



Firebird RDBMS

Einführung in das relationale Datenbanksystem Firebird



LUG Erding

Inhalt

- Einführung in Datenbanken
- Was ist Firebird
- Versionen
- Feature
- Betriebssysteme
- Installation
- Tools
- Architektur
- Pro's and Con's
- Stolperfallen
- Links / Hilfe
- Fragen



LUG Erding

Was sind Datenbanken

Datenbank:

Sammlung logisch zusammenhängender Daten, die ein Modell der Wirklichkeit darstellen

Datenbankmanagementsystem:

Das Datenbankmanagementsystem (DBMS) regelt den Zugriff auf die Daten

Datenbanksystem

Datenbank + DBMS



LUG Erding

Desktop vs. Client Server Datenbanken

Desktop Datenbanken

- Flat File Datenbank:
- z.B. dBase, Paradox, Access
- Schnell bei exklusiven Zugriff
- Fast keine Eigenintelligenz
- Nicht für gleichzeitigen Zugriff ausgelegt
- Oft proprietäre Abfragesprache
- Kopieren Tabelle zum Client, der dann verarbeitet
- Skalieren schlecht

Sind Dateien, die eine Datenbank darstellen

Client Server Datenbanken

- z.B. Oracle, Postgres, Firebird
- Langsamere Zugriff, da über Netzwerk
- Hohe Eigenintelligenz
- Gleichzeitiger Zugriff von Clients
- Abfrage erfolgt oft in SQL
- Liefern nur Ergebnis zum Client zurück
- Skalieren gut

Daemon, der die Daten und den Zugriff darauf kontrolliert



LUG Erding

Was ist Firebird

- Relationales Datenbank Management System
- Client Server Datenbank
- Open Source (IBPL – ähnlich MPL)
- Aus InterBase 6.0 hervorgegangen
- Für große Datenmengen geeignet
- Wartungsarm und betriebssicher
- Geringer Ressourcenverknappung (13 MB Vollinstallation, <2MB)
- Pro Verbindung 2MB (classic) und <300kb (Super Server)
- Multi Generationen Architektur
- Aktive Datenbank (Trigger, Stored Procedures)



LUG Erding

Geschichte von Firebird

- 1985 gründeten Jim Starkey, Ann Harrison und Don Depalma Groton Database Systems
- 1986 würde Groton in InterBase umbenannt
- 1991 Borland kauft InterBase
- 1994 InterBase 4
- 2000 InterBase 5.6
- 2000, 25 Juli wird InterBase 6.0 als Open Source veröffentlicht
- 2001 Borland kündigt InterBase 6.5 (Closed Source) an
- 2002, 12 März **FireBird 1.0** erscheint (wie InterBase 6.0 – nur weniger Fehler (z.B. kein hard coded User – politically / correct))
- 2004 Firebird 1.5 (Codebase c++, new features, less bugs)
- 2006 Firebird 2.0 RC
- Aktuell: Firebird 1.5.3



LUG Erding

Wer steckt hinter Firebird

- Kommerziell unabhängiges Projekt
- Firebird Foundation koordiniert Arbeitsabläufe, Test Cases
- www.firebirdsql.org / sourceforge.net/projects/firebird
- Support wird in div. Foren geleistet – es gibt auch kommerziellen Support
- Über 100 aktive Mitglieder (Spenden, Entwicklung, Support)
- Vollzeitentwickler gesponsort von der Foundation / von Firmen
- Entwickler sind in der Regel auch Datenbankbenutzer



LUG Erding

Versionen - Stable

- Firebird 1.0, 1.5
 - Embedded
 - Classic und embedded sind unter Unix identisch
 - Windows: dll stellt den Server dar – keine Installation
 - Classic
 - Prozess basiert (über xinetd)
 - SMP
 - Local Connection (sehr schnell im Zugriff)
 - Braucht aber mehr RAM (+2MB pro Verbindung)
 - Langsam beim Verbindungsaufbau
 - Super Server
 - Thread basiert (pro Verbindung)
 - Weniger Ressourcen, schnellerer Verbindungsaufbau
 - Kein SMP, keine lokalen Verbindungen



LUG Erding

Versionen - Development

- Firebird 2.0
 - Firebird 1.5 + extra Feature, weniger Bugs
 - Wird wahrscheinlich Stable auf der Firebird Conference
- Vulcan
 - 64 Bit port
 - Neue Architektur, neue Feature
 - Voll SMP (Faktor 3.6 auf 4x Prozessor System)
 - Code freeze von Firebird 1.5 aus 2005
- Firebird 3.0
 - Nächstes Jahr
 - Merge von Vulcan und Firebird 2.0



LUG Erding

Betriebssysteme

- FreeBSD i386
- Hp UX
- Linux – i386, amd64
- MacOS X
- Sinix
- Solaris – sparc, x86
- Windows (win32, nicht win64)

Andere Architekturen sind in der Regel kein Problem – anpassen des configure Scripts und ggf. alignment ändern.



LUG Erding

Quick installation – Schritt 1: Server installieren

- Rpm based:
 - RHEL 4 braucht compat package
 - Rpm -i FirebirdXX*.rpm
- Ubuntu
 - Apt-get install Firebird-Server
- Gentoo
 - Emerge firebird
- Source:
 - ./configure
 - make
 - make install



LUG Erding

Quick installation – Schritt 2: Konfiguration

- Bei Classic Installation: xinetd konfigurieren

```
service gds_db
{
    flags                = REUSE
    socket_type          = stream
    wait                 = no
    user                 = firebird
    log_on_success       += USERID
    log_on_failure       += USERID
    server               = /opt/firebird/bin/fb_inet_server
    disable              = no
}
```

- firebird.conf anpassen

```
vi /opt/firebird/firebird.conf
# specify firebird directory
DatabaseAccess = Restrict /var/db/firebird

    mkdir -p /var/db/firebird
    chown firebird:firebird /var/db/firebird
/etc/init.d/xinetd start
```



LUG Erding

Quick installation – Schritt 2: Konfiguration

- Läuft der Dienst

```
netstat -an | grep 3050
```

- Verbindung zur Datenbank aufbauen
- Datenbank erzeugen

```
/opt/firebird/bin/isql -user sysdba -password xxx  
Use CONNECT or CREATE DATABASE to specify a database  
SQL> create database '127.0.0.1:/var/db/firebird/testdatabase.fdb';  
SQL> commit;  
SQL> quit;  
ls -lh /var/db/firebird/  
-rw-r--r-- 1 firebird firebird 584K Aug 22 12:16 testdatabase.fdb
```



LUG Erding

Tools

- **Shipped:**

- Isql command line tool
- Gbak backup tool
- Nback seit 2.0 inkrementelles Backup
- Gstat Statistiken über die Datenbank
- Gfix Datenbank reparieren
- Qli command line interpreter
- Gpre c Pre-Prozessor

- **Other:**

- Flamerobin gui tool, cross platform, open source (flamerobin.sf.net)
- Eclipse clay database plugin
- Squirrel sql eclipse plugin
- IB Export, SQL Workbench (Windows only, closed source)



LUG Erding

Feature

- Sub Select / Inline Select / Joins / Views
- Domänen
- Generatoren / Sequenzen
- Trigger (INSERT, UPDATE, DELETE)
- VIEWS
- Stored Procedures, Exceptions
- UDF – User Defined Functions
- External Tables
- ACID compliant transactions, Savepoints
- Unicode
- Blob Filter
- Events
- Derived Tables, Raw Devices, new Protocol, Cursor (2.0)
- Inkrementelles Backup (1.5 addon, 2.0 dabei)



LUG Erding

Connectivity

- C native
- C++ IBPP (Cross Plattform), Borland Interbase Driver
- Delphi
- PHP (use InterBase driver)
- Python kinterbasdb.sf.net
- ODBC
- JDBC jaybird.sf.net
- Ruby <http://wiki.rubyonrails.org/rails/pages/Firebird+Adapter>
- .NET



LUG Erding

Architektur - Begriffe

- Transaktion
 - Ein oder mehrere zusammenhängende Aktionen die als eine Aktion gesehen werden sollen
- Commit
 - Transaktion wird gespeichert
- Rollback
 - Transaktion wird nicht gespeichert
- TIP
 - Transaction Inventory Page: Transaktionsnummer, Timestamp und Zustand der Transaktion werden hier gespeichert
- OIT
 - Oldest interesting Transaction: älteste Transaktion, die nicht COMMITTED wurde
- BDR
 - Back Difference Record: Delta zwischen dem letzten und akt.



LUG Erding

Architektur - Funktionsweise

- Jede Transaktion bekommt einen Timestamp und eine ID
 - Werden mit dem Zustand in der TIP gespeichert
- Jeder Datensatz hat die ID die ihn verändert hat
- Wird ein Datensatz verändert, erzeugt, oder gelöscht wird er mit der ID der Transaktion versehen

- ON UPDATE:
 - BDR (delta) mit der alten ID wird erzeugt und in der DB gespeichert
 - Datensatz wird mit den neuen Werten gespeichert und mit der neuen ID versehen
 - Die neue Version erhält eine Referenz auf das BDR



LUG Erding

Architektur – Funktionsweise II

- ON DELETE:
 - Der alte DS wird als gelöscht markiert
- ON INSERT:
 - Der neue Datensatz wird mit der ID versehen

Damit die Datenbank nicht ins unermessliche wächst wird ein Garbage Collect durchgeführt

- Classic Server: Kooperativer GC: jedes mal wenn auf den DS zugegriffen wird wird aufgeräumt
- Super Server: DS wird nur getagged, und dann von extra Thread aufgeräumt
- Rolled-Back Transactions werden nicht garbage collected – sie werden durch den Sweep als committed markiert.



LUG Erding

Architektur – sweep - housekeeping

- Manuell oder automatisch
 - Wenn auf einen DS über einen SELECT zugegriffen wird
 - Wenn Next Transaction – OIT > 20.001
- Der Prozess der die Transaktion gestartet hat ist der Looser und räumt alles auf (fixed in 2.0)
- Sweeping entfernt alle unnötigen BDRs und synchronisiert OIT mit dem aktuellen Wert



LUG Erding

Architektur – Internals

- Metadaten werden in RDB\$ Tabellen abgespeichert
- Stored Procedures und Trigger werden als BLR (Binary Language Representation) gespeichert
- SQL Anweisung wird beim 1. Prepare in BLR kompiliert und von da an als BLR ausgeführt
 - Nur bei parametrisierten SQL Anweisungen sinnvoll
 - Probleme mit LIKE -> bei FB 1.5 wird kein Index verwendet – lieber STARTING WITH verwenden



LUG Erding

Pro's and Con's

- Pro
 - ACID transaction support
 - Kleiner Footprint
 - Großer Einsatzbereich: von Embedded bis Client Server
 - Open Source
 - Feature Rich
 - Performant
- Con's
 - Wichtige Feature fehlen mir: clustered tables, clustering, blob text search
 - Nur Indices auf Textfelder mit < 254 Zeichen (FB < 2.0)
 - Kein Bulk import



LUG Erding

Firebird vs. MySQL

- Daten Import (LOAD DATA INFILE)
- Cluster fähig
- Viele Ergänzungen zum SQL Standard
- Besteht eigentlich aus mehreren Datenbanken – alle haben unterschiedliche Feature, Konfigurationsoptionen, Probleme
- Für mich: ungewohnte Bedienung: Useränderungen erst nach mysqladmin reload ersichtlich
- Transaktionen / Stored Procedures bei Firebird ausgereifter
- Firebird braucht kein / hat kein Redo Log -> keine aufwändige Administration nach crash



LUG Erding

Stolperfallen

- LIKE '%xzy' und LIKE ? Verwenden keinen Index
 - STARTING WITH ? Verwendet Index
- SELECT COUNT(*) FROM XYZ; ist tödlich
- Transaktionen
 - Immer so kurz wie möglich offen halten
 - Nicht vom Benutzerverhalten abhängig machen
 - Commit ist billiger als Rollback
 - Jeder Zugriff (auch SELECT findet im Kontext der Transaktion statt)
- Generatoren sind außerhalb des Transaktionskontextes



LUG Erding

Links

- WEB Seiten

- <http://www.firebirdsql.org>
- <http://www.ibphoenix.com>
- <http://flamerobin.sf.net>

Firebird Project Page

Support Firma, Knowledge Base

Datenbanktool

- Mailinglisten:

- firebird-support@yahoogroups.com
- firebird-general@yahoogroups.com
- firebird-architect@yahoogroups.com



LUG Erding

Fragen?



