

# Precopy Live Migration pada Virtualisasi Internet Protocol Multimedia Subsystem Core Network

## *Precopy Live Migration on Core Network Virtualization Internet Protocol Multimedia Subsystem*

Mizanul Ahkam, Rizal Munadi dan Teuku Yuliar Arif\*

Magister Teknik Elektro, Fakultas Teknik, Universitas Syiah Kuala Darrussalam, Banda Aceh, 23111

*Received December, 2018, Accepted May, 2019*

Internet Protocol Multimedia Subsystem (IMS) merupakan suatu arsitektur topologi framework untuk mengirimkan layanan Multimedia berdasarkan IP (Internet Protocol), yang telah menjadi arsitektur framework terkemuka yang sebagai enabler dari operator untuk menawarkan layanan VoIP dan layanan multimedia. Precopy Live Migration merupakan proses migrasi yang diterapkan pada IMS dengan cara page memory pada VM ditransmisikan pada fase iterasi ketika VM sedang dalam state running pada host sumber. Kemudian VM pada sumber di suspend menggunakan fase stop-and-copy dan sisa page memory ditransmisikan menuju host tujuan. Penelitian ini berfokus pada bagaimana IMS tetap dapat melayani pada saat terjadinya migrasi. Downtime yang terlalu lama akan berakibat pada pelayanan dari servis IMS tidak maksimal. Parameter yang akan diuji adalah Downtime, Waktu Migrasi, dan Jumlah data yang di transfer.

*Internet Protocol Multimedia Subsystem (IMS) is a framework architect topology to delivery multimedia services (Internet Protocol) based which has become the latest architect framework as operator services enabler that provide VoIP and Multimedia services. Precopy Live Migration implemented to IMS with VM's memory pages are transmitted in the iteration phase, while the VM is still running at the source. After that, the VM is suspended in the stop- and-copy phase, and the remaining pages are transmitted to the destination. This research focused on how IMS can provide user during migration process. Longest downtime process will impact on service level to user. During migration process, Downtime, Migration Time, and Amount transferred data were evaluated.*

**Keywords:** downtime, migration, IMS, Virtual, precopy live migration

### Pendahuluan

Konsep virtualisasi jaringan, seperti Network Functions Virtualization (NFV), telah menjadi perhatian yang besar dari operator telekomunikasi karena kemajuan teknologi virtualisasi server dan peningkatan kinerja server. Virtualisasi jaringan dapat berkontribusi pada *End of Live* (EoL) pada hardware ATCA, lebih jauh dapat mengurangi CAPEX, OPEX dan waktu pemberian *service*, dan mempercepat *recovery* dari bencana dan sebagainya. Faktanya, ada beberapa tantangan dalam penerapan teknologi virtualisasi pada arsitektur IMS. Dalam virtualisasi, setiap elemen IMS berjalan pada mesin virtual (VM) yang dibuat dan dikelola oleh Virtual Machine Monitor (VMM) yang diinstal pada mesin fisik. Live migration adalah fitur kunci dari teknologi

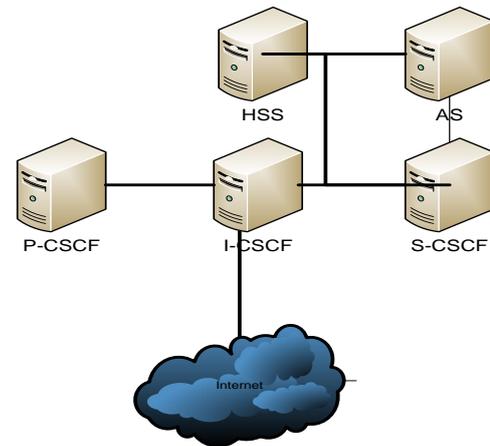
virtualisasi server. Pada teknik ini VM berpindah dari satu server fisik ke server fisik yang lain dengan waktu downtime mendekati nol (Lu et al., 2013). *Live migration* bekerja dengan cara meng-copy memory pada live VM berulang kali (iterasi) dan dapat digunakan untuk interupsi *maintenance*, *load balancing* atau penghematan energi pada *resource*. Selain itu, VM yang masih beroperasi pada server sumber juga dihentikan untuk menyelesaikan proses *live migration*, yang memiliki durasi waktu peralihan VM pada server tujuan yang mendekati nol. Oleh karena itu penting untuk menyelidiki dampak dari degradasi layanan dan penghentian operasi pada perilaku arsitektur IMS dan konfigurasi cluster HA untuk menerapkan teknik *live migration* pada arsitektur IMS yang membutuhkan *high availability*.

Fokus permasalahan pada penelitian ini adalah menerapkan *Precopy Live Migration* pada saat arsitektur IMS sedang melayani user untuk mencari nilai downtime. Tujuan penelitian ini adalah menganalisis penerapan metode *Precopy Live Migration* berdasarkan waktu downtime, waktu migrasi, serta jumlah data yang ditransfer.

*Internet Protocol Multimedia Subsystem* (IMS) merupakan suatu arsitektur topologi framework untuk mengirimkan layanan Multimedia berbasis IP (*Internet Protocol*), yang telah menjadi arsitektur framework terkemuka yang menjadi enabler dari operator untuk menawarkan layanan VoIP dan layanan multimedia. Standardisasi IMS dirancang oleh asosiasi yang bernama 3GPP yang didukung oleh *European Telecommunications Standards Institute* (ETSI) dan *Telecom & Internet Converged Service & Protocol for Advanced Network* (TISPAN)[2]. Dirancang untuk dapat bekerja dalam tipe *Multiple Access* seperti GSM, WCDMA, CDMA, Wireless dan Wireline, IMS menjadi solusi pilihan bagi operator sebagai pertumbuhan enabler substansial (Russell, 2007).

Arsitektur IMS dibagi menjadi dua bagian yaitu *Access Network* dan *Core Network*. *Core Network* dibagi menjadi dua bagian, yaitu *Home Subscriber Server* (HSS) dan *Call Session Control Function* (CSCF) (Alcatel-Lucent, 2009). HSS atau disebut juga sebagai *User Profile Server Function* (USPF) adalah database master dari user yang mendukung entitas jaringan IMS yang meng-handle panggilan (Alcatel\_Lucent, 2013). HSS terdiri dari informasi profil user yang menampilkan *authentication* dan *authorization* dari *user* dan dapat membuat informasi dari lokasi fisik user, berfungsi sama dengan *Home Location Register* (HLR) dan *Authentication Center* (AUC) pada jaringan GSM (Fokus, 2006). Hal ini dapat kita lihat pada Gambar 1.

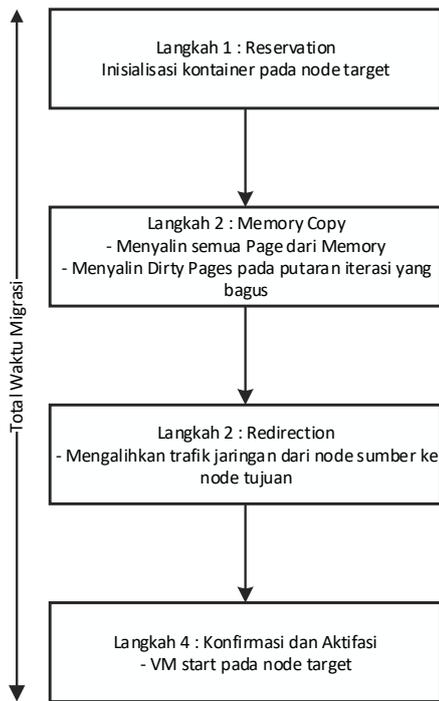
Salah satu ide terpenting dibalik *cloud computing* adalah skalabilitas, yang menjadi teknologi kunci yang terjadi, merupakan virtualisasi. Virtualisasi, dalam arti yang luas, merupakan emulasi satu dari banyak mesin/server dalam satu komputer fisik (Sampaio and Barbosa, 2016). Mudahnya, virtualisasi merupakan emulasi hardware platform software. Virtualisasi memungkinkan satu komputer mengambil banyak peran dari banyak komputer.



Gambar 1. Arsitektur IMS (Khosroshahy, 2006)

Langkah migrasi pada virtualisasi XEN adalah dipilih host (node) target yang memiliki resource untuk melakukan migrasi pada VM yang telah terjadi overload. Permasalahannya adalah, inisiasi dari ketersediaan resources. Tahap kedua terdiri dari copy seluruh page memory menuju node target sebagai iterasi pertama dan dirty page sebagai iterasi berikutnya. Pada tahap ketiga, trafik jaringan dialihkan dari VM yang telah overload menuju VM yang telah dikopi pada node tujuan. Tahap terakhir, yaitu langkah empat yaitu sebagai konfirmasi penerimaan dari VM yang telah di migrasi. Copy dari memory pada node target menjadi primary dan VM yang dimigrasi menjadi aktif. Waktu migrasi yang diambil adalah dimulai dari waktu inisial kontainer (tahap 1) hingga konfirmasi dan aktivasi VM pada node target (tahap 4) (Fakhfakh, 2009). Hal ini dapat digambarkan dengan skema bagan pada Gambar 2.

Pada saat ini, algoritma precopy merupakan algoritma live migration yang digunakan oleh hypervisor populer (XEN, KVM, VMWare, dan lain-lain). Algoritma precopy merupakan algoritma yang mentransmisikan page memory pada fase iterasi ketika VM sedang running pada host sumber. Kemudian VM di-suspend sehingga sisa page memory ditransmisikan ke host tujuan. Algoritma precopy dapat membuat waktu downtime lebih baik pada algoritma stop-and-copy, namun memiliki kekurangan dapat meningkatkan waktu migrasi dan konsumsi bandwidth (Ruan et al, 2016). Kelebihan utama algoritma precopy adalah reliabilitas dan kehandalan, karena jika proses live migration akan diulang ke kondisi awal. maka proses migrasi mengalami kegagalan.



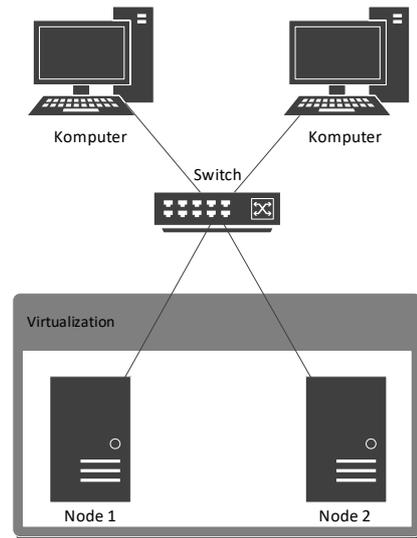
Gambar 2. Virtualisasi XEN pada saat Live Migration (Fakhfakh, 2009).

**Metode**

Pada bagian metode, terbagi menjadi beberapa bagian. Pada bagian pertama menjelaskan mengenai eksperimen model arsitektur yang ditunjukkan dengan Gambar 3. Berikutnya menjelaskan mengenai hardware dan software yang digunakan pada penelitian ini. Pada bagian terakhir menjelaskan mengenai precopy live migration dengan parameter yang diukur waktu migrasi, jumlah data yang ditransfer dan downtime.

**Test Bed**

Berikut merupakan model eksperimen yang diterapkan pada penelitian ini seperti yang digambarkan pada Gambar 3. Terdapat dua buah komputer sebagai pengujian sekaligus bertindak sebagai client, satu buah switch yang bertindak sebagai penghubung, dan dua buah node yang bertindak sebagai host masing-masing VM. Pada node 1 akan menyimpan 4 buah VM yang merepresentasikan masing-masing elemen IMS, yaitu HSS, PCSCF, ICSCF dan SCSCF. Keuntungan dalam menggunakan teknologi virtualisasi adalah dapat menghemat sumber daya resource dari server fisik, karena dalam satu server fisik dapat menampung lebih dari satu Virtual Machine yang bertindak sama dengan satu server fisik.



Gambar 3. Model jaringan test bed

Untuk memungkinkan terjadinya Live Migration, pada salah satu komputer (Gambar 3) akan dijalankan software pembebanan server. Sehingga, pada saat resource VM telah mencapai batas threshold, IMS akan melakukan Live Migration. Node 2 memiliki resource yang lebih besar dibanding node 1, sehingga VM yang berpindah ke node 2 akan memiliki resource yang lebih besar juga dibandingkan pada saat VM berada pada node 1.

**Software dan Peralatan**

Keseluruhan eksperimen ini menggunakan tiga buah PC x86\_64 dan 2 buah laptop yang terhubung melalui satu buah switch kecepatan 1000Mbps pada jaringan IPv4. Node 1 dengan spesifikasi prosesor Intel® Core™ i3 (4 thread) generasi ke 2, RAM 8GB. Sedangkan node 2 dengan spesifikasi yang lebih tinggi, Intel® Core™ i3 generasi ke 4 (4 thread), dan RAM 12GB. Seluruh node menggunakan sistem operasi debian 8.8.0. Laptop merepresentasikan sebagai client yang akan melakukan komunikasi. Software yang akan digunakan adalah OpenIMScore yang di-develop oleh FOKUS pada Fraunhofer Institute (Fokus, 2006) sebagai server arsitektur. Sedangkan klien yang melakukan koneksi komunikasi menggunakan software Boghe. SIPp digunakan untuk pembebanan yang akan diletakkan pada salah satu laptop atau terminal yang tersedia. Sedangkan untuk menganalisa dan meng-capture paket menggunakan software Wireshark dan Tshark untuk versi command line (Wireshark, 2018).

```
xentop - 03:23:31 Xen 4.4.1
5 domains : 1 running, 4 blocked, 0 paused, 0 crashed, 0 dying, 0 shutdown
Mem: 8298944k total, 6125556k used, 2173388k free CPUs: 4 @ 3092MHz
```

NAME	STATE	CPU(sec)	CPU(%)	MEM(k)	MEM(%)	MAXMEM(k)	MAXMEM(%)	VCPUS
Domain-0	-----r	18	0.6	4194304	50.5	6291456	75.8	1
hss	--b---	4	0.1	1048576	12.6	1049600	12.6	1
icscf	--b---	3	0.0	262144	3.2	263168	3.2	1
pcscf	--b---	3	0.0	262144	3.2	263168	3.2	1
scscf	--b---	3	0.0	262144	3.2	263168	3.2	1

NETS	NETTX(k)	NETRX(k)	VBDS	VBD_00	VBD_RD	VBD_WR	VBD_RSECT	VBD_WSECT
0	0	0	0	0	0	0	0	0
1	79	3	1	0	1816	93	116226	1584
1	74	3	1	0	1211	69	56794	1160
1	76	3	1	0	1218	68	51458	1200
1	70	3	1	0	1207	60	54490	1120

Gambar 4. Xentop pada Dom0

**Strategi Precopy Live Migration**

Strategi dalam menganalisis precopy Live Migration adalah dengan membuat script yang akan melakukan monitoring terhadap seluruh resource dari masing-masing VM dengan memanfaatkan fitur dari XEN yaitu xentop seperti pada Gambar 4. Ketika salah satu dari elemen arsitektur IMS mengalami overload pada resource, sehingga menyebabkan nilai resource telah melebihi 90% (Tsakalozos,2017), maka live migration akan terjadi dengan perintah live migration dari node 1 ke node 2. Setelah VM yang baru saja dimigrasikan telah diaktifkan kembali, mengikuti VM yang lain untuk bermigrasi. Gambar 5 merupakan gambar ketika terjadi live migration.

**Prosedur Pengukuran**

Parameter yang akan diukur adalah downtime, jumlah data yang ditransfer, serta lama waktu yang dibutuhkan untuk migrasi. Prosesi pengukuran dilakukan pada saat terjadinya prosedur live migration dari node sumber menuju node target. Pengukuran downtime dilakukan dengan menghitung berapa lama waktu yang dibutuhkan oleh VM selama proses mati hingga hidup kembali. Proses mati (halt) yang terjadi pada node 1 dan proses hidup (start) pada node 2. Waktu migrasi dihitung sejak inisial kontainer pada node target

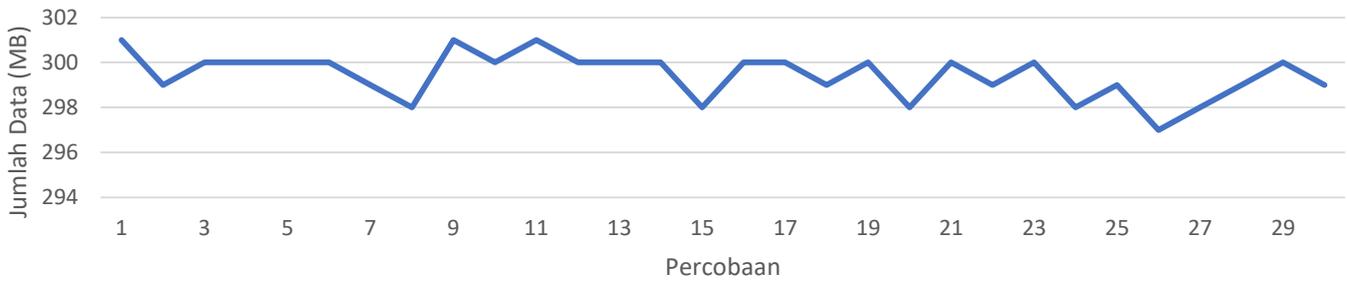
hingga waktu pengaktifan VM yang di migrasikan pada node target. Pengukuran jumlah data yang ditransfer merupakan pengukurann jumlah data dari node sumber menuju node target dalam bentuk byte.

```
root@n1:~/hasil# xl migrate hss n2.mzn.com
Saving to migration stream new xl format (info 0x0/0x0/624)
migration target: Ready to receive domain.
Loading new save file <incoming migration stream> (new xl
fmt info 0x0/0x0/624)
Savefile contains xl domain config
xc: progress: Reloading memory pages: 7168/131072 5%
xc: progress: Reloading memory pages: 13312/131072 10%
xc: progress: Reloading memory pages: 20480/131072 15%
xc: progress: Reloading memory pages: 26624/131072 20%
xc: progress: Reloading memory pages: 32768/131072 25%
xc: progress: Reloading memory pages: 39936/131072 30%
xc: progress: Reloading memory pages: 46080/131072 35%
xc: progress: Reloading memory pages: 53248/131072 40%
xc: progress: Reloading memory pages: 59392/131072 45%
xc: progress: Reloading memory pages: 65725/131072 50%
migration target: Transfer complete, requesting permission
to start domain.
migration sender: Target has acknowledged transfer.
migration sender: Giving target permission to start.
migration target: Got permission, starting domain.
migration target: Domain started successfully.
migration sender: Target reports successful startup.
Migration successful.
```

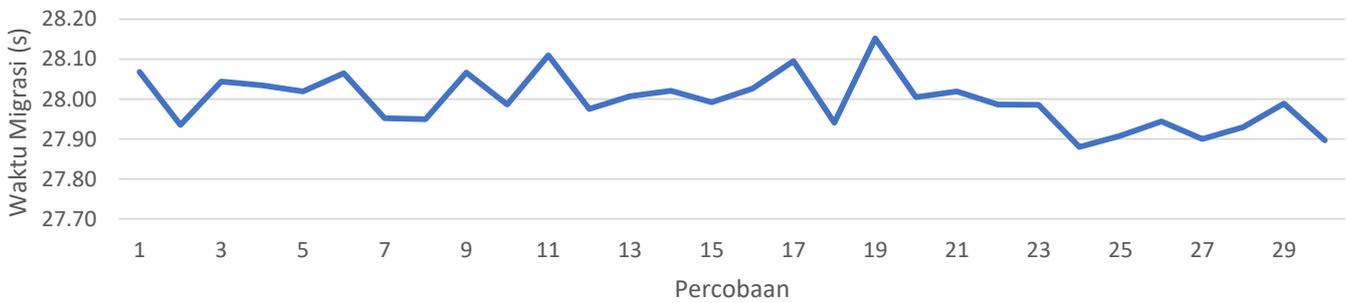
Gambar 5. Tampilan log pada terminal pada saat terjadi live migration

**Hasil dan Pembahasan**

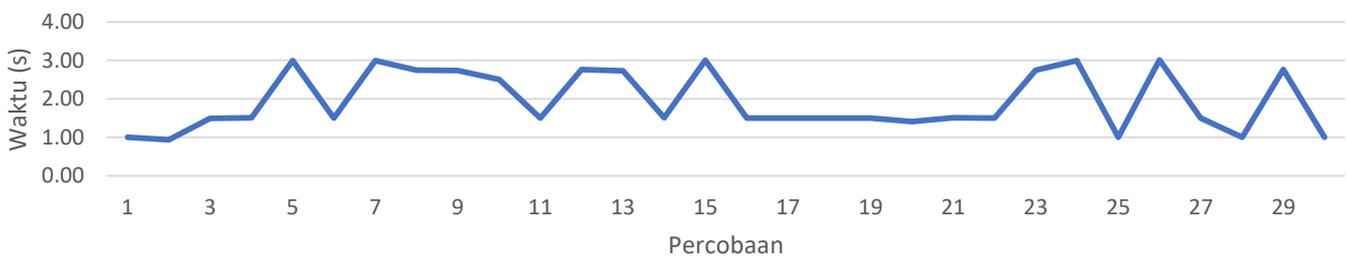
Setelah dilakukan pengukuran sesuai dengan prosedur pengukuran di atas, maka hasil yang didapat adalah sebagai berikut, dari grafik pada Gambar 6, jumlah data dari masing-masing



Gambar 6. Grafik jumlah data



Gambar 7. Grafik waktu migrasi



Gambar 8. Grafik waktu downtime

percobaan bernilai tidak tetap. Hal ini bergantung pada jumlah user yang sedang melakukan panggilan baru, dan yang sedang dalam status aktif panggilan. Pada percobaan 26 dengan nilai 297 MB merupakan jumlah data yang terkecil. Sedangkan jumlah data yang tertinggi ada pada percobaan 1, 9 dan 11 dengan nilai 301 MB. Pada grafik Gambar 6 memiliki garis tren menurun.

Pada percobaan pengukuran waktu migrasi, didapat hasil seperti pada Gambar 7. Jumlah data dari masing-masing percobaan bernilai tidak tetap. Hal ini juga disebabkan oleh hal yang sama yaitu bergantung pada jumlah user yang sedang melakukan panggilan baru dan yang sedang dalam status aktif panggilan. Pada percobaan 19 merupakan waktu migrasi dengan perolehan terlama dengan nilai 28,15 s, yang diikuti dengan percobaan 20 dengan nilai 28 s. Sedangkan percobaan 24

merupakan waktu migrasi tercepat dengan nilai 27,88 s. Pada gambar memiliki garis tren menurun.

Hasil perolehan waktu downtime pada precopy live migration terlihat pada Gambar 8, percobaan 16 hingga percobaan 22 memiliki nilai waktu downtime yang cenderung sama. Pada percobaan 2 merupakan waktu downtime tercepat yang dialami oleh arsitektur IMS pada saat terjadi live migration dengan nilai 0,94 s. Sedangkan downtime terlama terjadi pada saat percobaan ke 26 dengan nilai 3,01 s. Kecenderungan waktu downtime yang sama terjadi pada percobaan 16 hingga 22 bernilai antara 1,4 hingga 1,5 s.

**Kesimpulan**

Pada penelitian ini, kami mengimplementasikan algoritma precopy live migration pada jaringan arsitektur IMS-Core. Proses precopy live migration memiliki kelebihan

dapat mentoleransi kesalahan yang diakibatkan pada saat migrasi berlangsung. Jika terjadi kesalahan maka proses live migration akan dibatalkan.

### Referensi

- F. Lu, H. Pan, X. Lei, X. Liao, and H. Jin, "A virtualization-based Cloud infrastructure for IMS core network," *Proc. Int. Conf. Cloud Comput. Technol. Sci. CloudCom*, vol. 1, pp. 25–32, 2013.
- T. Russell, "the Ip Multimedia Subsystem ( Ims )," *Security*, vol. 45, no. 7, p. 224, 2007.
- Alcatel-Lucent, Internet Protocol Multimedia Subsystem (Ims) Solution Maintenance Release 07.00.05 And Controlled Introduction Release 07.01.00, no. 1. *Alcatel-Lucent*, 2009.
- Alcatel-Lucent, IP Multimedia Subsystem, no. September. *Alcatel-Lucent*, 2013.
- Fokus, "IP Multimedia System ( IMS ) - Principles , Architecture and Applications," pp. 1–69, 2006.
- M. Khosroshahy, "Architectural Overview of IP Multimedia Subsystem -IMS," no. June, pp. 1–35, 2006.
- A. M. Sampaio and J. G. Barbosa, *Energy-Efficient and SLA-Based Resource Management in Cloud Data Centers*, 1st ed., vol. 100. Elsevier Inc., 2016.
- M. Fakhfakh, I. Limam Bedhiaf, O. Cherkaoui, and M. Frikha, "High availability in IMS virtualized network," 2009 1st Int. Conf. Commun. Networking, ComNet 2009, pp. 0–5, 2009.
- Y. Ruan, Z. Cao, and Z. Cui, "Pre-Filter-Copy: Efficient and Self-Adaptive Live Migration of Virtual Machines," *IEEE Syst. J.*, vol. 10, no. 4, pp. 1459–1469, 2016.
- Wireshark.org, "Wireshark User ' s Guide," pp. 0–291, 2018.
- K. Tsakalozos, V. Verroios, M. Roussopoulos, and A. Delis, "Live VM Migration under Time-Constraints in Share-Nothing IaaS-Clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 8, pp. 2285–2298, 2017