

Distribuované systémy a výpočty (14)

Jan Janeček

KP FEL ČVUT

7/1/2008



Mobilita výpočtu

Mobilní agenti

- ▶ aktivní objekty
- ▶ nepreemptivní migrace určovaná agentem
- ▶ sběr a distribuovaná analýza dat

Aktivní síť

- ▶ "přetěžování" existujících metod
- ▶ redefinice chování síťových prvků

Mobilita výpočtu

Mobilní agenti

- ▶ aktivní objekty
- ▶ nepreemptivní migrace určovaná agentem
- ▶ sběr a distribuovaná analýza dat

Aktivní síť

- ▶ "přetěžování" existujících metod
- ▶ redefinice chování síťových prvků

Agenti

Autonomní agenti

AI - modifikace chování pro dosažení optima

Multiagentní systémy

AI - kooperace jinak autonomních agentů

Distribučování agentů

spolupráce komponent výpočtu v distribuovaném prostředí

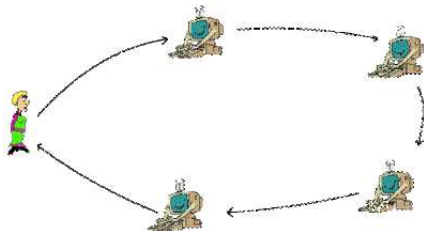
Mobilní agenti

distribučování agentů s možností migrace

Mobilní agenti

Mobilní agent

- ▶ mobilní programová komponenta
 - ▶ specifické výpočetní paradigma
- ▶ autonomní rozhodování o migraci
 - ▶ mobilní agent přeruší výpočet a odmigruje na jiný počítač
- ▶ omezená komunikace se zdrojem
 - ▶ agent doručí výsledek výpočtu migrací



Mobilní agenti

Úroveň mobility

- ▶ Mobilní kód
 - ▶ migrace kódu
- ▶ Mobilní objekt
 - ▶ migrace kódu a dat
- ▶ Mobilní agent
 - ▶ migrace kódu, dat a stavu

- ▶ Slabá mobilita - Weak Mobility
 - ▶ migrace kódu a dat
- ▶ Silná mobilita - Strong Mobility
 - ▶ migrace kódu, dat a stavu výpočtu

Mobilní agenti

Úroveň mobility

- ▶ Mobilní kód
 - ▶ migrace kódu
- ▶ Mobilní objekt
 - ▶ migrace kódu a dat
- ▶ Mobilní agent
 - ▶ migrace kódu, dat a stavu

- ▶ Slabá mobilita - Weak Mobility
 - ▶ migrace kódu a dat
- ▶ Silná mobilita - Strong Mobility
 - ▶ migrace kódu, dat a stavu výpočtu

Mobilní agenti

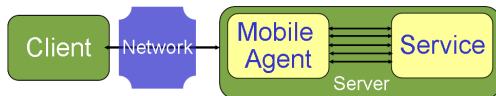
Klient - server

- ▶ úzká spolupráce klienta se serverem
- ▶ vysoká komunikační náročnost



Mobilní agent

- ▶ autonomní činnost agenta
- ▶ redukce přenášeného objemu dat



Mobilní agenti

Výhody

- ▶ redukce datových toků
- ▶ lokální komunikace agenta se serverem
- ▶ autonomní práce
- ▶ vhodné pro mobilní zařízení s omezenými možnostmi
- ▶ přesun náročnějšího výpočtu na výkonné prvky
- ▶ možnost práce i při odpojení klienta

Problémy

- ▶ bezpečnost
- ▶ uzavřenost - typicky u skriptových řešení

Mobilní agenti

Výhody

- ▶ redukce datových toků
- ▶ lokální komunikace agenta se serverem
- ▶ autonomní práce
- ▶ vhodné pro mobilní zařízení s omezenými možnostmi
- ▶ přesun náročnějšího výpočtu na výkonné prvky
- ▶ možnost práce i při odpojení klienta

Problémy

- ▶ bezpečnost
- ▶ uzavřenost - typicky u skriptových řešení

Infrastruktura pro mobilní agenty

Podpora

- ▶ spolupráce s lokálním systémem - kontext, platforma
- ▶ migrace mezi počítači
- ▶ komunikace mezi agenty - kanály

Toolkity

- ▶ Telescript - General Magic - speciální jazyky
- ▶ Agent Tcl / D'Agents - Dartmouth College
- ▶ Voyager - Object Space - JVM
- ▶ Odyssey - General Magic - JVM
- ▶ Aglets - IBM - JVM

Infrastruktura pro mobilní agenty

Podpora

- ▶ spolupráce s lokálním systémem - kontext, platforma
- ▶ migrace mezi počítači
- ▶ komunikace mezi agenty - kanály

Toolkity

- ▶ Telescript - General Magic - speciální jazyky
- ▶ Agent Tcl / D'Agents - Dartmouth College
- ▶ Voyager - Object Space - JVM
- ▶ Odyssey - General Magic - JVM
- ▶ Aglets - IBM - JVM

JVM platformy pro mobilní agenty

Java je vhodným prostředím pro technologii mobilních agentů, protože zajišťuje:

- ▶ přenositelnost mezi OS platformami
 - ▶ bytecode
- ▶ bezpečnost
 - ▶ sandboxing, security manager
- ▶ dynamické zavádění tříd
 - ▶ výpočet migrujícího kódu
- ▶ serializaci objektů
 - ▶ migrace objektů

JVM platformy pro mobilní agenty

Java je vhodným prostředím pro technologii mobilních agentů, protože zajišťuje:

- ▶ přenositelnost mezi OS platformami
 - ▶ bytecode
- ▶ bezpečnost
 - ▶ sandboxing, security manager
- ▶ dynamické zavádění tříd
 - ▶ výpočet migrujícího kódu
- ▶ serializaci objektů
 - ▶ migrace objektů

Aglets

O jazyk Java se opírající toolkit vyvinutý IBM (source code public license)

Aglet = Agent+applet

Vlastnosti:

- ▶ aglet je mobilní Java objekt migrující mezi servery Aglets
- ▶ výpočet omezený kontextem - Context
- ▶ více kontextů na serveru
- ▶ Aglet = Aglet state (variable values) + Aglet code (Java class)

Aglets

Autonomous

- ▶ Runs in a separate thread when it arrives at a host

Reactive

- ▶ Responds to messages that are sent to it. A message is an object exchanged by by aglets Message passing is synchronous and asynchronous

Unique

- ▶ A globally unique identifier is bound to each aglet

Mobility level = weak mobility

Aglets methods

Creation

- ▶ Create a new aglet in a context and start its execution

Cloning

- ▶ Create a copy of an aglet, with a unique ID

Dispatching

- ▶ Send an aglet to another context, either locally or on a new host

Retraction

- ▶ Pull an aglet from its current context back to the requesting context

Deactivation/ Activation

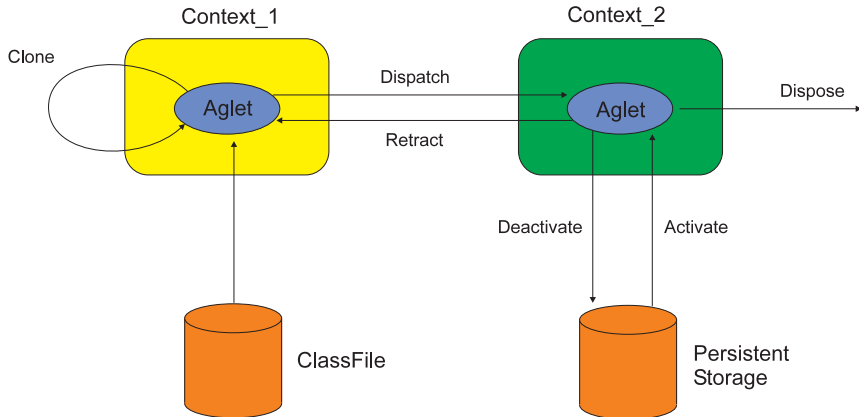
- ▶ Halt and restart the aglet's execution

Disposal

- ▶ Halt aglet execution and remove it from context

Metody podporující migraci

Životní cyklus mobilního agenta



Aglets - příklad

```
import com.ibm.aglet.*;

/**
 * DisplayAglet
 * Displays text in the message area on its aglet viewer.
 */
public class DisplayAglet extends Aglet {

    public void run() {
        setText("Hello,world! I am "+getClass().getName()+".");
    }
}
```

Aglets - příklad

```
public class HelloAglet extends Aglet {
    public void atHome(Message msg) {
        setText("I'm back.");
        dispose();
    }
    public void onCreate(Object init) {
        setMessage("Hello World!");
        itinerary = new SimpleItinerary(this);
        home = getAgletContext().getHostingURL().toString();
    }
    public void sayHello(Message msg) {
        setText(message);
        try {
            itinerary.go(home, "atHome");
        } catch (Exception ex) {ex.printStackTrace();}
    }
}
```

Aglets - příklad

```
/* Starts the trip of this aglet to the destination */
public synchronized void startTrip(Message msg) {

    // get the address for trip
    String destination = (String)msg.getArg();

    // Go to the destination and Send "sayHello" message to
    owner(this)
    try {
        itinerary.go(destination, "sayHello");
    } catch (Exception ex) {
        ex.printStackTrace();
    }
}
}
```

Aktivní síť

Podpora pro rozšiřování funkcí síťových služeb

Aplikační aktivita směrovačů paketů

Rozšiřitelnost funkce pro speciální pakety - Capsules

Mechanismus pro distribuci kódu

Aktivní prvky - směrovače

ANTS - Active Networks Transfer System

Programovatelné směrovače v IP síti

Cache pro kód aplikovaný na procházející data (protokoly)

Zajištění bezpečnosti aplikace kódu

Capsule

ANTS - Active Networks Transfer System

IP header

ANTS header

Payload



Type

identifikace aplikované rutiny

Previous address

adresa aplikované rutiny (pro její získání)

Parameters

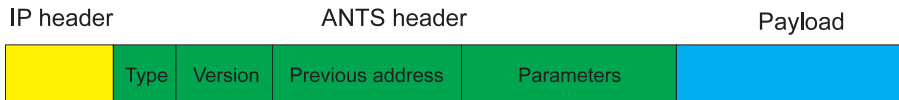
parametry pro zpracování

Payload

hlavička a data vyšších vrstev

Capsule

ANTS - Active Networks Transfer System



Type

identifikace aplikované rutiny

Previous address

adresa aplikované rutiny (pro její získání)

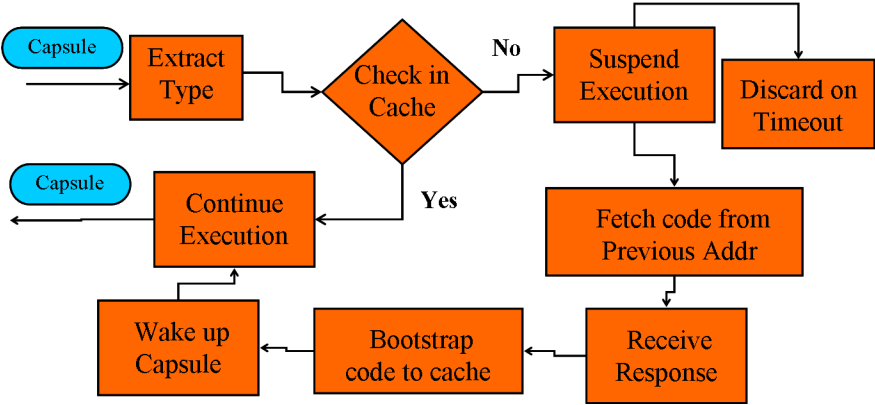
Parameters

parametry pro zpracování

Payload

hlavička a data vyšších vrstev

Distribuce kódu



Aktivní síť

neúspěšná myšlenka v prostředí IP

pomalost zpracování (Java sandbox)

latence při vzdáleném zavádění kódu do cache

bezpečnost

možné aplikace v sensorových sítích