



IBM Software Group

Vytváření uživatelských datových typů a funkcí v Javě

Jan Musil

(jan_musil@cz.ibm.com)

IBM Certified System Administrator - Informix Dynamic Server V11

Informix

CIDUG 

Information Management software

© 2008 IBM Corporation

Přehled prezentace

- Přehled technologie J/Foundation
- Popis Java Virtual procesoru
- Konfigurace databázového serveru pro technologii J/Foundation
- Vytváření Java uživatelských funkcí
- Kompilování a instalace Java UDR do databázového serveru
- Přehled J/Foundation knihoven pro psaní Java UDR
- Opaque datové typy a jejich implementace v Java prostředí
- Praktické ukázky

Přehled technologie J/Foundation

- Infrastruktura poskytovaná databázovým serverem pro psaní Java uživatelských funkcí (Java UDR) resp. podporu uživatelských datových typů
- Databázový server váže definici SQL funkce (procedury) se spustitelnou Java funkcí
- Databázový server provádí mapování mezi SQL datovým typem a Java objektem pro předávání argumentů a návratových hodnot
- Technologie poskytuje podporu pro řešení chybových stavů
- Domovský adresář: \$INFORMIXDIR/extend/krakatoa

Popis Java Virtual procesoru

- Java Virtual Procesor (Java VP) je specializovaný proces databázového serveru, který poskytuje prostředí pro vykonávání Java UDR
- V každém Java VP je zapouzdřen Java Virtual Machine (JVM)
- Prostřednictvím Java VP jsou vykonávány také všechny SQL příkazy zadávané prostřednictvím Java UDR
 - Součástí Java VP je CPU VP, který okamžitě vykonává všechny SQL příkazy
 - Nedochozí tak k předávání SQL příkazů mezi Java VP a CPU VP
- SINGLE_CPU_VP = 0 i v případě konfigurace 1 CPU VP

Konfigurace databázového serveru pro technologii J/Foundation

- Smart blob prostor pro Java JAR
- JVP properties konfigurační soubor
- ONCONFIG konfigurační parametry
- Proměnné prostředí
- Podpora GLS

Smart blob prostor pro Java JAR

- Databázový server ukládá Java JAR soubory do smart blob prostoru
- Smart blob prostor musí být vytvořen a nastaven konfiguračním parametrem SBSPACENAME

JVP properties konfigurační soubor

- Cesta a jméno jsou nastaveny ONCONFIG konfiguračním parametrem JVPPROFILE
- Obsahuje nastavení pro Java VP
- Předpis pro nastavení (přednastavené hodnoty) jsou v souboru \$INFORMIXDIR/extend/krakatoa/.jvpprops.template
- Nastavení
cp \$INFORMIXDIR/extend/krakatoa/.jvpprops.template
\$INFORMIXDIR/extend/krakatoa/.jvpprops
JVPPROFILE \$INFORMIXDIR/extend/krakatoa/.jvpprops

ONCONFIG konfigurační parametry

JVPHOME \$INFORMIXDIR/extend/krakatoa

JVPLOGFILE \$INFORMIXDIR/extend/krakatoa/jvp.log

JVPPROFILE \$INFORMIXDIR/extend/krakatoa/.jvpprops

JVPJAVAVM server

VPCLASS jvp,num=1

JVMTHREAD native

JVPJAVALIB /lib/sparc/

JVPCLASSPATH
\$INFORMIXDIR/extend/krakatoa/krakatoa.jar:
\$INFORMIXDIR/extend/krakatoa/jdbc.jar

(krakatoa_g.jar a jdbc_g.jar jsou knihovny pro debug mód)

Proměnné prostředí

- AFDEBUG (AFCRASH)
- JVM_MAX_HEAP_SIZE
- JAR_TEMP_PATH
- JAVA_COMPILER

Podpora GLS

- Data v databázi jsou uložena v DB_LOCALE kódové sadě
- Java UDR zpracovává data v Unicode
- Při předávání dat mezi databázemi a Java UDR dochází ke konverzi z DB_LOCALE na Unicode případně z Unicode na DB_LOCALE

Vytváření Java uživatelských funkcí

- Každá Java UDR se mapuje na statickou Java metodu, která je součástí JAR archivu
- Mapování SQL datových typů na Java objekty se provádí na základě specifikace Informix JDBC ovladače
- Rozlišujeme Java uživatelské funkce a procedury
 - Funkce vrací hodnoty
 - Procedura nevrací hodnoty
- Java UDR se mohou použít jako
 - Funkce nebo procedury koncového uživatele
 - Agregáční funkce
 - Funkce pro paralelní zpracování
 - CAST funkce
 - Operátorové funkce
 - iterátorové funkce
 - funkcionální indexy
 - podpora opaque datových typů
 - funkce negace

Kompilování a instalace Java UDR do databázového serveru

```
public class Myfce {  
    public static String myfce(String arg) throws SQLException  
    {  
        String regval;  
        ...  
        return regval;  
    }  
}
```

Myfce.java

```
$ javac Myfce.java  
$ jar cvf fce.jar Myfce.class
```

```
execute procedure  
install_jar("file:/home/informix/work/fce.jar", "fce_jar", 0);  
  
create function myfce(lvarchar) returns lvarchar  
external name 'myfce_jar:Myfce.myfce(java.lang.String)'  
language java;
```

myfce.sql

```
$ dbaccess db myfce.sql
```

Přehled J/Foundation knihoven pro psaní Java UDR

- Všechny běžné Java knihovny z JDK
 - Java.util.*, java.io.*, java.net.*, java.rmi.*, java.lang.*, java.sql.* a podobně
 - Nikoliv však grafické a s uživatelským rozhraním, jako java.awt.*, java.applet.* a podobně
- Informix JDBC Driver API knihovny (1.0)
- Informix rozšíření k JDBC Driver (2.0)
 - Rozšíření pro uživatelské datové typy resp. blob/clob objekty
- Další Informix rozšíření
 - com.informix.udr.*

Co je to opaque datový typ ?

- Opaque datový typ je uživatelsky definovaný datový typ jehož interní struktura je definovaná uživatelem a je neznámá databázovému serveru
- Funkce potřebné pro práci s opaque datovým typem musí být také uživatelsky definované
- Definice opaque datového typu vyžaduje (napsané v C nebo Java)
 - Datovou strukturu, která definuje, jak bude datový typ interně uložen v databázových strukturách
 - Podpůrné funkce, které umožňují databázovému serveru s ním pracovat
 - Volitelné modifikátory a další funkce

Implementace Opaque datových typů v Java prostředí

- Java třída rozšířená o SQLData interface
 - Základní metody třídy (konstruktor, konverzní metody,...)
- Metody SQLData interface
 - převod I/O datového toku do/z interní struktury
 - readSQL(), writeSQL() a getSQLTypeName()
- Metody pro podporu opaque datového typu
 - I/O metody pro podporu SQL (CAST operátorové funkce)
 - Metody pro implementaci základních operací (=,<,>,<=>,>=<,<>)
 - Metody pro podporu vytváření B-Tree indexů

Integrace Java opaque datových typů v IDS

- Vytvoření SQL opaque typu
 - CREATE OPAQUE TYPE
- Mapování SQL opaque typu na Java třídu
 - EXECUTE PROCEDURE setUDTExtName
- Registrace funkcí pro podporu opaque datového typu
- Vytvoření konverzních pravidel
 - CREATE IMPLICIT/EXPLICIT CAST

Praktická ukázka

- Vytvoření nového opaque datového typu pro ukládání rodných čísel
 - Jméno nového datového typu: **rodcis**
- Kontrola tvaru RČ
 - Délka
 - Datum narození a koncovku odděluje „/“
 - Kontrolní číslici pro RČ od 1.1.1954
 - Konzistence pomocí kontrolní číslice
- Implementace B-Tree indexování
- Implementace operátorů = <> < > =< =>
- Aplikace pravidla porovnávání podle data narození bez ohledu na pohlaví a koncovku RČ

Otázky a odpovědi

Děkuji za pozornost!

Jan Musil
jan_musil@cz.ibm.com