Repflow: minimalizovanie času doručenia toku pomocou replikácie tokov v dátových centrách

Overenie

Nadežda Juhásová
FIIT STU
Bratislava, Slovensko

Barbora Ungerová
FIIT STU
Bratislava, Slovensko

***Abstrakt*— V dátových centrách sa pomocou protokolu TCP prenášajú informácie. Tieto informácie sa prenášajú v podobe dlhých a krátkych tokov. Prenášanie tokov od zdroja k cieľu je možné po viacerých cestách siete. Prerozdelenie tokov na rôzne cesty má na starosti algoritmus ECMP. Môže sa však stať, že krátky tok sa bude nachádzať vo fronte s dlhým tokom, čo spôsobí tzv. head-of-line blokovanie, ktoré zvyšuje čas doručenia (FCT). Predchádzajúce práce, v ktorých sa pokúšali skrátiť čas dokončenia väčšinou vyžadovali zasahovanie do hardvéra alebo pozmenenie protokolu. Riešenie Repflow sa zameralo na softvérové riešenie tohto problému, pomocou replikácie každého krátkeho toku v sieti. Každý replikovaný tok pôjde v sieti inou cestou, čo zníži pravdepodobnosť, že originálny aj zreplikovaný tok budú čakať vo fronte s dlhými tokmi a tým sa skráti čas doručenia toku. Pomocou simulácie na nástroji NS-3 a implementácii riešenia na nástroji Mininet, Repflow poskytlo 50 až 70 percentné zlepšenie času doručenia tokov. My poskytujeme overenie tohto riešenia pomocou nástroja Mininet.**

Kľúčové slová—TCP,FCT, krátke toky

I. ANALÝZA

V článku bolo implementované Repflow na skrátenie FCT (Flow Completion Time) v TCP(Transmission Control Protocol) sieťach, v ktorom sa pomocou nástrojov Mininet a NS-3 replikovali krátke TCP toky. centrá, v ktorých sa ako jedným z hlavných protokolov používa protokol TCP a topológia Fat-tree.

Hlavným problémom na ktorý sa RepFlow zameral bolo čakanie krátkych tokov vo fronte s dlhými. Vďaka tomu, že v sieti v dátových centrách je viac rovnako dobrých ciest, ktorými je možné tok vyslať, krátke toky môžeme vyslať po inej ceste, kde nebudú čakať.

Cesty sa vyberajú pomocou ECMP (Equal-cost multi-path), ktorý zahashuje tok a pošle po jednej ceste. Na tejto ceste však môže nastať kolízia, ak sa na danej ceste už niečo posiela a krátky tok bude musieť čakať kým sa dokončí odosielanie paketov alebo dlhého toku. Tento jav sa nazýva head-of-line blocking. Tento problém je možné vyriešiť replikáciou toku, pretože je malá pravdepodobnosť, že oba toky budú odosielané po rovnakej ceste. Prvý doručený tok ukončí prenos.

A. FCT (Flow Completion Time)

Čas dokončenia prenosu je metrika na meranie doby prenosu informácií poslaných zo zdrojového bodu do cieľového bodu.

B. Head-of-line blocking

Je to dej, ktorý sa deje v sieťach pri posielaní paketov. Keď sa v sieťach posielajú pakety na cieľového používateľa môžu blokovať ďalšie prichádzajúce pakety. Táto situácia sa volá Head-of-line blocking.

C. Mininet

Mininet je nástroj, ktorý vytvára realistickú virtuálnu sieť. Na tejto sieti je možné zaviesť skutočný kernel, prepínač a aplikačný kód. Toto sa deje na jednom stroji napríklad na virtuálnom stroji. Na vytváranie sa využíva príkaz sudo mn, ktorý vie vytvoriť hostov, switche alebo controllery. Tento systém je založený na BSD Open Source licencí a teda je stiahnuteľný zadarmo, preto sa často využíva na vzdelávanie a testovanie.

D. RiplPOX

RiplPOX je jednoduchý kontroler v dátových centrách postavený na Ripl. RiplPOX poskytuje príklad Openflow kontrolera, ktorý používa statický opis siete na vytvorenie cesty. Nutnosťou použitia RiplPOX-u je, že musí používať rovnakú topológiu ako Mininet. Riplpox umožňuje viaccestovosť siete.

II. NÁVRH

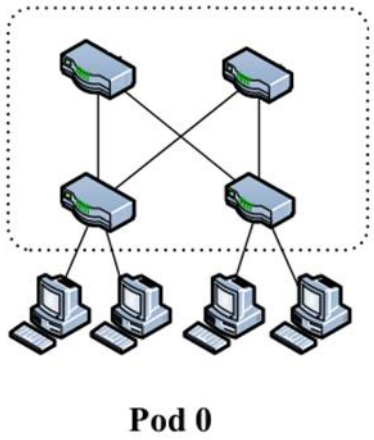
Na overenie implementácie použijeme nástroj Mininet, do ktorého implementujeme replikovanie krátkych tokov a toto

riešenie overíme na transportných štruktúrach Data mining a vyhľadávanie na webe.

Ako prvé vytvoríme topológiu, pre dané riešenie. Vytvorenie TCP spojenia v danej topológii. Po vytvorení TCP spojenia medzi jednotlivými komponentami, implementujeme replikácie krátkych tokov. Najprv vytvoríme funkciu, ktorá zhodnotí či daný tok je krátky tok, teda jeho veľkosť je do 100KB. Potom keď dorazí krátky tok, vytvoríme dva TCP sokety, cez ktoré pošleme identické pakety.

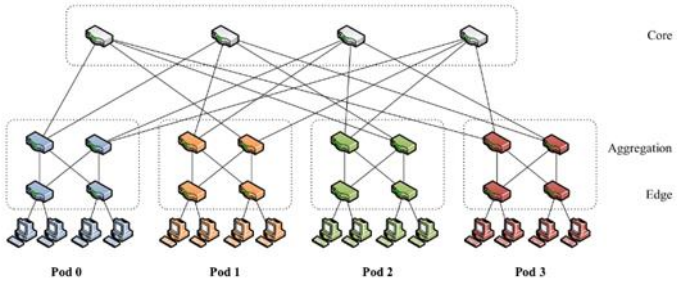
Využijeme štandrad TCP-New Reno, ktorý bol využitý v mnohých štúdiach. Inicializačné okno bude nastavené na 12KB a prepínače budú využívať DropTail queues s veľkosťou buffra 100 paketov.

Repflow využíva rovnaké parametre ako TCP, ale všetky toky menšie ako 100KB budú replikované.



Obr 1. jeden 4 pod

Náš návrh budeme testovať na začiatku na jednom pode z topológie 4-pod Fat-tree. Ako je na obrázku vyššie.



Obr 2. 4-pod Fat-tree

Z ktorých zistíme správnosť riešenia. Ďalej budeme testovať na celej štruktúre 4-pod Fat-tree(Obr. 2), ktorý bude vybalancovaný ECMP stratégiou.

A budeme porovnávať, aké výsledky dostaneme s použitím TCP a aké s použitím Repflow. Následne naše výsledky porovnáme s výsledkami, ktoré namerali tvorcovia článku RepFlow: Minimizing Flow Completion Times with Replicated Flows in Data Centers.

A. Nástroje

A budeme porovnávať, aké výsledky dostaneme s použitím TCP a aké s použitím Repflow. Následne naše výsledky porovnáme s výsledkami, ktoré namerali tvorcovia článku RepFlow: Minimizing Flow Completion Times with Replicated Flows in Data Centers.

B. Topológia a nastavenia:

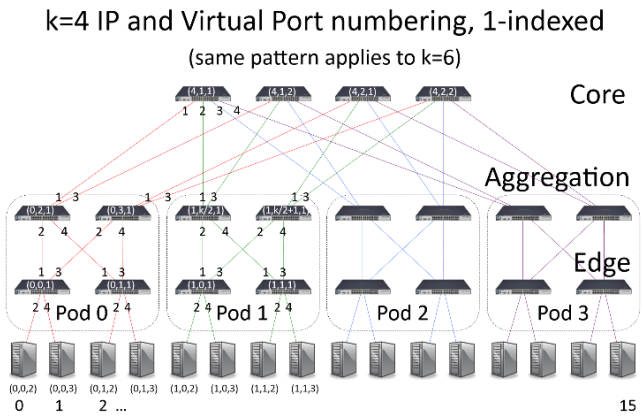
Budeme používať 4-pod Fat-tree so 16 hostami prepojené 20 prepínačmi (viď obrázok vyššie), každý so 4 portmi. Každý port má buffer s veľkosťou 50 paketov. Šírka pásma prenosu je nastavená na 20Mb a oneskorenie na 1ms, čo predstavuje minimálne oneskorenie, ktoré podporuje Mininet bez vysoko-presných časovačov.

C. Obmedzenia:

Mininet sa stáva nestabilným, keď zaťaženie presahuje 0,5, čo môže spôsobiť jeho obmedzenie v škálovateľnosti.

III. IMPLEMENTÁCIA

Našu implementáciu sme overovali na nasledujúcej topológií.



Obr 3. Topológia

Overenie riešenia Repflow bolo implemetované v jazyku python na nástroji mininet. Ako controller sme využili RiplPox, ktorý umožňuje viaccestovosť siete. Vytvorili sme topológiu Fat-three so 16-timi uzlami. Riplpox nie je kompatibilný s vyššími verziami Mininetu ako je verzia 2.0.0. Museli sme preto nainštalovať túto verziu. Následne sme

Nepodarilo sa nám však na Riplpoxe spustiť viaccestovosť. Rozhodli sme sa preto sledovať iné parametre, v tejto oblasti.

Implementovali sme posielanie správ pomocou klient-server aplikácie, v ktorej sme zaznamenávali čas odoslania paketu a čas prijatia odoslaného paketu. Tieto údaje sme potom spracovali a získali z nich vybrané parameter.

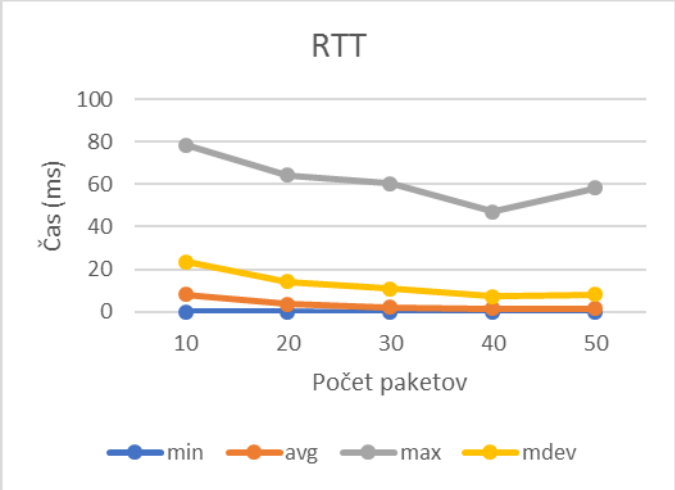
Ako ďalšiu metódu na meranie parametrov sme použili príkaz iperf. Iperf nam umožnil sledovanie priepustnosti v sieti, ktoré sme následne analyzovali a vytvorili z neho primer.

Ďalší parameter ktorý sme sledovali bola doba odozvy, tento parameter sme sledovali pomocu zaznamenaného času v aplikácii klient server. Tento čas sme potom spracovali a získali priemernú dobu odozvy.

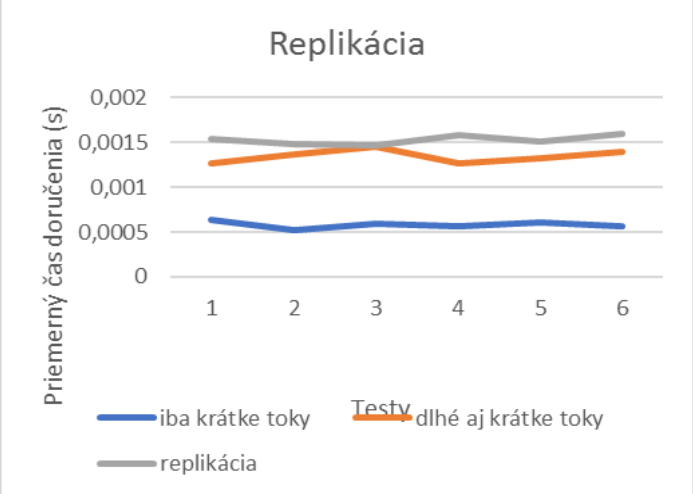
Pri tvorbe klient-server sme vytvárali dva sokety, v ktorých sa nachádzali rovnaké pakety a tie sa potom odosielali. Sledovali sme dané toky pomocou nástroja WireShark, v ktorom sme zistili, že napriek využitiu Riplpoxu sa pakety neodosielajú po rôznych cestách.

IV. Výsledky riešenia Repflow

Naše zadanie sme overovali na nástroji mininet s kontrolérom Riplpox. Pomocou príkazu ping sme zisťovali latenciu siete a zisťovali ako sa chová pri väčšom počte paketov, čo znázorňuje graf 1. nižšie.



Graf 1. RTT



Graf 2. Replikácia

Potom sme overovali ako sa pakety správajú keď je v sieti viacej komunikácií, ktoré sme kontrolovali našim skriptom, čo znázorňuje graf 2.

V. Zhodnotenie

Náš experiment nebol úspešný lebo sme nedosiahli viaccesnosť ako sme chceli a teda sme otestovali len vedľajšie parametre. Naše výsledky sme neporovnali s výsledkami tvorcov článku kvôli abstraktnosti a zložitosti výpočtov, ktoré oni sami robili.

VI. Referencie

[1] Nandita Dukkipati, Nick McKeown : Why Flow-Completion Time is the Right metric for Congestion Control and why this means we need new algorithms: Computer Systems Laboratory, Stanford University, Stanford, CA 94305-9030, USA

[2] Valter Popeskic. HOL Head-of-line blocking. https://howdoesinternetwork.com/2015/hol-head-of-line-blocking

[3] http://mininet.org/

[4] MurphyMc. 2013. RipL-POX (Ripcord-Lite for POX): A simple network controller for OpenFlow-based data centers https://github.com/MurphyMc/riplpox

[5]Karishma Sureka. 2014. Datacenters - Reduction of Broadcast traffic using SDN http://sdn-in-datacenters.blogspot.sk/2014/04/literature-survey-portland-design.html