Optimising Virtual Network Functions Migrations: A Flexible Multi-Step Approach

Nguyễn Tuấn Khải, Andreas Baumgartner, Thomas Bauschert
Technische Universität Chemnitz
Chair of Communication Networks
09126 Chemnitz, Germany
Email: [tuan-khai.nguyen | andreas.baumgartner | thomas.bauschert]@etit.tu-chemnitz.de

***Abstract*—In this paper, we introduce a novel optimisation model for *virtual network functions* (VNFs) migration in multiple steps. Since VNF migration can be enabled by VM migration, we model the course of VM migration by applying the concept of *time-expanded networks*. Our model is related to virtual network embedding (VNE), i.e., the problem of mapping virtual networks, or service chains (SCs), onto a capacitated substrate network. Contrary to the classical static VNE problem, we focus on finding an optimum transition from one mapping to another through one or multiple intermediate mappings. The concept of multi-step VM migration was first introduced in a previous publication, in which a fixed migration deadline per step was imposed. In this paper we remove this restriction, making the migration model more flexible. The VNF migration problem is formulated as a mixed integer linear program (MILP) and solved by a commercial solver. The performance of our new approach is evaluated via simulations assuming a realistic substrate network topology and SCs. The results show significant performance improvements w.r.t. migration time, failure sensitivity, feasibility, and cost of migration, especially in the case of high network utilisation.**

***Index Terms*—virtual network embedding, virtual machine migration, virtual network function, multi-step VM migration**

I. Introduction and Related Work

VM migration is driven by the necessity for flexible adaptation of virtual networks due to fluctuations in client demands or changes in the substrate network [1]. It can take place either within a data center (DC) (intra-DC level) or across DCs (inter-DC level). In this paper, we focus on the latter case, where the possibilities of relocating resources amongst different DC locations are considered.

We are aware that migrating VMs between DCs could pose a number of technical problems. In general, migrating VMs across DCs entails altering the VMs' IP addresses, which might break the involved ongoing applications. The problem of keeping the IP addresses during a migration of VMs (i.e., live migration) can be tackled by certain virtual networking technologies such as VXLAN, vCDNI, and NVGRE [2].

While *pre-copy* migration [3] is widely applied, the implementation of *post-copy* migration [4] is still somewhat challenging and known for being prone to failure. The pre-copy method, on the other hand, ensures low service downtime, and is therefore less affected by failure.

978-1-5386-9376-6/19/$31.00 ©2019 IEEE

In our previous publication [5], a mathematical optimisation model for pre-copy VM migration was proposed, introducing the concept of *soft bottlenecks*, which refers to the preoccupation of resources by existing VNFs. When several VM migrations take place, some of them could result in a release of resources that other migrations can use, so rather than settling at their destinations, some VMs migrate to so-called *stop-over nodes* (interim locations). However, a hard restriction on the "migration dealine" [6] could pose efficiency and feasibility issues. In this paper, we remove this restriction and thereby obtain a larger solution space, yielding more adaptable and flexible solutions.

Espositio et al. in [7] cope with pre-copy migration of multiple VMs by developing a GeoMig program to optimise and balance both the migration time and service downtime. Their concept of *total migration time* refers to the sum of migration time over the copy rounds of a single step only, rather than the overall time of many steps which we seek to optimise. In our paper, we give a conservative estimation of the migration time based on the VM size and transmission rate. To realise pre-copy migration, it is crucial to keep the page dirtying rate below the transmission rate [8]. As long as this is fulfilled, the migration time of a VM is bounded and converges to a finite value, no matter how many copy rounds might be taken.

Schaffrath et al. in [9] assign a variety of costs to different parts of the VM migration procedure. While this consideration could be valuable for any future migration approaches, the mathematical model presented in [9] aims to give an incentive to migrate VMs, purely by setting migration flags on the VMs that need to migrate, rather than indicating which traffic routes or how much bandwidth is used.

To the best of our knowledge, no work has analysed the prospects of multi-step VM migration. In the subsequent sections, we show that our multi-step approach could outperform single-step migration methods in several aspects. The rest of this paper is structured as follows: Chapter II provides a description of the problem statement, the optimisation model and its mathematical formulation. In Chapter III, performance evaluation results are outlined. Lastly, Chapter IV gives a brief summary and a discussion of ideas for future research.

II. Problem Description & Mathematical Model

A. Network Models

1) Substrate Network: The *substrate network* describes a network of interconnected DCs (inter-DC scenario) and is modelled as a directed graph $G^{\mathbf{S}} = (N^{\mathbf{S}}, L^{\mathbf{S}})$ with $N^{\mathbf{S}}$ being the set of *substrate nodes* and $L^{\mathbf{S}} \subseteq N^{\mathbf{S}^2}$ of *substrate links*. Each substrate node $s \in N^{\mathbf{S}}$ represents a DC that accommodates servers and switching devices which are responsible for hosting VMs and realising *virtual networks*. The parameter ζ^s_γ denotes the total capacity of a substrate node s w.r.t. a certain type of resource $\gamma \in \Gamma$, where $\Gamma = \{\text{‘pro’}, \text{‘mem’}\}$, standing for “processing power” and “memory”. For full-duplex connections, the links $(s, \tilde{s})$ and $(\tilde{s}, s)$ are set to have the same bandwidth $b^{(s,\tilde{s})} = b^{(\tilde{s},s)}$.

2) Virtual Networks: The set of virtual networks is denoted by R. A virtual network $r \in R$ is also represented by a directed graph $G^{\mathbf{V}}_r = \left(N^{\mathbf{V}}_r, L^{\mathbf{V}}_r\right)$ where $N^{\mathbf{V}}_r$ is the set of virtual nodes and $L^{\mathbf{V}}_r \subseteq {N^{\mathbf{V}}_r}^2$ of virtual links. Each virtual node $v_r \in N^{\mathbf{V}}_r$ represents a VM which is embedded on a physical node and demands an amount $d^{v_r}_\gamma$ of resources of type γ. The virtual links $L^{\mathbf{V}}_r$ model the communication relationships amongst the virtual nodes, whose bandwidth demands are denoted by $d^{(v_r,\tilde{v}_r)}$ and assumed to be fixed.

B. Problem Statement

A common objective of a VNE problem is to find for each $r \in R$ a mapping $g_r : N^{\mathbf{V}}_r \to N^{\mathbf{S}}$ and $h_r : L^{\mathbf{V}}_r \to \mathcal{P}(L^{\mathbf{S}})$, typically referred to as *node mapping* and *flow mapping* [1], respectively. A direct transition of one (node and flow) mapping to another by means of VM migration is considered single-step migration. If this transition involves many intermediate mappings, it is a multi-step migration.

Given the current mapping, our VM migration optimisation approach involves two phases. Phase I represents a classical VNE problem where the optimal target mapping is determined (which might be different from the current mapping). Phase II copes with the VM migration process from the current mapping towards the optimum one. Essentially, it extends the VNE concept by employing the idea of *time-expanded networks* [10], in which multiple copies of the same graph are generated.

In Phase I, a target embedding is found such that a certain objective is fulfilled. We define by $y^{v_r}_s \in \{0, 1\}$ and $y^{(v_r,\tilde{v}_r)}_{(s,\tilde{s})} \in \mathbb{R}^+_0$ the node- and flow-mapping variables which respectively decide whether a virtual node v_r is embedded in DC s and how much bandwidth is occupied by a virtual link $(v_r, \tilde{v}_r)$ on a substrate link $(s, \tilde{s})$. The values of these decision variables are obtained as the solution from Phase I. The parameters $o^{v_r}_s \in \{0, 1\}$ and $o^{(v_r,\tilde{v}_r)}_{(s,\tilde{s})} \in \mathbb{R}^+_0$ indicate the current mapping state of the nodes and links. The objective of Phase II is to find multiple intermediate migration steps while maintaining the running services, such that the total migration time (TMT) is minimised and the embedding in the final step coincides the target embedding derived from Phase I.

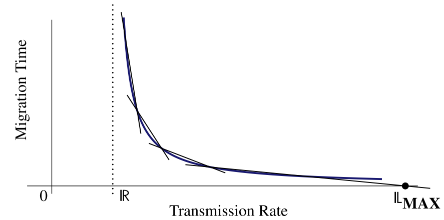


Fig. 1. Piece-wise linear approximation of the migration time.

C. Mathematical Problem Formulation

As many models are already available for the classical VNE problem (see [1]), any of which could be used for Phase I, we focus on Phase II, i.e., the multi-step VM migration optimisation. We adopt the concept of time-expanded networks by expanding—in a mathematical sense—the substrate network to a structure composed of multiple layers, where the transition from one layer to the next represents a migration step. The substrate network $G^{\mathbf{S}}$ is replicated for each layer indexed with $i \in \Psi = \{0, 1, \ldots, \psi\}$, where ψ denotes the maximum number of migration steps. Mathematically speaking, we now model a time-expanded graph with $N^{\mathbf{S}}_\psi = N^{\mathbf{S}} \times \Psi$ and $L^{\mathbf{S}}_\psi = L^{\mathbf{S}} \times \Psi$. The following auxiliary index sets are defined for convenience:

$$\Psi^0 = \{0, 1, \ldots, \psi - 1\} \text{ and } \Psi^+ = \{1, 2, \ldots, \psi\}$$

In Phase II, the former variables $y^{v_r}_s$ and $y^{(v_r,\tilde{v}_r)}_{(s,\tilde{s})}$ from Phase I now become input parameters.

We adopt the approach outlined in [11] to determine the time for pre-copy VM migration. The migration time of a VM $t_{\mathbf{mig}}$ depends on factors such as its image size (or memory footprint), page dirtying rate, transmission bandwidth, switch-over goal time, and minimum progress, denoted by $\mathbb{M}, \mathbb{R}, \mathbb{L}, \mathbb{T}$, and $\mathbb{X}$, respectively. For a live VM migration with these given parameters, the migration time is then $t_{\mathbf{mig}} = \max\left(\frac{\mathbb{M}(\mathbb{L}+\mathbb{R})}{\mathbb{L}^2}, \min\left(\frac{\mathbb{M}-\mathbb{T}\mathbb{R}}{\mathbb{L}-\mathbb{R}}, \frac{\mathbb{M}\mathbb{L}(\mathbb{L}-\mathbb{R})-\mathbb{X}\mathbb{R}^2}{\mathbb{L}(\mathbb{L}-\mathbb{R})^2}\right)\right)$.

In this work, we simplify the formula above by setting both $\mathbb{T}$ and $\mathbb{X}$ of all VMs to zero[1]. It then reduces to $t_{\mathbf{mig}} = \frac{\mathbb{M}}{\mathbb{L}-\mathbb{R}}$, yielding a conservative estimation of the VM migration time, which is a convex function w.r.t. the transmission rate $\mathbb{L}$ and hence can be conveniently approximated in a piece-wise linear manner (see Figure 1 and Constraints (9)).

For each layer $i \in \Psi$ we define $\tilde{f}^{(v_r,\tilde{v}_r)}_{(s,\tilde{s}),i} \in \mathbb{R}^+_0$ as a migration mapping variable indicating how much bandwidth of $(s, \tilde{s})$ on layer i is used for migrating v_r. We further define $x^{v_r}_{s,i} \in \{0, 1\}$ and $f^{(v_r,\tilde{v}_r)}_{(s,\tilde{s}),i} \in \mathbb{R}^+_0$ as node- and flow-mapping variables, specifying whether v_r is mapped onto s, as well as how much bandwidth on $(s, \tilde{s})$ is allocated to $(v_r, \tilde{v}_r)$ on layer i. An auxiliary variable $\eta^{v_r}_{s,i}$ represents the result of $\left(x^{v_r}_{s,i} \vee x^{v_r}_{s,i+1}\right)$.

Our goal in Phase II is to minimise the total migration time (TMT), whose benefit could be two-fold. First, since the result

[1]The meaning of setting these parameters to zero is two-fold: 1) no disruption time and 2) a complete make-before-break transition; both are ideal conditions, but give an upper bound to the formula.

of Phase I is often dictated by impermanent factors such as energy prices or customer demands, the faster this embedding is achieved, the less likely they may have changed. Second, the cost of migration (CoM) relates (either directly or indirectly) to the data volume transmitted during the migration. For a single VM and a fixed number of migration hops:

$$\begin{aligned} \text{CoM} &\sim t_{\textbf{mig}} \cdot \mathbb{L} \\ &\sim t_{\textbf{mig}} \cdot \mathbb{R} + \mathbb{M} \end{aligned}$$

In this case, one could expect that minimising the migration time also reduces the CoM. This however may neither apply to concurrent migrations nor ensure a direct relationship between the CoM and $\mathbb{M}$, since the CoM generally relies on the number of hops traversed by the traffic. Facing the objective of migration time minimisation, we also seek to investigate its implications for the CoM.

The mathematical formulation of Phase II is as follows:

Minimise $$\sum_{i \in \Psi^0} t^i_{\textbf{mig}} \qquad (0)$$

Subject to

Boundary Condition Constraints

$$f^{(v_r,\tilde{v}_r)}_{(s,\tilde{s}),0} = o^{(v_r,\tilde{v}_r)}_{(s,\tilde{s})}, \quad f^{(v_r,\tilde{v}_r)}_{(s,\tilde{s}),\psi} = y^{(v_r,\tilde{v}_r)}_{(s,\tilde{s})}, \quad \forall r \in R, (v_r,\tilde{v}_r) \in L^{\mathbf{V}}_r, (s,\tilde{s}) \in L^{\mathbf{S}}$$

$$x^{v_r}_{s,0} = o^{v_r}_s, \quad x^{v_r}_{s,\psi} = y^{v_r}_s, \qquad \forall r \in R, v_r \in N^{\mathbf{V}}_r, s \in N^{\mathbf{S}} \qquad (1)$$

Let $\varphi^{v_r}_{s,i} = \mathbb{L}^{v_r}_i \cdot x^{v_r}_{s,i}$ and $\tilde{\varphi}^{v_r}_{s,i} = \mathbb{L}^{v_r}_i \cdot x^{v_r}_{s,i+1}$, then these products are linearised as follows

$$\begin{aligned} &\varphi^{v_r}_{s,i} \geq \mathbb{L}^{v_r}_i + \mathbb{L}^{v_r}_{\textbf{MAX}} \cdot x^{v_r}_{s,i} - \mathbb{L}^{v_r}_{\textbf{MAX}} \\ &\tilde{\varphi}^{v_r}_{s,i} \geq \mathbb{L}^{v_r}_i + \mathbb{L}^{v_r}_{\textbf{MAX}} \cdot x^{v_r}_{s,i+1} - \mathbb{L}^{v_r}_{\textbf{MAX}} \\ &\varphi^{v_r}_{s,i} \leq \mathbb{L}^{v_r}_{\textbf{MAX}} \cdot x^{v_r}_{s,i}, \qquad \tilde{\varphi}^{v_r}_{s,i} \leq \mathbb{L}^{v_r}_{\textbf{MAX}} \cdot x^{v_r}_{s,i+1} \\ &\varphi^{v_r}_{s,i} \leq \mathbb{L}^{v_r}_i, \quad \tilde{\varphi}^{v_r}_{s,i} \leq \mathbb{L}^{v_r}_i, \qquad \forall r \in R, v_r \in N^{\mathbf{V}}_r, s \in N^{\mathbf{S}}, i \in \Psi^0 \qquad (2) \end{aligned}$$

Flow Conservation Constraints

$$\sum_{(s,\tilde{s}) \in L^{\mathbf{S}}} f^{(v_r,\tilde{v}_r)}_{(s,\tilde{s}),i} - \sum_{(\tilde{s},s) \in L^{\mathbf{S}}} f^{(v_r,\tilde{v}_r)}_{(\tilde{s},s),i} = d^{(v_r,\tilde{v}_r)} \left(x^{v_r}_{s,i} - x^{\tilde{v}_r}_{s,i} \right), \quad \forall r \in R, (v_r,\tilde{v}_r) \in L^{\mathbf{V}}_r, s \in N^{\mathbf{S}}, i \in \Psi^+ \qquad (3)$$

Migration Flow Constraints

$$\sum_{(s,\tilde{s}) \in L^{\mathbf{S}}} \tilde{f}^{v_r}_{(s,\tilde{s}),i} - \sum_{(\tilde{s},s) \in L^{\mathbf{S}}} \tilde{f}^{v_r}_{(\tilde{s},s),i} = \mathbb{L}^{v_r}_i \left(x^{v_r}_{s,i} - x^{\tilde{v}_r}_{s,i+1} \right), \quad \forall r \in R, v_r \in N^{\mathbf{V}}_r, s \in N^{\mathbf{S}}, i \in \Psi^0 \qquad (4)$$

Resource Capacity Constraints

$$\sum_{r \in R} \sum_{v_r \in N^{\mathbf{V}}_r} d^{v_r}_{\gamma} \cdot \eta^{v_r}_{s,i} \leq \zeta^s_{\gamma}, \quad \forall s \in N^{\mathbf{S}}, \gamma \in \Gamma, i \in \Psi^0 \qquad (5)$$

Since $\eta^{v_r}_{s,i} = x^{v_r}_{s,i} \vee x^{v_r}_{s,i+1}$, this operation is linearised as follows

$$\begin{aligned} &\eta^{v_r}_{s,i} \geq x^{v_r}_{s,i+1}, \qquad \eta^{v_r}_{s,i} \geq x^{v_r}_{s,i} \\ &\eta^{v_r}_{s,i} \leq x^{v_r}_{s,i} + x^{v_r}_{s,i+1}, \qquad \forall r \in R, v_r \in N^{\mathbf{V}}_r, s \in N^{\mathbf{S}}, i \in \Psi^0 \qquad (6) \end{aligned}$$

Bandwidth Constraints

$$\sum_{r \in R} \sum_{v_r \in N^{\mathbf{V}}_r} \tilde{f}^{v_r}_{(s,\tilde{s}),i} + \sum_{r \in R} \sum_{(v_r,\tilde{v}_r) \in L^{\mathbf{V}}_r} f^{(v_r,\tilde{v}_r)}_{(s,\tilde{s}),i} \leq b^{(s,\tilde{s})}, \quad \forall (s,\tilde{s}) \in L^{\mathbf{S}}, i \in \Psi^0 \qquad (7)$$

Node Mapping Constraints

$$\sum_{s \in N^{\mathbf{S}}} x^{v_r}_{s,i} = 1, \qquad \forall r \in R, v_r \in N^{\mathbf{V}}_r, i \in \Psi^+ \qquad (8)$$

Piece-wise Migration Time Linearisation (where J is the set of linear segment indices)

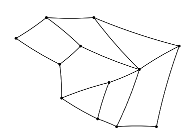
$$t^i_{\textbf{mig}} \geq g^{v_r}_j \left[\mathbb{L}^{v_r}_i \right], \quad \forall r \in R, v_r \in N^{\mathbf{V}}_r, i \in \Psi^0, j \in J \qquad (9)$$

III. Performance Evaluation

A. Simulation Setup

For the substrate network setup, we use the Polska network topology from SNDlib [12], whose characteristics are given in Table I. The node resource capacities and link bandwidths are set randomly based on a realistic set of values given by a German network operator.

TABLE I
POLSKA NETWORK CHARACTERISTICS

Topology	# Nodes	# Links
	12	18

We focus on telco services which are realised by chains of stateful VNFs where each VNF runs within a VM, and assume different service types represented by the service chains (SCs) S_1, S_2, and S_3:

$$S_1 : src \leftarrow f_0 \leftarrow f_1, \qquad S_2 : src \leftrightarrows f_0 \leftrightarrows f_1.$$
$$src \leftrightarrows f_2. \qquad S_3 : src \leftrightarrows f_0 \leftrightarrows f_1 \leftrightarrows f_2.$$

An SC represents a virtual network. In each SC, src is the location from which the users are accessing the service. An arrow and its direction denote a virtual link between two VNFs and the traffic flow direction. Two opposite arrows denote a bidirectional link. The actual resource demands of each SC are a random scale-up of the base-line values shown in Tables II and III, in which the bandwidth demand of any virtual bidirectional link is the same in both directions. Some VNFs do not have significant demands (expressed by 0 demand in Table II) but need to be instantiated to handle the traffic flow. Hence, migrating these nodes only means redirecting the traffic flow to and reinstantiating them at other locations. An SC is selected randomly from the three SC models above whenever a service is to be generated.

The page dirtying rate is randomised following a uniform distribution in the range 2,000–15,000 pages/second for each VM, assuming page size of 4kB. In addition, we assume that when migrating, the entire memory of a VM is relocated.

We use the number of SCs as a crude indicator of the network utilisation; i.e., more SCs implies more resources are utilised. Initially, a defined number of SCs are generated and randomly mapped onto the substrate network; this is done in

TABLE II
BASE-LINE VIRTUAL NODE RESOURCE DEMANDS FOR THE THREE SCS

	Processing Demands				Memory Demands [MB]			
	src	f_0	f_1	f_2	src	f_0	f_1	f_2
S_1	0	150	0	150	0	1500	0	75
S_2	0	1000	0	-	0	2000	0	-
S_3	0	1000	400	50	0	2000	1000	0.5

TABLE III
BASE-LINE VIRTUAL LINK DEMANDS FOR THE THREE SCS

	Bandwidth Demands [Mbps]			
	src–f_0	f_0–f_1	f_1–f_2	src–f_2
S_1	450	50	-	1.5
S_2	250	250	-	-
S_3	200	200	200	-

*These values apply to any link direction.

the pre-processing phase. In Phase I, the objective is to find a placement to minimise the total cost of energy. Phase II is solved multiple times for different values of ψ in order to observe its influence on the outcomes.

It is straightforward that with increasing value of ψ, the minimum total migration time (TMT) can only decrease. The obtained TMT corresponding to each value of ψ therefore is an upper bound for the true minimum (i.e., having no restriction on ψ). However, the computation time also increases with more steps allowed. Thus, we face a trade-off between flexibility and computation time.

Both phases are solved using CPLEX MIP solver version 12.6.3 running on an Intel i7-3930K machine at 3.2GHz and 6 cores. To restrict the computation time we set the time limit to 1.5 hours and the relative MIP gap tolerance to 2%.

B. Results

We first determine the best TMT[2] w.r.t. different numbers of embedded SCs and compare it with the TMT obtained with the minimum number of migration steps (see Figure 2). While there is an upward trend in the TMT in both cases, for low network utilisation, not many steps seem to be required to minimise it; therefore the two curves almost coincide in the beginning. However, for high network utilisation a significant gap can be observed.

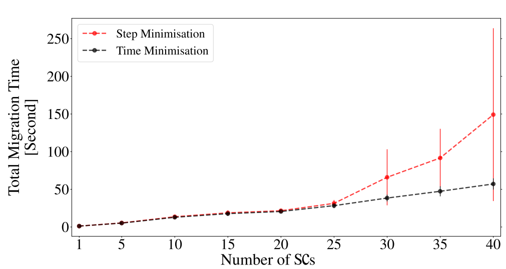


Fig. 2. Best total migration time vs. number of SCs

Figure 3 shows, for a single realisation of 30 SCs, the detailed composition of the TMT w.r.t. different migration steps, marked by different colours. While the TMT is about 60 seconds, the individual steps are much shorter. Note that this is desirable as the failure probability decreases with less migration time.

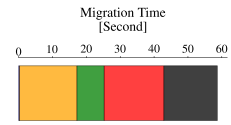


Fig. 3. Migration time breakdown of a multi-step VM migration realisation

Figure 4 shows the dependency of the migration steps on the number of SCs (i.e., the network utilisation) for both the TMT minimisation and the migration step minimisation strategies. In order to minimise the TMT, often more migration steps are required. As the red curve shows, even to migrate VMs without considering the TMT, a multi-step migration approach might be the only solution, especially in case of high network utilisation, because a single-step migration strategy would then be infeasible. As can also be observed from Figure 5, the feasibility of single-step migrations (i.e., the portion of feasible single-step migrations out of all migrations) significantly declines when a larger number of SCs are present.

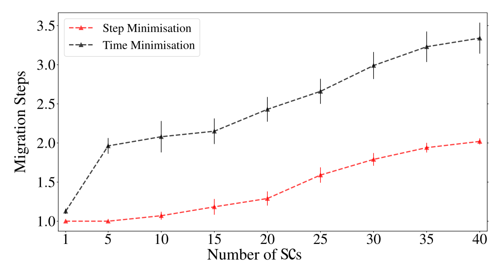


Fig. 4. Averaged best number of migration steps vs. number of SCs

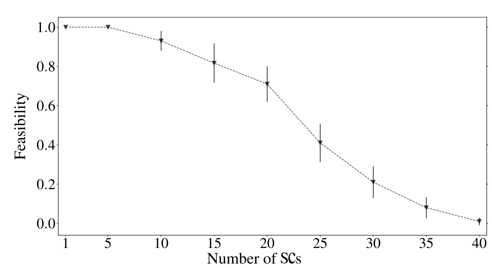


Fig. 5. Feasibility of single-step migrations vs. number of SCs

Figure 6 shows relative TMT improvements w.r.t. the number of SCs for different ψ values. For instance, the blue bars show the relative reduction of the single-step TMT by allowing double-step migrations; the orange bars show the relative reduction of the double-step TMT by allowing triple-step migrations, and so on. Essentially, as long as non-zero values are shown, we still have an improvement by increasing ψ. Moreover, the size of the blue bars increases with the number of SCs. This is to be expected since for higher

[2]Due to the termination criteria, the best TMT is not necessarily the same as the minimum one.

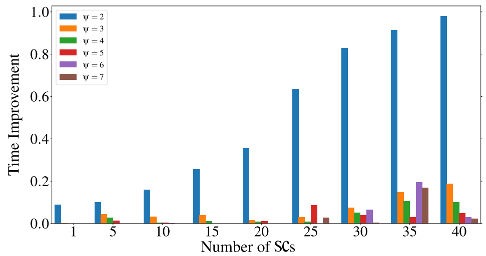


Fig. 6. Incremental total migration time improvement

utilisation, the TMT for single-step migrations will rapidly increase compared to multi-step migrations. Thus, a lot of TMT could be saved by applying the multi-step approach.

To evaluate the CoM, we define it as follows:

$$\sum_{i\in\Psi^0}\sum_{r\in R}\sum_{v^r\in N_r^{\mathrm{V}}}\sum_{(s,\tilde{s})\in L^{\mathrm{S}}} t^i_{\mathbf{mig}}\cdot\tilde{f}^{v_r}_{(s,\tilde{s}),i}$$

This is equivalent to the total traffic volume handled by the entire substrate network due to VM migration. Figure 7 shows the CoM improvement w.r.t. the number of SCs for different ψ values. While here it is possible to have negative bars, a loss in CoM when allowing multi-step migrations is small and rare, and could be alleviated or compensated by allowing even more migration steps.

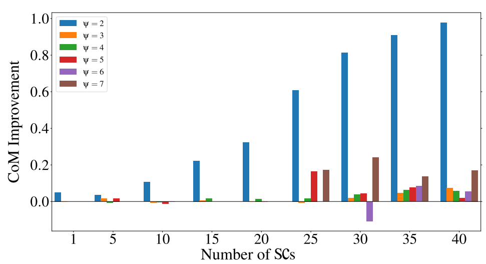


Fig. 7. Incremental CoM improvement

The observation that more migration steps yield lower CoM can be explained as follows. For single-step migrations, the available bandwidth for VM migration is too cramped, causing a fierce competition amongst the VMs for capacity. This leads to longer migration paths, larger TMT, and more overall transmitted data volume. With more steps, on the other hand, the flexibility is higher, and it seems that even the total number of hops is still smaller than in the single-step case.

IV. Conclusion & Future Work

In this paper we describe a novel mathematical optimisation model for VM migration over multiple steps. The concept of multi-step migration was first introduced in our previous work [5] which imposes a restriction on migration deadlines and hence poses efficiency and feasibility issues. The work in this paper addresses these issues by removing this restriction and considers a more flexible migration scheme.

The performance of the novel model is evaluated by means of the Polska network topology, which is an example network topology taken from SNDlib [12]. No matter what type of single-step migration approach is adopted, for larger numbers of SCs there always will be a bandwidth bottleneck that can only be circumvented by allowing more degrees of freedom for the VM migration through a multi-step VM migration approach. Its effectiveness is apparent particularly for high network utilisation.

As our MILP model is $\mathcal{NP}$-hard, the solution time limit may lead to some loss of optimality of the obtained solutions (especially for larger problem instances). A potential way to tackle the scalability problem would be the development of suitable heuristic methods. We realise that some standard heuristic procedures could be leveraged for this purpose and intend to continue our work in this direction. Moreover, we plan to develop a further optimisation model based on a model-predictive control approach taking into account the dynamic nature of the input variables which might supersede the separate Phase I and Phase II models.

Acknowledgment

This work has been performed in the framework of the Celtic-Plus project SENDATE Secure-DCI, funded by the German BMBF (ID 16KIS0481), and the project SyNErgIt, funded by ESF (ID 100270062).

References

[1] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.

[2] S. Azodolmolky, P. Wieder, and R. Yahyapour, "Cloud computing networking: challenges and opportunities for innovations," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 54–62, 2013.

[3] M. I. Biswas, G. Parr, S. McClean, P. Morrow, and B. Scotney, "A practical evaluation in openstack live migration of vms using 10gb/s interfaces," in *Service-Oriented System Engineering (SOSE), 2016 IEEE Symposium on*. IEEE, 2016, pp. 346–351.

[4] M. R. Hines, U. Deshpande, and K. Gopalan, "Post-copy live migration of virtual machines," *ACM SIGOPS operating systems review*, vol. 43, no. 3, pp. 14–26, 2009.

[5] N. T. Khải, A. Baumgartner, and T. Bauschert, "A multi-step model for migration and resource reallocation in virtualized network infrastructures," in *Computer Communications Workshops (INFOCOM WKSHPS), 2017 IEEE Conference on*. IEEE, 2017, pp. 730–735.

[6] H. Maziku and S. Shetty, "Towards a network aware vm migration: Evaluating the cost of vm migration in cloud data centers," in *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*. IEEE, 2014, pp. 114–119.

[7] F. Esposito and W. Cerroni, "Geomig: Online multiple vm live migration," in *Cloud Engineering Workshop (IC2EW), 2016 IEEE International Conference on*. IEEE, 2016, pp. 48–53.

[8] R. Bradford, E. Kotsovinos, A. Feldmann, and H. Schiöberg, "Live wide-area migration of virtual machines including local persistent state," in *Proceedings of the 3rd international conference on Virtual execution environments*. ACM, 2007, pp. 169–179.

[9] G. Schaffrath, S. Schmid, and A. Feldmann, "Optimizing long-lived cloudnets with migrations," in *Utility and Cloud Computing (UCC), 2012 IEEE Fifth International Conference on*. IEEE, 2012, pp. 99–106.

[10] M. R. Silver and O. L. De Weck, "Time-expanded decision networks: A framework for designing evolvable complex systems," *Systems Engineering*, vol. 10, no. 2, pp. 167–188, 2007.

[11] V. Mann, A. Gupta, P. Dutta, A. Vishnoi, P. Bhattacharya, R. Poddar, and A. Iyer, "Remedy: Network-aware steady state vm management for data centers," *NETWORKING 2012*, pp. 190–204, 2012.

[12] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, "Sndlib 1.0—survivable network design library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010.