

Forgalom igény tudatos hálózat tervezés minimális torlódással és úthosszal

Tudáskezelő rendszerek II. labor összefoglaló

Szecsődi Imre

2019

Tartalomjegyzék

1. Bevezetés	2
1.1. Motiváció	2
1.1.1. Hálózat tervezési stratégiák	2
1.1.2. Adattárházak hálózati felépítése	3
1.1.3. Újrakonfigurálás megvalósítása	3
1.2. Labor célja	3
1.3. Laborban megvalósított munka	3
2. Modell	4
2.1. Forgalom igény tudatos hálózat tervezés probléma	4
2.2. Formális felírás	4
2.2.1. Torlódás	5
2.2.2. Úthossz	5
2.2.3. Skálázhatóság	5
2.3. EgoTree	5
2.4. cl-DAN algoritmus	5
3. Megvalósítás	6
4. Teszt eredmények	7
5. Összefoglalás	8
6. Irodalomjegyzék	9

1. fejezet

Bevezetés

A labor munka a Demand-Aware Network Design with Minimal Congestion and Route Lengths [3] cikk alapján készült.

1.1. Motiváció

- A technika előrehaladásával egyre nagyobb lett a feldolgozandó adatok mennyisége
- Adattárházakban a szerverek közötti kommunikáció is ezáltal megnövekedett
- A jelenlegi hálózatok a legrosszabb esetre vannak tervezve, azaz, hogy majdnem teljes sávszélességű, kétirányú kapcsolat álljon fent bármelyik két szerver között
- A valós kommunikáció nem ezt a sémát követi, hanem túlnyomó részt megadott párok között történik a legtöbb kommunikáció

Microsoft Research ProjectORe [4].

- Nézzünk meg pár valós példát, Microsoft adattárházában 250 ezer szervert 5 production klaszterben elosztva

1.1.1. Hálózat tervezési stratégiák

- A technika fejlődésével elérhetővé váltak eszközök arra, hogy egy adott hálózatot újra konfiguráljunk, attól függően milyen terhelés éri
 - pl, korábbi kommunikációs minták alapján
- Két fő optimalizációs lehetőség van, legyen rövid az út (a) vagy legyen minimális a torlódás (b)

- A cikk bemutat egy módszert arra, hogy lehet mindkettőre majdnem optimális megoldást adni egyszerre (c)

1.1.2. Adattárházak hálózati felépítése

- Core switch
- Aggregation Swiches
- Top of Rack Switches
- In-Rack Switches

1.1.3. Újrakonfigurálás megvalósítása

- Átlag hálózatok statikusan vannak konfigurálva, nem sok lehetőséget adva annak, hogy változtassunk
 - pl. Ethernet switchek
- Optikai switchek már újra tudják konfigurálni magukat, de ezek "lassúak"
- Microsoft Research - ProjecToR[4], lézer segítségével kiváltani az optikai switcheket
 - 12 μs váltás idő (2500x gyorsabb mint egy optikai hálózati switch)

1.2. Labor célja

A labor célja a cikkben[3] bemutatott algoritmus implementálása, és annak alkalmazása különböző véletlenszerűen generált gráfokra. A kapott eredményeket össze lehet hasonlítani a megadott elméleti korlátokkal.

1.3. Laborban megvalósított munka

A labor ideje alatt elkészült egy keretrendszer, ami segítségével tesztelhető a szerzők által felvázolt algoritmus. A keretrendszer Python [2] nyelven íródott. Egy véletlen gráfok generálására egy külső csomag lett használva [1]

2. fejezet

Modell

2.1. Forgalom igény tudatos hálózat tervezés probléma

- Vegyünk egy hálózatot meghatározott számú csomóponttal
- A hálózathoz tartozik egy demand mátrix, ami leírja a valószínűségét annak, hogy i forrásból mekkora eséllyel lesz adat küldve j célba
- A cél, hogy ezen adatból egy olyan hálózati séma készítése, ami kis torlódást és rövid utakat eredményez, ez mellett még skálázható is

2.2. Formális felírás

- Adott N darab csúcspont $V = \{1, \dots, N\}$, és egy kommunikációs séma M_D ami egy $N \times N$ mátrix
- A mátrix (i, j) eleméhez tartozik egy $p(i, j)$ valószínűség, ahol i a forrás csomópont és j a cél
- A bemeneti mátrix ábrázolható egy irányított G_D gráfban, ahol az élsúlyok a két pont közötti kommunikációs valószínűség
- Az algoritmus feltétele, hogy a mátrix ritka legyen
- Egy N hálózatra a torlódást és az úthosszt útválasztási sémával fogjuk definiálni
- Egy útválasztási séma az N hálózatra $\Gamma(N)$, ami Γ_{uv} utak halmaza, ahol (u, v) párok különböző utakat jelölnek
- Γ_{uv} egy útsorozat, ami összeköti az u pontot v ponttal

2.2.1. Torlódás

1. Definíció. *A torlós jó*

2.2.2. Úthossz

2. Definíció. *Az úthossz is jó*

2.2.3. Skálázhatóság

1. Tétel. *Itt a kapitány tétel, vétel*

2.3. EgoTree

2.4. cl-DAN algoritmus

3. fejezet

Megvalósítás

4. fejezet

Teszt eredmények

5. fejezet

Összefoglalás

6. fejezet

Irodalomjegyzék

- [1] NetworkX - <http://networkx.github.io/>.
- [2] Python - [Python.org](http://python.org).
- [3] C. Avin, K. Mondal, and S. Schmid. Demand-Aware Network Design with Minimal Congestion and Route Lengths. page 9.
- [4] M. Ghobadi, D. Kilper, R. Mahajan, A. Phanishayee, N. Devanur, J. Kulkarni, G. Ranade, P.-A. Blanche, H. Rastegarfar, and M. Glick. ProjecToR: Agile Reconfigurable Data Center Interconnect. In *Proceedings of the 2016 conference on ACM SIGCOMM 2016 Conference - SIGCOMM '16*, pages 216–229, Florianopolis, Brazil, 2016. ACM Press.