

Distribovaný systém pre algoritmické obchodovanie na burze

Michal Hornický

Brno University of Technology, Faculty of Information Technology
Božetěchova 1/2. 602 00 Brno - Královo Pole
xhorni14@fit.vutbr.cz



18. júna 2019

- Webová aplikácia

- Webová aplikácia
- Používateľské stratégie

```
1 local slow = ta.sma(20);  
2 local fast = ta.ema(20);  
3 local macd = ta.macd(1,2,3);  
4 local rsi = ta.rsi(10);  
5 local eee = ta.sma(10);  
6  
7 if fast() > slow() then  
8     return "long"  
9 elseif slow() > fast() then  
10    return "short"  
11 else  
12    return 'neutral'  
13 end
```

- Webová aplikácia
- Používateľské stratégie
- Automatické
vyhodnocovanie stratégie
a obchodovanie systémom

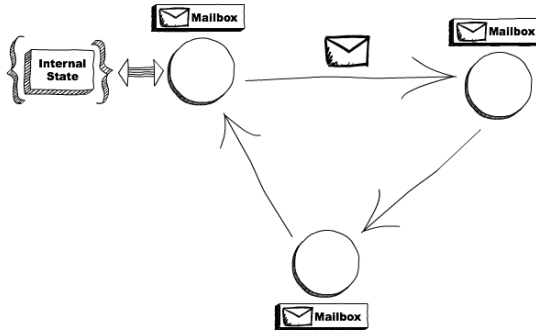
```
1 local slow = ta.sma(20);  
2 local fast = ta.ema(20);  
3 local macd = ta.macd(1,2,3);  
4 local rsi = ta.rsi(10);  
5 local eee = ta.sma(10);  
6  
7 if fast() > slow() then  
8     return "long"  
9 elseif slow() > fast() then  
10    return "short"  
11 else  
12    return 'neutral'  
13 end
```

- Webová aplikácia
- Používateľské stratégie
- Automatické
vyhodnocovanie stratégie
a obchodovanie systémom
- Požiadavky na
škálovať elnosť a latenciu

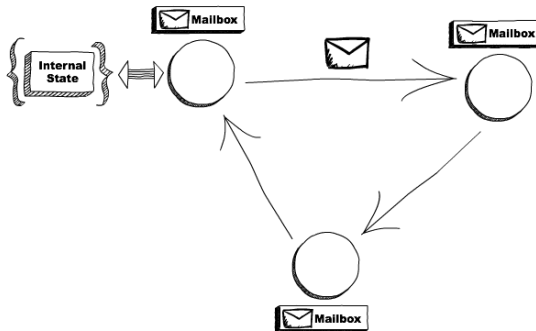
```
1 local slow = ta.sma(20);  
2 local fast = ta.ema(20);  
3 local macd = ta.macd(1,2,3);  
4 local rsi = ta.rsi(10);  
5 local eee = ta.sma(10);  
6  
7 if fast() > slow() then  
8     return "long"  
9 elseif slow() > fast() then  
10    return "short"  
11 else  
12    return 'neutral'  
13 end
```


- Distribuovaná architektúra

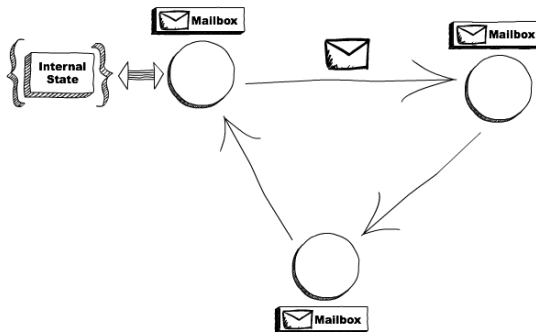
- Distribuovaná architektúra
- Actor model

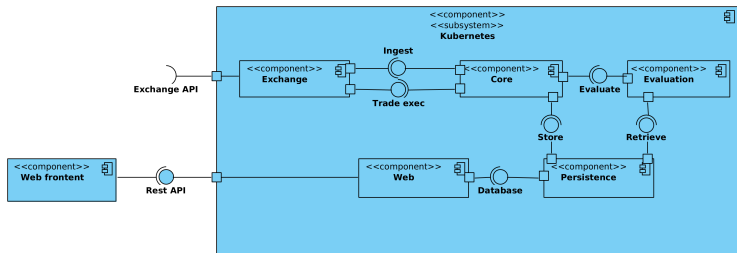


- Distribuovaná architektúra
- Actor model
- Cloudové výpočetné prostredie

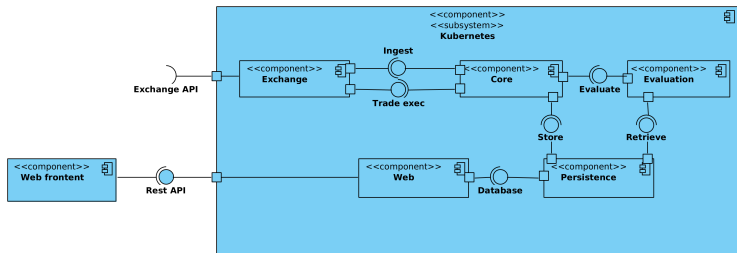


- Distribuovaná architektúra
- Actor model
- Cloudové výpočetné prostredie
- Moderný implementačný jazyk

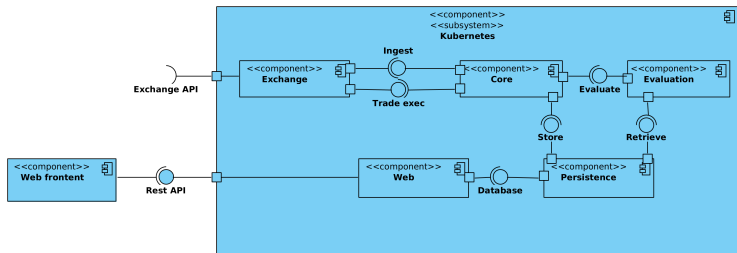




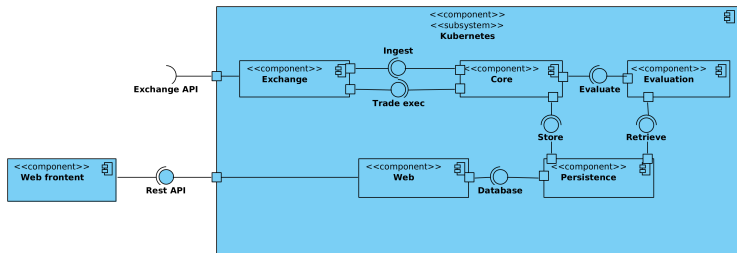
- Systém rozdelený do komponentov



- Systém rozdelený do komponentov
- Komponenty spravované systémom Kubernetes
 - Monitoring a správa jednotlivých komponentov
 - DNS pre service discovery
 - Dynamické škálovanie na základe zát'áže



- Systém rozdelený do komponentov
- Komponenty spravované systémom Kubernetes
 - Monitoring a správa jednotlivých komponentov
 - DNS pre service discovery
 - Dynamické škálovanie na základe zát' až e
- Komunikácia pomocou ZeroMQ



- Systém rozdelený do komponentov
- Komponenty spravované systémom Kubernetes
 - Monitoring a správa jednotlivých komponentov
 - DNS pre service discovery
 - Dynamické škálovanie na základe zát'áže
- Komunikácia pomocou ZeroMQ
- Jazyk Rust s knižnicou Actix

- 2 Open source knižnice:
 - Knižnica Actix-comm pre komunikáciu aktérov pomocou ZeroMQ
 - Knižnica Actix-arch pre architekturné komponenty

- 2 Open source knižnice:
 - Knižnica Actix-comm pre komunikáciu aktérov pomocou ZeroMQ
 - Knižnica Actix-arch pre architekturné komponenty
- Systém dostupný na *trader.semtexzv.com*
 - Podporuje burzu Bitfinex

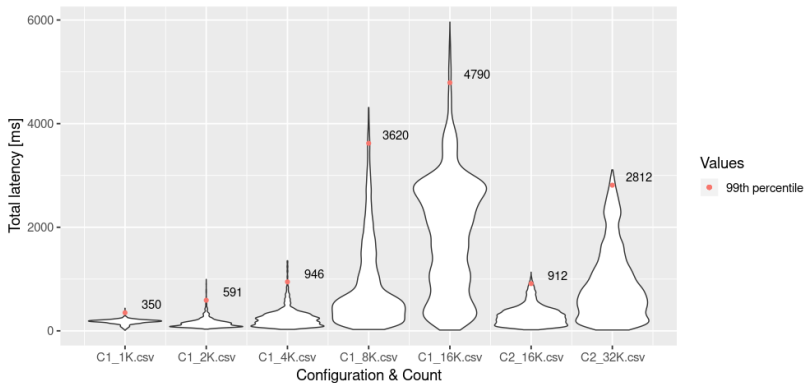
Prístup :

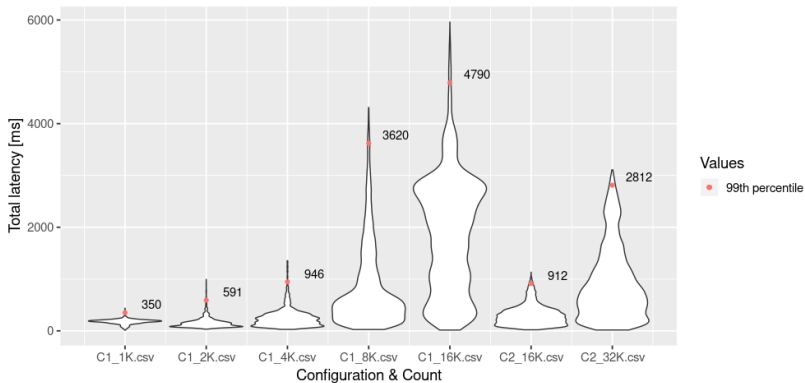
- Virtuálny používatelia
- Zmena počtu používateľov a konfigurácie systému
- Meranie latencie

Prístup :

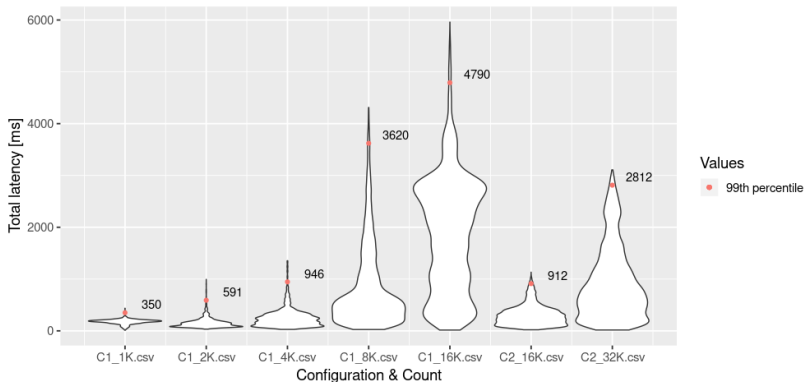
- Virtuálny používatelia
- Zmena počtu používateľov a konfigurácie systému
- Meranie latencie

Primárny cieľ: latencia < 1s





- Konfigurácia C1: \$ 30 - 4 000 priradení



- Konfigurácia C1: \$ 30 - 4 000 priradení
- Konfigurácia C2: \$ 60 - 16 000 priradení
- Náročnejšie konfigurácie budú vyžadovať automatické škaloovanie komponentov

2 aspekty:

2 aspekty:

- ① Samotný projekt
 - Používateľské rozhranie
 - Simulácia na historických dátach (Backtesting)

2 aspekty:

① Samotný projekt

- Používateľské rozhranie
- Simulácia na historických dátach (Backtesting)

② Architektúra

- Úprava knižníc actix-comm a actix-arch
- Používanie pri ďalších projektoch

```
1 -- Fast exponential moving average closely follows price while smoothing out
2 -- random swings
3 local ema_fast = ta.ema(3)
4 -- Slow exponential moving average tracks longer term trend
5 local ema_slow = ta.ema(29)
6
7 -- We use RSI to determine when an asset is overbought or oversold
8 local rsi = ta.rsi(14)
9
10 -- Triggering signal for buying and selling
11 local buy_signal = ema_fast() > ema_slow() * 1.001
12 local sell_signal = ema_fast() < ema_slow() * 0.999
13
14 -- RSA guards for overbought or oversold markets
15 local rsi_overbought = rsi() > 80
16 local rsi_oversold = rsi() < 20
17
18 if buy_signal and not rsi_overbought then
19     return 'long'
20 elseif sell_signal and not rsi_oversold then
21     return 'short'
22 else
23     return 'neutral'
24 end
```

Strata 2% po týždni používania

```
1 -- Fast exponential moving average closely follows price while smoothing out
2 -- random swings
3 local ema_fast = ta.ema(3)
4 -- Slow exponential moving average tracks longer term trend
5 local ema_slow = ta.ema(29)
6
7 -- We use RSI to determine when an asset is overbought or oversold
8 local rsi = ta.rsi(14)
9
10 -- Triggering signal for buying and selling
11 local buy_signal = ema_fast() > ema_slow() * 1.001
12 local sell_signal = ema_fast() < ema_slow() * 0.999
13
14 -- RSI guards for overbought or oversold markets
15 local rsi_overbought = rsi() > 80
16 local rsi_oversold = rsi() < 20
17
18 if buy_signal and not rsi_overbought then
19     return 'long'
20 elseif sell_signal and not rsi_oversold then
21     return 'short'
22 else
23     return 'neutral'
24 end
```

Strata 2% po týždni používania

Ďakujem za pozornosť

- 1 Prečo ste zvolili ZeroMQ ?
- 2 Popíšte ako funguje Load balancing v komponente ktorý vyhodnocuje stratégie.

Možné alternatívy:

- 1 TCP
- 2 ZeroMQ
- 3 HTTP

Výhody ZeroMQ:

- Komplexné topológie - Pub-sub
- Jednoduchý formát správ
- Nízka latencia

- Nemožnosť použitia DNS distribúcie záťaže kvôli dlhodobým pripojeniam
- 1 distribučný kontajner, niekoľko výkonných
- Implementované ako sada aktérov v knižnici Actix-arch
- Distribúcia Round-robin na úrovni požiadavkov

