# DON'T PANIC

## HITCHHIKER'S GUIDE TO THE JVM MONITORING

# O CZYM BĘDZIEMY MÓWIĆ

- Gdzie możemy napotkać problemy
- Jak diagnozować przyczyny problemów
- Jakich narzędzi użyć
- Jak ich użyć

# KTO BĘDZIE MÓWIĆ?

# GDZIE MIESZKAJĄ PROBLEMY?

# GDZIE MIESZKAJĄ PROBLEMY?

Niewydajne algorytmy

Niewłaściwe kolekcje


Wycieki pamięci

Niewydajny model danych

# GDZIE MIESZKAJĄ PROBLEMY?

Pule wątków

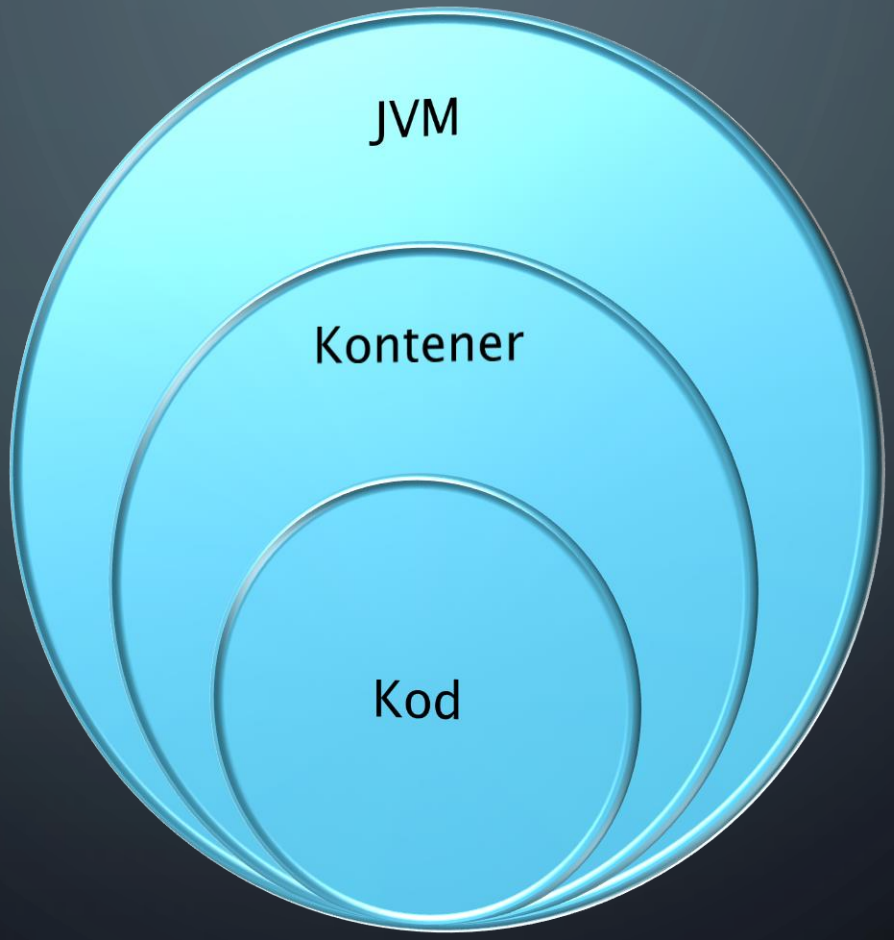
Pule połączeń


Cache

# GDZIE MIESZKAJĄ PROBLEMY?

GC i zarządzanie pamięcią

Lock'i i synchronizacja



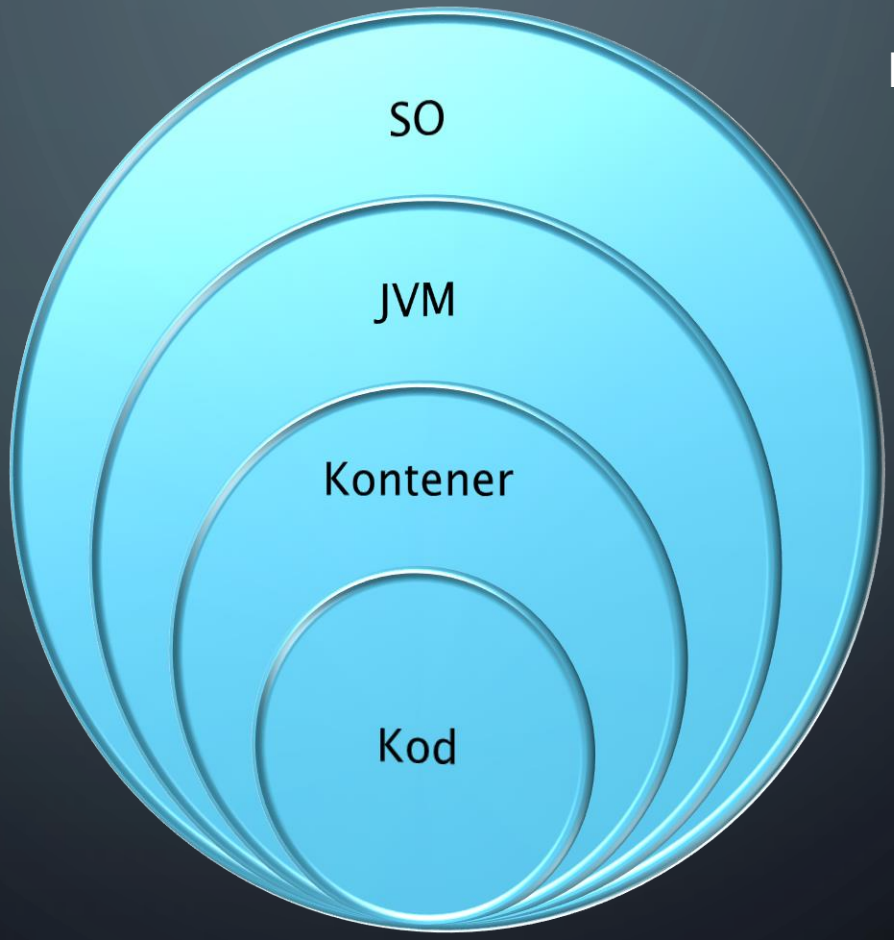

JIT

# GDZIE MIESZKAJĄ PROBLEMY?

Inne aplikacje

I/O

Konfiguracja pamięci

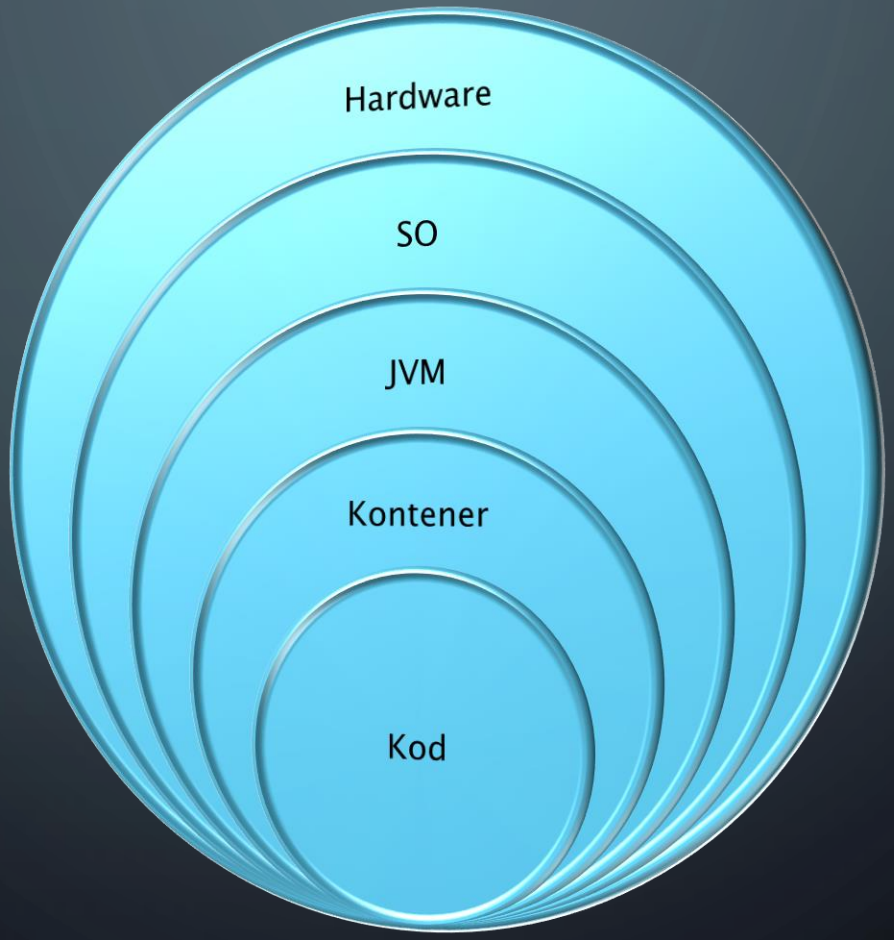


# GDZIE MIESZKAJĄ PROBLEMY?

Rzeczywista ilość zasobów

Wydajność dysków

Wydajność i latency sieci



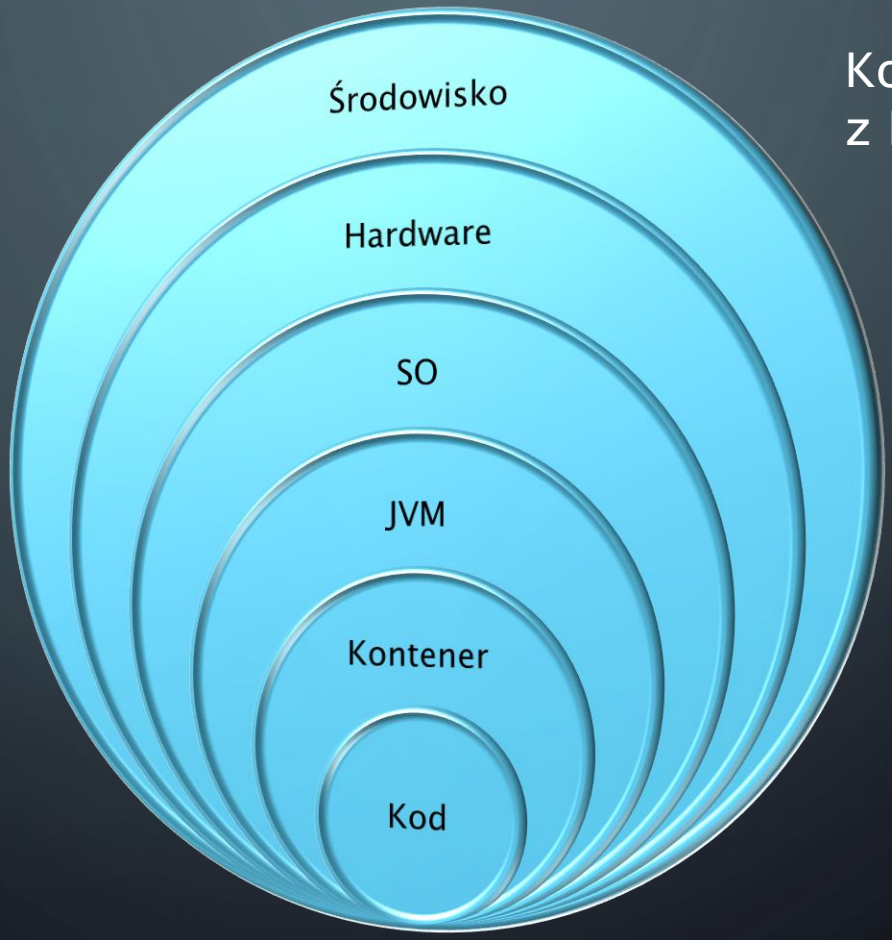

Cache procesora

Wydajność pamięci

# GDZIE MIESZKAJĄ PROBLEMY?

Komunikacja z systemami zewnętrznymi

Komunikacja z IDM




Komunikacja z bazą danych

Komunikacja z przeglądarką

# WNIOSKI

- Szukanie problemów w kodzie jest jak szukanie igły w stogu siana.
- Warto sięgnąć po narzędzia które poprowadzą nas dokładnie do źródła problemu.
- Czasami problemy nie wynikają z kodu a z konfiguracji całego otoczenia aplikacji.

# NARZĘDZIA

- Monitorowanie systemu operacyjnego
  - top
  - vmstat
  - iostat
  - iotop
  - strace
  - pidstat

# NARZĘDZIA

- Monitorowanie JVM:

- Pamięć:
  - jmap
  - jhat
  - Eclipse MAT
- GC:
  - jstat
  - gcviewer
- Kod:
  - jstack
  - TDA
  - Profilery

- Ogólne:
  - jcmd
  - jinfo
  - jconsole
  - jvisualvm
  - JMission Control
- APM:
  - Dynatrace
  - AppDynamics
  - Plumbr

# CZAS NA …

# DEMO

# PODSUMOWANIE

- Problemy nie wynikają jedynie z kodu.
- Szybka identyfikacja przyczyn problemów wymaga sięgnięcia po właściwe narzędzia.
- W razie problemów sprawdź:
  - Czy to problem SO (wysoki system time, wysoki iowait)? - <vmstat, iostat, top>
  - Czy to problem z GC i pamięcią ( częste FGC, wysokie zużycie procesora, freeze aplikacji, OOM )? - <jstat, top, vmstat, gcviewer, jvisualvm, jmap, mat>
  - Czy to problem z Lockami ( duża ilość zblokowanych wątków w zrzucie stosu wątków, niskie zużycie procesora )? - <jstack, thread dump analyzer, jvisualvm, top, vmstat>
  - Gdzie system spędza najwięcej czasu? ( profiler, zrzut stosu wątków ) <jstack, , jtop, jvisualvm, jprofiler, yourkit>

# DODATKOWE INFORMACJE

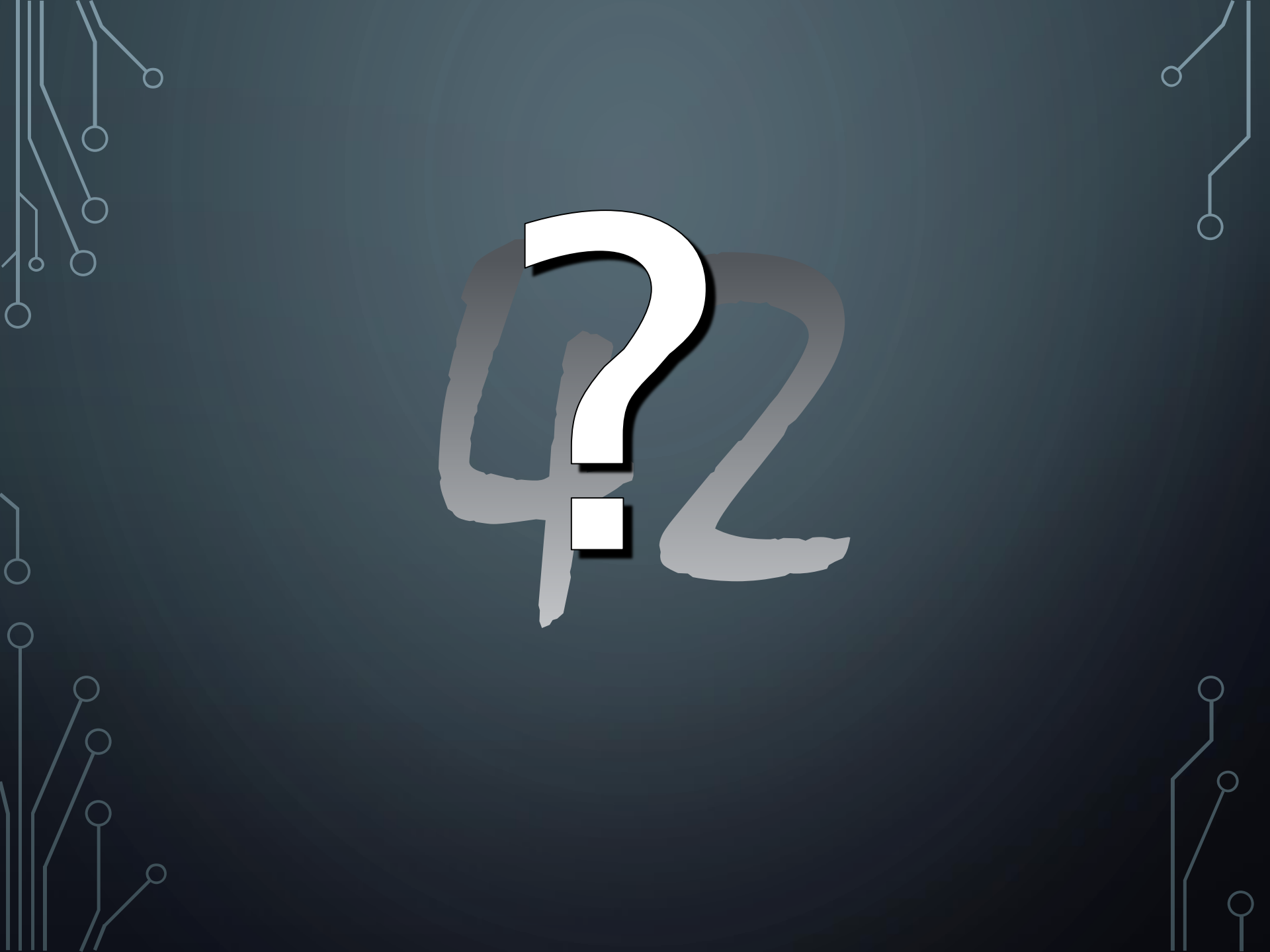
- Kirk Pepperdine – „Live Java Performance Troubleshooting”

https://vimeo.com/35387464

- Leonid Igolnik – "The dark art of performance tuning or how to become a performance hero ..."

https://www.youtube.com/watch?v=dqg0R3gYGac

- Scott Oaks – „**Java Performance: The Definitive Guide** – Getting the Most Out of Your Code”



# SO LONG AND THANKS FOR ALL THE FISH