

# Ako dokázať viac za menej

Rastislav Pečák  
Softec

# Ako sa dá zvýšiť výkon aplikácie?

## ▪ Škálovateľnosť

- „Schopnosť **udržať si** výkonnostnú úroveň po pridaní ďalších procesorov“ (jadier, počítačov) a „... nielen udržať, ale využiť **celý potenciál prostredia**“

## ▪ Posilnenie hardware

- Horizontálne škálovanie - zvýšenie počtu procesorov / jadier
- Vertikálne škálovanie - paralelný beh na viacerých počítačoch

## ▪ Optimalizácia aplikácie

## ▪ Použite nových nástrojov a techník

# Problémy zvýšenia výkonu JEE aplikácie

- **Plytvanie zdrojmi**
  - Pre zvýšenie výkonu rozširujeme infraštruktúru
- **Nákladný vývoj na dosiahnutie škálovateľnosti**
  - Neefektívny **synchronny** paralelizačný model
- **Drahá udržiavateľnosť**
  - Nedostatočná adaptabilita na zmenu infraštruktúry



# Problémy zvýšenia výkonu JEE aplikácie

## ■ Plytvanie zdrojmi

- Pre zvýšenie výkonu rozširujeme infraštruktúru

■ **Jazyk Java neposkytuje vhodný paralelizačný model**



■ **Lepší paralelizačný model = Scala (scalable language) a Closure**

# Ako dosiahnuť viac?

*„What are you waiting for? You're faster than this.  
Don't think you are, know you are.“*

The Matrix

- Využiť zdroje a infraštruktúru **naplno** použitím správneho, moderného prístupu



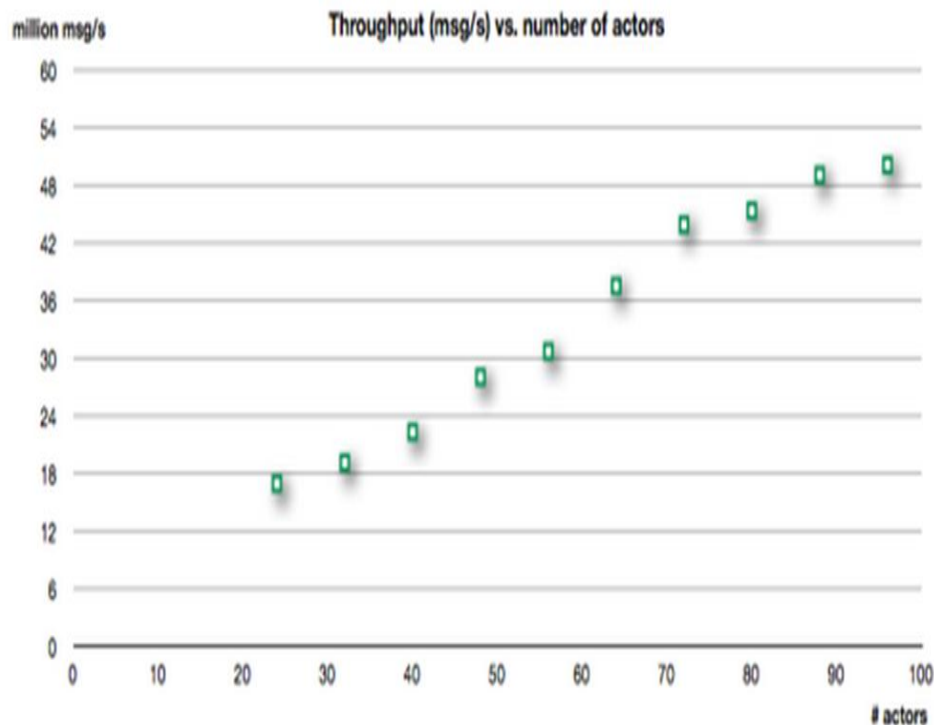
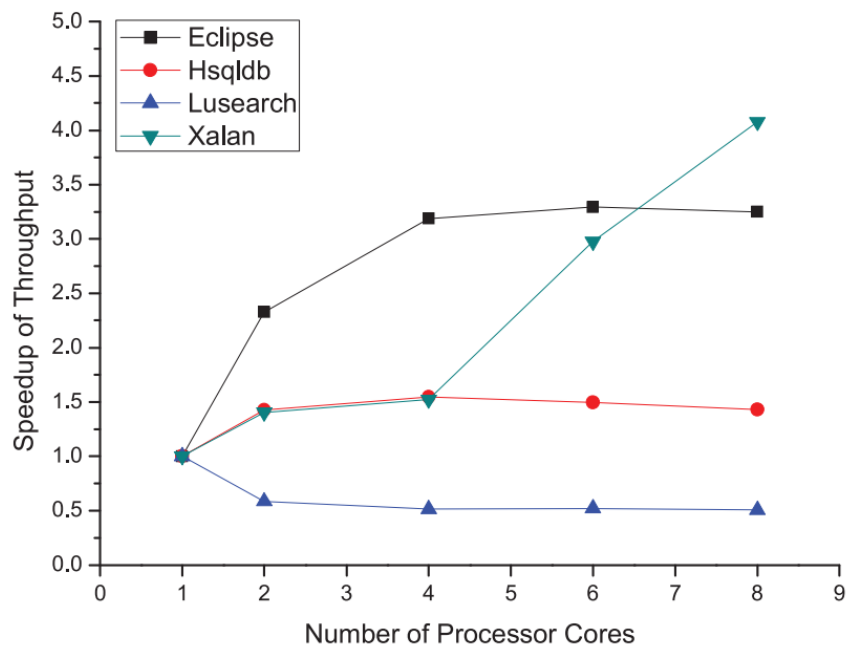
Asynchrónny event-based  
paralelizačný model

[www.akka.io](http://www.akka.io)

- toolkit pre písanie paralelných aplikácií
- **efektívnejší paralelizačný model** jazyka Scala
  - Nepoužíva zdieľané zdroje
  - Používa štruktúry actors a futures
- middleware bežiaci na infraštruktúre
- knižnica, ktorá je súčasťou aplikácie



# Škálovateľnosť Akka.io a Java aplikácií



# Akka.io – asynchrónny paralelizačný model

## ■ Škálovateľnosť

- na úrovni CPU
- na úrovni infraštruktúry

## ■ Efektívnejší vývoj

- efektívne odchyťovanie chýb
- abstrakcia nad vláknami
  - efektívnejšie písanie business funkcionality
- **Java/Scala API**





## Využívanie zdrojov

- Lacnejší vývoj a udržiavanie
- Lepšia horizontálna a vertikálna škálovateľnosť oproti JEE
- Prípady použitia
  - Transakčné spracovanie
  - Request ( - response) Service
  - Batch processing
  - Complex Event Stream Processing



◦ Efektívnejšie písanie business funkcionality

- API pre Javu

## Dosiahneme **viac** ...

- Rýchlejšiu odozvu
- Viac obslužených klientov

## ...za **menej**...

- Hardvéru
- Licencií
- Úsilia
- => Nákladov

## ...pomocou **moderných technológií**

- Akka.io