Minosse RDBMS

Antonello Provenzano

(antonello@minosse.com)

Minosse

Stavvi Minòs orribilmente, e ringhia:
essamina le colpe ne l'intrata;
giudica e manda secondo ch'avvinghia.

Dante Alighieri (Divina Commedia, Inferno -Canto V)

There Minos stands, grinning with ghastly feature:
he, of all who enter, strict examining the crimes;
gives sentence, and dismisses them beneath,
according as he foldeth him around.

Dante Alighieri (Divine Comedy, Inferno – Chant V)

What is it Minosse?

Relational Database System

- Implements ISO/ANSI SQL-92 Standard
- Relational Model

Database Management System

- Server/Client structure
- Service

Embedded Database

The Language

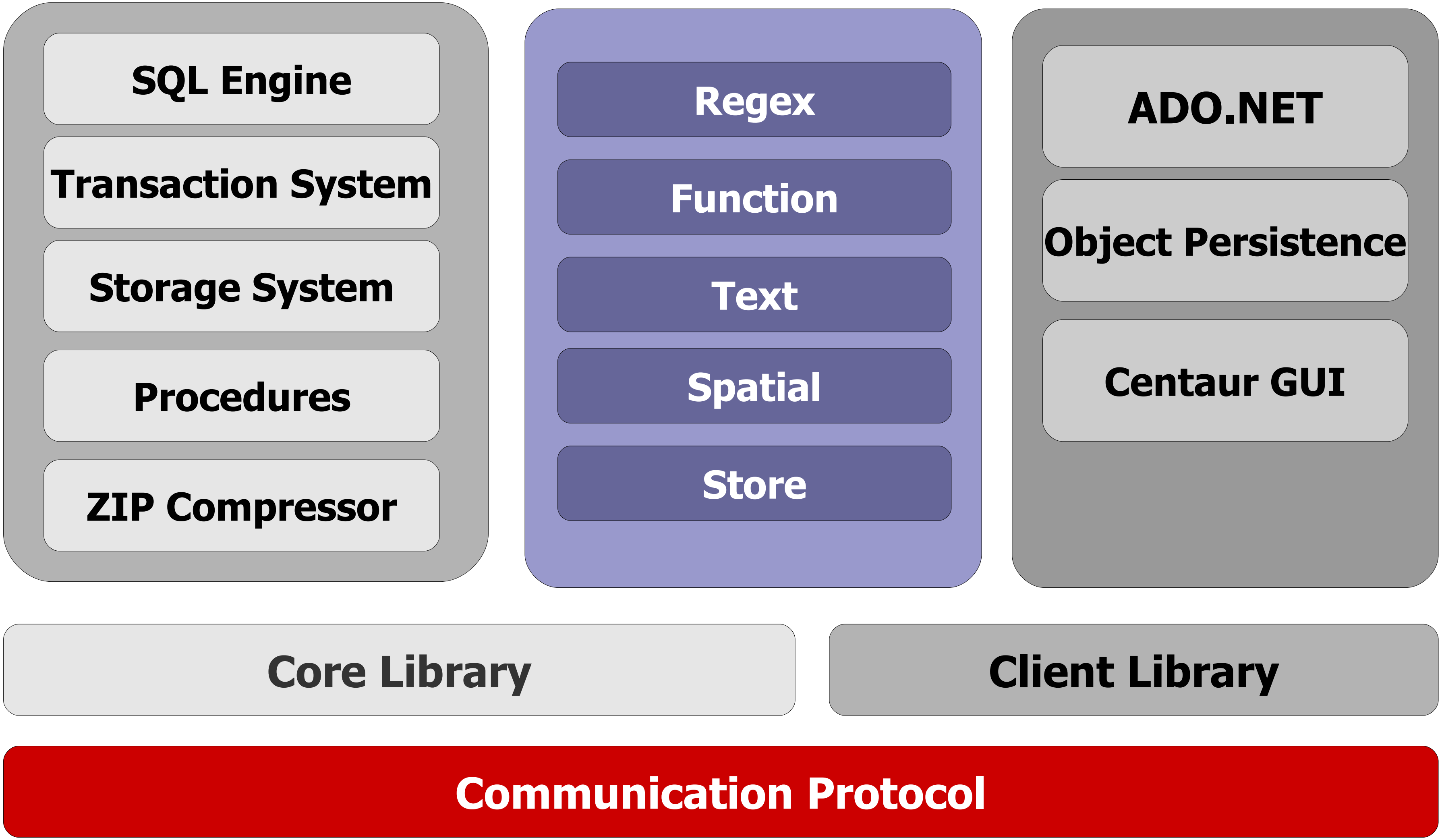
Entirely written in pure C#

- High performances
- Integration with .NET/Mono applications
- Large development community

Development ease

- Fast development of code parts
- Wide range of existing libraries

Component Graph



SQL Engine

Transaction System

Storage System

Procedures

ZIP Compressor

Regex

Function

Text

Spatial

Store

ADO.NET

Object Persistence

Centaur GUI

Core Library

Client Library

Communication Protocol

The pourpose

Building a brand new winning database system

- Strong data-storage system
- Component expandibility
- Complex queries execution

Increase productivity

- Communications on TCP/IP protocol
- Use through accessible API
- Indipendency from server/client structure

Motivations.

Database Systems Proliferation

Trivial Implementations

- Standards not supported in most of the cases
- Poor data-structures
- Large scale database not supported

Systems oriented to just one development framework

- Java/JDBC implementations
- API inaccessible
- Complex portability

The System and C#

Simple and intuitive language

- Easy to be read and interpret
- Provides easy extension of the code
- “Unsafe” code for performance boost
- Fastly learnable (specially from Java)

Development Community

- Growing fastly inlast years: one of the biggest
- Open-source and propertary code development

At the Beginning

Minosse is a personal project

- Build an integrated data management system for other applications
- Manage a large amount of data highly scalable
- Language and runtime probing

OpenSource project

- Contribute to the development of Mono community
- GPL: core libraries, server and clients

Today.

The Development Team

Companies:

- Deveel Ltd
- Novell Inc.
- Mainsoft Inc.

Developers:

- 4 developers
- 3 contributors

SQL Support

ISO/ANSI SQL-92 Implementations

- Schemas, tables, temporary tables, views, triggers, indices, savepoints
- Standard types plus `BLOB` and `CLOB`
- Sub-queries joins (`INNER JOIN`, `OUTER JOIN`, etc.)
- Constraints (`PRIMARY KEY`, `FOREIGN KEY`, `UNIQUE`, `CHECK`)
- Transactions

Additions to SQL Standards

Types

- Spatial types (`POINT`, `LINESTRING`, `POLYGON`, etc.)
- Interval type

Functions

- Spatial functions based on GIS SFS
- String, Mathematical and various other
- Sequences

Data Storage

Storage API

- Expandibility of the system
- Custom implementations

Supported Storage

- File storage system supporting transactions
- Memory storage system

FileSystem storage system

Synchronization of data with underlying file-system

- Data saved at each transaction commit
- Maximum size of tables 4TB (2Gb file)

Index based archives

- Fast indexing and selection
- Strong data integrity

The Integrated System

Small size of the application

- Core libraries (mdblib) ~ 1Mb
- Database Server (minossed) ~ 300Kb
- Database Client (mdbcli) ~ 400Kb

Small memory requirement

- No need for big system for optimal performances
- Small table sizes

MinosseCC: The Parser Generator

Heavily based on JavaCC

- Written in Java
- Starts from the same grammar structure as JavaCC

Parser production

- Input grammar files (.mcc)
- Standalone C# parsers output (no requirement of external libraries)
- Fast and easy to manage

Relational Model

Relational System

- Multiple concurrent users
- Multiple concurrent process
- Transactions
- Isolation levels (Serializable)
- Tables/rows lock
- Backup and restore

Statement and data caching

Relational Model: SQL

Relational System on SQL Standardization

- Tables, views, temporary tables, triggers, procedures and functions
- Relational constraints
  - Foreign Key
  - Check
- Table Joins

Server/Client Communication Structure

Open-designed protocol

- Multiple communation protocol (TCP/IP, SOAP, etc.)
- Error codes
- Local/remote connections

Parameter support with '?' placemark

Geographic Information System

Geo-spatial implemantation

- OpenGIS Simple Features Specifications (SFS)
- GIS types
- Spatial functions

Support for geographical types

- Database system types
- Client side types

Text Management Enchancements

String collation

- SQL grammar enchancements (`COLLATE`, `DECOMPOSITION`, `STRENGTH`, `ENCODING`)
- Unicode collation

Regular expressions

- RegEx API

Examples.

SQL: Table Management

```
// Create a simple table
CREATE TABLE IF NOT EXISTS foo
          (column1 INT(5) NOT NULL UNSIGNED,
           column2 VARCHAR(255) NOT NULL,
          PRIMARY KEY(column1), UNIQUE(column2));

// Alterate the tabella
ALTER TABLE foo ADD COLUMN column3 VARCHAR(150) NULL;
ALTER TABLE foo ADD CONSTRAINT const1 UNIQUE (column1);
ALTER TABLE foo DROP COLUMN column3;
ALTER TABLE foo DROP CONSTRAINT const1;
ALTER TABLE foo DROP PRIMARY KEY;
ALTER TABLE foo ALTER COLUMN column1 SET 0;
ALTER TABLE foo ALTER COLUMN column1 DROP DEFAULT;

// Drop the table
DROP IF EXISTS TABLE foo;

// Drop multiple tables
DROP IF EXISTS TABLE foo, goo, doo;
```

SQL: Querying Data

```
// A simple select
SELECT * FROM foo WHERE column1 > 5 ORDER BY column2;

// A more complex selection
SELECT DISTINCT column1, column2, CONCAT('No.', ROUND(column1, 2)
    FROM foo WHERE column1 > 5 ORDER BY column2;

// Select a column value into a variable
SELECT column1 INTO var1 FROM foo WHERE column2 > ABS(column3);

// Insert into a table the result of a selection
INSERT INTO goo SELECT * FROM foo
        WHERE column1 = LTRIM(CONCAT(var1, '.', var2));
```

SQL: Inserting Data

```
// Standard data insertion
INSERT INTO foo (column1, column2, column3)
       VALUES (0, 'doo', 'goo');
INSERT INTO foo VALUES (1, 'doo1', 'goo1');
INSERT INTO foo (column1, column2, column3)
       VALUES (4 + 5, CONCAT('i', 'u', 'p'),
               CONCAT(column1, 'p'));

// Insertion from select
INSERT INTO foo (column1, column2, column3)
       SELECT column4, column5, column6
       FROM foo2 WHERE column6 LIKE '%r';

// Insertion from SET variant
INSERT INTO foo SET column1 = 0, column2 = 'test';
INSERT INTO foo SET column1 = 1, column2 = CONCAT(column1,
'set');
```

The Future.

Things To Do (Small Term)

Database manager

- Graphical User Interface based on Centaur
- Web interface based on ASP.NET

Object Persistence Layer

Command Line Interface Client

Things To Do (Middle Term)

XML Support

- XQuery client
- XML:DB Native Support

Stored Procedures

- PL/SQL-like parser/executor
- .NET/Mono language procedure enchancements

Things To Do (Long Term)

Diagnostics Tools

- Performance Profiling
- Logger enchancements

SQL Language Enchancements

- Type definitions
- Object support

How to Contribute to the Project

C# Developers

- System development
- Debuging and testing

Editors

- User and development manuals/articles
- Management of the site minosse.com

External contributors

Further Information

Minosse RDBMS Project

- http://www.minosse.com
- http://forge.novell.com/modules/xfmod/project/?minosse
- http://www.deveel.com

Email Contacts:

- antonello@minosse.com
- tomi@deveel.com

deveel

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. Deveel Ltd, makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Deveel Labs, reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All Deveel marks referenced in this presentation are trademarks or registered trademarks of Deveel Labs srl in Italy and other countries. All third-party trademarks are the property of their respective owners.

No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of Deveel Ltd. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

minosse databasesystem

deveel