



RĪGAS TEHNISKĀ UNIVERSITĀTE
DATORZINĀTNES UN INFORMĀCIJAS TEHNOLOĢIJAS FAKULTĀTE
DATORVADĪBAS, AUTOMĀTIKAS UN DATORTEHNIKAS INSTITŪTS

JAVA PERSISTENCE SPACE4J

Vitālijs
Hodiko



IEVADS



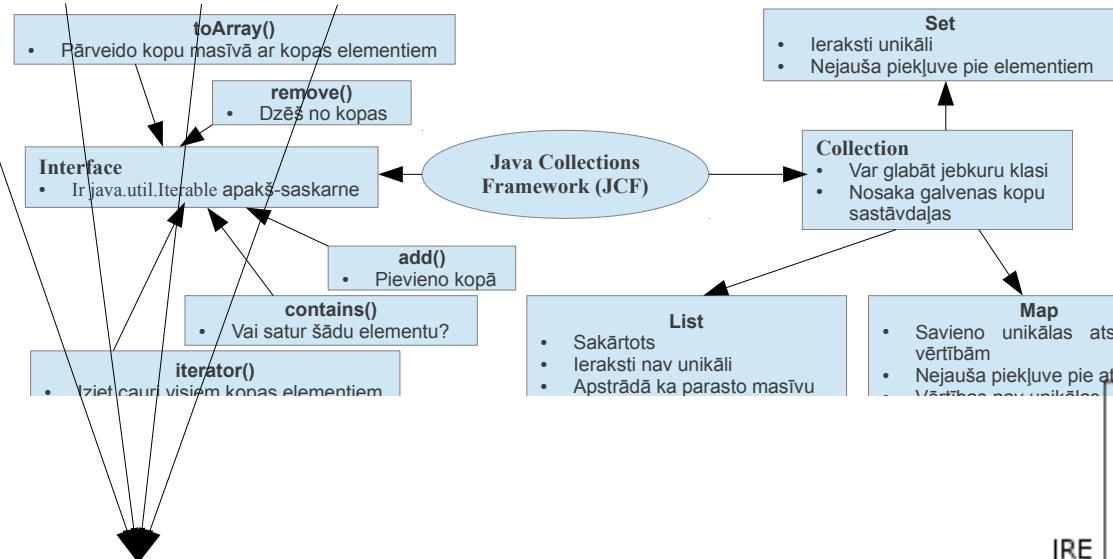
levads

James Gosling, Java radītājs



(vairāku platformu)

WORA



Java source files
(.java)

```
class Foo {
    /* ... */
}
```

javac

Java bytecode files
(.class/.jar)

```
...
iconst_0
iload
istore_1
jsr 19
iconst_0
iload
...
```

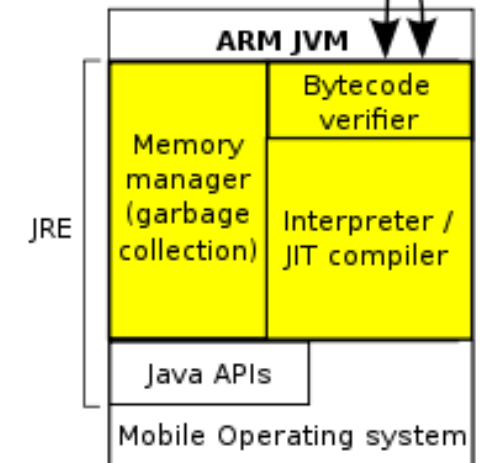
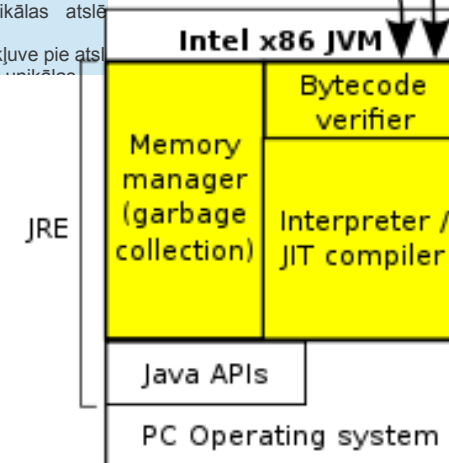
Python source files
(.py)

```
def f(x):
    print x
...
```

jython

Java bytecode files
(.class/.jar)

```
...
istore_1
iload_1
jsr 19
iconst_0
iload
...
```



VĒSTURE



Vēsture

- Andrew D. Birrell, Michael B. Jones, Edward P. Wobber. A Simple and Efficient Implementation for Small Databases, 1987
- The Prevalyer - Object Prevalence by Klaus Wuestefeld, realizācija Java valodā

database

id	name	lastname	birthdate	name	birthdate
7700	Jackson	Stephen	11-08-1987	John	11-08-1987
7701	Reynolds	Randy	01-04-1988		11-01-1981
7702	Armstrong	Stephen	05-10-1989		11-05-1987
7703	Jackson	Linda	19-09-1989		10-09-1986



JAUDĪGS

ĀTRS

KAS PREVALENTA IR SISTĒMA ! ?



Kļūdu noturīgums
Caurskatāmība
Ātrums

...

ATOMICITY

CONSISTENCY

ISOLATION

DURABILITY

Īpašības

~~SQL~~

~~ATSKAITES~~

~~RDBMS~~

~~LATENCY~~

RAM(!?)

Efektīva programmēšana

Pareiza datu organizēšana

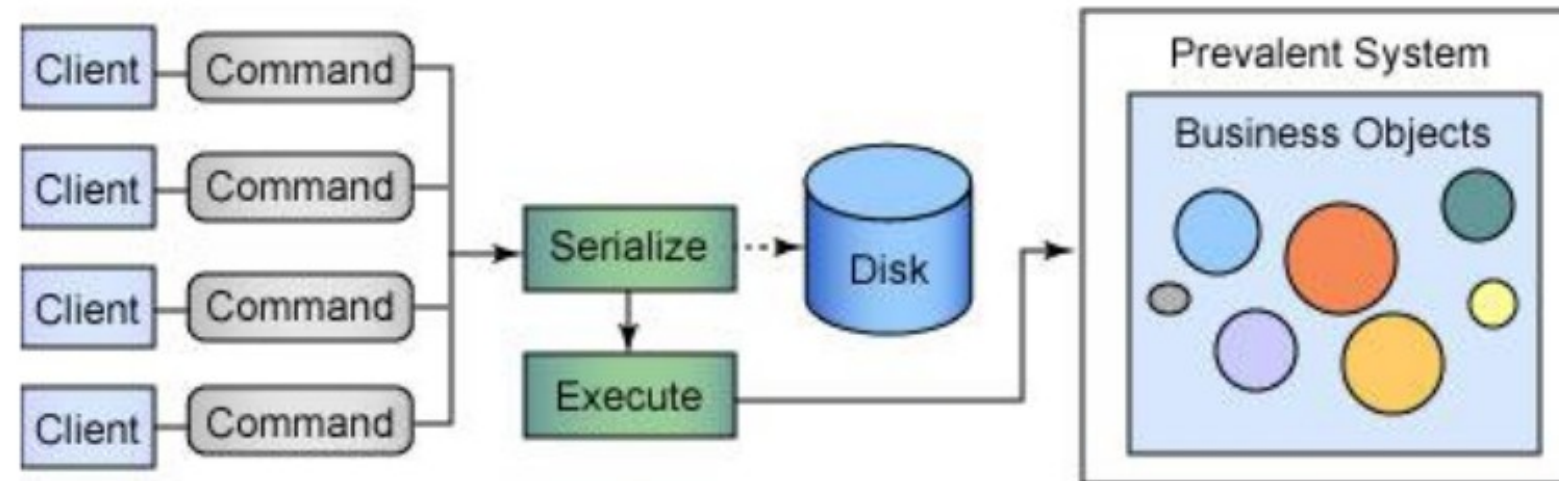
PREVALENTA SISTEMA
BIZNESA OBJEKTI

PRIEKŠNOSACĪJUMI

KOMANDA



Priekšnosacījumi



- Sistēmu prevalence ir pēc būtības vienkārša un to var īstenot jebkura OO valoda, tikai, ja tā un pamata sistēma atbalsta operāciju secīgu izpildi.
- Sistēma pārvalda pilnība savus datus
- Datu apjoms ir samēra mazs un lasīšanas īpatsvars ir lielāks salīdzinot ar rakstīšanu

KĀPĒC

KĀ

IEGUVUMI

SPACE4J



leguvumi

“...biznesa sistēma, kura spēj apstrādāt **6 miljonu operāciju sekundē** vienā pavediena.” **Martin Fowler**(pasaules mēroga lektors un autors)

leguvumi

Indeksācija

NOLASĪT (**Biznesa**
obj)



leguvumi



Dati
netiek
nozaudēti
ārkārtas
situācijās

*(servera uzlaušana vai ugunsgrēks neskaitas)

leguvumi

“Space4J is like Oracle: **Updates only block updates.
Reader don't block or get blocked by anything.”**

Kāpēc

“

You can **EASILY DISTRIBUTE** with your applications.

You work with Java collections and NO SQL. There is no relational-to-object bridge.

Read access is **SUPER-FAST**, if you care for that.

”

You can easily cluster it.

Sergio Oliveira Jr. - SPACE4J aktīvs izstrādātājs un autors

Kā

Video goes here

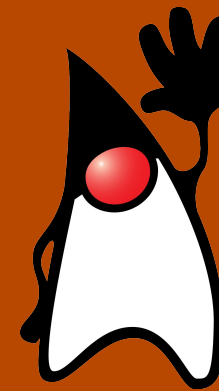
Zināmie pielietojumi

- **Lehman Brothers Holdings Inc.** - izmantoja RAPTOR sistēmā(Akciju pasūtījumu pārvaldības sistēma)
- **Blizzard Entertainment** – izmanto Starcraft RST spēlē
- Un daudz citu piemēru....



**VIENKĀRŠI
RĪKI** + **EFEKTĪVI
RISINĀJUMI** = **LIELISKI
REZULTĀTI**

Paldies!





RĪGAS TEHNISKĀ UNIVERSITĀTE
DATORZINĀTNES UN INFORMĀCIJAS TEHNOLOĢIJAS FAKULTĀTE
DATORVADĪBAS, AUTOMĀTIKAS UN DATORTEHNIKAS INSTITŪTS

JAVA PERSISTENCE SPACE4J

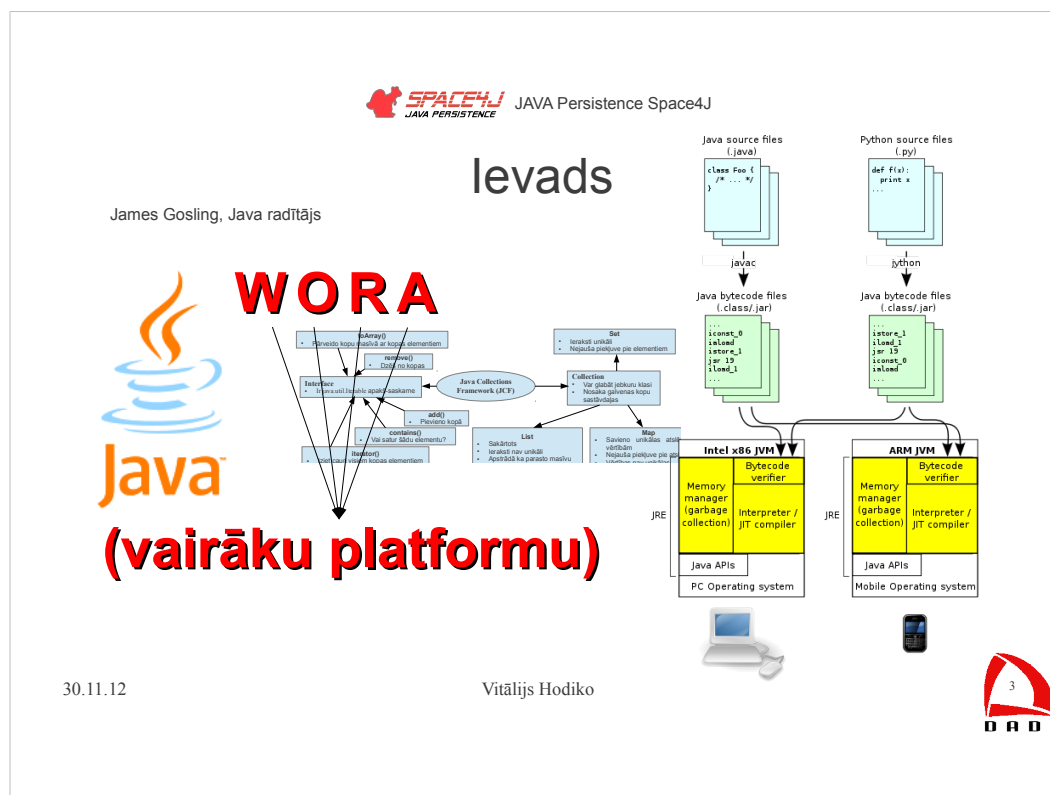
Vitālijs
Hodiko





IEVADS





DUKE, Java talismans

Pirms sākt runāt par prevalentam sistēmām es gribu atgādināt Java valodas piecus galvenos veidošanas mērķus:

- Vienkārša, OO, pazīstama (sintakse ir liela mēra pārņemta no C++)
- Stabila un droša
- Arhitektūras neitrāla un pārnēsājama
- Interpretējama, paralēli izpildāma, dinamiska
- Augsta veikspēja (Java 7 ir 1,5x lēnāka par C)
-
- ```
public class OddEven {
 • public OddEven() {}
 • public static void main(final String[] args) {}
}
```



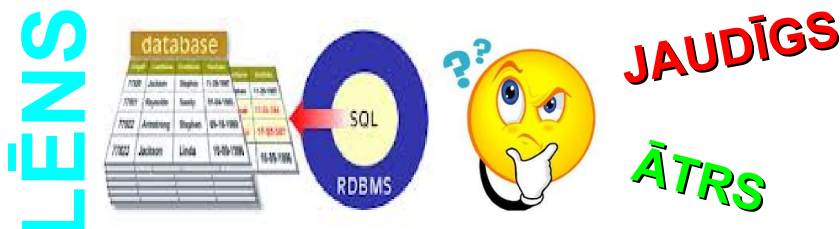
# VĒSTURE





## Vēsture

- Andrew D. Birrell, Michael B. Jones, Edward P. Wobber. A Simple and Efficient Implementation for Small Databases, 1987
- The Prevalyer - Object Prevalence by Klaus Wuestefeld, realizācija Java valodā



30.11.12

Vitalijs Hodiko



Ka ir zināms ir divu veidu lietas: LIELAS un mazas.

Ka arī, ka strādāt ar rīkiem, kas ir domātas LIELAM lietam, ir pārmērība, jā tādas nav vajadzīgas.

Pienāca laiks un cilvēki saka domāt: kā izveidot ātru un jaudīgu datu pārvaldības sistēmu, kura tajā pašā laikā saglabās relāciju datubāzes pozitīvas iezīmes.


Pirmo reizi ideja par prevalentu sistēmu bija izteikta šajā darbā (Viegla un efektīva mazu datu bāzu realizācija). Sākotnējā jēdziena aprakstīja datu glabāšanu atmiņā, bet nekas nebija teikts par datu struktūru.

Prevalent[prevelent]dominējošs, izplatīts

Un mēs zinām ka Ram atmiņa ir vairākas reizes ātrāka par HDD diskem

Vēlāk modelis bija piemērots objekt-orientētas sistēmas un nosaukts par Objektu Prevalenci. To paveica Klaus Wuestefeld, The Prevalyer projekta dibinātais, modeļa implementācija Java valodā.

Prevalences hipotēze - prevalentai sistēmai jābūt pietiekami liela operatīva atmiņā, lai tā varētu glabāt sistēmas stāvokli. Prevalences aizstāvji apgalvo ka tas tiek nepatraukti atvieglots samazinot RAM cenu, vel jo vairāk jo biznesa datubāzes jau ir pietiekami mazi, lai ietilptu atmiņā.

 JAVA Persistence Space4J


ICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY  
STENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY  
ATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION  
BILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY

**KAS PREVALENTA  
IR SISTĒMA !?**

ICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY  
STENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY  
ATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION  
BILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY  
ICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY ATOMICITY  
STENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY CONSISTENCY  
ATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION ISOLATION  
BILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY DURABILITY

30.11.12

Vitālijs Hodiko



Tātad, kas tad īsti ir prevalenta sistēma?

- Tā uztur visus objektus atmiņā
- Jebkurai operācijai kas rada iespaidu uz stabilu stāvokli jābūt secīgam komandas objektam. Kad šī komanda tiek izpildīta, tā secīga forma tiek ierakstīta žurnālā. Un kad sistēmu pārstartē, tā atjauno objektu stāvokļus izpildot žurnāla ierakstītas komandas.
- Var saglabāt pilnu sistēmas stāvokli (tā saucamais “snapshot” vai “momentuzņēmums”), tā, lai pārstartējot sistēmu tā būtu up-to-date

**A** T O M I C I T Y  
**C** O N S I S T E N C Y  
**I** S O L A T I O N  
**D** U R A B I L I T Y

SPACE4J JAVA PERSISTENCE Space4J

Īpašības

SQL

ATSKAITES

RDBMS

LATENCY

Kļūdu noturīgums  
Caurskatāmība  
Ātrums  
...

ATOMICITY  
CONSISTENCY  
ISOLATION  
DURABILITY

RAM(!?)

Efektīva programmēšana

Pareiza datu organizēšana

30.11.12

Vitalijs Hodiko

7

Jim Gray - Definēja un attīstīja tehnoloģiju 1970 gados

Saglabā ACID(atomicity, consistency, isolation, durability - garantē, ka datu bāzes operācijas tiek apstrādātas ticami) īpašības

Momentuzņēmumi un žurnālēšana – durability

Secīga komandu izpilde – isolation

Consistency – par to atbild izstrādātājs

Atomicity – par to arī atbild izstrādātājs

+Atomitāte - katrs darījums ir “viss vai neko”: ja daļa no darījuma bija neveiksmīga, tad viss darījums ir neveiksmīgs un datubāzes stāvoklis nemainās. Tai jānodrošina atomitāti jebkura situācija, arī sistēmas sabrukuma, kļūmes, strāvas zuduma gadījumos. (the developer, who has to implement the commands in atomic fashion)

+Saskaņotības īpašība nodrošina, ka jebkurš darījums izraisīs datubāzes pāreju no viena stāvokļa uz citu. Jebkuru datu ierakstīšana jābūt spēka esošas saskaņas ar visiem noteikumiem, ieskaitot, bet ne aprobežojoties ar ierobežojumiem, kaskadem, triggeriem un jebkuriem thereof kombinācijām.

+Izturība nozīmē, ka tiklīdz darījums ir noticis, un tā tam arī jāpaliek jebkuras situācijās ietekmē. (ugunsgrēks, elektrības zudums, sistēmas kļūme un t.t.)

+Visi dati glabājas operatīvajā atmiņā, kas nodrošina atraku piekļuvi salīdzinājuma ar cieto disku.

-Programmētajam jāmaksā strādāt ar atmiņu.

-Taču ar to nepietiek, arī izstrādātajam jāorganizē pareizi datus un jāraksta efektīvu kodu, lai sasniegtu max efektivitāti. Ar to palīdz heša optimizācijas tabulas, kuras izmantošana nozīmīgi palielina lasīšanas operāciju ātrumu salīdzinājuma ar RDBMS.

-Nav nepieciešamības pēc SQL pieprasījumiem un atskaitēm (jo to visu pievada programmēšanas valoda)- datu apstrāde un attēlošana ir aprobežota tikai ar valodas iespējām

•Viena no fundamentālām problēmām ir momentuzņēmumu saskaņotība



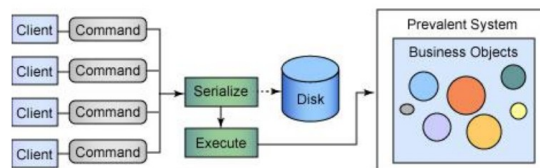
PREVALENTA SISTEMA  
BIZNESA OBJEKTI

# PRIEKŠNOSACĪJUMI

KOMANDA



## Priekšnosacījumi



- Sistēmu prevalence ir pēc būtības vienkārša un to var īstenot jebkura OO valoda, tikai, ja tā un pamata sistēma atbalsta operāciju secīgu izpildi.
- Sistēma pārvalda pilnība savus datus
- Datu apjoms ir samēra mazs un lasīšanas īpatsvars ir lielāks salīdzinot ar rakstīšanu

Prevalentas sistēmas modelis sastāv no trim blokiem:

- Biznesa objekts – secīgas izpildes un deterministisks (izpildot vienu un to pašu komandu rezultāts nemainas)
- Komanda – secīgas izpildes komanda sadarbojas ar prevalentu sistēmu
- Prevalenta sistēma – saglabā sistēmas kopijas un komandu žurnālu

 SPACE4J  
JAVA PERSISTENCE

JAVA Persistence Space4J

KĀPĒC  
KĀ  
IEGUVUMI

**SPACE4J**





30.11.12

Vitālijs Hodiko

10

Tagad es jums pastāstīšu, pirmkārt, kāpēc ir jāizvēlas tieši šī bibliotēka un kāpēc vispār kaut ko ir jāmaina

Otrkārt, es atbildēšu uz jautājumu KĀDS ir jūsu ieguvums

Un, protams, es jums parādīšu kā var uzinstalēt un izmantot šo bibliotēku

## leguvumi


**“...biznesa sistēma, kura spēj apstrādāt 6 miljonu operāciju sekundē vienā pavediena.”** Martin Fowler (pasaules mēroga lektors un autors)

30.11.12

Vitālijs Hodiko



Saglabāt datus dabiska, valodai raksturīga formāta ir daudz ātrāk un programmētājam draudzīgāk nekā vairāki pārveidojumi, kas ir nepieciešami, lai saglabāt un iegūt datus no DBMS datubāzēm. Martin Fowler ir autors un starptautiska mēroga lektors programmatūras izstrādē, kas specializējas objektorientētā analīze un plānojuma, UML, modeļu un elastīga programmatūras izstrādes metodoloģija, tostarp ekstremālā programmēšanā.


 **SPACE4J** JAVA Persistence Space4J

leguvumi

Indeksācija →

**NOLASĪT (Biznesa  
obj)**

30.11.12 Vitālijs Hodiko

 12

## NOLASĪT NO OBJEKTA DATU STRUKTŪRAS IR EFEKTĪVI

Space4j ir apgādāts ar viegli izmantojamo indeksēšanas struktūru. Tas nozīmē, ka jūs varat izmantot četrus pilnīgi atšķirīgus indeksēšanas veidus, lai meklētu nepieciešamos datus jebkura virziena.



## Ieguvumi



30.11.12

Vitalijs Hodiko

\*(servera uzlaušana vai ugunsgrēks neskaitās)



Laiku pa laikam jūsu sistēma/lietojumprogrammatūra var veikt datu saglabāšanu(momentuzņēmons). Tā kā Space4J tur visus datus Space objekta, tad veicot momentuzņēmonu objekts tiek serializēts un pēc tam saglabāts cietā diskā. Tādēļ pēc pārstartēšanas tiek atjaunotas komandas kopš pēdējas datu saglabāšanas

Pirmkārt, mums jānoskaidro dažas lietas. Ko nozīmē “pārāk milzīgs” jūsu sapratnē. Vai tas ir 1MB, 1GB, 1TB? Cik ātri cietais disks var saglabāt 1GB datu uz tā? Un vai tik tiešām jums ir vajadzīga tik milzīga datubāze? Lielākai lietojumprogrammu daļai tā nav. Piemēra pēc: ja jūsu lietotāja objekts ir 128 baitu liels un jūsu datubāze ir 1 miljons lietotāju, tad tas aizņems mazāk par 130MB operatīvas atmiņas.



JAVA Persistence Space4J

## leguvumi

**“Space4J is like Oracle: Updates only block updates.  
Reader don't block or get blocked by anything.”**

30.11.12

Vitalijs Hodiko

\*(servera uzlaušana vai ugunsgrēks neskaitās)



Laiku pa laikam jūsu sistēma/lietojumprogrammatūra var veikt datu saglabāšanu(momentuzņēmons). Tā kā Space4J tur visus datus Space objekta, tad veicot momentuzņēmonu objekts tiek serializēts un pēc tam saglabāts cietā diskā. Tādēļ pēc pārstartēšanas tiek atjaunotas komandas kopš pēdējas datu saglabāšanas

Pirmkārt, mums jānoskaidro dažas lietas. Ko nozīme “pārāk milzīgs” jūsu sapratnē. Vai tas ir 1MB, 1GB, 1TB? Cik ātri cietais disks var saglabāt 1GB datu uz tā? Un vai tik tiešām jums ir vajadzīga tik milzīga datubāze? Lielākai lietojumprogrammu daļai tā nav. Piemēra pēc: ja jūsu lietotāja objekts ir 128 baitu liels un jūsu datubāze ir 1 miljons lietotāju, tad tas aizņems mazāk par 130MB operatīvas atmiņas.



## Kāpēc

“ You can **EASILY DISTRIBUTE** with your applications.

You work with Java collections and NO SQL. There is no relational-to-object bridge.

Read access is **SUPER-FAST**, if you care for that.

”

You can easily cluster it.

Sergio Oliveira Jr. - SPACE4J aktīvs izstrādātājs un autors

30.11.12

Vitālijs Hodiko



Forums ir dzīvs un pats autors atbild uz visiem jautājumiem un ir gatavs palīdzēt ja rodas problēmas.



# Kā

Video goes here

30.11.12

Vitālijs Hodiko





## Zināmie pielietojumi

- **Lehman Brothers Holdings Inc.** - izmantoja RAPTOR sistēmā(Akciju pasūtījumu pārvaldības sistēma)
- **Blizzard Entertainment** – izmanto Starcraft RST spēlē
- Un daudz citu piemēru....



JAVA Persistence Space4J

VĒLREIZ APKOPOJOT

**VIENKĀRŠI  
RĪKI**    **+**    **EFEKTĪVI  
RISINĀJUMI**    **=**    **LIELISKI  
REZULTĀTI**

30.11.12

Vitālijs Hodiko





# Paldies!





- +Atomitāte - katrs darījums ir “viss vai neko”: ja daļa no darījuma bija neveiksmīga, tad viss darījums ir neveiksmīgs un datubāzes stāvoklis nemainās. Tai jānodrošina atomitāti jebkura situācija, arī sistēmas sabrukuma, kļūmes, strāvas zuduma gadījumos. (the developer, who has to implement the commands in atomic fashion)
- +Saskaņotības īpašība nodrošina, ka jebkurš darījums izraisīs datubāzes pāreju no viena stāvokļa uz citu. Jebkuru datu ierakstīšana jābūt spēka esošas saskaņa ar visiem noteikumiem, ieskaitot, bet ne aprobežojoties ar ierobežojumiem, kaskadem, trigeriem un jebkuriem thereof kombinācijām.
- +Izturība nozīmē, ka tiklīdz darījums ir noticis, un tā tam arī jāpaliek jebkuras situācijās ietekmē.
- +Visi dati glabājas operatīvajā atmiņā, kas nodrošina atraku piekļuvi salīdzinājuma ar cieto disku.
- Taču ar to nepietiek, arī izstrādātajam jāorganizē pareizi datus un jāraksta efektīvu kodu, lai sasniegtu max efektivitāti. Ar to palīdz heša optimizācijas tabulas, kuras izmantošana nozīmīgi palielina lasīšanas operāciju ātrumu salīdzinājuma ar RDBMS.