



GETTING AQUAINTED WITH APACHE DERBY

Kristian Waagan

Developer, Database Group

Sun Microsystems

Kristian.Waagan@Sun.COM



Agenda

- Introduction of the ASF and myself
- Apache Derby
 - > Introduction
 - > History
 - > Community
- Product Overview
- New Features
- Questions and Answers

The Apache Software Foundation

“The Apache Software Foundation provides support for the Apache community of open-source software projects.

The Apache projects are characterized by a collaborative, consensus based development process, an open and pragmatic software license, and a desire to create high quality software that leads the way in its field.”

About Me

- Software Engineer
- Based in Norway
- Working on Apache Derby for 4 years
 - > Daily work performed in the Apache community
 - > Recent areas of focus: in-memory DB, performance and LOBs
- Derby committer, PMC member

LOB: large object (Blob/Clob)
PMC: Project Management Committee

Apache Derby – Introduction

Apache Derby

- A well-featured relational database, 100% Java
 - > Runs on Java ME CDC 1.1, Java SE 1.4.2 and up
- Small footprint, easy to use
- Aims for high level of standards compliance
- Transactional, multi-user engine
- Platform independent on-disk format
- Originally written to be an embeddable database
 - > Invisible to the end-user
 - > Zero maintenance

Apache Derby History

- 1996: Cloudscape founded
- 1999: Cloudscape acquired by Informix
- 2001: Database part of Informix acquired by IBM
- 2004: IBM donated Cloudscape to Apache as Derby
- Early 2005: Sun starts contributing to Apache Derby
- July 2005: Derby graduated from Apache Incubator
- Released 10.1 in August 2005
- Current version 10.5 (April 2009)

Apache Derby Community

- Using Apache infrastructure
- Communication mostly through mailing lists (archived)
 - > Some communication on IRC
- Groups of contributors
 - > Larger company involvement (Sun Microsystems, IBM)
 - > Independent
 - > Google Summer of Code (GSoC) students
- User feedback
 - > Users mailing list
 - > Issue tracker (JIRA)
 - > Company internal

Apache Derby Community (cont.)

- Release process
 - > Two release types: feature and maintenance
 - > Committer volunteers as release manager
 - > Schedules key points
 - Branch creation
 - Beta build and/or release candidate
 - Release note draft circulation
 - Period of increased testing
 - Community voting
 - > Approved release is published at end of vote
- Maintenance releases easier to produce

Apache Derby Community (cont.)

- Tests run and published
 - > Tinderbox test
 - > Daily tests on multiple platforms and JVMs
 - > Performance regression tests
 - > Large data volume test
 - > Long-running test
- Hudson
- Current community activities
 - > New 10.5 maintenance release (10.5.3, August)
 - > Ongoing work for next feature release (10.6)

Distributions

- Current distributions
 - > Apache Derby by the Apache Software Foundation



- > Java DB by Sun Microsystems

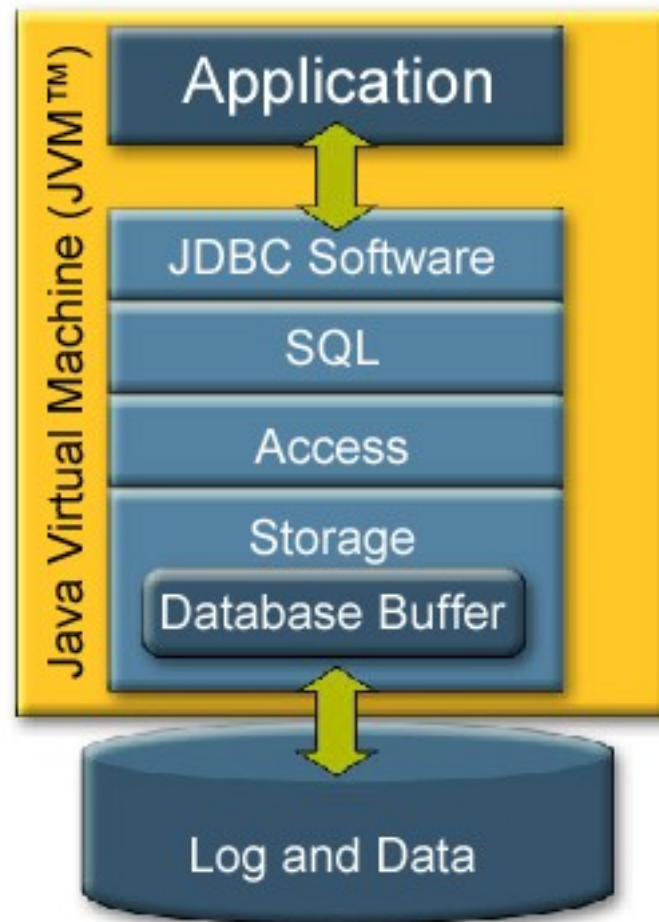
Java DB

- Discontinued distributions
 - > Cloudscape by IBM
(IBM is still involved in the ASF community)

Apache Derby – Product Overview

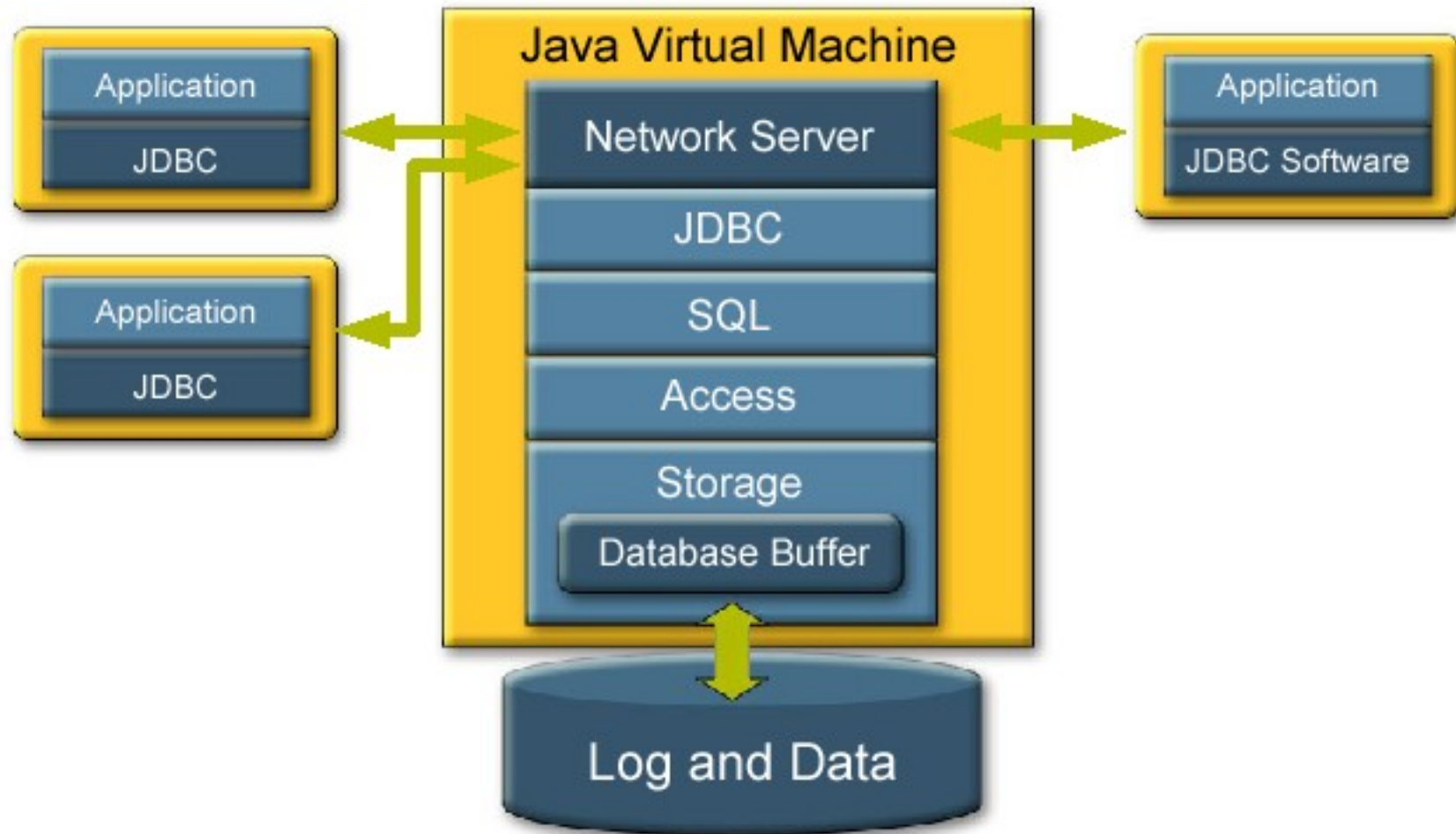
Derby Modes

- Embedded



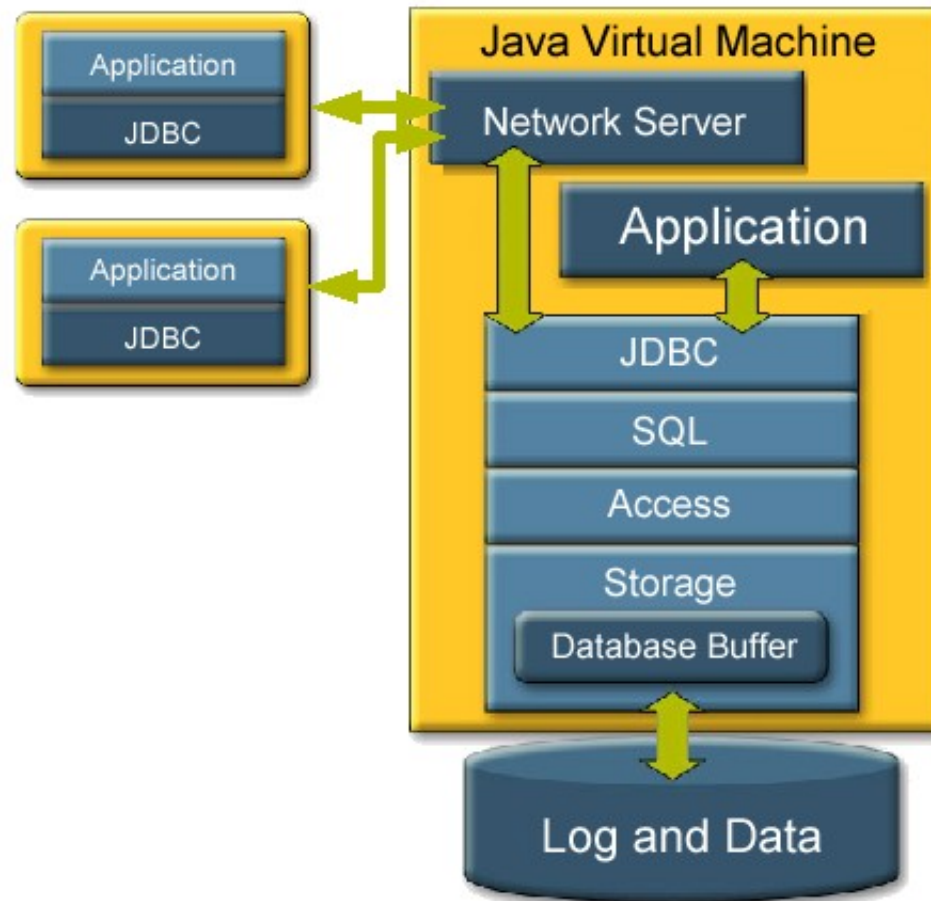
Derby Modes (cont.)

- Client/Server



Derby Modes (cont.)

- Embedded + Client/Server



Features

- Multi-user, multi-connection, multi-threaded
- ACID properties
 - > Transactions
 - > Foreign keys, check constraints
 - > 4 isolation levels
 - > Crash recovery, write-ahead-logging
- Row-level locking, deadlock detection, lock timeouts
- Online/offline backup, restore
- Encryption
- Stored procedures and triggers

Features (cont.)

- SQL
 - > Tables, indexes, read-only views, joins, triggers, procedures, functions, foreign keys, constraints, temporary tables
 - > Cost based optimizer

NOTE: SQL queries are compiled into Java bytecode

- Read-only databases in a JAR/ZIP
- Client side JDBC statement cache
- XA support

Derby Table Functions

- Allows SQL access to external resources
 - > Non-relational data
 - Plain text files
 - XML
 - > Legacy data sources
 - > Other databases
 - Tear-off subsets
 - Import data
 - Migration
 - “Distributed joins”

Derby Table Functions (cont.)

- Accessed through a *public static void* method
- Returns a *java.sql.ResultSet* implementation
- *Define and invoke a function*

create function myFunc(x, y) returns table (...) ...

select ... from table(myFunc('/tmp/a.dat', 7))

- *Use with SQL*
 - > Simple select: *select ...*
 - > Import: *insert into ... select ...*
 - > Create table from VTI shape: *create table ... as select ...*

Apache Derby – New Features

Generated Columns

- Automatically generated and updated
- Can be indexed
- Allows faster search/ordering on expressions

```
CREATE TABLE CIRCLES(  
RADIUS INT,  
AREA GENERATED ALWAYS AS (  
PI()*RADIUS*RADIUS))
```

Generated Columns (cont.)

```
CREATE FUNCTION MY_POW(  
    A DOUBLE, B DOUBLE)  
    RETURNS DOUBLE  
    DETERMINISTIC  
    PARAMETER STYLE JAVA  
    NO SQL LANGUAGE JAVA  
    EXTERNAL NAME 'java.lang.Math.pow';
```

```
CREATE TABLE CIRCLES(  
    RADIUS INT,  
    AREA GENERATED ALWAYS AS (  
        PI()*MY_POW(RADIUS,2));
```

SQL Roles

- Modeled after the SQL:2003 spec
- Bundle sets of privileges as roles
 - > Assign roles to users
 - > Roles can inherit from other roles
- Simplifies administration
- No default role, use *set role*
- Must enable SQL authorization mode
 - > Limitation: Only database owner can manage roles

In-memory Database back end

- Stores data in memory only
- For transient / temporary data
- Use cases:
 - > Development and testing
 - > Ad-hoc queries
 - > Database initialization
- Not feature complete – feedback appreciated!

`jdbc:derby:memory:myDb;create=true`

Update Statistics Stored Procedure

- Non-unique indexes have cardinality statistics
- Outdated statistics can lead to bad optimizer decisions
- Must be manually updated
 - > Call SYSCS_UTIL.SYSCS_UPDATE_STATISTICS
 - > Compress tables
 - > Recreate indexes
- Work being done on automatic update

OFFSET/FETCH

[OFFSET integer-literal {ROW | ROWS}]

[FETCH { FIRST | NEXT } [integer-literal] {ROW | ROWS} ONLY]

- Standardized way of LIMIT/TOP functionality
- Next increments:
 - > Allow OFFSET/FETCH in subqueries
 - > Allow dynamic parameters with OFFSET/FETCH

Replication of Encrypted Databases

- Derby supports basic master – slave replication
 - > Asymmetric
 - > Asynchronous
- Manual failover
- Added support for replication of encrypted dbs in 10.5

LOB Improvements

- LOBs are read/write
- Previous work
 - > Avoiding materialization
 - > Locator based implementation
- Performance improvements
 - > Streaming
 - > Positioning (due to modified UTF-8 encoding)
 - > Avoiding unnecessary work
- Freeing resources on the server

Where to get Derby / Java DB

- Apache: db.apache.org/derby
- Sun Microsystems: developers.sun.com/javadb
- Bundled in Sun JDK for Java SE 6
 - > See under directory `<install_dir>/db/`
- Ubuntu: `sun-java6-javadb`
- OpenSolaris: `SUNWjavadb`



GETTING AQUAINTED WITH APACHE DERBY

Thank You for Your Attention!

Kristian Waagan

Developer, Database Group

Sun Microsystems

Kristian.Waagan@Sun.COM



GETTING AQUAINTED WITH APACHE DERBY

Questions & Answers



GETTING AQUAINTED WITH APACHE DERBY

Thank You for Your Attention!

Kristian Waagan

Developer, Database Group

Sun Microsystems

Kristian.Waagan@Sun.COM

