 357, 367
 Button, 788
 BW Administrator Workbench, 591, 595, 608, 652, 777
 BW aggregate maintenance, 777
 BWA index, 652
 checking and switching off, 653
 maintain, 654
 BW check report, 777

BW query, 562, 565
 analysis, 580
 background print, 601
 BW Query Monitor, 777
 BW workload statistics, 579

C

Calc engine, 684
 CALL, 272
 Catalog buffer, 556
 Catalog cache, 732, 746
 CA Wily Introscope, 65, 452, 457
 Enterprise Manager, 454
 WebView, 454
 Workstation, 454
 Central monitoring system, 174
 Central SAP monitor, 65, 778
 Central single-record statistics, 218
 Change, 537, 545
 Change and Transport Organizer, 788
 Change and Transport System, 531
 Change Log, 720
 Change request, 531
 Change run, 612, 656
 Changing secondary indexes, 487
 Characteristic, 563, 575
 Characterizing parameters, 146
 Checking group, 401
 Checkpoint, 92
 Check set, 627
 Class loader, 409
 Class statistic, 457
 CLEAR statement, 213
 Client, 788
 Client destination statistics records, 333
 Client/server architecture, 38
 scalability, 46
 Client statistics records, 333
 Cloud application, 662
 Cluster coding, 671
 Clustered index, 774
 CO, 788
 Code cache, 416
 Code completion, 703
 Code Inspector, 183, 216, 226, 527, 702
 Code push-down, 659, 681
 Column-based data storage, 616, 659, 665, 667
 Column store, 683, 694
 Command monitor, 734
 Common Programming Interface
 Communication, 788
 Compilation, 416
 Compressibility, 680
 Compression, 616, 625, 666, 668
 cluster coding, 671
 prefix coding, 671
 run-length coding, 671
 sparse coding, 671
 Computer, 76, 124
 Computer Center Management System (CCMS), 29, 35, 63, 134, 484, 788
 Alert Monitor, 135, 330
 monitoring, 134
 monitoring tree, 138
 System Component Recovery (SCR), 174
 Concordance, 620
 Concurrent mark-sweep collector, 412
 Condition table, 535, 550
 Connection view, 422
 Consolidation strategy, 261
 Container, 89
 Context switch, 80, 271, 788
 Continuous performance optimization, 57
 Controls, 347
 CO-PA Accelerator, 661
 Core, 80, 742
 Cost-based optimizer (CBO), 476
 Coupling
 hard, 321
 soft, 321
 CPI-C, 788
 CPU, 46, 80, 788
 bottleneck, 81, 84, 106, 165
 load, 79, 82, 106, 166, 250
 requirement, 245, 343
 resources, 305
 sizing, 686
 time, 156, 159, 162, 168, 187, 306
 trace, 201
 wait time, 307

Creating secondary indexes, 487
 Cursor, 193
 Cursor cache, 270
 Cursor ID, 193
 Customer interaction center, 212
 Customizing, 788
 Customizing data, 534
 Customizing Organizer, 788

D

Data archiving, 788
 Database, 76, 788
 analysis, 728
 buffer, 91, 665, 729
 error log file, 105, 750
 standstill, 108
 tasks, 733
 Database access, 183
 fully qualified, 191
 Database administration, 110
 Database administrator cockpit *see* DBA
 Cockpit, 727
 Database analysis, 89
 Database Analyzer, 738
 Database buffer, 91, 665
 Database consolidation, 256, 260
 Database global memory, 746
 Database heap, 746
 Database index
 missing, 107
 Database instance, 76, 289, 788
 Database level, 40
 Database load, 499, 722
 Database lock, 104, 382, 383, 385, 450,
 750, 789
 IBM DB2 for z/OS, 762
 Informix, 743
 MaxDB, 737
 monitor, 124, 386
 SQL Server, 773
 Database monitor, 64, 89, 118, 386, 727
 Database object
 missing, 777
 Database operation, 192

Database optimizer, 107, 472, 475, 492,
 702, 789
 Cost-based (CBO), 107
 Database performance monitor, 90, 779
 Database performance problem, 167
 Database procedure, 156
 calls, 156
 subrecord, 156
 time, 156
 Database process, 82, 698
 monitor, 96, 124, 189, 450, 457,
 733, 742, 755, 762, 773
 Database processor, 106
 Database query parallelization, 666
 Database response times
 long, 125
 Database server, 40, 76, 789
 Database service, 343
 Database system, 76, 382, 788
 parallel, 76
 Database table, 561
 Database time, 154, 162, 167, 168, 185
 Database tuning, 30
 Database view, 509, 660, 731, 739, 745,
 752, 759, 765, 769, 775
 Data buffer, 92, 93, 94, 746, 761, 771
 Data cache, 92, 732, 771
 Data Control Language (DCL), 789
 Data Definition Language (DDL), 789
 Datafile, 89
 Data locality, 675
 Data Manipulation Language (DML),
 789
 Data mining, 562, 664
 Data modeling
 SAP HANA, 679, 684
 Data package, 612
 Data retention
 centralized, 642
 decentralized, 642
 DataStore object, 563, 564, 575
 HANA-optimized, 717, 719
 indexing, 593
 Data volume, 737
 transferred, 349
 Data warehouse, 563
 DB2 for LUW, 747

DBA, 789
 DBA Cockpit, 90, 125, 386, 481, 483,
 687, 727, 732
 DB2 for IBM i, 755
 DBA log, 777
 DBA Planning Calendar, 483
 Deadlock, 388, 789
 Debugger, 417
 Debugging, 211
 Decomposition, 625
 vertical, 625
 Default data cache, 741
 DELETE statement, 213
 Delta index, 636, 639, 655, 672, 708,
 710
 activate, 640
 integrate, 641
 Demilitarized Zone (DMZ), 40, 264
 Deoptimization, 416
 Dequeue module, 384
 Destination, 322, 326
 Developer log, 295
 Developer trace, 779
 Development Workbench, 35
 DIAG protocol, 789
 Dialog load, 165
 Dialog response time, 319
 Dialog service, 262, 302, 304
 Dialog user, 230, 312
 Dialog work process, 154, 314, 339, 789
 Dictionary, 677
 Dictionary coding, 668
 Difference coding, 671
 Dimension ID, 566
 Dimension table, 566
 index, 591
 Direct read, 191, 730, 739, 744, 758,
 763
 Disk read, 465
 Dispatcher, 302, 789
 process, 262
 queue, 130, 153
 wait time, 153, 162, 167, 306
 Dispatching, 263
 Displacement, 114
 Displacement (swap), 531, 535
 Display attribute, 569

Display statistics on table accesses (table
 call statistics), 779
 disp+work, 262
 Distributed Statistics Record (DSR), 173,
 218
 Document, 230
 Documentation, 61
 Dominator tree, 447
 Drilldown, 564
 Dual-stack installation, 260
 Dump, 291
 Dynamic statement cache, 761
 Dynamic user distribution, 264, 308
 Dynpro, 789

E

Easy Web Transaction (EWT), 370, 789
 eCATT, 789
 Eclipse, 789
 Eden, 409, 411
 EDI, 789
 EDM DSC cache, 761
 EDM pool, 761
 E fact table, 596
 Electronic Data Interchange, 789
 Embedded Search, 618
 em/initial_size_MB, 126
 End-to-end diagnostics, 69
 End-to-end runtime analysis, 220, 223,
 226
 SAP GUI transaction, 222
 End-to-end trace, 206, 360, 451
 End-to-end workload analysis, 178
 End user experience monitoring, 54
 End User Experience Monitoring, 365
 Enqueue, 382, 789
 module, 384
 operation, 188
 server, 302
 service, 262, 302, 303
 table, 390
 trace, 188, 199
 Enterprise File Search (EFS), 617
 Enterprise IMG, 789
 Enterprise JavaBeans, 40

- Entity, 789
 - Error code
 - DBIF_RSQL_NO_MEMORY*, 297
 - EXPORT_NO_SHARED_MEMORY*, 293
 - EXSORT_NOT_ENOUGH_MEMORY*, 297
 - PXA_NO_SHARED_MEMORY*, 292
 - SET_PARAMETER_MEMORY_OVERFLOW*, 292
 - STORAGE_PARAMETERS_WRONG_SET*, 291, 295, 297
 - SYSTEM_NO_MORE_PAGING*, 280
 - SYSTEM_NO_ROLL*, 292
 - TSV_TNEW_PAGE_ALLOC_FAILED*, 291
 - TSV_TNEW_PG_CREATE_FAILED*, 280
 - Escalation procedure, 56
 - Event, 368
 - Event data, 664
 - Exclusive database locks, 104
 - Exclusive lock wait, 104, 112, 131, 144, 385, 404, 762
 - SQL Server*, 773
 - Execution plan, 472, 473, 702, 729, 789
 - DB2 for IBM i*, 757
 - IBM DB2 for z/OS*, 762
 - Microsoft SQL Server*, 774
 - Oracle*, 767
 - Executive Information System (EIS), 561
 - Expert monitor, 63
 - Export/import buffer, 399, 522, 551
 - Export/import SHM buffer, 293, 522, 551
 - Extended global (EG) memory, 279
 - Extended memory, 294, 790
 - Extended memory area (EM), 293
 - Extensible Markup Language (XML), 798
 - Extraction, 566
- F**
-
- Fact table, 566, 611
 - compress*, 595, 655
 - Fact table (Cont.)
 - index*, 591
 - SAP HANA*, 718
 - Failover recovery, 259
 - Failover solution, 304
 - FE Net Time, 350
 - Fetch, 537
 - Fetch operation, 192, 505
 - F fact table, 596
 - Fiber Distributed Data Interchange (FDDI), 790
 - File and network I/O analysis, 439
 - File operation, 439
 - Files statistics, 439
 - File system cache, 85
 - Firewall, 790
 - Flow, 563
 - FOR ALL ENTRIES, 505, 507, 509
 - Fragmentation, 547, 555
 - FREE statement, 213
 - Frontend
 - time*, 350
 - trace*, 221
 - Full table scan, 473, 480, 488, 589, 730
 - Function builder, 318
 - Function module, 317
- G**
-
- Garbage collection (GC), 408, 409, 456, 790, 793, 809
 - analysis*, 441, 457
 - collector*, 412
 - compacting*, 411
 - full*, 410
 - JVM*, 409
 - log*, 415
 - mark and copy*, 411
 - mark and sweep*, 411
 - partial*, 410
 - Gateway monitor, 324
 - Generation, 409
 - Generation time, 154
 - Global cache hit ratio, 762
 - Globally Unique Identifier, 790
 - GoingLive Check, 25, 295

- GoingLive Functional Upgrade Service, 254
 - Gross-time optimization, 210
 - GUI, 790
 - communication*, 350
 - controls*, 348
 - time*, 155, 187, 349, 352
 - GUID, 790
- H**
-
- Hard disk, 674
 - access*, 737
 - monitor*, 773
 - Hardware, 46
 - analysis*, 77
 - Hardware bottleneck, 82, 165
 - analysis*, 79
 - Hardware capacity, 166
 - Hardware consolidation, 256, 258
 - Hardware landscape, 303
 - Hardware monitoring, 75
 - Hardware partner, 231, 234
 - Hardware sizing, 229, 231, 401
 - Hardware tuning, 29
 - Harmonization, 256
 - Hash table, 214
 - Heap, 272
 - dump*, 447
 - dump analysis*, 457
 - memory*, 790
 - Hierarchy, 571, 575
 - High availability, 303, 304, 790
 - High-availability cluster, 259
 - High water mark, 115
 - Hints, 493
 - Hit ratio, 92
 - HotSpot, 102, 430, 676, 773
 - HotSpot Java Virtual Machine, 411
 - HPROF file, 447
 - HTML business template, 372
 - HTML Control, 348
 - HTTP call, 376
 - HTTP trace, 199, 225, 359
 - HTTPWatch, 364
 - HybridProvider, 563, 564
- I**
-
- IAC, 790
 - IBM DB2
 - for IBM i*, 753
 - for Linux, UNIX, and Windows*, 746
 - for z/OS*, 760
 - IBM DB2 for Linux, UNIX, and Windows
 - execution plan*, 751
 - IBM i, 282, 754
 - IBM Informix Dynamic Server
 - execution plans*, 744
 - IDES, 790
 - Index, 603, 659, 681, 707
 - administration*, 480
 - B* index*, 677
 - bitmap index*, 590, 593
 - B* tree index*, 593
 - BWA index*, 652
 - concatenated*, 679, 707
 - create*, 480
 - divide*, 637
 - efficient*, 193
 - fragmentation*, 485
 - inverted*, 677, 707
 - logical*, 623, 637
 - main memory*, 616
 - maintain*, 480
 - missing*, 107
 - physical*, 623, 637
 - primary index*, 108
 - quality*, 486
 - reorganize*, 485
 - TREX index*, 616
 - Indexing, 588, 645
 - Index range scan, 473, 480
 - Index scan, 590
 - Index server, 621, 642, 683
 - Index unique scan, 473
 - Index update, 645

InfoCube, 563, 566, 575, 605
HANA migration, 777
HANA-optimized, 717
indexing in the BWA, 650
overview, 777
transactional, 591
InfoObject, 563
InfoProvider, 563, 575, 597
InfoSet, 563
Initial sizing, 232, 236
In-memory application, 662
In-memory computing, 616, 666
In-memory database, 674
Insert-only approach, 680
Installation
 central, 303
 distributed, 303
Integer ID, 668
Integrity, 48
Interaction model, 348
Interfaces, 301
Internal Document (IDoc), 791
 IDoc type, 790
Internal table, 210
International Demo and Education
 System, 790
Internet Application Component, 790
Internet Communication Framework
 (ICF), 367, 370
Internet Communication Manager
 (ICM), 133, 262, 357, 359, 367, 621
 monitor, 65, 133, 360, 373, 779
Internet connection, 347
Internet of Things, 663
Internet Pricing and Configuration (IPC),
 261, 271, 341, 791
Interpreter, 415, 416
Inter Process Communication (IPC), 791
Intranet, 791
Introscope statistics, 174
Introscope Trace, 456
Invalidation, 114, 530, 531
I/O bottleneck, 86, 103, 194
I/O operation, 457
iSeries, 282
iView, 38

J

Java, 792
Java application instance, 76, 265
Java bytecode, 416
Java Development Kit (JDK), 408
Java Dictionary, 521, 556
Java heap, 409, 414, 457
 analysis, 447
Java runtime, 421
JavaScript, 683
Java Server Page (JSP) , 40, 347, 357
Java Servlet, 40, 347
Java statistics, 173
Java trace, 65
Java Virtual Machine (JVM), 341, 407,
 457, 792, 797
 memory area, 409
 work processes, 418
Java workload monitor, 178
Job analysis, 778
Job overview, 611
Join, 575, 590
Join engine, 684
Just-in-Time (JIT) compiler, 407, 408,
 416, 431

K

Key figure, 146, 563
Key performance indicator (KPI), 135
Knowledge management (KM), 617

L

Landscape reorganization, 636, 638
Landscape replication, 684
Large IO pool, 741
Latency time, 359, 440, 674
Leaf, 590
Linearity, 238
Line item dimension, 575, 719
Linux, 281
liveCache, 261

Load, 160, 170
 inbound, 330
 outbound, 330
Load distribution, 147, 166, 301
 ABAP, 302
 incorrect, 82, 127
 new, 338
Load from external systems, 176
Loading, 566
Load profile, 147, 454
Load time, 154
Local Area Network (LAN), 45, 792
 LAN check 87
Local memory, 270
Lock, 381, 382, 437, 457
 database lock, 382
 lock conflict, 385
 locking with quantities, 400, 401
 lock object, 383
 monitoring, 385
 SAP enqueue, 382, 398
 shared enqueues, 400
 table lock, 389
 wait situation, 381
Lock escalation, 737
Lock handler, 382
Locking with quantities, 304
Lock list, 746
Log area, 108, 737
Logical analysis, 61
Logical changes, 164
Logical unit of work (LUW), 336, 792
Logistic Information System (LIS), 317,
 561
Logon group, 264, 308, 329
 maintenance, 779
 monitor, 303
Loop
 nested, 214
Low-speed connection, 355
LRU, 792

M

Machine code, 416
Main memory, 46, 269, 659, 665, 674,
 707

Main memory (Cont.)
 bottleneck, 81, 85, 165
 buffering, 392
 configuration monitor, 249
 requirement, 183, 287
 sizing, 685
 workload, 80
Maintenance view DBDIFFVIEW, 482
Master data, 534
 Master data table indexing, 593
Master index server, 636, 642
Master service, 715
Memory
 allocation, 117, 275, 691
 area, 283, 298
 available, 286
 configuration, 112, 269
 configuration monitor, 290, 295,
 403, 543, 553, 555, 598, 779
 extract, 212
 fixed allocated (heap), 293
 Inspector, 210, 212
 leak, 446, 449, 457
 local, 270, 792
 management, 792
 physical, 118, 690
 pipe, 133, 370
 profile, 168
 shared, 270
 used, 691
 virtual, 270, 691, 797
 virtual required, 286
Memory Analyzer, 446, 457
Memory management, 116, 168
 IBM i, 282
 integrated, 409
 Linux, 281
Merge, 640, 672, 710
 auto merge, 711
 critical merge, 712
 hard merge, 711
 smart merge, 712
Message server, 229, 263, 344
Message service, 262, 303
Metadata, 91
Method, 416, 457
 memory requirement, 426
Method Parameter Trace, 457

Microsoft SQL Server, 470
 Microsoft Windows, 280, 307
 Migration, 255
 Missing index, 481
 Mobile client, 38
 Mode, 272, 793
 PRIV mode, 290
 Mode list (mode list), 290
 Model View Controller (MVC), 367
 Modularization unit, 201, 210
 Monitoring, 47, 55
 central, 50
 Monitoring agent, 140, 218
 Monitoring plan, 47, 50
 Monitoring system
 central, 174
 Monitoring tree, 138
 Moore's law, 793
 Multicore processor, 80
 Multilinguism, 575
 MultiProvider, 563, 565, 575
 Multithreaded CPU, 80

N

Name server, 621, 683
 Nametab buffer, 522
 Navigation attribute, 569, 575
 Nested loop join, 510, 590, 745
 Net time, 350
 Network, 87, 194, 306
 alert monitor, 779
 connection, 439
 graphics, 777
 I/O and file I/O trace, 457
 monitor, 779
 operation, 439
 problems, 194
 transfer time, 365
 tuning, 29
 Non-dialog work processes, 277
 Nonoptimal load distribution, 167
 NRIV, 397
 Number range, 391
 buffering, 391, 393, 396

Number range (Cont.)
 buffer mode, 397
 interval, 391
 number range level, 396
 object, 391

O

Object Linking and Embedding (OLE),
 793
 OLAP, 246, 261, 562, 793
 OLAP cache, 553, 576
 configuration, 598, 777
 monitor, 599, 777
 monitoring, 596
 OLAP engine, 683
 OLAP processor, 576
 Old generation, 409, 413
 collectors, 412
 Online Transaction Processing (OLTP),
 246, 261, 562, 793
 Open object, 772
 Open operation, 192
 Open SQL monitor, 451
 Operating mode, 778, 793
 Operating system, 269, 298, 793
 command, 779
 configuration parameter, 291
 file, 777
 limit, 295
 monitor, 64, 77, 86, 124, 189, 249,
 353, 753, 779
 paging, 271
 parameter, 87
 process, 82
 restrictions, 284
 swap space, 291
 Operation mode, 312
 Optimization
 technical, 20
 Optimization plan, 47, 50
 Oracle, 93, 767
 Oracle wait event, 98
 Outsourcing, 24

P

Package cache, 746
 Package dimension, 572
 Page in, 796
 Page out, 796
 Pages, 92
 Paging, 793
 Paging file, 283, 298
 Paging rate, 81, 85, 753
 Parallelization, 666, 675, 715
 Parameter change, 87
 Parameter maintenance, 360
 Parameters
 characterizing, 146
 Parsing, 472
 Partitioning, 315, 625, 675, 716
 area-based, 676
 horizontal, 675
 round robin, 676
 vertical, 675
 PASE runtime environment, 282
 Passport, 173, 218, 796
 Pending period, 529, 538
 PERFMON program, 364
 Performance, 48, 793
 measurement, 348
 Performance forum, 72
 Performance HotSpot analysis, 430
 Performance HotSpot Trace, 457
 Performance indicator, 134
 Performance management
 proactive, 21
 Performance problem
 general, 161
 specific, 161, 169
 Performance trace, 188, 352, 585, 635
 Performing sizing project, 238
 Permanent generation, 409, 414
 Permanent performance problem, 163
 Physical main memory (RAM), 269,
 283, 298
 Physical read access, 92
 Pivoting, 564
 Planning application, 562
 Plug-in, 37
 Pool size, 649
 Pop-up window, 793
 Prefix coding, 671
 Prepare, 193
 Prepared statement, 761
 Prepared templates, 372
 Preprocessing
 distribution, 637, 648
 Preprocessor, 621, 637, 683
 capacity, 649
 configure, 648
 modes, 648
 Presentation level, 38
 Presentation server, 41
 analyses, 360
 Presentation server trace, 362
 Primary index, 471, 482, 730
 Priority class, 339
 Private mode, 126, 277
 Probe, 453
 Procedure cache, 771
 Process
 complete, 127
 external, 84
 stopped, 126
 Process after input (PAI), 793
 Process before output (PBO), 793
 Process chain, 566
 Process ID, 124
 Processing time, 156, 159, 162
 Processor, 80, 255, 788
 Processor thread, 345
 Profile parameter, 291
 abap/atrapath, 203
 abap/atrasizequota, 203
 abap/heap_area_dia, 275, 278, 294,
 296
 abap/heap_area_nondia, 275, 278,
 294, 297
 abap/heap_area_total, 275, 287, 295
 abap/heaplimit, 275
 dbs/io_buf_size, 193
 em/address_space_MB, 281
 em/blocksize_KB, 274
 em/initial_size_MB, 116, 274, 289, 295
 em/max_size_MB, 281
 maintain, 778
 PHYS_MEMSIZE, 117, 281, 289

Profile parameter (Cont.)
rdisp/atp_server, 303
rdisp/bufrefmode, 530
rdisp/bufreftime, 530
rdisp/enqname, 303
rdisp/max_wprun_time, 309
rdisp/mshost, 303
rdisp/PG_MAXFS, 280
rdisp/PG_SHM, 280
rdisp/ROLL_MAXFS, 274, 295
rdisp/ROLL_SHM, 116, 274
rdisp/vb_dispatching, 316
rdisp/vbstart, 314
rsdb/max_blocking_faktor, 508
rsdp/obj/buffersize, 401
rsdp/obj/max_objects, 401
rstr/file, 189
rstr/max_diskspace, 190
setting, 295
stat/dbprorec, 156
ztta/roll_area, 274, 278
ztta/roll_extension, 274, 278, 290, 294
ztta/roll_first, 275, 276, 278

Program
RSCOLL00, 152
SAP_COLLECTOR_FOR_PERFORMANCE, 152
saposcol, 79

Program buffer, 292, 522
Program counter, 80
Program error, 291
Program Global Area (PGA), 93
Program termination, 291
Promotion, 409
PTF package, 757
Python trace, 634

Q

Quadratic dependency, 216
Quality analysis, 216
Quantity structure, 234
Query
analytic, 673
parallelization, 625

Queue server, 622, 623
configure, 637, 645
Quick Sizer, 232, 239, 298

R

R/3, 794
Radio-frequency identification (RFID), 663
RAID, 794
Ranking, 620
RBO, 475
rdisp/atp_server, 401
Read sequential vs. record-by-record, 479
Read access, 398
logical, 92
Read random hit ratio, 761
READ TABLE, 214
READ TABLE ... WITH KEY ..., 214
Read/write (I/O) problems, 86, 102, 750
Record-by-record read, 479
Recovery, 48, 53, 259
Recursive call, 95
Redo log file, 164
REFRESH statement, 213
Region
generic, 524
Relational database, 40
Relational database management system (RDBMS), 794
RemoteCube, 597
Remote function call (RFC), 37, 188, 231, 339, 794
ABAP coding, 322
asynchronous, 322, 778
asynchronous (aRFC), 310, 335
background, 322
background RFC (bgRFC), 338
call, 188
client profile, 333
connection, 778
cycle, 323
destinations monitoring, 330
fundamentals, 320

Remote function call (RFC) (Cont.)
queued (qRFC), 322
statistics, 333
synchronous, 322
time, 326
trace, 187, 188, 198
transactional, 797
transactional (tRFC), 322, 336
Rendering time, 365
Reopen operation, 192
Reorganization, 716
Replication, 642, 684
triggering, 643
Report, 561, 562
precalculation, 601
Report RSSNROA1, 394
Request, 537, 545
Required field, 512
Reset or check the number range buffer, 778
Resource management, 259
Resource monitor, 736
Resource requirements, 231, 238, 802
Response time, 54, 55, 145, 156, 183, 305, 729
Response time distribution, 176
RFID, 794
Roll buffer, 288
Roll-in, 154, 273, 788
Roll memory, 294
Roll-out, 154, 273, 788
Rollup, 564
Roll wait time, 155, 187, 326, 349
Root recognition, 620
Roundtrip, 349, 440
Row-based data storage, 665
Row cache, 95
Row-ID, 471
Row store, 683, 697
Rule-based optimizer (RBO), 478
Run-length coding, 671
Runtime, 430
analysis, 146, 188, 201, 453
constant, 216
logarithmically increasing, 216

S

SAP Advanced Planning & Optimization (SAP APO), 35
SAP application instance, 76
SAP Application Performance Standard (SAPS), 243, 794
SAP Basis, 35
SAP BEx web analyzer, 359
SAP buffering, 113, 155, 288, 298, 499, 521, 524
access, 188
activate, 531
full, 524
generic, 524
SAP buffer trace, 189
SAP Business ByDesign, 662
SAP Business Connector (SAP BC), 787
SAP BusinessObjects with SAP HANA, 565, 661, 788
SAP BusinessObjects Business Intelligence platform, 565
SAP BusinessObjects Dashboards, 566
SAP BusinessObjects Explorer, 566, 653
SAP BusinessObjects Voyager, 566
SAP BusinessObjects Web Intelligence, 566
SAP Business Suite, 34
introduction, 256
SAP HANA, 661
SAPCCMSR, 218
SAP client plug-in, 221, 226, 362, 364
SAP component, 34
SAPconnect, 141
SAP Crystal Reports, 566
SAP cursor cache, 194
SAP Customer Relationship Management (SAP CRM), 35, 791
SAP EarlyWatch Alert, 57, 68, 70, 247, 295
SAP Easy Access menu, 355
SAP enqueue, 382, 383, 389, 390, 398
monitor, 403
SAP Enterprise Resource Planning (ERP), 35
SAP extended memory, 115, 272, 274, 276, 279, 288, 289, 298, 372

SAP GoingLive Check, 231, 235, 247
 SAP GoingLive Functional Upgrade Check, 248
 SAP GoingLive Migration Check, 248
 SAP GUI, 347, 358, 794
 controls, 347, 348
 end-to-end runtime analysis, 222
 for HTML, 37, 358
 for Java environment, 37, 38, 358
 for Windows, 37, 38, 358
 transaction, 222
 SAP HANA, 500, 659, 794
 administration, 719
 cloud application, 662
 compression, 668
 database platform, 661
 indexing, 676
 main memory area, 690
 scalability, 500
 scaling, 46
 sizing, 685
 SAP HANA Studio, 687
 SAP heap memory, 115, 272, 274, 294, 298, 342
 SAP host agent, 755
 SAP Implementation Guide (IMG), 598, 779, 791
 SAP instance, 76, 149, 257, 288, 791
 maintenance, 777
 overview, 778
 SAP Internet Transaction Server (ITS), 39, 261, 271, 310, 356, 370, 791
 external, 371
 integrated, 279, 370, 779
 monitor, 65
 performance analysis, 373
 status monitor, 372
 SAP J2EE Engine, 35
 SAP Java Virtual Machine (JVM), 408
 SAP Java Virtual Machine Profiler, 419, 447, 457
 SAP kernel, 295
 SAP liveCache, 36, 147, 262, 728, 792
 SAP logical unit of work (SAP LUW), 383
 SAP Management Console, 417, 456
 SAP MaxDB, 470, 732
 command monitor, 735
 execution plan, 738
 resource monitor, 736
 SAP memory area, 287
 SAP memory configuration, 112, 126
 monitor, 64, 112
 SAP memory management, 168
 SAP NetWeaver 7.40, 35, 528
 SAP NetWeaver 7.30, 31
 SAP NetWeaver Administrator, 64, 66
 SQL trace, 450
 SAP NetWeaver Application Server, 38, 356, 265
 SAP NetWeaver Application Server (AS) ABAP, 76, 265
 SAP NetWeaver Application Server (AS) Java, 35, 76, 265, 407
 server node, 421
 SQL trace, 450
 SAP NetWeaver BW, 36, 359, 553, 561, 794
 administration tools, 578
 data selection, 594
 frontend, 565
 indexing, 588
 monitor, 777
 optimization, 586
 performance optimization, 574
 SAP HANA, 661
 statistics, 581, 654
 SAP NetWeaver BW Accelerator (BWA), 575, 578, 660, 795, 619, 624
 creating indexes, 652
 delta index, 655
 indexing InfoCubes, 650
 monitor, 777
 sizing, 626
 Workbench, 578
 SAP NetWeaver Cloud, 662
 SAP NetWeaver Enterprise Search, 36, 618, 794
 SAP NetWeaver Portal, 36, 38
 SAP NetWeaver Process Integration, 36, 787
 Saposcol, 79

SAP paging, 271
 memory, 279, 298, 555, 793
 SAP parameter
 change, 119
 SAP performance menu, 779
 SAP performance trace, 183
 SAP Product Lifecycle Management (SAP PLM), 35
 SAP R/3, 22
 SAP repository browser, 778
 SAP roll area
 local, 273
 roll buffer, 273
 roll file, 273
 roll memory, 115, 272, 273, 298
 shared, 273
 SAProuter, 794
 SAP Service, 229, 303
 SAP Service Marketplace, 298
 Service, 70
 SAP Solution Landscape, 35
 SAP Solution Manager, 71, 139, 148, 173, 206, 207, 220, 223, 226, 365, 451, 728
 analysis, 69
 monitoring, 66
 performance optimization, 69
 service-level management, 57
 tracing, 69
 workload analysis, 69
 SAP Standard Application Benchmark, 229, 242
 SAP Supplier Relationship Management (SAP SRM), 35
 SAP Supply Chain Management (SAP SCM), 35
 SAP Sybase, 796
 SAP Sybase ASE, 740
 database process, 742
 data buffer, 741
 engine, 742
 SAP system, 35, 76
 SAP System Identifier (SID), 570, 795
 table, 570
 SAP system service, 794
 SAP system trace, 779
 SAP transaction, 383
 SAP Web Dispatcher, 40, 229, 262, 264, 310
 SAP work process, 41, 82, 104, 121, 289, 298, 733
 overview, 121
 Savepoint, 92
 Scalability, 238, 716, 795
 horizontal, 46, 795
 program, 215
 vertical, 46, 795
 SD benchmark, 242
 Search Engine Service (SES), 617
 Secondary index, 469, 472, 480, 482
 Select
 identical, 195
 SELECT * clause, 505
 Selection screen, 511
 Selectivity, 476, 478, 488
 SELECT, nested, 504
 Semaphore, 125
 Sensor data, 664
 Sequential read, 192, 479, 730, 738, 745, 751, 757, 758, 762, 764, 768, 774
 Serialization, 125
 Server, 77, 795
 Server consolidation, 256, 259
 Server destination statistics records, 333
 Server profile, 167
 Service, 302
 maintenance, 202
 Service-level agreement, 51
 Service-level management (SLM), 51, 68, 71, 795
 Service-level reporting, 51, 53, 56, 71
 Service maintenance, 372, 376
 Session, 341
 external, 272
 internal, 272
 Session monitor, 96
 Shadow process, 96, 98
 Shared cursor cache, 95, 99
 Shared memory, 270, 292, 341, 370, 398, 553, 746, 795

Shared objects, 551
 area configuration, 554
 area monitor, 554
 buffering, 522, 553
 Shared pool, 93, 95
 Shared SQL area, 99, 516
 Shared SQL cache, 99
 Simulation application, 562
 Single-core processor, 80
 Single-level storage, 282
 Single point of failure (SPOF), 304
 Single-record buffer, 521, 523
 Single-record statistics, 65, 146, 183,
 225, 325, 342, 350, 375, 580, 779
 central, 175, 218
 Single-transaction analysis, 207
 Sizing, 229, 685
 initial, 232, 236
 throughput-based, 233, 234
 T-shirt sizing, 232
 user-based, 233
 Sizing Plausibility Check, 235
 Sizing process, 231
 Skeleton, 761
 Slave index server, 636, 642
 Slave service, 716
 Slice & dice, 564
 Snapshot, 642
 Social network, 663
 Socket, 439
 Socket statistics, 439
 Solution monitoring, 47, 67
 Sorted table, 214
 Sparse coding, 671
 Spool service, 262, 302, 304
 SQL (Structured Query Language), 795
 native SQL, 494
 Open SQL, 494, 556
 SQL code, 504
 SQL editor, 703
 SQL monitor
 HANA migration, 702
 SQL Plan Cache, 701
 SQL programming
 efficient, 496
 golden rules, 496, 668, 680, 721
 SQL Query Engine, 755
 SQLScript, 684
 SQL Server, 771
 primary index, 774
 SQL statement, 70, 95, 96, 98, 99, 188,
 461, 469, 488, 489, 495, 702, 730,
 734
 buffer, 741
 dynamic, 761
 expensive, 95, 101, 462, 749, 779
 nested, 505
 optimization, 461
 SQL statistics, 99, 100, 450, 463, 464,
 468, 495, 550, 701, 779
 Analysis, 102
 DB2, 748, 749
 DB2 for IBM i, 755
 IBM DB2 for z/OS, 762
 Informix, 743
 SAP MaxDB, 734
 SQL Server, 773
 SQL trace, 65, 102, 185, 188, 189, 190,
 225, 457, 463, 464, 467, 496, 501,
 550, 779
 Stack pointer, 80
 ST-A/PI, 207
 Star join execution plan, 588
 Star schema, 563, 670
 extended, 566
 Statement cache, 771, 773
 Statement string, 761
 Statistics
 derived, 428
 Statistics record, 173, 184, 795
 distributed, 173, 218
 Statistics server, 717
 Statistics single record, 536
 Stemming, 620
 Stock, 563
 Stop-the-world, 411
 Stored procedure, 660, 772, 796
 SUBMIT, 272
 Subrecord, 147, 156
 Suite Accelerator, 661
 Support package, 796
 Survivor, 411
 Swap, 114

Swap space, 80, 116, 270, 282, 283,
 285, 298, 796
 Synchronization analysis, 436
 Synchronization trace, 457
 System availability, 304
 System buffers, 521, 522
 System consolidation, 256, 260
 System Global Area (SGA), 93
 System landscape, 256, 796
 System log, 778
 System message
 sending, 778
 System monitoring, 49

T

 Table
 analysis, 546, 777
 ARFCSDATA, 336
 ARFCSSTATE, 336
 buffered, 540, 544
 buffered objects, 557
 condition table, 535
 created by SAP, 542, 546
 customer-developed, 542, 546
 *D010**, 516
 D010S, 516
 *D020**, 516
 DBDIFF, 482
 DBSTATC, 483
 DDLOG, 528, 549, 554
 DDNTF, 516, 545
 DDNTT, 516, 545
 hash table, 214
 internal, 210, 803
 KAPOL, 516, 539
 NRIV, 385, 391
 NRIV_LOKAL, 394
 reorganize, 485
 RESB, 62, 398, 485
 sorted, 214
 SWNCMONI, 152
 TCURR, 547
 update table, 315, 319
 VBBE, 398
 VBDATA, 315
 Table (Cont.)
 VBHDR, 315, 545
 VBMOD, 315
 Table access statistics, 107, 477, 483,
 536, 540, 545, 777
 Table buffer, 44, 186, 521, 557
 ABAP server, 535
 generic, 515
 generic (TABL), 521
 Java, 556
 partial (TABLP), 521
 Table call statistic, 536
 Table maintenance, 483, 710
 Table operation, 201
 Table pool
 ATAB, 550
 KAPOL, 550
 Table size, 545
 Tablespace, 89
 Task type, 150
 Technical analysis, 61
 Technical tuning, 61
 Temporary performance problem, 163
 Temporary sequential objects, 796
 TemSe, 796
 Teraspace, 282
 Text dictionary, 671
 Text search, 617
 Think time, 305
 Thread, 79, 80, 457, 698, 732, 742
 dump, 418
 hardware, 80
 monitor, 701
 software, 80
 Throughput, 159, 381
 Time-based sampling, 430
 Time dependency, 570, 575
 Time dimension, 572
 Time-flow hierarchy, 208
 Time profile, 176
 Trace frontend, 221
 Trace level, 222
 Trainings, 800
 Transaction, 383, 796
 AL11, 777
 AL12, 777

- Transaction (Cont.)
- BALE, 777
 - critical, 54
 - DB01, 91, 104, 386, 404
 - DB02, 91, 477, 777
 - DB05, 546, 777
 - DB12, 91, 777
 - DB13, 91, 777
 - DB13C, 91
 - DB20, 777
 - DB21, 483
 - DB24, 91
 - DB50, 732
 - DBACOCKPIT, 64, 124, 386, 481, 483, 687, 728, 732, 777, 789
 - DSWP, 178, 223, 454
 - DWDM, 348
 - LISTCUBE, 579, 777
 - Locking, 778
 - ME57, 511
 - OS06, 77
 - OS06N, 77
 - OS07, 77
 - OS07N, 77
 - OSS1, 777
 - response time, 183
 - RSA1, 578, 581, 591, 608, 652, 777
 - RSCUSTV14, 777
 - RSDDBWAMON, 627, 655, 777
 - RSDDV, 578, 627, 653, 777
 - RSMIGRANADB, 721, 777
 - RSODSO_SETTINGS, 777
 - RSRCACHE, 599
 - RSRT, 579, 580, 585, 597, 627, 653, 777
 - RSRV, 655, 777
 - RZ01, 777
 - RZ02, 777
 - RZ03, 777
 - RZ04, 777
 - RZ10, 777
 - RZ11, 360, 778
 - RZ12, 778
 - RZ20, 65, 135, 330, 343, 778
 - SAT, 201
 - SBGRFCCONF, 338
 - SBGRFCMON, 338
- Transaction (Cont.)
- SCII, 216
 - SE11, 330, 480, 482, 503, 778
 - SE12, 200, 778
 - SE14, 481, 778
 - SE15, 778
 - SE16, 582, 778
 - SE24, 216, 778
 - SE30, 65, 215
 - SE37, 216, 318
 - SE38, 216, 296, 386, 512, 778
 - SE80, 369, 515, 778
 - SEU, 778
 - SHMA, 554
 - SHMM, 554
 - SICF, 202, 372, 376
 - SITSMON, 779
 - SITSPMON, 65, 372
 - SM01, 778
 - SM02, 778
 - SM04, 324, 778
 - SM12, 391, 403, 778
 - SM13, 109, 126, 315, 778
 - SM21, 109, 126, 275, 778
 - SM24, 553
 - SM30, 483
 - SM36, 778
 - SM37, 611, 778
 - SM39, 778
 - SM49, 778
 - SM50, 64, 121, 211, 295, 303, 318, 324, 373, 386, 733, 778
 - SM51, 77, 121, 130, 778
 - SM56, 396, 778
 - SM58, 336, 778
 - SM59, 326, 327, 337, 778
 - SM63, 778
 - SM65, 778
 - SM66, 122, 128, 303, 318, 324, 373, 386, 733, 778
 - SM69, 779
 - SMGW, 324
 - SMICM, 65, 133, 360, 373, 779
 - SMLG, 127, 303, 309, 779
 - SNRO, 396
 - SPRO, 401, 779
 - SQLM, 702, 779

- Transaction (Cont.)
- ST01, 779
 - ST02, 64, 112, 249, 286, 290, 293, 295, 403, 543, 553, 779
 - ST03, 64, 147, 149, 175, 303, 325, 330, 342, 373, 579, 582, 779
 - ST03G, 64, 173, 175
 - ST03N, 149, 251
 - ST04, 91, 118, 779
 - ST05, 188, 352, 463, 779
 - ST06, 64, 77, 82, 86, 103, 119, 124, 249, 286, 353, 753, 779
 - ST06N, 77
 - ST07, 779
 - ST08, 779
 - ST09, 779
 - ST10, 779
 - ST11, 779
 - ST12, 207, 208
 - ST14, 779
 - ST22, 291, 779
 - STAD, 147, 175, 184, 325, 333, 342, 350, 375, 779
 - STAT, 536
 - STATTRACE, 173, 175, 219
 - STMS, 779
 - STUN, 63, 779
 - SWLT, 779
 - SXMB_MONI, 779
 - TREXADMIN, 125, 626, 630, 639, 643, 779
 - TU02, 779
 - VA01, 169
 - variant, 514
- Transaction code, 797
- Transaction data, 533
- Transaction profile, 169
- Transaction step, 151, 153, 272
- Transformation, 566
- Transmission Control Protocol/Internet Protocol (TCP-IP), 796
- Transparency, 672
- Transport, 797
- Transport domain, 797
- Transport Domain Controller (TDC), 796
- Transport management system (TMS), 779, 796
- Transport Organizer (TO), 796
- Tree, 590
- Tree Control, 348
- TREX, 36, 44, 262, 615, 660, 794, 797
- administration, 779
 - applications, 617
 - architecture, 620
 - BWA, 624
 - compression, 668
 - distributed installation, 637
 - fundamentals, 616
 - monitoring server utilization, 628
 - optimization, 636
 - performance analysis, 628
 - process of a query, 622
 - Python trace, 634
 - reorganization, 638
 - replication, 642
 - RFC server, 650
 - search functions, 619
 - server overview, 630, 631
 - sizing, 624
 - tools, 626, 681
 - workload overview file, 633, 634
- TREX index, 617
- tRFC table, 337
- Troubleshooting, 55
- T-shirt sizing, 232
- Tuning, 519
- application tuning, 58, 72
 - program optimization, 59
 - technical, 57
- Tuple reconstruction, 668
- ## U
- Unicode, 253, 271, 288
 - Unicode conversion, 253
 - Uniform Resource Locator (URL), 797
 - Unit dimension, 572
 - UNIX, 85, 307
 - standard implementation, 282
 - Update, 109, 313, 344
 - asynchronous, 314
 - deactivated, 126
 - dispatching, 316

Update (Cont.)
local, 318
priority decisions, 317
record, 778
request, 314
synchronous, 318
service, 262, 302
table, 314
time, 319
type, 316
V1 update, 316
V2 update, 316
V3 update, 316
work process, 314, 315

Upgrade, 253

User
active, 160
occasional, 161
power user, 161
transactional, 161

User call, 95

User connection, 37

User context, 154, 272, 339, 797

User exits, 237

User interface
graphical, 790

User kernel thread (UKT), 733

User list, 778

User mode
expert, 149
service engineer, 149

User monitor, 324

User profile, 161, 176

V

Value per unit, 563

Variant, 205

Verification, 61

Virtualization, 258

Virtual Machine Container (VMC), 261, 262, 301, 341, 797

Virtual memory, 270, 298

Virtual provider, 563

W

Web application, 358, 365

Web browser, 787

Web Dynpro, 797

Web Dynpro ABAP, 39, 347, 357, 367, 373
runtime analysis, 202

Web Dynpro Java, 347, 357

Web GUI, 358

Web reporting, 565

Web service, 37, 231
ABAP, 372
monitoring calls, 376

Web services, 357

Web template
precalculation, 602

Web transaction, 220

WHERE clause, 498

Wide Area Network (WAN), 45, 797

Wily Introscope, 803

Windows, 85

Workbench Organizer (WBO), 797

Work group, 308

Workload analysis, 25, 58, 145, 146, 152, 161, 453, 457
global, 175

Workload monitor, 64, 147, 149, 152, 168, 219, 251, 303, 325, 342, 373, 580, 779
BW system workload, 583
central, 174, 175
Java, 178

Workload overview, 176

Work process, 41, 125, 127, 262, 273, 305, 343, 797
ABAP trace, 201
increasing the number, 345
Java Virtual Machine, 418

Work process monitor, 318, 373

Work process overview, 64, 97, 121, 189, 211, 295, 303, 323, 386, 733, 778
system-wide, 128
systemwide, 779

Work process type, 151

World Wide Web (WWW), 798

WRKSYSSTS, 753

X

XS server, 683

Y

Young generation, 409, 413

Z

Zero Administration Memory
 Management, 116, 280, 284, 299



Thomas Schneider

SAP Performance Optimization Guide

837 Pages, 2013, \$79.95

ISBN 978-1-59229-874-7

 www.sap-press.com/3360

Dr. Thomas Schneider started his career at SAP AG in 1996. Among other things, he was in charge of the Center of Expertise for Performance in the Service & Support group and the responsible Support Alliance manager for key accounts. Since 2004 he has worked in the Research & Breakthrough Innovation group, where he first was responsible for IT Service & Application Management. Since 2009 he has focused on the Partner Development Infrastructure in the SAP Cloud organization as the responsible architect.

We hope you have enjoyed this reading sample. You may recommend or pass it on to others, but only in its entirety, including all pages. This reading sample and all its parts are protected by copyright law. All usage and exploitation rights are reserved by the author and the publisher.