Working Set Theorems for Routing in Self-Adjusting Skip List Networks

Chen Avin[1] **Iosif Salem**[2] Stefan Schmid[2]

[1] Communication Systems Engineering Department, Ben Gurion University of the Negev
[2] Communication Technologies, Faculty of Computer Science, University of Vienna

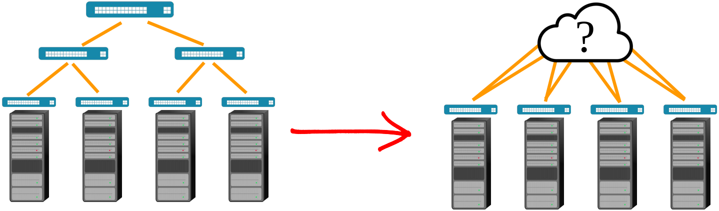
אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

universität
wien

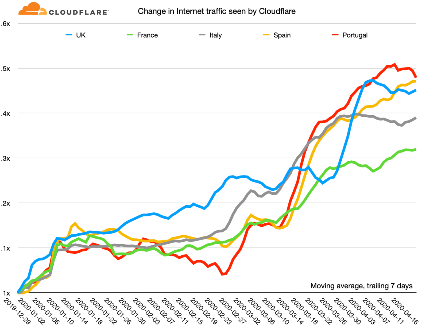
In a nutshell

Context: Self-Adjusting Networks

Goal: adjust a non-tree topology over unknown demand, minimize routing+adjustment costs



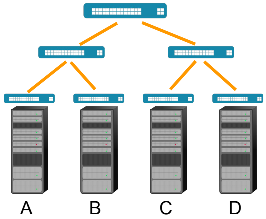
Data center traffic on the rise



- internet/data center traffic is increasing (even more in lockdowns!)

- packet switch bandwidth is increasing slower than the traffic increase rate!

A look inside: data center interconnects

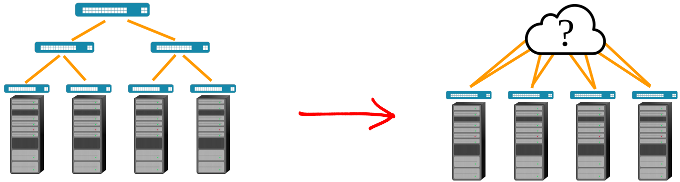


- data center top-of-the rack switch interconnects are currently **static**
- good design only for uniform demand patterns
- what if there is “elephant” traffic between (A, C) and (B, D)?

- Demand is skewed! [BAM10, GMP^{+}16]

- Need **dynamic** physical topologies!

Hardware support for dynamic connectivity

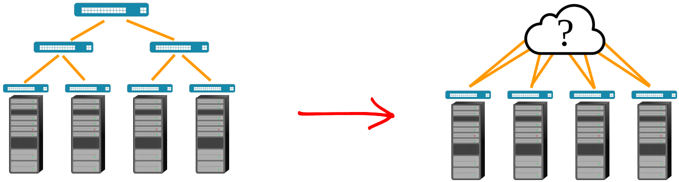


- dynamic physical topologies so far use: circuit switches, 60 GHz wireless, and **free-space optics** [GMP^{+}16]

- large number of topologies are possible (high maximum degree), low reconfiguration time

How should topology **adjust** over time to better serve the demand?

Emergence of Self-adjusting Networks (SANs)



Challenges:

1. **How** should topology change upon serving a request?

2. Is it possible to support **non tree-based** topologies?
   existing work focuses on tree-based topologies, e.g. SplayNet [SAS$^+$16]

3. What are the **performance** guarantees?

Roadmap

1. Model
2. Self-Adjusting Skip List Networks
3. Proving working set property
4. Concurrent requests

Roadmap

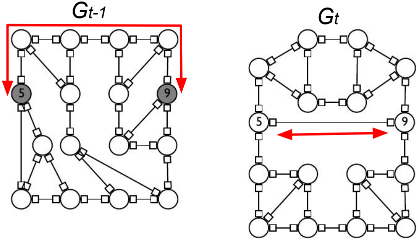
1. Model
2. Self-Adjusting Skip List Networks
3. Proving working set property
4. Concurrent requests

Abstracting SANs: input & algorithms

Input: G_0 an initial graph, $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_m)$ a sequence of communication requests

An **online SAN algorithm** $\mathcal{A}$ takes input G_0 and upon $\sigma_t = (s_t, d_t)$

- serves σ_t
- decides how to transform G_{t-1} to G_t

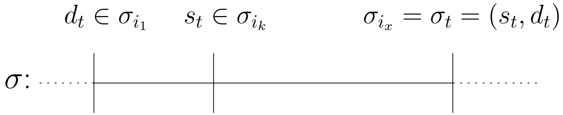


Based on [SAS+16, AS19]

SAN complexity

- $\mathit{cost}(\sigma_t)$: routing cost in G_{t-1} + cost of adjusting G_{t-1} to G_t

(pairwise) working bag $\mathit{WB}(\sigma_t)$: smallest subsequence ending in σ_{t-1} that contains both source and destination of σ_t



(pairwise) working set $\mathit{WS}(\sigma_t)$: distinct elements in working bag
(pairwise) working set number $|\mathit{WS}(\sigma_t)|$: size of working set

(pairwise) working set property: $\forall \sigma_t$: $\mathit{cost}(\sigma_t) = \mathcal{O}(\log |\mathit{WS}(\sigma_t)|)$

Roadmap

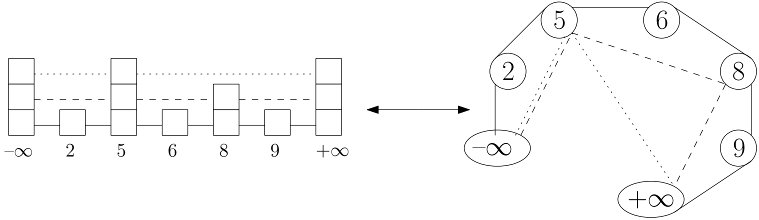
1 Model

2 Self-Adjusting Skip List Networks

3 Proving working set property

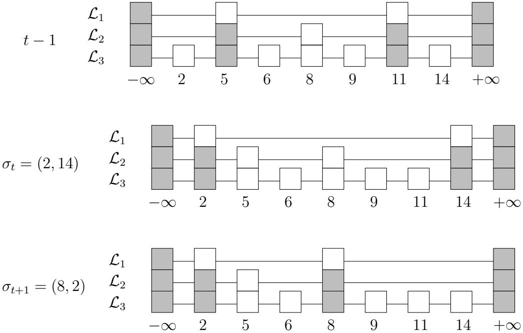
4 Concurrent requests

Skip List Networks



- element/link of the skip list = node/edge in graph (no duplicates)
- routing according to skip list finger search
- Good fit for networks due to: **local routing**, more resilient to link failures than trees, alternative to tree-based self-adjusting networks

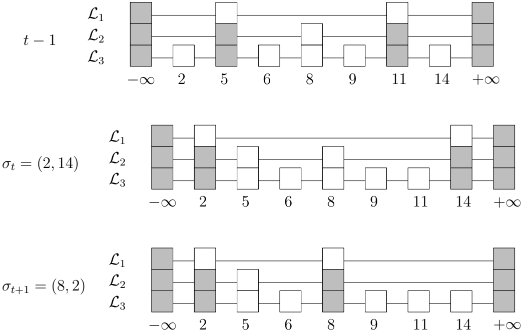
SASL^2: Self-Adjusting Skip List Network



- based on *SASL*, a statically optimal (for search sequences) self-adjusting skip list by Ciriani et al. [CFLM07]

- adjustment: promotion/demotion of nodes:
    - higher levels $\Longrightarrow$ shortest distance

SASL^2: Self-Adjusting Skip List Network



SASL^2: upon request (s, d): route (s, d), *adjust*(s), *adjust*(d)

- demoted nodes selected uniformly at random
- demotion is **graceful** and proportional to originating level

Roadmap

1 Model

2 Self-Adjusting Skip List Networks

3 Proving working set property

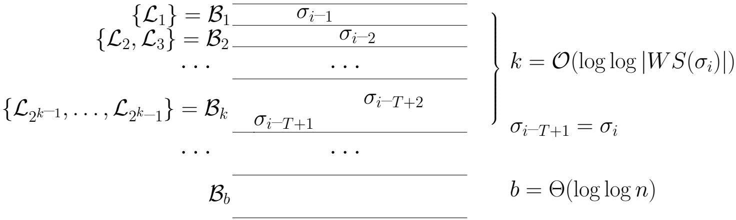
4 Concurrent requests

SASL^2 has the working set property

Step 1: Working set property for *SASL*

Fix a **search** request σ_i

Consider working bag of size T: $(\sigma_{i-T+1} \dots, \sigma_i)$



$\{\mathcal{L}_1\} = \mathcal{B}_1$ σ_{i-1}

$\{\mathcal{L}_2, \mathcal{L}_3\} = \mathcal{B}_2$ σ_{i-2}

$\dots$ $\dots$

$\{\mathcal{L}_{2^{k-1}}, \dots, \mathcal{L}_{2^k-1}\} = \mathcal{B}_k$ σ_{i-T+2} σ_{i-T+1}

$\dots$ $\dots$

$\mathcal{B}_b$

$k = \mathcal{O}(\log\log|WS(\sigma_i)|)$

$\sigma_{i-T+1} = \sigma_i$

$b = \Theta(\log\log n)$

[CFLM07]: items in working bag pushed down $\mathcal{O}(\log T)$ bands

This work: items in working bag pushed down $\mathcal{O}(\log|WS(\sigma_i)|)$ bands

$\Longrightarrow$ *SASL* has the working set property!

SASL^2 has the working set property

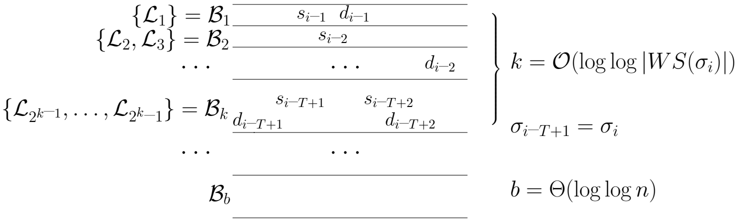
Step 2: Extending to *SASL*^{2}

Fix a communication request σ_i

Consider working bag of size T: $(\sigma_{i-T+1} \dots, \sigma_i)$

Step 2a: convert to sequence of search requests $(s_{i-T+1}, d_{i-T+1}, \dots, s_i, d_i)$, where $\sigma_t = (s_t, d_t)$

Step 2b: apply **pairwise** working set property definition!


$\{\mathcal{L}_1\} = \mathcal{B}_1$
s_{i-1} d_{i-1}
$\{\mathcal{L}_2, \mathcal{L}_3\} = \mathcal{B}_2$
s_{i-2}
$\dots$
$\dots$ d_{i-2}
$\{\mathcal{L}_{2^{k-1}}, \dots, \mathcal{L}_{2^k-1}\} = \mathcal{B}_k$
s_{i-T+1} s_{i-T+2}
d_{i-T+1} d_{i-T+2}
$\dots$ $\dots$
$\mathcal{B}_b$
$k = \mathcal{O}(\log\log|WS(\sigma_i)|)$
$\sigma_{i-T+1} = \sigma_i$
$b = \Theta(\log\log n)$

Roadmap

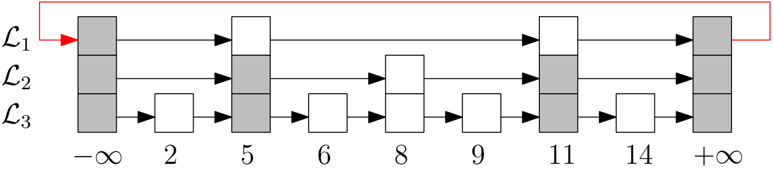
1 Model

2 Self-Adjusting Skip List Networks

3 Proving working set property

4 Concurrent requests

Serving requests concurrently



- Combine *SASL*^{2} with a **concurrent skip list implementation**, e.g. Herlihy et al. [HLLS07]

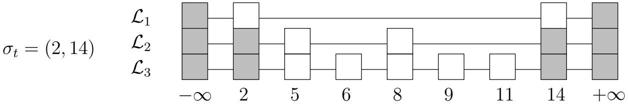
- Routing: use search routine (`findNode()`)

- Node promotion/demotion: use modified node add/delete routines

Wrap-up

- Existing/developing technology supports dynamic physical topologies [GMP+16]

- **Our contribution**: a self-adjusting skip list network with the (pairwise) working set property



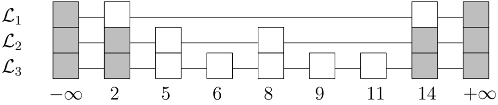
Future directions

- Lower bounds? (beyond the ones in SplayNet [SAS$^+$16])
- Extend other data structures to SANs

Thank you for your attention!

- Existing/developing technology supports dynamic physical topologies [GMP+16]
- **Our contribution**: a self-adjusting skip list network with the (pairwise) working set property

$\sigma_t = (2, 14)$



References I

- Chen Avin and Stefan Schmid, *Toward demand-aware networking: a theory for self-adjusting networks*, ACM SIGCOMM Computer Communication Review **48** (2019), no. 5, 31–40.

- Chen Avin, Iosif Salem, and Stefan Schmid, *Brief announcement: On self-adjusting skip list networks*, 33rd International Symposium on Distributed Computing, DISC 2019, October 14-18, 2019, Budapest, Hungary, 2019.

- Theophilus Benson, Aditya Akella, and David A Maltz, *Network traffic characteristics of data centers in the wild*, Proceedings of the 10th ACM SIGCOMM conference on Internet measurement, 2010, pp. 267–280.

References II

- Prosenjit Bose, Karim Douïeb, and Stefan Langerman, *Dynamic optimality for skip lists and b-trees*, Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms, Society for Industrial and Applied Mathematics, 2008, pp. 1106–1114.

- Valentina Ciriani, Paolo Ferragina, Fabrizio Luccio, and S. Muthukrishnan, *Static optimality theorem for external memory string access*, 43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings, IEEE Computer Society, 2002, pp. 219–227.

- Valentina Ciriani, Paolo Ferragina, Fabrizio Luccio, and S Muthukrishnan, *A data structure for a sequence of string accesses in external memory*, ACM Transactions on Algorithms (TALG) **3** (2007), no. 1, 6.

References III

- Monia Ghobadi, Ratul Mahajan, Amar Phanishayee, Nikhil R. Devanur, Janardhan Kulkarni, Gireeja Ranade, Pierre-Alexandre Blanche, Houman Rastegarfar, Madeleine Glick, and Daniel C. Kilper, *Projector: Agile reconfigurable data center interconnect*, Proceedings of the ACM SIGCOMM 2016 Conference, Florianopolis, Brazil, August 22-26, 2016 (Marinho P. Barcellos, Jon Crowcroft, Amin Vahdat, and Sachin Katti, eds.), ACM, 2016, pp. 216–229.

- Maurice Herlihy, Yossi Lev, Victor Luchangco, and Nir Shavit, *A simple optimistic skiplist algorithm*, Structural Information and Communication Complexity, 14th International Colloquium, SIROCCO 2007, Castiglioncello, Italy, June 5-8, 2007, Proceedings (Giuseppe Prencipe and Shmuel Zaks, eds.), Lecture Notes in Computer Science, vol. 4474, Springer, 2007, pp. 124–138.

References IV

- Stefan Schmid, Chen Avin, Christian Scheideler, Michael Borokhovich, Bernhard Haeupler, and Zvi Lotker, *Splaynet: Towards locally self-adjusting networks*, IEEE/ACM Transactions on Networking (TON) **24** (2016), no. 3, 1421–1433.
- Daniel Dominic Sleator and Robert Endre Tarjan, *Amortized efficiency of list update and paging rules*, Commun. ACM **28** (1985), no. 2, 202–208.
- Daniel Dominic Sleator and Robert Endre Tarjan, *Self-adjusting binary search trees*, Journal of the ACM (JACM) **32** (1985), no. 3, 652–686.