

Middleware

1. cvičení

<http://d3s.mff.cuni.cz>

Department of
Distributed and
Dependable
Systems



CHARLES UNIVERSITY IN PRAGUE

faculty of mathematics and physics

Petr Tůma

tuma@d3s.mff.cuni.cz

Jaroslav Keznikl

keznikl@d3s.mff.cuni.cz

Vojtěch Horký

horky@d3s.mff.cuni.cz

- Cvičení
 - Jednou za 14 dní, celkem 6 cvičení
 - Úterý 17:20 v lichých (sudých kalendářních) týdnech
 - Jaroslav Keznikl, keznikl@d3s.mff.cuni.cz
 - Úterý 17:20 v sudých (lichých kalendářních) týdnech
 - Vojtěch Horký, horky@d3s.mff.cuni.cz
 - Jaroslav Keznikl, keznikl@d3s.mff.cuni.cz
- Web
 - <http://d3s.mff.cuni.cz/teaching/middleware/>
- Mailing list
 - mwy@d3s.mff.cuni.cz
 - <https://d3s.mff.cuni.cz/mailman/listinfo/mwy>

Podmínky udělení zápočtu

- Podrobné podmínky na webu předmětu
- Alespoň **7 bodů** za semestr
 - **2 body** za správně vyřešenou úlohu odevzdanou včas
 - **1 bod** za úlohu odevzdanou včas, ale s chybami
 - Nutno předvést opravenou verzi na cvičení
 - **1 bod** za úlohu odevzdanou pozdě (cca do 4 dnů po deadline)
 - Nutno předvést na cvičení
- Celkově bude **5 úloh**, možnost získat 2 body za každou
- **Alternativní úlohy**
 - Preferovaný způsob řešení
 - Zábavnější než povinný „hello world“ z cvičení
 - Zadání, bodování i termíny **dohodnout s přednášejícím**

- **Po předchozí dohodě**
- Hodnocení 2 až 6 bodů (podle obtížnosti)
- Termín do konce zkouškového období LS
 - Jen ve výjimečných případech později
- Okruhy témat
 - Benchmark (2), sada benchmarků (4 - 6)
 - Rozchození IDL/IIOP testeru s aktuálním software (3 - 6)
 - Zlepšení výkonu (2 - 6)
 - Úloha na základě vlastního námětu

Důležité upozornění

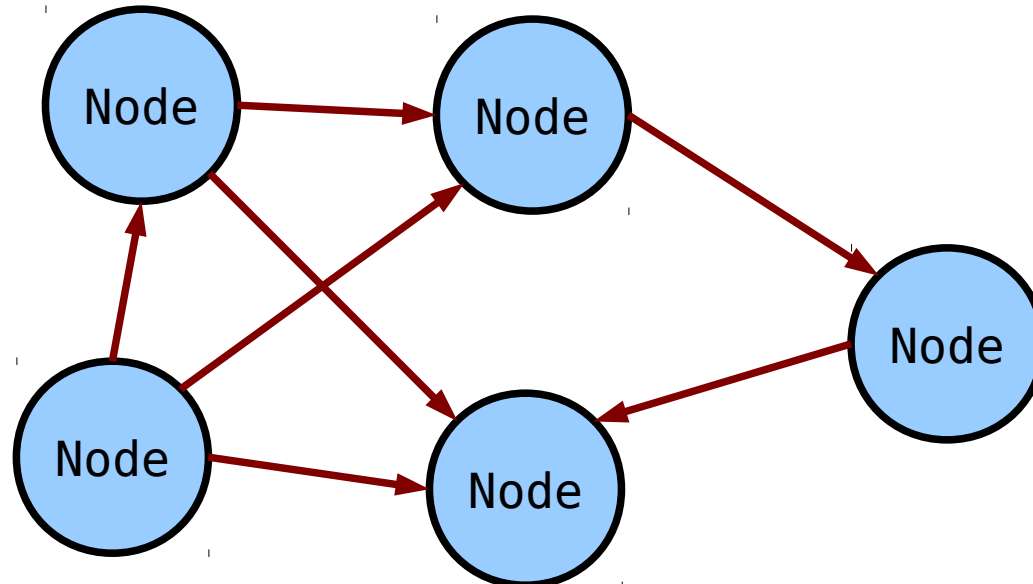
- **Nepodvádějte!**
 - Na úlohách pracujte samostatně
 - Nejedná se o povinný předmět
 - Chcete-li podvádět, dělejte to jinde
 - Nechcete se naučit, jak modifikovat cizí kód
 - Chcete se naučit používat middleware technologie
 - Pokud vám úlohy připadají nudné...
 - ...domluvte se na alternativních úlohách
- Pokus o podvod → nelze získat zápočet

Java RMI

Úloha: Vzdálenost mezi uzly v grafu

```
public interface Searcher
{
    public int getDistance (Node oFrom, Node oTo);
}
```

```
public interface Node
{
    Set<Node> getNeighbors ();
    void addNeighbor (Node oNeighbor);
}
```



1. Lokální implementace (1/3)

<http://d3s.mff.cuni.cz/teaching/middleware/files/as1.zip>

- Verze Javy
 - `java -version`
 - `java-config -s sun-jdk-1.6`
- Eclipse (spustit `eclipse-3.7`, či `eclipse-4.2`)
 - Výhody - autocomplete, nápověda k funkcím, okamžitá kompilace a detekce chyb v kódu
 - Nevýhody - pracnější spouštění s parametry
- Interface `Node` a `Searcher`
 - Třídy `NodeImpl` a `NodeSearcher`
- Spustitelná třída `Main` (`java Main`)
 - Měření rychlosti na náhodném grafu

1. Lokální implementace (2/3)

- Vaše implementace musí být snadno spustitelná
 - Použijte poskytnutou lokální implementaci
 - Nepřidávejte zbytečně package apod.
 - Ujistěte se, že implementace funguje v laboratoři
- Rozšiřujte jeden projekt místo čtyř různých verzí
 - Práce s interface zakrývá rozdílné implementace
 - S Remote interface lze dále pracovat i lokálně
 - Stačí odchylovat výjimky, které stejně nenastanou
 - Např. vzdálený graf je jen další pole `Node []`;
 - Snadno vytvoříte (logicky) stejný lokální i vzdálený
 - Podobně `Searcher`
 - Měřit všechny varianty najednou usnadňuje porovnání
 - Stačí přidat měření a sloupce výsledků do cyklu v `searchBenchmark ()`

1. Lokální implementace (3/3)

- Používejte spouštěcí skripty dle Hello World
 - Předávají důležité parametry
 - Jednoduché Run as.../Application v Eclipse nestačí!
 - Ale po chvíli klikání fungovat bude
- Pročtěte si tyto slajdy
 - Předejdete opakovanému řešení typických problémů
- Hlaste nejasnosti a problémy
 - Včas před odevzdáním
 - Pro dotazy používejte výhradně mailing list
 - Tentýž dotaz může zajímat více lidí

2. Vzdálený Searcher (1/4)

- Rozšíření interface Searcher (viz Example)
 - Interface `java.rmi.Remote`
 - Výjimky metod `java.rmi.RemoteException`
- Vzdáleně přístupný objekt (viz ExampleImpl)
 - Buď – dědění z třídy `java.rmi.server.UnicastRemoteObject` – export zajistí konstruktor předka
 - Nebo – ruční export statickou metodou `UnicastRemoteObject.exportObject(obj)`
 - Neřeší ale sémantiku `hashCode()`, `equals()`, `toString()` – u Searcheru nevadí, je jen jeden

2. Vzdálený Searcher (2/4)

- Spustitelný server (viz ExampleServer)
 - Security Manager (zde *optional*, ale při jeho použití je potřeba také *policy file*! - viz dále)
 - Vytvoření instance (a export) vzdáleného objektu
 - Registrace pomocí `java.rmi.Naming.[re]bind()`
- Překlad (viz skript make) a spuštění
 - `javac` - s Eclipse netřeba, kompiluje sám
 - `rmic`
 - Vytvoří *stuby* vzdáleným objektům
 - Parametry ← názvy tříd implementujících vzdálené objekty
 - Za určitých podmínek není potřebný
 - Klient má třídy k dispozici (zde ano)
 - Třídy dědí `UnicastRemoteObject`
 - Lépe používat vždy - odhalí některé chyby

2. Vzdálený Searcher (3/4)

- Aplikace `rmiregistry` – spustit na pozadí
 - Ideálně z jiného adresáře nebo s prázdnou CLASSPATH
 - Neměla by sama vidět implementační třídy (cesta se předává pomocí `java.rmi.server.codebase`)
 - Obsazený port? – spouštět s jiným číslem portu (>1024)
 - Upravit ve voláních `[re]bind()` a `lookup()` cestu
 - `localhost` → `localhost:1234`
- Spuštění serveru – viz skript `run-server`
 - *Policy file* – potřeba pouze se Security Manager
 - Nastaví `java.rmi.server.codebase`
 - Bez něj nebude registrace do `rmiregistry` fungovat

2. Vzdálený Searcher (4/4)

- Rozšíření Main o RMI (viz ExampleClient)
 - Security Manager (*optional*)
 - Získání reference na vzdálený Searcher
`java.rmi.Naming.lookup(path)`
 - Přidat volání vzdáleného Searcher.getDistance()
s lokálními objekty NodeImpl v metodě searchBenchmark()
 - Jak se server dostane k sousedům předávaných uzlů?
- Spuštění klienta (optional policy file) – viz run-client

3. Vzdálené objekty Node (1/2)

- Rozšíření iface Node o RMI (jako Searcher)
- Třída dědicí z `UnicastRemoteObject` a implementující Node
 - Kvůli sémantice `hashCode()`, `equals()`, `toString()`
 - Jiná třída než lokální `NodeImpl`!
 - Stačí copy/paste a upravit
 - Chceme, aby se lokální Nodes chovaly stále lokálně v předchozí variantě

3. Vzdálené objekty Node (2/2)

- Jak vytvářet a vracet instance na žádost klienta?
- Implementací NodeFactory s metodou **createNode()**
 - Podobně jako vzdálený Searcher – interface s RMI, implementační třída, vytvoření a **Naming.bind()** v rámci **stávajícího** serveru
 - Nevytvářejte další samostatný spustitelný server, pro 4. variantu chceme mít jeden
- Klient získá referenci pomocí **lookup()** a vytváří kromě lokálního grafu také vzdálené objekty Node
- Jak přistupuje lokální Searcher na vzdálené Nodes? Co přesně vrací NodeFactory klientovi?

4. Vzdálený Searcher na vzdálených Nodes

- Vše už máme připraveno, stačí přidat tuto variantu do **searchBenchmark()** a porovnat rychlost
- Jak přistupuje Searcher na serveru k objektům Node na (tom samém) serveru?

5. Vliv rychlosti sítě

- Dopusud jsme spouštěli klient a server na stejném stroji
 - Overhead RMI komunikace, ale bez latence sítě
- Spouštění na více strojích
 - Server na stroji suseda, klient lokálně
 - Upravit cesty ve voláních **[re]bind()** a **lookup()**
 - Název vzdáleného stroje místo localhost
 - Spuštění **rmiregistry** a Serveru v SSH relaci na vzdáleném stroji
 - Spuštění klienta lokálně

6. Kombinace předáváníí hodnotou a odkazem

- Předchozí úkoly řeší extrémní případy
- Možnost kombinace obou přístupů...
- Idea: „dávkový“ přenos větších částí grafu
- `getTransitiveNeighbors(int distance)`
 - Vrátí nejen přímé sousedy, ale sousedy do určené vzdálenosti
- Úprava implementace rozhraní `Searcher`
 - V `getDistance()` se neomezovat na přímé sousedy
 - V každém kroku žádat sousedy do určité vzdálenosti
- Zkusit různé hodnoty parametru `distance`
 - Porovnat naměřené časy s předchozími variantami

- Funkční implementace
- **Dokumentace**
 - Odpovědi na všechny otázky v zadání
 - Popis výsledků měření
- E-mailem na adresu cvičícího (do deadline uvedeného na webu)
- Ujistěte se, že vaše řešení funguje v labu
- Řešení by mělo být snadno (standardním způsobem) spustitelné
 - V opačném případě to může způsobit negativní první dojem ;-)