

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Primož Hrovat

Mikrostoritve v decentraliziranem okolju

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Matjaž Branko Jurič

Ljubljana, 2018

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Besedilo teme diplomskega dela študent prepíše iz študijskega informacijskega sistema, kamor ga je vnesel mentor. V nekaj stavkih bo opisal, kaj pričakuje od kandidatovega diplomskega dela. Kaj so cilji, kakšne metode uporabiti, morda bo zapisal tudi ključno literaturo.

*Na tem mestu zapišite, komu se zahvaljujete za izdelavo diplomske naloge.
Pazite, da ne boste koga pozabili. Utegnil vam bo zameriti. Temu se da
izogniti tako, da celotno zahvalo izpustite.*

Posvetilo

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Arhitekturni koncepti mikrostoritev	3
2.1	Arhitektura mikrostoritev	3
2.2	Vzorci	4
2.3	Vsebniki in orkestracija	4
3	Tehnologija veriženja podatkovnih blokov	5
3.1	Razlaga osnovnih konceptov	6
3.2	Ethereum	6
3.3	Hyperledger	6
4	Koncepti registracije in odkrivanja storitev v decentraliziranem svetu	7
5	Implementacija in validacija rešitve	9
6	Zaključek	11
	Literatura	13

Seznam uporabljenih kratic

kratica	angleško	slovensko
CA	classification accuracy	klasifikacijska točnost
DBMS	database management system	sistem za upravljanje podatkovnih baz
SVM	support vector machine	metoda podpornih vektorjev
...

Povzetek

Naslov: Mikrostoritve v decentraliziranem okolju

Avtor: Primož Hrovat

TODO

Ključne besede: decentralizacija, distribuirane storitve, tehnologija veriženja podatkovnih blokov.

Abstract

Title: Microservices in decentralized environment

Author: Primož Hrovat

TODO

Keywords: decentralization, distributed services, blockchain.

Poglavje 1

Uvod

Poslovne storitve se danes selijo v oblak. S pojavom arhitekture mikro-storitev in podporne tehnologije, kot so vsebniki in okolja za orkestracijo vsebnikov, so, nekdam ogromni, kosi programske opreme pričeli razpadati na manjše, logično ločene sestavne dele. Razmeroma majhne in neodvisne aplikacije, specializirane za opravljanje točno določenih nalog, omogočajo hiter vzpostavitevni čas in puščajo majhen odtis na porabi strojne opreme. Neodvisnost teh aplikacij nam omogoča tudi skaliranje teh posameznih delov celotne storitve, ko je to potrebno. Vsebniki so razmeroma stara tehnologija, ki je s pojavom okolja Docker doživela pravi razcvet. Gre za "lahko" obliko virtualizacije, virtualizacija tu poteka na nivoju procesov in ne operacijskega sistema. Pravi potencial vsebnikov izkoristimo šele z uporabo orkestratorjev, kot so Kubernetes, Amazon ECS, Google Container Engine (GKE), Docker Swarm, Azure Container Service in podobni. Ta orodja omogočajo monitoring, zaganjanje, zaustavljanje in preverjanje storitev, skladno s podanimi zahtevami. Storitve se izvajajo distribuirano (porazdeljeno) in replicirano, lahko tudi na fizično ločenih sistemih, kar zagotavlja visoko stopnjo odzivnosti in dosegljivosti. Stopnja dosegljivosti se danes meri na peti ali šesti decimalki („number of nines“).

S pojavom Bitcoina se je začel razvoj tehnologije, ki v decentraliziranem okolju omogoča varno in nespremenljivo hrambo podatkov. Veriženje

podatkovnih blokov je v zadnjih letih s pojavom različnih organizacij kot so Hyperledger in Ethereum doživelo razcvet. Nova tehnologija omogoča shranjevanje podatkov, repliciranih na vseh sodelujočih entitetah v omrežju, obenem pa ohranjajo nespremenljivost.

Tu se pojavlja vprašanje. Ali je možno tudi poslovno logiko prestaviti v decentraliziran svet na način, da bo sodelujoči v omrežju ob klicu želene storitve vedno dobil odgovor, izvedla pa ga bo katerakoli druga entiteta, obenem pa zagotoviti pravilnost izvajana? Z uresničitvijo tega cilja bi miselnost decentraliziranih podatkov prestavili nivo višje, na nivo poslovne logike. Odzivnost in dosegljivost storitve bi s številom sodelujočih v omrežju dosegla 100%, prav tako bi bilo praktično nemogoče izvesti napade DOS.

Poglavje 2

Arhitekturni koncepti mikrostoritev

Razvoj monolitov je enostaven in danes dobro podprt v vseh danes prisotnih razvojnih okoljih. Prenos in namestitev teh storitev na strežniške sisteme je enostaven, rešitev v obliki izvršljivih datotek ali s kopiranjem direktorijske strukture prenesemo v produkcijsko izvajalno okolje. Če želimo tako storitev skalirati, kot vstopno točko v naše zaledne sisteme nastavimo izenačevalnika obremenitev (ang. load balancer), ki poskrbi za enakomerno porazdeljevanje dela med posameznimi instancami storitve.

Slabosti te arhitekture se pojavijo, ko storitev postane kompleksnejša. Podaljša se zagonski čas (start up time), ob preobremenitvi le enega dela storitve je potrebno pognati celotno aplikacijo. Proces sprotne dostave (Continuous Delivery) je otežen, za posodobitev enega dela sistema, je potrebno celotno storitev zaustaviti, namestiti novo različico in jo zagnati. [3]

2.1 Arhitektura mikrostoritev

Arhitektura mikrostoritev omogoča

2.2 Vzorci

2.2.1 metrike

2.2.2 health

2.2.3 discovery

2.2.4 fault tolerance

2.3 Vsebniki in orkestracija

Poglavje 3

Tehnologija veriženja podatkovnih blokov

Veriženje podatkovnih blokov je peer-to-peer porazdeljena podatkovna shramba, dosežena s konsenzom, sistemom "pametnih" pogodb ter drugih pomožnih tehnologij [1]. V središču omrežja je glavna knjiga "ledger", ki beleži vse transakcije, izvedene na omrežju. [2] Vsak blok v verigi predstavlja zbirko transakcij, te člene pa med seboj povezuje zgoščevalna (hash) funkcija. V verigo je blok moč le dodajati. Vsak nov blok mora biti pred zapisom potrjen s strani skupnosti in, preko vrednosti zgoščene funkcije prejšnjega člena, povezan v verigo blokov. Celotna veriga blokov je shranjena pri vsaki sodelujoči entiteti. Kombinacija teh pristopov omogoča, da nobena izmed sodelujočih entitet ne more spreminjati že zapisanih blokov. S temi mehanizmi se zagotovi veljavnost in nespremenljivost podatkov, v okolju, ki mu apriori ni potrebno zaupati. Ni več potrebe po zunanji, zaupanja vredni, entiteti.

Za interakcijo z glavno knjigo in zapisovanje novih informacij, omrežje uporablja t.i. "pametne pogodbe".

3.1 Razlaga osnovnih konceptov

3.2 Ethereum

Ethereum je decentralizirana platforma, ki izvaja "pametne" pogodbe (smart contracts) - aplikacije, ki se izvajajo natanko tako, kot so bile zapisane. Platforma je osnovana na verigi podatkovnih blokov, ki omogoča reprezentacijo in prenos vrednosti. Lahko si ga predstavljamo kot svetovni računalnik, izvajanje programske kode pa poteka na vseh sodelujočih računalnikih. Pametne pogodbe ponujajo možnost interakcije s podatkovno verigo, določeni deli kode pa se izvajajo le pod točno določenimi pogoji.

3.3 Hyperledger

Hyperledger je družina odprtokodnih projektov, namenjenih razvoju tehnologije veriženja podatkovnih blokov. Projekt deluje pod okriljem organizacije The Linux Foundation, v sodelovanju s skupnostjo. Med prvimi in najbolj znanimi izmed Hyperledger projektov je Hyperledger Fabric, prvotno razvit v podjetju IBM in Digital Asset. Pod okrilje organizacije Hyperledger spadajo še Sawtooth, Iroha, Burrow ter Indy. Vsak izmed projektov na svoj način rešuje izzive s področja podatkovnih verig ali pa naslavlja ozko problemsko domeno, za primer: projekt Indy se ukvarja s problematiko spletne identitete uporabnika. [1] Trenutno najbolj znana in razširjena platforma je Fabric, trenutno v različici 1.1. Od ostalih podobnih projektov se loči predvsem v privatnih omrežjih, pri katerih je sodelovanje omejeno s sistemom dovoljenj. Omogoča modularno izbiro načina soglasja in ga je moč prilagajati zahtevam poslovnih uporabnikov. [?]

Poglavje 4

Koncepti registracije in odkrivanja storitev v decentraliziranem svetu

Poglavje 5

Implementacija in validacija rešitve

Poglavje 6

Zaključek

Literatura

- [1] Hyperledger. <https://www.hyperledger.org>. Dostopano: 22. 05. 2018.
- [2] Hyperledger Fabric documentation. <https://hyperledger-fabric.readthedocs.io>. Dostopano: 22. 05. 2018.
- [3] Pattern: Monolithic Architecture. <http://microservices.io/patterns/monolithic.html>. Dostopano: 22. 05. 2018.