

## Oponentský posudek bakalářské práce

### Martin Berka: Výkonové testy v Javě v paralelním a distribuovaném prostředí

Hlavním cílem hodnocené BP bylo navrhnout aplikaci pro spouštění testů v paralelním a distribuovaném prostředí. Práce je tedy výrazně prakticky zaměřena.

V teoretické části autor nejprve představuje paralelní prostředí se sdílenou pamětí a prostředky pro synchronizaci. Následně je popsáno distribuované prostředí a možnosti distribuce dat. Dále zde autor představuje sadu algoritmů, jejichž pomocí bude později v praktické části testovat paralelní a distribuovaný běh.

V realizační části autor nejprve specifikuje požadavky na aplikaci a představuje případy užití aplikace. Následně vybírá technologie pro distribuci dat. V další části textu je již popsána samotná implementace systému a algoritmů pro testování. V závěru praktické části jsou přestaveny výsledky testů, které autor uskutečnil.

K práci je přiloženo CD, které obsahuje text diplomové práce, zdrojové texty aplikace, přeloženou aplikaci a příklady konfiguračních souborů.

Práce je dobře strukturovaná. Autor nejprve popisuje dostupné technologie, u nichž hodnotí vhodnost jejich dalšího použití v práci. Také představuje algoritmy, pomocí nichž bude testovat rozdíly mezi paralelním a distribuovaným prostředím. V práci se také vyskytuje minimum chyb, což zlepšuje její čitelnost.

Na autorovi oceňuji, že musel nastudovat větší množství technologií a algoritmů, které pak použil ve své práci. Dále na autorovi oceňuji, že do práce zahrnul UML diagramy, na kterých je vidět navržená architektura programu. Je škoda, že autor nezahrnul do práce i algoritmus, který by byl málo náročný na distribuci dat, ale výrazně časově náročný (např. hledání hamiltonovské kružnice v relativně malém grafu, u kterého bychom věděli, že tam kružnice existuje). V tomto případě by se minimalizovaly nároky na distribuci dat a mohla by být vidět vhodnost použití distribuovaného výpočtu.

Při čtení práce jsem narazil na několik věcných nedostatků menšího významu:

- Při popisu synchronizačního primitiva `monitor` a jeho implementací v Javě autor neuvedl druhou možnost vytvoření monitoru, tj. odvození podmínkové proměnné z proměnné zámku a následné vytvoření monitoru pomocí těchto primitiv.
- U procesu serializace autor neuvedl, že v Javě existuje klíčové slovo `transient`, kterým lze označit proměnné, které nebudou serializovány (např. deskriptory souborů).
- Diagram užití je velice jednoduchý a neobsahuje všechny případy užití. Takto zakreslen znamená, že aplikace má pro uživatele jen jednu funkci. Aplikace jich ale nabízí více (např. výběr paralelní nebo distribuovaný, konfigurace testu).
- V práci autor používá spojení „protokol NIO“ a „protokol RMI“. V obou případech se ale nejedná o protokoly, nýbrž o technologie, které jsou postaveny nad TCP/IP.

Veškerou odkazy na citovanou literaturu vedou k relevantním zdrojům.

K předložené práci mám následující dotaz:

- Z výsledků testů vyplývá, že paralelního výpočty, při kterých bylo použito NIO a RMI mají lepší výsledky než testy, které jste uskutečnil pomocí TCP/IP. Obě tyto technologie ale využívají TCP. Jak si toto vysvětlujete?

Závěrem konstatuji, že diplomant při zpracování BP prokázal jak odpovídající teoretické znalosti, tak i potřebnou implementační zručnost. Práce splňuje zadání ve všech bodech. Všechny mé poznámky k výsledné práci jsou pouze minoritního charakteru. Na základě těchto faktů **doporučuji práci k obhajobě a hodnotím stupněm:**

**Výborně**

V Plzni dne 18. srpna 2014



.....

Ing. Pavel Bžoch