Jedno Rešenje CloudSim Simulacije Distribuirane STM

Dragan Brkin, Branislav Kordić, Miroslav Popović

***Apstrakt*—U ovom radu predstavljeno je proširenje IaaS Cloud simulatora CloudSim. Prvo je modelovan zadatak u vidu transakcije nad transakcionom memorijom i komunikacija između centara za obradu podataka upotrebom protokola ažuriranja u dve faze. Zatim je implementiran model prototipa distribuirane STM. Na kraju je urađena evaluacija i dobijeni rezultati poređeni su sa prethodno izvedenim teorijskim rezultatima. Prezentovani rezultati su pozitivni i stimulišu budući rad u razvoju distribuirane STM.**

***Ključne reči*—CloudSim; Softverska Transakciona Memorija; Distribuirani Sistemi; Protokol Ažuriranja u Dve Faze.**

I. Uvod

Transakciona memorija – TM (engl. *Transactional Memory*) predstavlja mehanizam, odnosno paradigmu paralelnog programiranja za višejezgarne i nadolazće mnogojezgarne arhitekture. Softverska Transakciona Memorija (STM) predstavlja programsku implementaciju TM paradigme, dok na sličan način Distribuirana Transakciona Memorija (DTM) generalizuje isti mehanizam za distribuirane sisteme [1].

Računarstvo u oblaku (engl. *Cloud Computing*) je tehnologija koja podržava novi način korišćenja hardverske infrastrukture. Dobavljači oblaka nude ove infrastrukture u vidu virtuelnog hardvera upravljanog od strane odgovarajućeg softvera. Oni nude svoje usluge u formi virtuelnih mašina (VM) spremnih za korišćenje na zahtev korisnika [5] i kao takve predstavljaju osnovu za distribuirane sisteme.

Infrastruktura kao usluga – IaaS (engl. *Infrastructure as a Service*) predstavlja jedan od tri osnovna modela u računarstvu u oblaku. Ovaj model omogućava korišćenje računarske infrastrukture u vidu VM. Korisnik je u mogućnosti da upravlja VM, njihovim umrežavanjem kao i skladištenjem podataka. U virtuelnoj infrastrukturi korisnik može pokrenuti različite vrste programske podrške, od operativnog sistema do aplikacija. Za pristup infrastrukturi koristi se Internet. Da bi ovaj servis bio dostupan korisnicima, neophodan je softver koji omogućava administraciju infrastrukture, jednostavno dodeljivanje resursa, upravljanje infrastrukturom i merenje performansi.

CloudSim predstavlja jedan od alata za modelovanje i simulaciju računanja u oblaku, odnosno infrastrukture kao servisa [2]. CloudSim alat je korišćen za simulaciju i analizu aplikacija velikih razmera poput društvenih mreža u oblaku [6], evaluaciju strategija za raspoređivanje poslova zasnovanih na SLA (engl. *Service Level Agreement*) u okviru distribuiranih i centara za obradu podataka u oblaku [7], kao i za analizu performansi i energetsku procenu I/O (engl. Input/Output) operacija unutar centara za obradu podataka na osnovu VM [5]. Prema našim najboljim saznanjima, ovo je prvi rad koji doprinosi modelovanju i simulaciji distribuiranih sistema zasnovanih na STM.

Protokol ažuriranja u dve faze 2PC (engl. *Two-phase commit*) je protokol neprekidivog ažuriranja korišćen u distribuiranim sistemima. Protokol se, kao što i sam naziv govori, sastoji iz dve faze. Prva faza predstavlja zahtev za ažuriranje koji transakcioni rukovaoc (koordinator) šalje svim transakcionim resursima kako bi se odredilo da li će se transakcija izvršiti ili prekinuti. Druga faza je izvršenje ili prekid transakcije, zavisno da li je odgovor, koji transakcioni koordinator dobije na zahtev, pozitivan ili negativan, respektivno.

U sledećem odeljku opisan je CloudSim, alat koji je korišćen za simulaciju distribuiranog sistema. Zatim su opisani model i arhitektura simuliranog sistema, nakon čega je islustrovan primer ponašanja sistema. Na kraju, predstavljena je evaluacija i dati su rezultati.

II. CloudSim alat

CloudSim je alat za modelovanje i simulaciju okruženja računarskog oblaka i evaluaciju algoritama za obezbeđivanje resursa [5-6]. Ovaj alat radi kao simulator zasnovan na diskretnim događajima i implementiran je u programskom jeziku Java. Osnovni elementi CloudSim-a čine entiteti: *CloudInformationService*, *Datacenter*, *DatacenterBroker* i *CloudsimShutdown*. Ovi entiteti predstavljaju osnove elemente arhitekture centra za obradu podataka (engl. *Datacenter*). Komunikacija između entiteta se ostvaruje slanjem definisanih poruka događaja (npr. VM_CREATE, CLOUDLET_SUBMIT itd). Ovi događaji mogu biti spoljašnji (poslati od strane jednog entiteta drugom) ili unutrašnji (poslati i primljeni od strane istog entiteta). Po primanju, svaka od poruka događaja se preuzima i sprovode se određene akcije, pre nego što se pošalje poruka potvrde (npr VM_CREATE_ACK) [5]. Simulacija u CloudSim-u

Dragan Brkin – Univerzitet u Novom Sadu, Fakultet tehničkih nauka, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija (e-mail: dragan.brkin@live.com).

Branislav Kordić – RT-RK Institute for Computer Based Systems, Narodnog fronta 23a, 21000 Novi Sad, Srbija, (e-mail: branislav.kordic@rt-rk.com).

Miroslav Popović – Univerzitet u Novom Sadu, Fakultet tehničkih nauka, Trg Dositeja Obradovića 6, 21000 Novi Sad, Srbija, (e-mail: miroslav.popovic@rt-rk.uns.ac.rs).

zasnovana je na izvršavanju objekta klase Cloudlet. Cloudlet modeluje proces opisan veličinom ulaznih i izlaznih podataka, potrebnom RAM memorijom, brojem procesorskih jezgara kao i opterećenja procesora, načinom raspoređivanja, itd. Cloudlet se izvršava na VM u okviru centra za obradu podataka.

Broker modeluje krajnjeg korisnika koji serveru, odnosno VM prosleđuje proces i prihvata rezultat obrade.

III. CloudSim model PSTM sistema

U ovom odeljku opisana je korišćena arhitektura simuliranog sistema. Zatim je dat opis odgovarajućeg CloudSim modela.

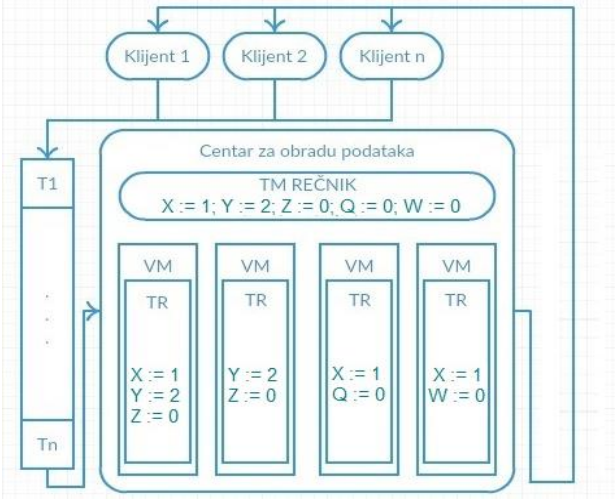
A. Arhitektura PSTM sistema

Pajton softverska transakciona memorija – PSTM (engl. *Python Software Transactional Memory*) je realizovana pomoću rečnika unutar centra za obradu podataka. Svakom korisniku se, na zahtev, prosleđuje kopija t-promenljive i na zahtev, ukoliko je to moguće, ažurira se stanje rečnika. Za prijem, odnosno slanje zahteva koristi se red (engl. *Queue*) kako bi se zahtevi mogli obrađivati po redosledu kojim su poslati [4]. Klijenti koji koriste transakcionu memoriju zovu se transakcione aplikacije.

PSTM-zasnovana aplikacija sadrži skup transakcija koje obavljaju operacije nad lokalnim promenjivama, odnosno kopijama t-promenljivih. Transakcija zahteva kopije t-promenljivih na početku i ažurira neke od njih na kraju, koristeći PSTM API spregu. Ovu spregu čini skup javnih funkcija definisanih u PSTM modulu koje zahtev za obradu šalju udaljenom serveru upotrebom RPC (engl. *Remote Procedure Call*) mehanizma. Rečnik koji se koristi je skup torki (*key*, *ver*, *val*), gde je *key* naziv t-promenljive, *ver* trenutna verzija, a *val* vrednost. PSTM server poslužuje dolazne zahteve automatski. Po prijemu, zahtev se obradi i odgovor se odmah šalje nazad klijentu [3].

B. CloudSim model

Arhitektura simuliranog distribuiranog sistema se sastoji od više servera predstavljenih u [3]. 2PC protokol omogućuje komunikaciju između centara za obradu podataka.



Sl. 1. Blok dijagram distribuiranog STM sistema.

Prilikom modelovanja sistema uvedeno je nekoliko pretpostavki. Smatra se da je komunikacija između centara za obradu podataka zanemarljiva u odnosu na lokalnu komunikaciju, pa se 2PC protokol obavlja ne narušavajući redosled novo-pristiglih zahteva za obradu transakcije. U početnom trenutku, svi centri za obradu podataka raspolažu sa istim podacima, odnostno imaju iste rečnike. Ova pretpostavka je uzeta radi pojednostavljenja modelovanja i simulacije obrade transakcija. Potrebno je napomenuti da se svaka transakcija izvršava na zasebnoj VM, na istom ili različitim centrima za obradu podataka. Blok dijagram CloudSim modela prikazan je na Sl. 1.

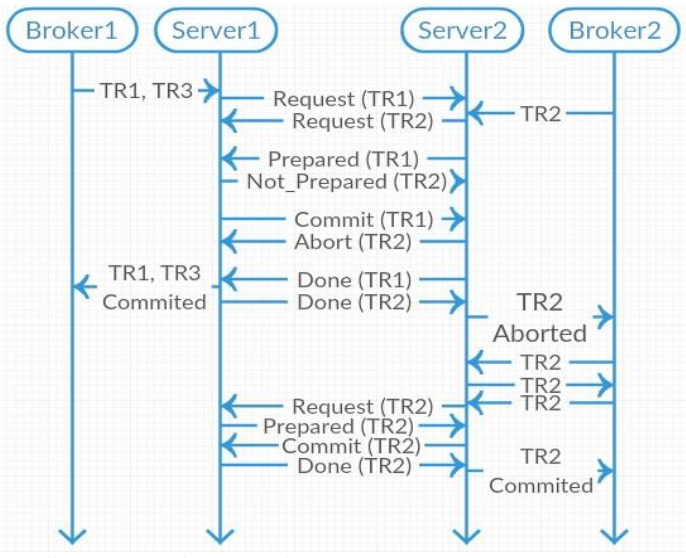
Model transakcije: predstavlja proširenje klase Cloudlet koja, kao što je već pomenuto, modeluje proces koji se izvršava na VM. Transakcija je definisna pomoću tipa transakcije, trenutnim statusom i skupovima t-promenljivih nad kojima se obavlja operacija čitanja, odnosno pisanja. Transakcija je neprekidiva operacija i ne zavisi od tipa transakcije, kao ni od veličine skupova t-promenljivih nad kojima se obavljaju operacije čitanja, odnosno pisanja.

Modelovanje rečnika STM servera: Rečnik koji se koristi u distribuiranom sistemu modelovan je kao par (*key*, *ver*) gde *key* predstavlja jedinstven ključ t-promenljive, a *ver* trenutnu verziju. Rečnik iz [3] je pojednostavljen, jer vrednost t-promenljive za potrebe simulacije sistema nije relevantna. U simulaciji takođe nisu uključena moguća zagušenja mreže kao ni otkazi servera na kojima se transakcija obrađuje.

Modelovanje 2PC protokola: Kako bi se 2PC protokol realizovao, svaki od centara za obradu podataka u simulaciji mora posedovati adrese svih ostalih centara za obradu podataka. Popunjavanje ove liste obavlja se pre početka simulacije, tako da u početnom trenutku, unutar simulacije, svi centri za obradu podataka imaju sve potrebne informacije za izvršavanje 2PC.

IV. Primer ponašanja modela

U ovom odeljku dat je jedan slučaj izvršenja transakcija u distribuiranom okruženju.



Sl. 2. Dijagram komunikacije između centara za obradu podataka i brokera.

Na Sl. 2 prikazana su dva centra za obradu podataka na koje su poslate tri transakcije *TR1, TR2, TR3* u približno isto vreme. Transakcije *TR1* i *TR3* obrađuju se na serveru *Server1*, dok se transakcija *TR2* obrađuje na serveru *Server2*.

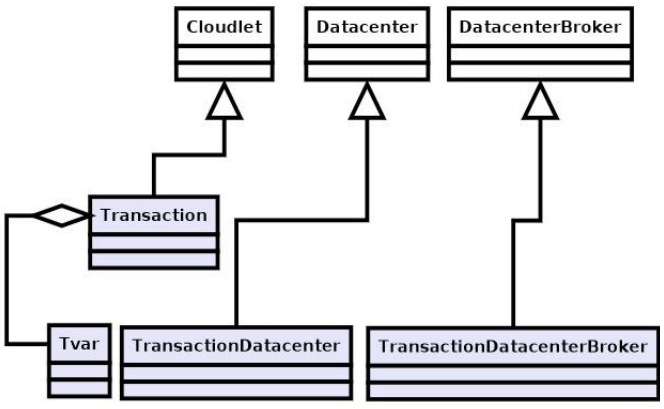
Dve transakcije *TR1* i *TR2* pokušavaju da ažuriraju skup istih t-promenljivih, dok jedna transakcija *TR3* obavlja čitanje, odnosno *READ* operaciju. Kako transakcija *TR3* koja obavlja *READ* operaciju ne može biti u konfliktu ni sa jednom drugom transakcijom, ona se uspešno izvršava. Transakcije *T1* i *T2* koje pokušavaju da ažuriraju istu t-promenljivu, dolaze u konflikt prilikom 2PC. Kako i jedna i druga mogu lokalno uspešno da izvrše ažuriranje, pokreće se 2PC sa obe strane. Centri za obradu podataka jedan drugom šalju zahtev za ažuriranje skupa t-promenljivih. Međutim u ovom trenutku jedan server lokalno brže obrađuje zahtev od drugog i on šalje negativan odgovor, dok server koji još nije stigao zahtev lokalno da obradi šalje pozitivan odgovor. Ažuriranje obavlja server (*Server1*) koji primi pozitivan odgovor i tada se izvršava poravnanje rečnika u sistemu. Transakcije *TR1* i *TR3* izvršavaju se istovremeno i uspešno, dok je transakcija *TR2* neuspešna.

Po neuspešnom ažuriranju, transakcija *TR2* obavlja operaciju čitanja kako bi preuzela poslednje verzije t-promenljivih, a potom ponovo pokušava da ažurira date t-promenljive. Kako ni jedna druga transakcija ne pokušava da ažurira isti skup t-promenljivih, konflikta neće biti. Server, nakon uspešne lokalne obrade, zahtev za ažuriranjem šalje drugom serveru upotrebom 2PC protokola. Po prijemu pozitivnog odgovora izvršava se ažuriranje rečnika u sistemu.

V. Implementacija

U procesu implementacije proširene su već postojeće CloudSim klase sa malim izmenama originalne implementacije. Na Sl. 3 prikazan je UML dijagram proširenih klasa. Na dijagramu su date samo proširene i naknadno dodate klase neophodne za simulaciju i rukovanje transakcijama.

Tvar klasa predstavlja objekat t-promenljive, koja čini osnovni gradivni element transakcije. Kako bi se transakcija od strane brokera mogla poslati serveru, odnosno VM, uvedene su poruke događaja i rukovaoci ovim događajima,



Sl. 3. UML dijagram dodatih i izmenjenih CloudSim klasa.

čime je obezbeđeno pokretanje obrade transakcije na VM i vraćanje rezultata obrade brokeru. Svi rukovaoci događajima neophodni za slanje i prihvat obrađene transakcije implementirani su u klasi *TransactionDatacenterBroker*.

Klasa *Transaction* sadrži implementaciju prethodno pomenutog modela transakcije. Objekat ove klase je sastavni deo poruke događaja koju broker šalje VM i obrnuto. Transakcija može da izvršava operacije čitanja i/ili pisanja.

Centar za obradu podataka poseduje lokalne računare unutar kojih se nalaze VM na kojima se izvršavaju transakcije. Kako bi transakcije mogle da se izvršavaju na VM, potrebno je bilo proširiti originalnu implementaciju klase *Datacenter* dodavanjem novih poruka događaja obrade transakcije kao i poruka neophodnih za realizaciju 2PC protokola. Rukovaoci ovim događajima implementirani su u klasi *TransactionDatacenter*.

VI. Evaluacija i rezultati

U ovom odeljku data je evaluacija sistema i rezultati simuliranih relevantnih slučajeva izvršenja. Tabela I sadrži rezultate grupe transakcija koje obavljaju operaciju čitanja. Sve transakcije izvršavaju se na istom centru za obradu podataka, pri čemu svaka od njih sadrži bar jednu istu t-promenljivu kao i ostale. Po vremenu početka *Start* i vremenu trajanja transakcija *Trajanje*, može se videti da se transakcije izvršavaju paralelno. Vremena u tabelama data su u neimenovanoj jedinici. Pošto se izvršavaju samo transakcije koje obavljaju operaciju čitanja, koja ne izaziva konflikt, sve transakcije se izvršavaju uspešno.

TABELA I
REZULTATI GRUPE TRANSAKCIJA KOJE OBAVLJAJU OPERACIJU ČITANJA NA JEDNOM CENTRU ZA OBRADU PODATAKA

Start	Trajanje	Čitanje/Upis	Neuspešno
0.1	1600.1	5/-	0
0.1	1600.1	5/-	0
0.1	1600.1	5/-	0
0.1	1600.1	5/-	0
0.1	1600.1	5/-	0

TABELA II
REZULTATI GRUPE TRANSAKCIJA KOJE OBAVLJAJU OPERACIJU ČITANJA ODNOSNO PISANJA NA JEDNOM CENTRU ZA OBRADU PODATAKA

ID	Start	Trajanje	Tip	Neuspešno
TR1	0.1	1600.1	Čitanje	0
TR3	0.1	1600.1	Čitanje	0
TR4	0.1	1600.1	Upis	0
TR2	1600.1	3200.1	Upis	1
TR5	3200.1	4800.1	Upis	2

Vreme trajanja transakcije u simulaciji preračunava se na osnovu definisanih karakteristika procesa. Svaka transakcija definisana je, kao što je već pomenuto, brojem procesorskih jedinica na kojima se izvršava i brojem instrukcija

neophodnim da bi se izvršila. Broj instrukcija izražava u jedinici *MI* milion instukcija (engl. *Million Instruction*). VM takođe je definisana brojem procesorskih jedinica, gde je svaka procesorska jedinica definisana jedinicom MI [2]. Na osnovu karakteristika VM i definisanog procesa, CloudSim automatski preračunava vreme potrebno da se dati proces, odnosno transakcija, izvrši.

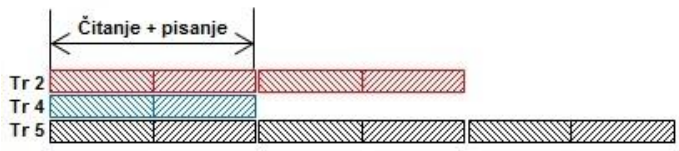
U Tabeli II dati su rezultati grupe transakcija koje se izvršavaju na jednom centru za obradu podataka i sve transakcije se pokreću u isto vreme. Transakcije *TR1* i *TR3* obavljaju operaciju čitanja, dok transakcije *TR2*, *TR4* i *TR5* obavljaju operaciju pisanja. Kako sve transakcije rade nad istim skupom t-promenljivih, dolazi do konflikta među transakcijama koje obavljaju operaciju pisanja. Među transakcijama koje su u konfliktu samo jedna može uspešno da se izvrši, dok se ostale izvršavaju neuspešno. U ovom slučaju transakcija *TR4* izvršila se uspešno i ona se paralelno izvršava sa transakcijama koje vrše operaciju čitanja *TR1* i *TR3*. Vremena početka i trajanja transakcija *TR2* i *TR5* posledica su obavljanja operacije čitanja pre ponovne operacije pisanja. Svakim neuspešnim ažuriranjem transakcija izvršava operaciju čitanja, kako bi dobavila najnovije verzije za skup t-promenljivih koje želi da ažurira.

TABELA III
REZULTATI GRUPE TRANSAKCIJA KOJE OBAVLJAJU OPERACIJU ČITANJA I PISANJA NA RAZLIČITIM CENTRIMA ZA OBRADU PODATAKA

ID	Start	Trajanje	Čitanje/Upis	Neuspešno
TR1	0.1	1600.1	5/5	0
TR2	1600.2	3200.2	5/5	1
TR3	3200.2	4800.3	5/5	2
TR4	4800.3	6400.4	5/5	3
TR5	6400.4	8000.5	5/5	4

U Tabeli III prikazani su rezultati za grupu od pet transakcija koje obavljaju i čitanje i pisanje, svaka na različitom centru za obradu podataka. Sve transakcije koriste isti skup t-promenljivih i počinju u približno isto vreme. Svaka od transakcija pokušava da ažurira predhodno čitanje t-promenljive, pa su stoga sve transakcije u konfliktu. Ovaj konflikt utiče na vreme izvršavanja transakcija. Vreme izvršavanja druge najkraće transakcije duplo je veće od vremena izvršavanja prve, što je posledica prvog neuspešnog ažuriranja.

Na Sl. 4 dat je grafički prikaz izvršavanja transakcija *TR2*, *TR4* i *TR5* iz Tabele II. Posle svake neuspešno završene transakcije mora se izvršiti operacija čitanja, kako bi se dobavile najnovije verzije t-promenljivih. Kao što se i sa grafika može videti, vreme trajanja transakcije direktno je proporcionalno broju prethodno neuspešnih izvršavanja.



Sl. 4. Prikaz izvršavanja transakcija na tri centra za obradu podataka.

VII. ZAKLJUČAK

U ovom radu predstavljena je CloudSim simulacija distribuiranog STM sistema. Predstavljen je način funkcionisanja simulatora i ponašanje sistema. Implementiran je model za neoptimizovani prototip. Evaluacija je urađena kroz simulaciju najkritičnijih slučajeva koji mogu narušiti ispravan rad sistema. Dobijenim rezultatima verifikovana je ispravnost algoritma obrade transakcija i komunikacije između distribuiranih centara.

Dalji rad mogao bi pokazati uticaj stohastičnog zagušenja mreže na performanse sistema. Takođe bi se mogao staviti akcenat na modelovanje zavisnosti trajanja transakcije od tipa transakcije, kao i od broja t-promenljivih nad kojima se transakcija obavlja.

ZAHVALNICA

Ovaj rad je delimično finansiran od strane Ministarstva za prosvetu, nauku i tehnološki razvoj Republike Srbije, na projektu broj III_044009_2.

LITERATURA

[1] T. Harris, J. R. Larus, i R. Rajwar, "Transactional Memory", 2nd edition, Morgan and Claypool, 2010.

[2] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, i R. Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing enviroments and evaluation of resource provisoning algorithms", *Software: Practice & Expirience*, 41, Jan. 2011.

[3] M. Popovic, B. Kordic, "PSTM: Python software transactional memory", 22nd Telecommunications Forum Telfor (TELFOR), 25-27 Nov. 2014, Belgrade, pp. 1106-1109

[4] M. Popovic, B. Kordic, I. Bašičević "DPM-PSTM: Dual-port Memory Based Python software transactional memory", 23nd Telecommunications Forum Telfor (TELFOR), 27-28 Aug. 2015, Belgrade, pp. 1106-1109

[5] H. Quarnoughi, J. Bokhobza, F. Singhoff, S. Rubini, "Integrating I/Os in Cloudsim for Performance and Energy Estimation", ACM SIGOPS Operating Systems Review - Special Topics, 3, December 2016, New York, NY, USA

[6] B. Wickremasinghe, R. N. Calheiros, R. Buyya, "CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications", 24th IEEE Internationa Conference on Advanced Information Networking and Applications (AINA). 22-23 April 2010.

[7] A. Kohne, D. Pasternak, L. Nagel, O. Spinczyk, "Evaluation of SLA-based Decision Strategies for VM Scheduling in Cloud Data Centers", In Proceedings of the 3rd Workshop on CrossCloud Infrastructures & Platforms, CrossCloud '16, pages 6:1-6:5, New York, NY, USA, 2016.

ABSTRACT

This paper presents an extension of the IaaS Cloud simulator CloudSim. First task is modeled in the form of a transaction on transactional memory and communications between the data center using the Two-Phase commit protocol. Then model of a prototype of distributed STM is implemented. At the end evaluation has been done and obtained results are compared with the previously derived theoretical results. The presented results are positive and stimulate future work in development of distributed STM.

A Solution of CloudSim Simulation of Distributed STM

Dragan Brkin, Branislav Kordić, Miroslav Popović