

---

# Optimalizace a ladění výkonu

## Obsah

Principy .....	1
Kdo optimalizuje .....	1
Co se optimalizuje .....	1
Co nás bude zajímat v PV168 .....	2
Optimalizace Javy .....	2
Trendy .....	2
Obecná zlepšení .....	2
Java 6 .....	2

## Principy

### Kdo optimalizuje

- Programátor, vývojář
- Správce nasazení
- Správce cílového (operačního) systému
- Poskytovatelé middleware (aplikačních serverů), databází apod.
- Producenti knihoven
- Dodavatelé JDK (překladač), JRE (Core API, JVM)
- Návrháři procesorů, paměti a výpočetních architektur vůbec

### Co se optimalizuje

Aplikační architekt a programátor	vlastní kód aplikace, především jeho architekturu, příp. i kód na mikroúrovni vč. doprogramování kódu nativního
Správce nasazení	ve vazbě na běhové prostředí (HW platformu, operační systém, aplikační server atd.) navrhuje vhodnou architekturu nasazení vč. distribuce do svazků (cluster), na "cloud" apod.
Správce cílového (operačního) systému	ladění HW konfigurace, operačního systému, volby při spuštění javové aplikace (paměťové limity, počty vláken), bezpečnostní nastavení (Security Manager)
Poskytovatelé middleware	middleware (např. aplikační server) do značné míry určuje celkovou architekturu řešení, sám poskytuje mnoho služeb, kde výpočty tráví hodně času
Producenti knihoven	totéž, výpočty mohou být v kódu knihoven převážnou část doby běhu, zde má smysl optimalizovat i na mikroúrovni; navíc jsou

	většinou používány jako black-box a přehlednost kódu není absolutní prioritou
JDK, JRE	překladač a běhové prostředí Javy jsou enormně důležité - podpora cílové platformy (64 bitů, vícejádrové a víceprocesorové stroje, NUMA architektury paměti...); řadu optimalizací lze provést při překladu do bajtkódu, zbytek pak při spouštění (JIT) a běhu na cílové platformě
Procesory, paměti, výpočetní architektury	64 bitů, vícejádrové a víceprocesorové stroje, NUMA

## Co nás bude zajímat v PV168

Aplikační architekt a programátor	-> ano
Správce nasazení	-> spíše pro PA165 a ani tam se nestihne
Správce cílového (operačního) systému	-> mimo rozsah PV168, PA165
Poskytovatelé middleware	-> PA165
Producenti knihoven	-> ano
JDK, JRE	-> ano
Procesory, paměti, výpočetní architektury	-> mimo rozsah PV168, PA165

## Optimalizace Javy

### Trendy

- Sun směřuje k optimalizaci i pro desktopy a RIA
- Hlavní doménou zůstávají "serverové" JVM

### Obecná zlepšení

- Oba garbage collectors se vylepšují.
- **Parallel Scavenger** podporuje nyní *Improved NUMA* (Non Uniform Memory Access) *Support* pro moderní počítačové architektury (např. víceúrovňové cache paměti)
- **HotSpot JIT**: Eliminace zbytečných převodů mezi celočíselnými typy
- Optimalizace obvyklých postupů při zřetězování

## Java 6

Změny v nejnovější *Java 6 Update 18*

garbage collection                      vylepšený sběr nepotřebných objektů

classloading	rychlejší zavádění a inicializace tříd
rychlejší verifikace kódu	při teplých startech JavaFX
start aplikací s GUI	použitím Direct 3D
rychlejší Java Web Start	souběžné stahování JAR balíků při aktualizacích



### **Poznámka**

Blíže viz Sun Releases Java 6 Update 18 With Significant Performance Improvements and Windows 7 Support [<http://www.infoq.com/news/2010/01/java6u18>]