CSC 309 Pointbase Tutorial

Nan Niu

http://www.cdf.toronto.edu/~radu/csc309/guide/pointbase.html

July 18, 2007

Overview

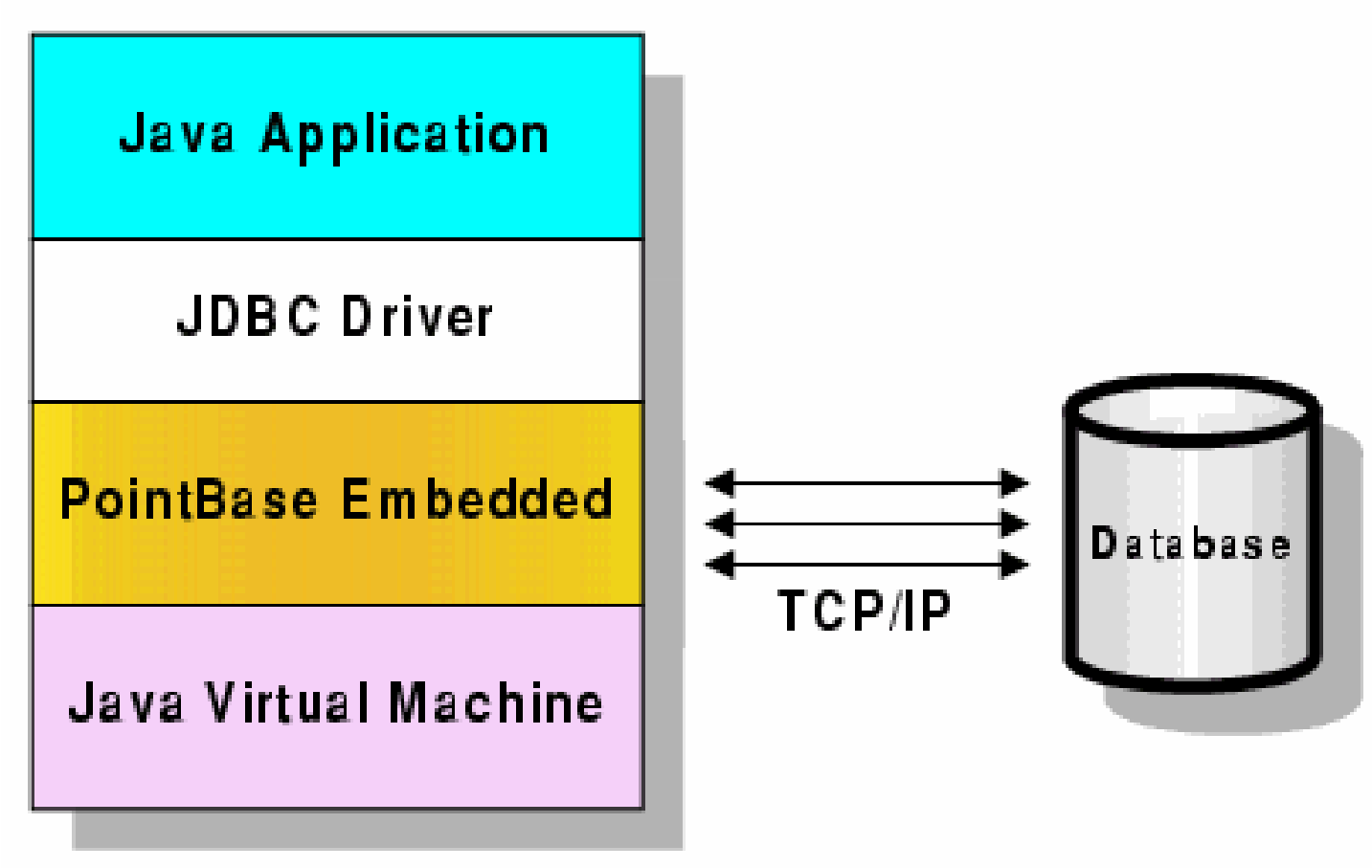
⇨Brief introduction

⇨Simple examples

⇨Servlet + Pointbase

Brief Introduction

- One of the successful Java-based RDBMS's
- Pointbase fully supports JDBC 1.x API, and adheres to ANSI and ISO standards



Commander and Console

- Create a pointbase.ini file for configuring (e.g., storing) your database
  - database.home = /h/u1/delara/tomcat/webapps/jdbc
  - Sample ini file available from the course website

- Create a "sample" database by using Commander or Console
  - select * from systables;
  - create table student (id integer, name character (40), birthday date);
  - insert into student values (1, 'John', date '1980-1-1');

Autocommit

- By default, Commander runs with the "autocommit" option off to prevent Pointbase from writing any modifications made to the database to disk (unless you explicitly "commit")

- To fix this problem, use "set autocommit on;"

- By default, Console runs with the "autocommit" option on

Simple Examples

- DBTest – Demonstrates a simple way to check that the DB is working fine. Program prints the names of all tables in the DB.

- DBTable – Demonstrates how to add/delete tables and how to insert/query data.

- DBInteractive – Interactive program that demonstrates how to add/delete tables and how to insert/query data.

Servlet + Pointbase

- Untar jdbc.tar from the course website
- Edit pointbase.ini by using a fully qualified path
  - database.home = /h/u1/delara/tomcat/webapps/jdbc
- Copy pointbase.ini from $CATALINA_HOME/webapps/jdbc/WEB_INF/classes to $CATALINA_HOME/bin
- Complie the servlet SQLGateway
- Start tomcat
- Run the servlet SQLGateway in the browser to query the databse

Notes

- Lock file – generated automatically; delete before running next DB application
- Quotation marks (e.g., ", \", ', etc.) when handling character/string type fields
- Remember to "commit;"
- Remember to close SQL statement and DB connection at the end

- Be familiar with Commander or Console
- Play with the demos (at least 4 java files available), and start from there

Questions?

Backup – Connection Pooling

- Maintain a pool of open connections that time themselves out
- The SharedPoolSource allocates a new Connection object
- Client returns Connection to the pool when done with it
- Client may block awaiting a freed connection if some maximum upper limit of open connections is reached
- Init method of main webapp servlet:
  - //Create a data source
- Each servlet doGet() or doPost() calls:
  - SharedPoolSource dbcp = getServletContext().getAttribute("dbConPool");
  - Connection con = dbcp.getConnection();
  - Statement s = con.createStatement();
  - con.close();