

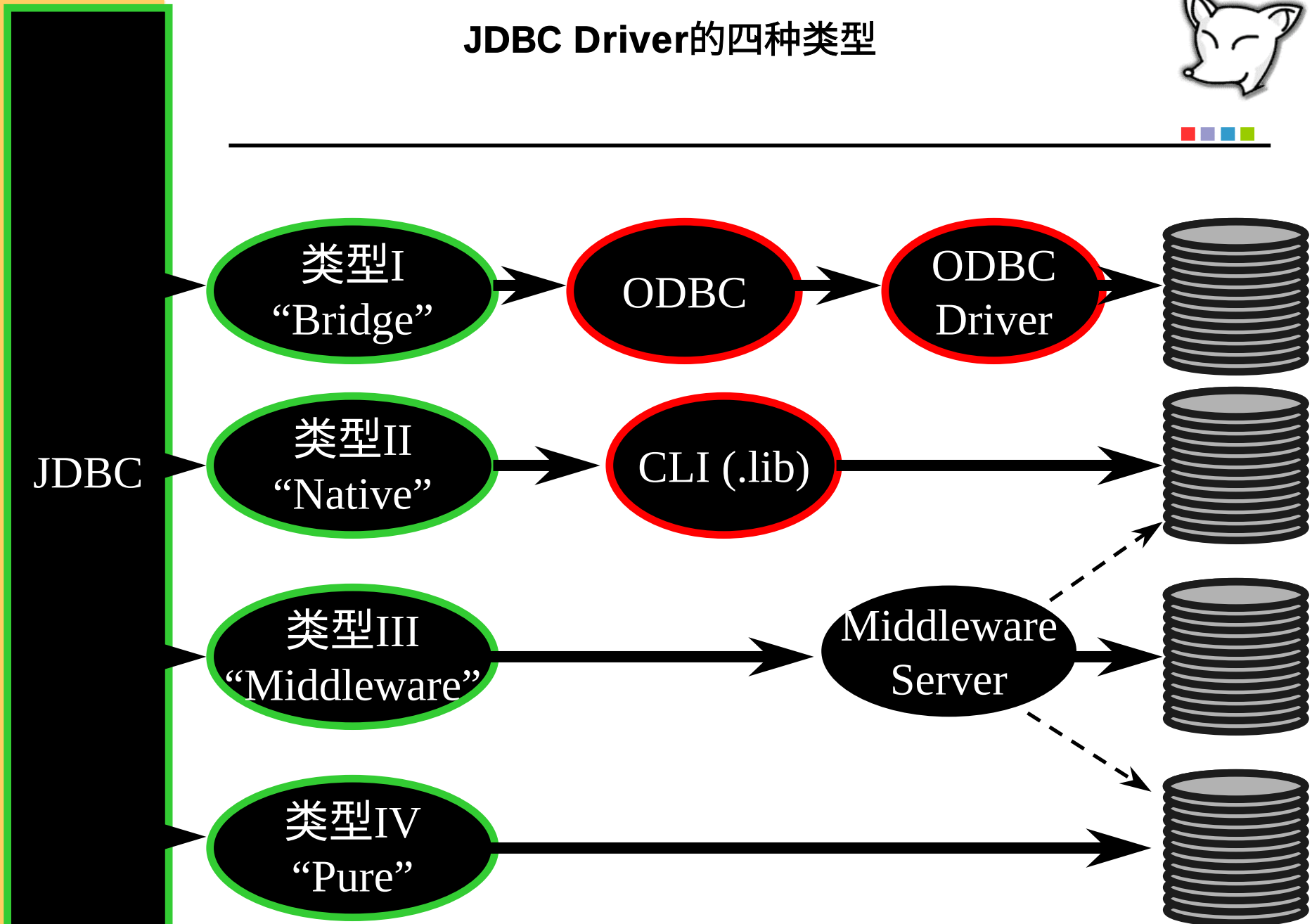

JDBC

Allen Long
Email: allen@huihoo.com
http://www.huihoo.com
2004-04

内容安排

- JDBC概述
- JDBC APIs
- 其他数据库技术

JDBC Driver的四种类型



类型一

- Use bridging technology
- Requires installation/configuration on client machines
- Not good for Web
- e.g. ODBC Bridge

类型二

- Native API drivers
- Requires installation/configuration on client machines
- Used to leverage existing CLI libraries
- Usually not thread-safe
- Mostly obsolete now
- e.g. Intersolv Oracle Driver, WebLogic drivers

类型三

- Calls middleware server, usually on database host
- Very flexible -- allows access to multiple databases using one driver
- Only need to download one driver
- But it’s another server application to install and maintain
- e.g. Symantec DBAnywhere

类型四




- 100% Pure Java -- the Holy Grail
- Use Java networking libraries to talk directly to database engines
- Only disadvantage: need to download a new driver for each database engine
- e.g. Oracle, mSQL

JDBC API

java.sql

JDBC is implemented via classes in the java.sql package

DriverManager




- DriverManager tries all the drivers
- Uses the first one that works
- When a driver class is first loaded, it registers itself with the DriverManager
- Therefore, to register a driver, just load it!

注册一个Driver

- statically load driver

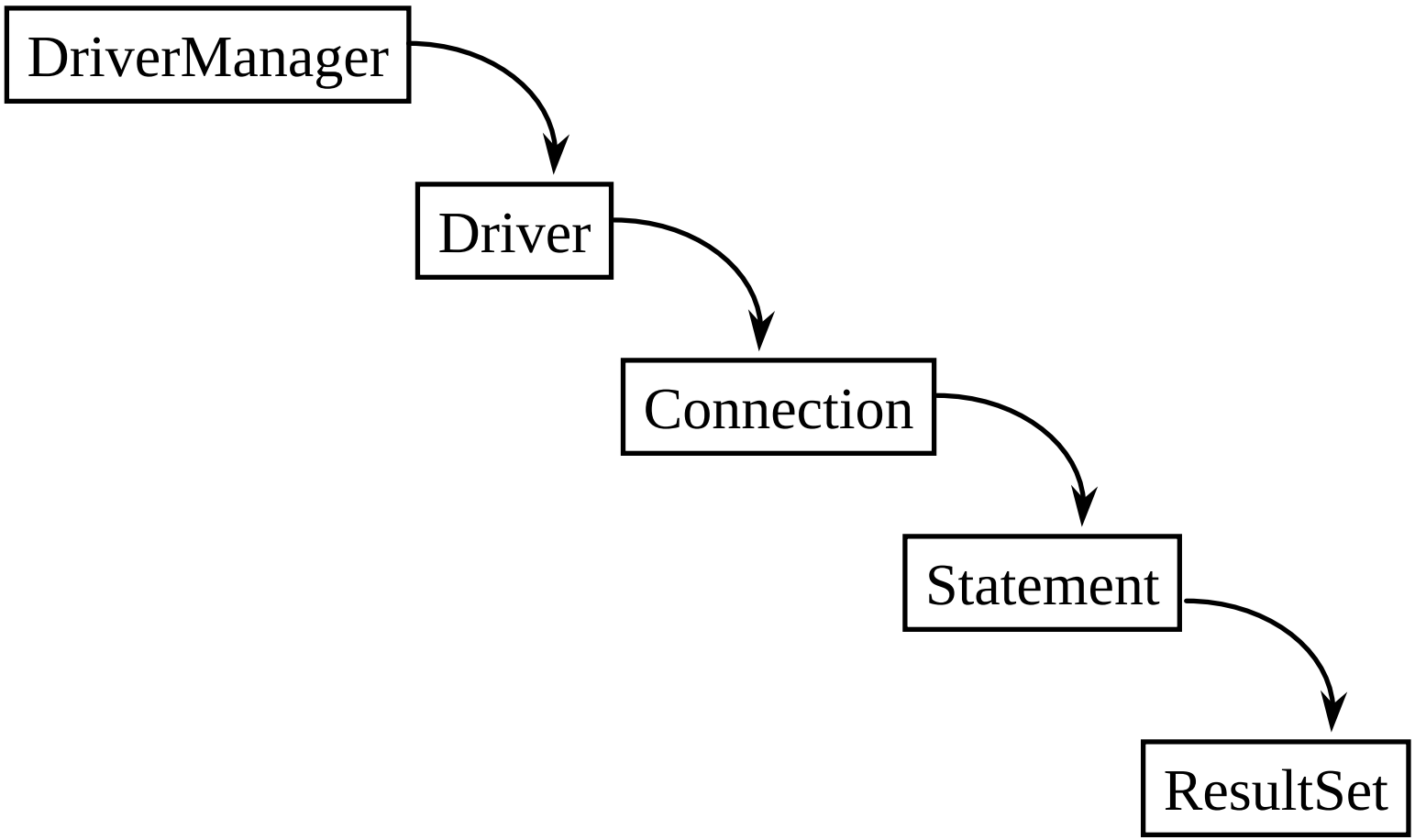
```
Class.forName("foo.bar.MyDriver");
Connection c = DriverManager.getConnection(...);
```

- or use the `jdbc.drivers` system property

JDBC Object Classes

- DriverManager
  - Loads, chooses drivers
- Driver
  - connects to actual database
- Connection
  - a series of SQL statements to and from the DB
- Statement
  - a single SQL statement
- ResultSet
  - the records returned from a Statement

JDBC类的使用



JDBC URLs

`jdbc:subprotocol:source`

- each driver has its own subprotocol
- each subprotocol has its own syntax for the source

`jdbc:odbc:`*`DataSource`*

- e.g. `jdbc:odbc:Northwind`

`jdbc:msql:`*`//host[:port]/database`*

- e.g. `jdbc:msql://foo.nowhere.com:4333/accounting`

DriverManager

```
Connection getConnection
    (String url, String user, String
    password)
```

- Connects to given JDBC URL with given user name and password
- Throws java.sql.SQLException
- returns a Connection object

Connection




- A Connection represents a session with a specific database.
- Within the context of a Connection, SQL statements are executed and results are returned.
- Can have multiple connections to a database
  - NB: Some drivers don't support serialized connections
  - Fortunately, most do (now)
- Also provides "metadata" -- information about the database, tables, and fields
- Also methods to deal with transactions

获得一个Connection

```
String url   = "jdbc:odbc:Northwind";
try {
  Class.forName ("sun.jdbc.odbc.JdbcOdbcDriver");
   Connection con =
   DriverManager.getConnection(url);
}
catch (ClassNotFoundException e)
   { e.printStackTrace(); }
catch (SQLException e)
   { e.printStackTrace(); }
```

连接方法

`Statement createStatement()`

- returns a new Statement object

`PreparedStatement prepareStatement(String sql)`

- returns a new PreparedStatement object

`CallableStatement prepareCall(String sql)`

- returns a new CallableStatement object

- Why all these different kinds of statements? Optimization.

Statement

- A Statement object is used for executing a static SQL statement and obtaining the results produced by it.

Statement Methods

`ResultSet executeQuery(String)`

- Execute a SQL statement that returns a single ResultSet.

`int executeUpdate(String)`

- Execute a SQL INSERT, UPDATE or DELETE statement. Returns the number of rows changed.

`boolean execute(String)`

- Execute a SQL statement that may return multiple results.

- Why all these different kinds of queries? Optimization.

ResultSet

- A ResultSet provides access to a table of data generated by executing a Statement.
- Only one ResultSet per Statement can be open at once.
- The table rows are retrieved in sequence.
- A ResultSet maintains a cursor pointing to its current row of data.
- The 'next' method moves the cursor to the next row.
  - you can’t rewind

事务管理




- Transactions are not explicitly opened and closed
- Instead, the connection has a state called *AutoCommit* mode
- if *AutoCommit* is true, then every statement is automatically committed
- default case: true

setAutoCommit

Connection.setAutoCommit(boolean)

- if *AutoCommit* is false, then every statement is added to an ongoing transaction
- you must explicitly commit or rollback the transaction using Connection.commit() and Connection.rollback()

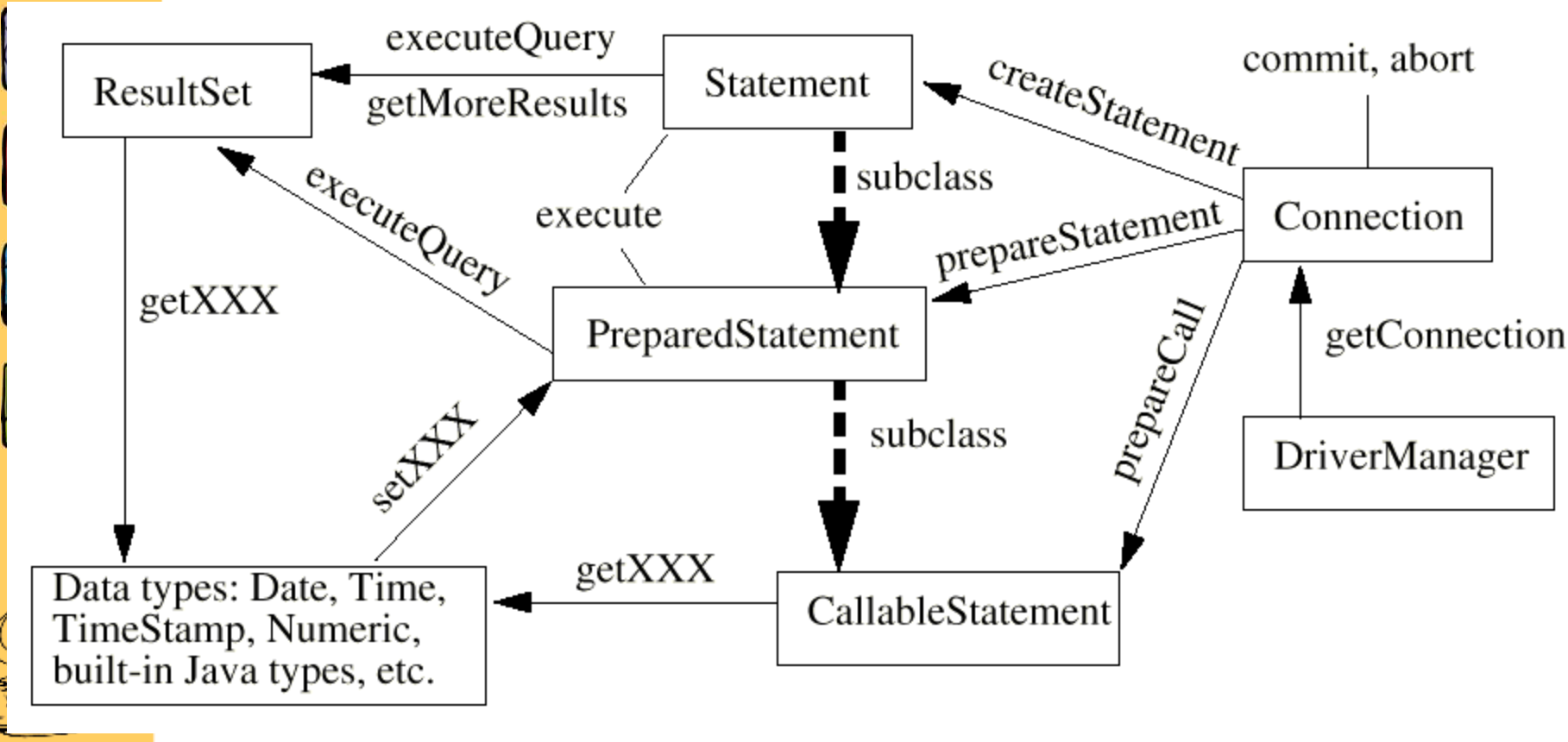
连接管理

- Hint: for a large threaded database server, create a Connection Manager object
- It is responsible for maintaining a certain number of open connections to the database
- When your applications need a connection, they ask for one from the CM's pool
- Why? Because opening and closing connections takes a long time
- Warning: the CM should always setAutoCommit(false) when a connection is returned

优化Statements

- Prepared Statements
  - SQL calls you make again and again
  - allows driver to optimize (compile) queries
  - created with Connection.prepareStatement()
- Stored Procedures
  - written in DB-specific language
  - stored inside database
  - accesed with Connection.prepareCall()

JDBC类图


executeQuery
ResultSet
getMoreResults
Statement
createStatement
commit, abort
executeQuery
execute
subclass
Connection
prepareStatement
getXXX
PreparedStatement
getConnection
setXXX
subclass
prepareCall
DriverManager
Data types: Date, Time,
TimeStamp, Numeric,
built-in Java types, etc.
getXXX
CallableStatement

Metadata

- Connection:
  - DatabaseMetaData getMetaData()
- ResultSet:
  - ResultSetMetaData getMetaData()

JDBC 2.0

- Scrollable result set
- Batch updates
- Advanced data types
  - Blobs, objects, structured types
- Rowsets
  - Persistent JavaBeans
- JNDI
- Connection Pooling
- Distributed transactions via JTS

数据库技术

- Hierarchical
  - obsolete (in a manner of speaking)
  - any specialized file format can be called a hierarchical DB
- Relational (aka SQL) (RDBMS)
  - row, column
  - most popular
- Object-relational DB (ORDBMS)
  - add inheritance, blobs to RDB
  - NOT object-oriented -- “object” is mostly a marketing term
- Object-oriented DB (OODB)
  - data stored as objects
  - high-performance for OO data models

内容回顾

- JDBC概述
- JDBC APIs
- 其他数据库技术

参考资料

- http://java.sun.com/products/jdbc/

sun公司的jdbc站点

- http://www.huihoo.com
  国内一个关于中间件的专业站点

结束

谢谢大家！

Allen@huihoo.com
http://www.huihoo.com