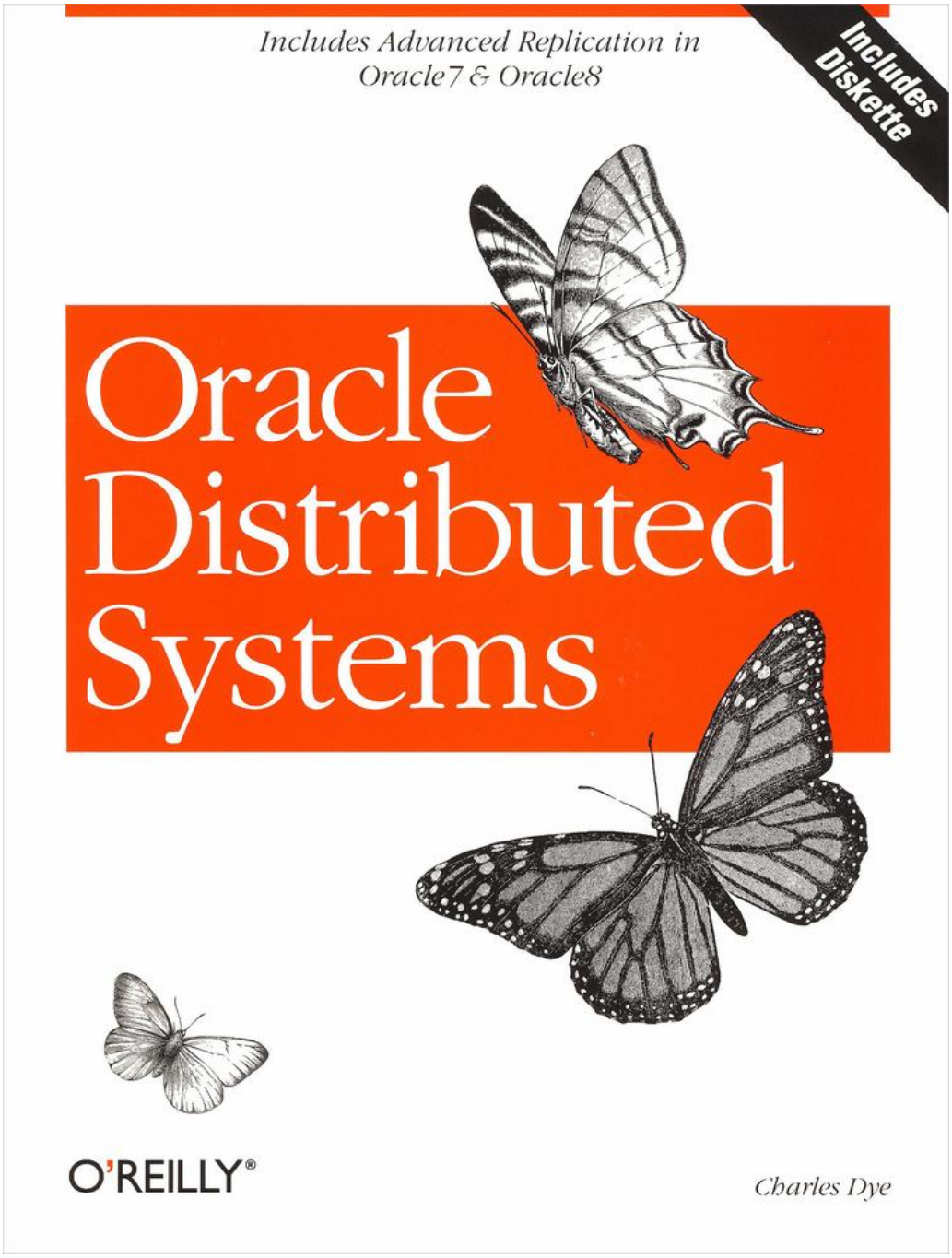

Oracle Distributed Systems

Charles Dye

Editor

Debby Russell

Chapter 1. Introduction to Distributed Systems

Any organization that uses the Oracle relational database management system (RDBMS) probably has multiple databases. There are a variety of reasons why you might use more than a single database in a distributed database system:

- Different databases may be associated with particular business functions, such as manufacturing or human resources.
- Databases may be aligned with geographic boundaries, such as a behemoth database at a headquarters site and smaller databases at regional offices.
- Two different databases may be required to access the same data in different ways, such as an order entry database whose transactions are aggregated and analyzed in a data warehouse.
- A busy Internet commerce site may create multiple copies of the same database to attain horizontal scalability.
- A copy of a production database may be created to serve as a development test bed.

Sometimes the relationship between multiple databases is part of a well-planned architecture, in which distributed databases are designed and implemented as such from the beginning. In other cases, though, the relationship is unforeseen; it is quite common for distributed databases to evolve as businesses expand, requirements grow, and applications spawn. But common to all cases is the need to copy or reference data in one or more remote databases.

A distributed database system will meet one or more of the following objectives:

Availability
Data must be available at the local site even when a remote site is unreachable.

Survivability
The failure of any single database instance must not impact the ongoing business.

Data collection
Regional data such as sales receipts is consolidated and aggregated at a single site.

Data extraction
A data warehouse extracts transaction records from an online transaction processing (OLTP) system.

Decentralized data
Data may be updated in several databases.

Maintenance
There must be support for activities such as load testing with data from production in a benchmarking database.

Oracle Corporation introduced interdatabase connectivity with SQL*Net in Oracle Version 5 and simplified its usage considerably with the database links feature in Oracle Version 6, opening up a world of distributed possibilities. Oracle now supplies a variety of techniques that you can use to establish interdatabase connectivity and data sharing. Each technique has its advantages and disadvantages, but in many cases the best solution is not immediately obvious.

Before delving into Oracle's offerings in the distributed database systems area, I'll clarify some terminology and concepts.

Terminology and Concepts

I have found thatthere is a great deal of confusion surrounding the various products and terminology from Oracle. I think it's worthwhile to clarify some of these terms up front so you'll get the most benefit from this book.

Database/ database instance
These terms are often used interchangeably, but they are not the same thing. In Oracle parlance, a *database* is the set of physical files containing data. These files comprise tablespaces, redo logs, and control files. A *database instance* (or simply *instance*) is the set of processes and memory structures that manipulate a database.

A database may be accessed by one or more database instances, and a database instance may access exactly one database.

Oracle parallel server
Oracle parallel server(OPS) is a technology that allows two or more database instances, generally on different machines, to open and manipulate one database, as shown in Figure 1.1. In other words, the physical data files (and therefore data) in a database can be seen, inserted, updated, and deleted by users logging on to two or more different instances; the instances run on different machines but access the same physical database.

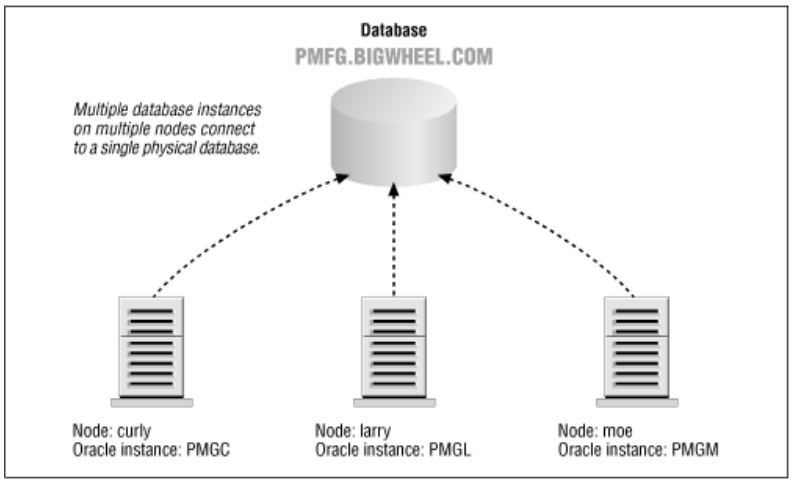

Figure 1-1. Parallel server architecture

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.