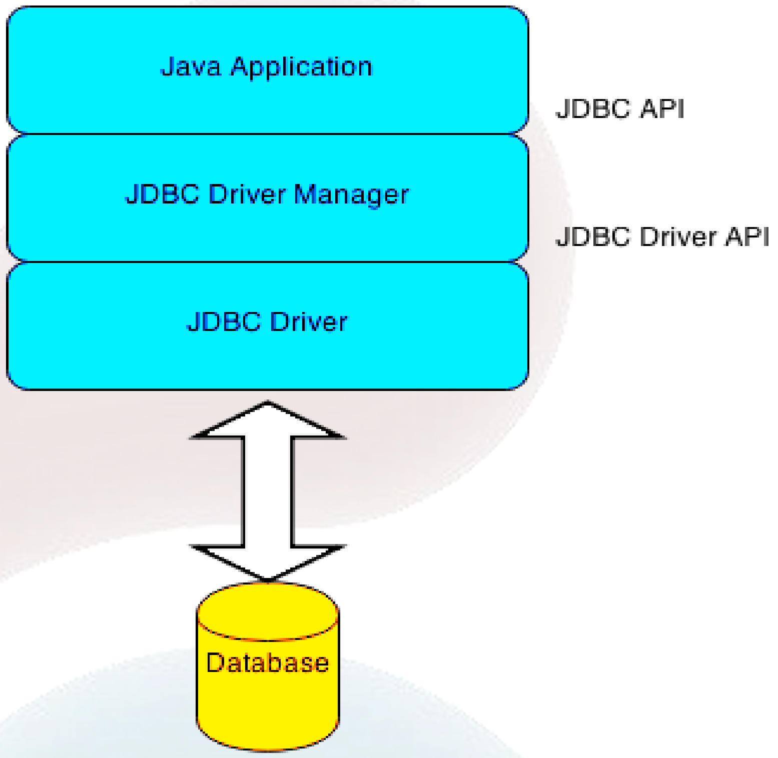
Query Processing and JDBC

J. Pereira R. Vilaça B. Sonntag

JDBC

- JDBC - Java Database Connectivity
- Set of Java interfaces and classes in package java.sql to access relational databases.
- SQL code is used implicitly inside Java code as Strings.



JDBC – Initialization

- Initialization:
  - Load and register JDBC driver in Driver Manager;
  - Obtain a Connection to the database using an URL;
  - It’s also possible to use a DataSource instead.
- Connections are:
  - Single-threaded;
  - Heavy weight;
  - Should be closed when no longer needed.

JDBC - Drivers

- The Driver interface is used by the vendors that implement JDBC drivers.
- It is needed to load the driver class in the application. Can be done using Class.forName("org.postgresql.Driver");
- The DriverManager class handles objects of type Driver
  - Register, remove and list drivers
- Use the DriverManager to obtain a Connection:
  - Connection connection = DriverManager.getConnection(url, user, pw);

JDBC – URLs

- Uses syntax similar to net URLs
  - jdbc:mysql://localhost:3306/mydb
  - jdbc:derby:test//localhost/testdb
- General syntax
  - jdbc:subprotocol_name:other_stuff
  - Where subprotocol selects the specific driver
  - Format for other_stuff depends on subprotocol, but in general:
    - jdbc:subprotocol://hostname:port/database_name
- jdbc:postgresql://localhost:5432/colorfulWidget would connect to the colorfulWidget database of localhost, using the Postgres driver

JDBC - Statement

- Used to execute SQL.
- Created using the createStatement() method on a Connection object.
  - Statement stmt = connection.createStatement();
- Executing DDL statements: CREATE, DROP, ALTER TABLE
  - stmt.execute("CREATE TABLE widget (id INT PRIMARY KEY, color CHAR(20))");
- Executing DML update statements: INSERT, UPDATE
  - int modifiedRows = stmt.executeUpdate("INSERT INTO widget (id,color) VALUES (1, ‘Red')");
  - Returns the number of modified rows.

JDBC – Prepared Statements

- Compiled once and executed several times. More efficient when several similar queries with different parameters are executed
- Accepts SQL queries with parameters specified as “?”.
- The parameters are set using the setXXX() where XXX is the parameter data type.
  - PreparedStatement stmt = connection.prepareStatement(“INSERT INTO widget (id, color) VALUES (?, ?)”);
  - stmt.setInt(1, 1);
  - stmt.setString(2, "Red");

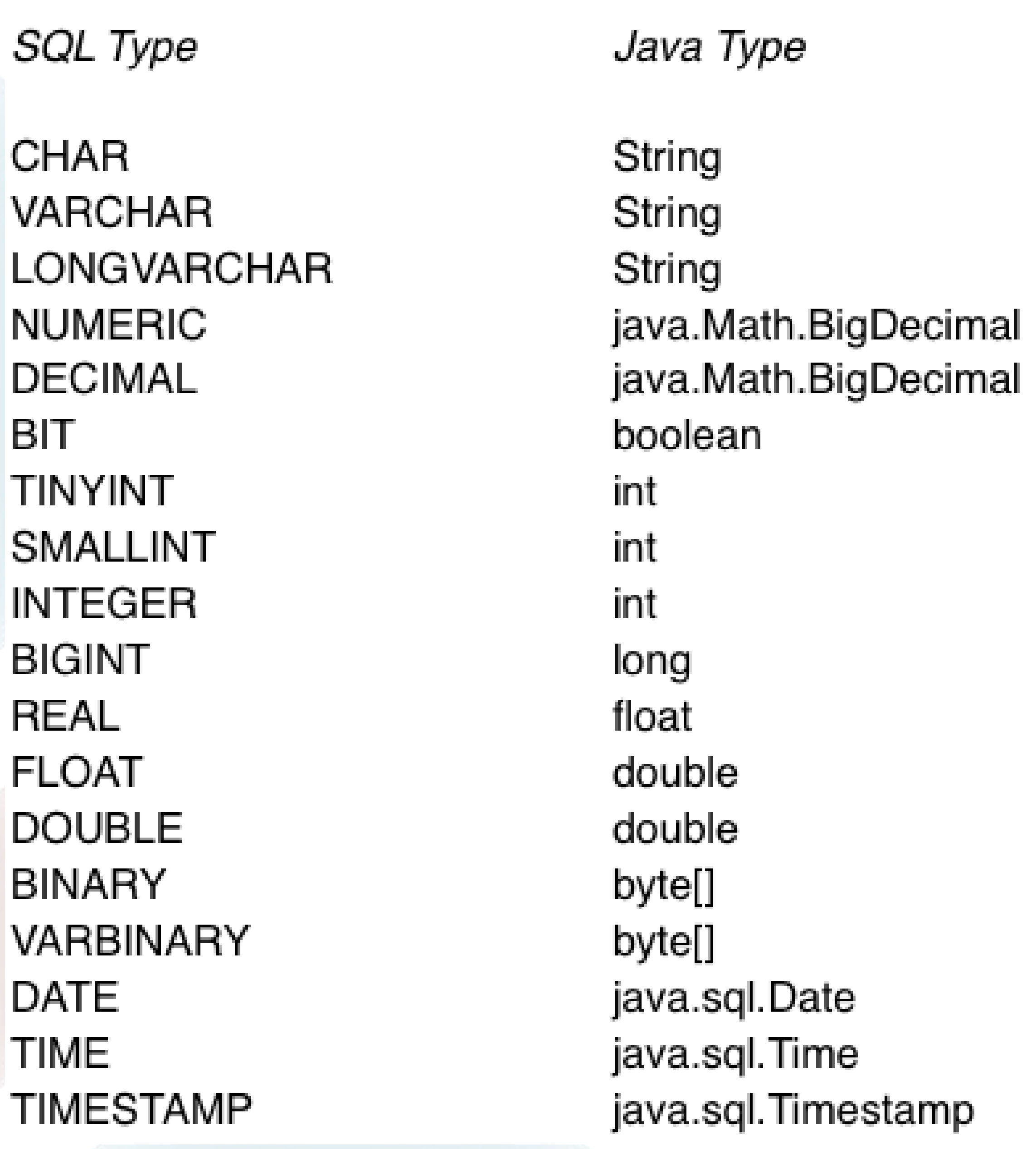
JDBC - Queries

- Queries returning data are executed using executeQuery:
  - ResultSet rs = stmt.executeQuery("SELECT * FROM widget");
- The ResultSet is similar to the Iterator:
  - Allows to navigate through the retrieved data and read the columns selected in the query;
  - Main methods: next(), previous(), first() and last().
  - Columns are fetched using the appropriate method for the type of that column and either the column number or the column name:
    - rs.getInt("id");
- Use Prepared Statements to avoid compilation overhead.

JDBC - Isolation

- Delimiting transactions:
  - setAutoCommit(false) and commit()/rollback()
  - Undo modifications on rollback
  - Be prepared for conflicting transactions
- The application must be ready for concurrency aborts
  - Handling possible errors

JDBC – SQL/Java Types



SQL Type	*Java Type*
CHAR	String
VARCHAR	String
LONGVARCHAR	String
NUMERIC	java.Math.BigDecimal
DECIMAL	java.Math.BigDecimal
BIT	boolean
TINYINT	int
SMALLINT	int
INTEGER	int
BIGINT	long
REAL	float
FLOAT	double
DOUBLE	double
BINARY	byte[]
VARBINARY	byte[]
DATE	java.sql.Date
TIME	java.sql.Time
TIMESTAMP	java.sql.Timestamp

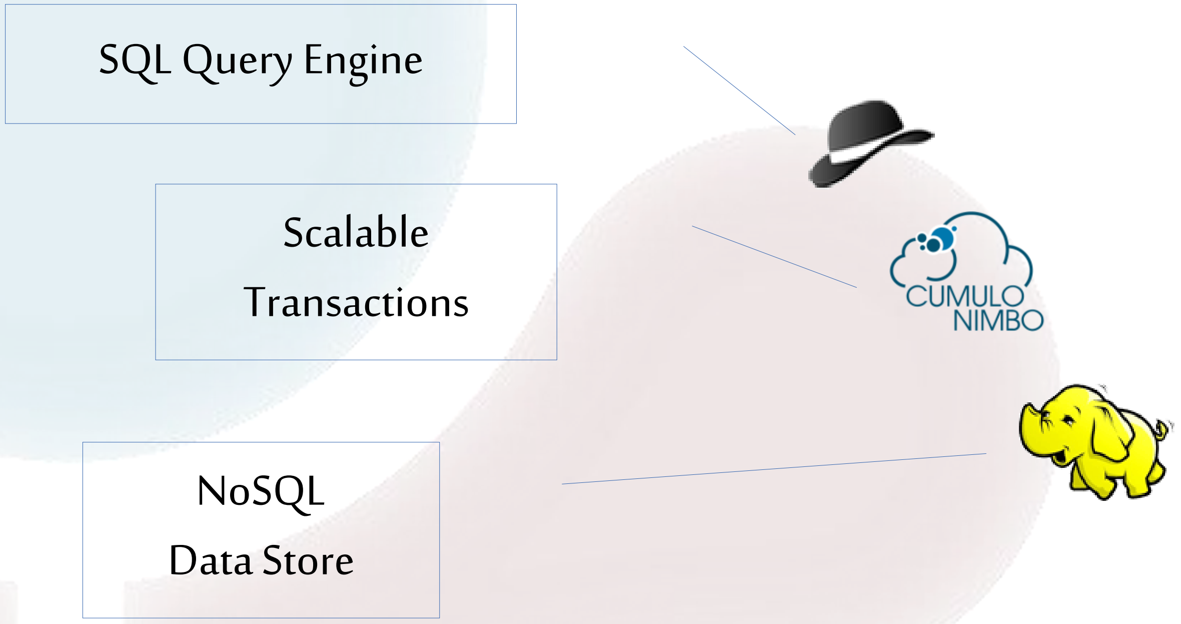
Distributed databases @ MEI

- Distributed Systems and Cryptography
  - Reliable Distributed Systems

- Applications Engineering
  - Database Administration

Research @ HASLab

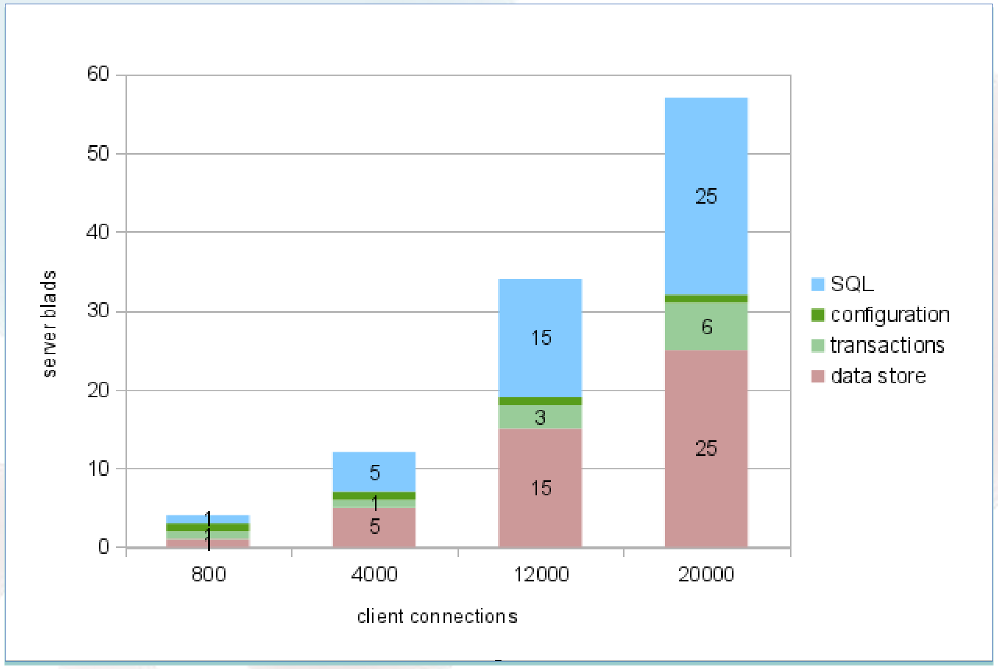
› CumuloNimbo: Highly Scalable Transactional Multi-Tier PaaS

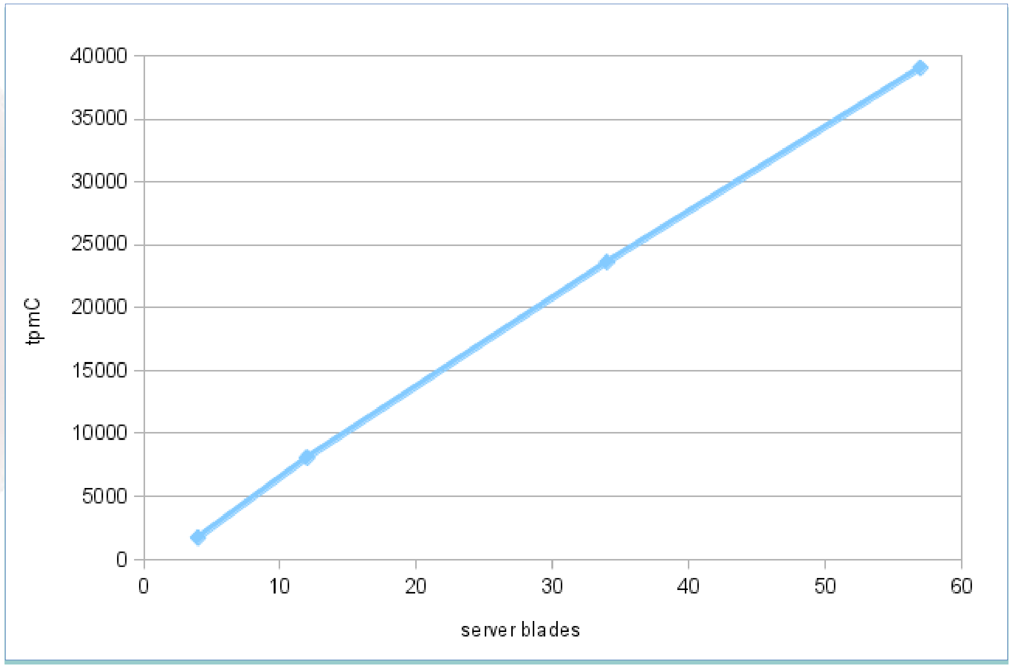


https://www.youtube.com/watch?v=2ShTc0v-3Aw

Research @ HASLab

- CumuloNimbo:
  - Standard SQL / JDBC
  - Independently elastic layers
  - Evaluated with TPC-C (OLTP standard)





quad-core Intel Xeon X3220 @ 2.40GHz / 8GB RAM

500GB SATA disk / 1Gbits Ethernet

Research @ HASLab

- ›CoherentPaaS
  - ›A Coherent and Rich PaaS with a Common Programming Model
- ›LeanBigData
  - ›Ultra-Scalable and Ultra-Efficient Integrated and Visual Big Data Analytics

HIGH-ASSURANCE
SOFTWARE LABORATORY

IMPROVING
PRACTICE
THROUGH
THEORY

HASLab
HIGH-ASSURANCE
SOFTWARE LABORATORY