Cloudscape and the Internet

Table Of Contents

Introduction

This is an introduction to a series of white papers on using Cloudscape in a web environment titled *i*.Cloudscape. The series will include application examples (including source code), architecture discussions and information on performance tests. Documents will continue to be added over time to help educate and inform about how to use Cloudscape in your web based application.

Java and the Internet go hand-in-hand. Java application/web servers are all the rage and for good reason. Java provides a quick-deployment, cross-platform development mechanism. OK, we all know how cool Java is, the real question is how does Cloudscape fit into the picture?

A Java Developer Story

Let us look at a fictitious, but not-so-far-from-home, example about Joe the java programmer.

Joe’s application, like most, starts with small data needs. Joe is tasked with creating a web page application to manage a list of equipment in his lab. Joe creates a couple of JavaServer Pages (JSP) to update and display the current lab equipment inventory. Joe thinks, there are only going to be 10-20 pieces of equipment to manage, so he decides to save the data in a flat file on the server. He creates some Java beans to parse the flat file, adds them to his JSP and his application is complete - or so he thinks. His manager is pleased with the result.

In fact, Joe’s manager is so pleased that he shows the application to all the other managers and they decide to use it to manage equipment in all the labs at their location totaling 200-300 pieces of equipment.

Joe creates another flat file to manage the new lab information. Then adds code to make sure only one user at a time can update the file. He then adds a third file to manage the information about each lab.

Then his manager asks for reports on the equipment. “How many X Brand Servers do we have?” “How many network ports are there?” Joe adds some routines to parse the flat file and produce the results requested by his manager. A week later, he shows the reports to his manager. His manager is so pleased, he takes the application to the VP of Lab Management.

The VP is so pleased, he says, “We shall use this for all the lab equipment company wide”. This includes 30,000 – 35,000 pieces of equipment, 200 locations worldwide and 200 lab managers. Joe puts an ad in the paper for Java developers.

This is where Cloudscape comes in. Let us look at how the story changes if Joe had decided to use Cloudscape

from the start to develop his web application.

Joe creates his JSP to enter and display the equipment information. Joe thinks, "I need only to save information on 20-30 pieces of equipment, but that may change," so he decides to use a Cloudscape database. He completes his application by creating tables for equipment (lab_equip), lab information (lab_info) and lab administrator information (lab_admin) in Cloudscape. Joe uses standard JDBC calls in his JSP to access the data, the same way he does with his other databases. His manager is pleased with the results.

In fact, Joe's manager is so pleased, that he shows the application to all the other managers and they decide to use it for all the labs at their location which includes 200-300 pieces of equipment. Joe adds the new lab administrators to the **lab_admin** table and the lab information to the **lab_info** table. Using Joe's JSP interface the other lab managers then add their own equipment to the **lab_equip** table. Joe does not have to change any of his code.

Then his manager asks for reports on the equipment. "How many X Brand Servers do we have?" "How many network ports are there?" Joe uses a web-based query tool and standard SQL to create the reports. He delivers the reports to his manager that same afternoon. Again Joe does not have to change his JSP or add additional code. His manager is so pleased he takes the application to the VP of Lab Management.

The VP is so pleased, he says "We shall use this for all the lab equipment company wide" which includes 30,000 – 35,000 pieces of equipment, 200 locations worldwide and 200 lab managers. Joe adds a DIMMS to his server, gets a list of all of the lab managers and locations and loads them into the database. Each lab manager enters their equipment data and Joe gets a promotion.

Then the President (of the United States) hears about Joe's application and decrees that we shall track all US government equipment information in this database. So Joe purchases a large server, installs iFoundation.2000, changes the connection URL in his application to point to the new iFoundation.2000 server and imports the information from the Cloudscape Database. The application is up and running.

OK, the second example got a little out of hand, but you get the point. By using Cloudscape to manage the data, Joe created a scalable solution from the start with less work. He wrote his application using standard JDBC which allowed it to be easily moved to a larger iFoundation.2000 database with negligible code changes.

Why Cloudscape?

Java Rules!

Cloudscape is a relational database engine written entirely in Java. Being 100% Java allows easy integration with JSP, servlets, Java web/application servers and Java programmers. Java's cross-platform support allowed Joe to move his application to any available server to meet his processing needs. With Cloudscape's small footprint (less than 2MB), Joe was able to develop his application on his laptop and copied the code to his server when he was ready for deployment.

Near Zero-Administration

In the beginning, Joe was looking for a personal tool to manage his lab. In that type of environment, he wanted to set it up and forget about it. Cloudscape offers near zero-administration so Joe gets a full featured relational database without requiring that he become a database administrator.

Scalability and Performance

Cloudscape is a small-footprint database, but don't be fooled, it is full-featured and scalable. In an embedded web application test using Enhydra 3.0.1 on SunOS 5.7 (demonstrated at Java One in June 2000), we were able to run more than 300 concurrent users against a 6GB Cloudscape database maintaining an average 1.2-second response time. See "Bubba's CloudBooks: A Cloudscape ePerformance Test" for more details.

The Embedded Model: What does it mean to "Embed" Cloudscape in your Web server?

Most web applications communicate with the database via TCP/IP or some other network protocol. Cloudscape can certainly be used in the traditional client-server mode (Figure 1).

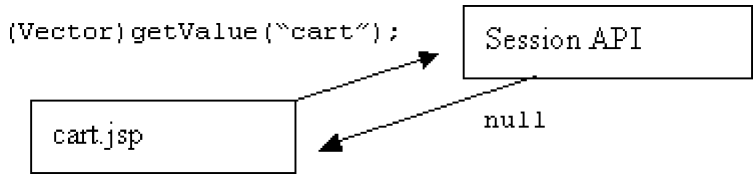

Figure 1

However, because Cloudscape is a 100% Java, embeddable database the traditional rules need not apply. Cloudscape does not need to run in a separate instance of the Java Virtual Machine (JVM) or on a different server. When we say that Cloudscape is embedded, we mean that the web/application server, your application code, and Cloudscape are all running in the same instance of the JVM (Figure 2).

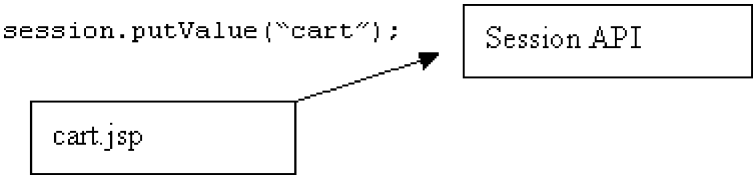

Figure 2

Running in embedded mode, your application still communicates to Cloudscape using standard JDBC. Cloudscape ships with an embedded JDBC driver, *COM.cloudscape.core.JDBCDriver*. This driver is provided to boot the database when Cloudscape is embedded in your application.

Embedded Model: Performance Benefits

When running Cloudscape embedded in your web/application server you do not have the communication overhead normally associated with communicating over other types of connections. When running embedded, each JDBC request is a direct Java method call instead of a network request. Using the embedded driver you can open and close hundreds of connections per second with little system impact. This makes connections faster and frees up the communication resources to handle web client connections.

Summary

Cloudscape was designed from the start to be an easy to use full-featured, scalable, 100% Java, object-relational database management system--without the need to hire a database administrator (DBA). Cloudscape is well suited to providing for the database needs of small to mid-size web applications.

Resource Links

Reference

Bubbas CloudBooks: A Cloudscape ePerformance Test

i-Cloudscape-Bubbas.html

Enhydra:

http://www.enhydra.org

Author:

Scott Fadden

Technical Marketing

Informix Software, Inc.

scottfa@informix.com