Optimal VM placement in data centres with architectural and resource constraints

Deze Zeng, Song Guo* and Huawei Huang

School of Computer Science and Engineering,
The University of Aizu,
Aizuwakamatsu, Fukushima, 965-8580, Japan
Email: deze-z@u-aizu.ac.jp
Email: sguo@u-aizu.ac.jp
Email: d8152101@u-aizu.ac.jp
*Corresponding author

Shui Yu

School of Information Technology,
Deakin University,
221 Burwood Hwy, Burwood VIC 3125, Australia
Email: syu@deakin.edu.au

Victor C.M. Leung

Department of Electrical and Computer Engineering,
University of British Columbia,
2329 W Mall, Vancouver, BC V6T 1Z4, Canada
Email: vleung@ece.ubc.ca

Abstract: Recent advance in virtualisation technology enables service provisioning in a flexible way by consolidating several virtual machines (VMs) into a single physical machine (PM). The inter-VM communications are inevitable when a group of VMs in a data centre provide services in a collaborative manner. With the increasing demands of such intra-data-centre traffics, it becomes essential to study the VM-to-PM placement such that the aggregated communication cost within a data centre is minimised. Such optimisation problem is proved NP-hard and formulated as an integer programming with quadratic constraints in this paper. Different from existing work, our formulation takes into consideration of data-centre architecture, inter-VM traffic pattern, and resource capacity of PMs. Furthermore, a heuristic algorithm is proposed and its high efficiency is extensively validated.

Keywords: inter-VM traffic minimisation; VM placement; data centre network.

Optimal VM placement in data centres

Biographical notes: Deze Zeng received his PhD and MS degrees in Computer Science from University of Aizu, Aizu-Wakamatsu, Japan, in 2013 and 2009, respectively. He received his BS degree from School of Computer Science and Technology, Huazhong University of Science and Technology, China in 2007. He is currently a research supporter in The University of Aizu, Japan.

Song Guo received his PhD degree in Computer Science from the University of Ottawa, Canada in 2005. He is currently a Senior Associate Professor at the School of Computer Science and Engineering, the University of Aizu, Japan. His research interests are mainly in the areas of protocol design and performance analysis for reliable, energy-efficient, and cost effective communications in wireless networks. He is an Associate Editor of the IEEE Transactions on Parallel and Distributed Systems and an editor of Wireless Communications and Mobile Computing. He is a Senior Member of the IEEE and the ACM.

Huawei Huang received his Master degree in Computer Science from the China University of Geoscience (Wuhan) in 2013. He is currently a PhD student at the School of Computer Science and Engineering, The University of Aizu, Japan. His research interests are mainly in the area of cloud computing.

Shui Yu received his BE and ME degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 1993 and 1999, respectively, and a PhD degree from Deakin University, Burwood, Vic., Australia in 2004. He is currently a Lecturer with School of Information Technology, Deakin University.

Victor C.M. Leung received his BASc and PhD degrees both in Electrical Engineering from the University of British Columbia in 1977 and 1981, respectively. He is Professor and holder of the TELUS Mobility Research Chair in the Department of Electrical and Computer Engineering of the same university. His research interests are in wireless networks and mobile systems. He is a Senior Editor of the IEEE Wireless Communications Letters.

1 Introduction

With the recent fast development of cloud computing, many companies such as Amazon, Google and Microsoft have made large investments to data centre deployment for the support of services in a diversity of forms, such as infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS) and service-as-a-service (SaaS). Virtualisation has been proved as a key technology in cloud computing as it provides a fundamental support in data centres by consolidating several virtual machines (VMs) into a single server, i.e., physical machine (PM). By this means, the physical resources can be provided in a flexible way to tenants by leasing VMs for different service deployment.

Thanks to the flexibility and efficiency of cloud computing, many service providers have adopted commercial data centres as their service hosting platforms. With the increasing communication demands in a data centre, it has been shown that to minimise the data centre communication cost is significant to its scalability (Meng et al., 2010b). This has raised a number of concerns from the perspective of data centre architecture, in particular, network architecture (Al-Fares et al., 2008; Greenberg et al., 2009, 2008). For a given architecture, the inter-PM communication cost is determined. To improve the
communication efficiency between PMs, quite a few data centre architectures have been proposed in the literature, e.g., VL2 (Greenberg et al., 2009), fat-tree (Leiserson, 1985), BCube (Guo et al., 2009), etc. Figure 1 shows two representative architectures. Generally, PMs in a data centre are organised in racks as shown by dotted rectangles in Figure 1. Racks and servers in each rack are interconnected by inter-rack and intra-rack switches, respectively.

Note that the inter-PM traffic is actually determined by the inter-VM communications that vary from services to services. For example, MapReduce (Dean and Ghemawat, 2008) requires performing map operations across many VMs before proceeding to the reduce phase, which may be conducted on some other VMs. A mashup web service (e.g., Wikimapia) must aggregate data from different services (e.g., Google map and Wikipedia) to compose the final service. With more communication-intensive applications, e.g., streaming services like YouTube, the bandwidth between VMs rapidly becomes a principal bottleneck in data centres. A straightforward idea is that all related VMs are placed in the same PM such that the communication cost is minimised without involving any switches. Unfortunately, the limited resources (i.e., CPU, memory and uplink/downlink bandwidth) on each server make the all-in-one solution impractical.

To tackle this communication cost issue, the approach by VM placement, i.e., the mapping between VMs and PMs, has been studied recently in the literature (Meng et al., 2010b; Jiang et al., 2012; Zhang et al., 2012a, 2012b). However, most existing work either relies on an over-simplified resource assumption model (e.g., Meng et al., 2010b) or ignores the actual data centre architecture (e.g., Zhang et al., 2012a). This motivates us to re-investigate the communication cost minimisation problem via VM placement in a data centre by taking both factors into consideration. Our contributions are summarised as follows.

- We formulate the VM placement problem as an integer programming with quadratic constraints. Inputs to this formulation include the resource requirement of each VM, pairwise VM traffic and data centre architecture description (i.e., PM-rack relationship). The solution to our formulation provides the optimal VM-PM placement such that the communication cost is minimised. We also prove the formulated problem NP-hard.

- To address the computational complexity, we propose a low-complexity heuristic algorithm to solve the VM placement problem. Experimental results show the high efficiency of our algorithm by the fact that it performs close to the optimal solution.

Figure 1 Examples of data centre architecture, (a) tree (b) fat-tree (see online version for colours)
The remainder of this paper is structured as follows. Section 2 provides a brief literature survey on VM placement. Section 3 introduces the data centre model studied in this paper. Section 4 develops our formulation of the virtual machine placement problem. Section 5 presents a heuristic algorithm for this problem. Section 6 gives the evaluation results on both exact and heuristic algorithms. Finally, Section 7 concludes this paper.

2 Related work

Due to the massive investments in data centre for the support of cloud computing, its scalability in terms of aggregated inter-PM communication cost has become a major concern. Therefore, much effort has been contributed to address this problem from different aspects, e.g., data centre architecture, communication protocol design, VM placement and migration, etc.

Several data centre architectures, e.g., VL2 (Greenberg et al., 2009), fat-tree (Leiserson, 1985), BCube (Guo et al., 2009), Portland (Mysore et al., 2009), etc., have been proposed with different inter-PM communication costs, as summarised in Meng et al. (2010b). Besides, it is worth noting that inter-VM traffic is a dominant factor to inter-PM traffic and therefore is another critical issue to the data centre scalability (Meng et al., 2010b). To improve inter-VM communication efficiency, Ren et al. (2012) propose an inter-VM communication protocol for the co-resident VMs on the same physical host.

One more critical issue to the data centre scalability is on the VM placement. For any data centre architecture and inter-VM communication protocol, the scalability could be much improved by exploiting the desired locations of VMs in a data centre. Therefore, various VM placement and migration algorithms have been proposed. Ajiro and Tanaka (2007) propose a VM placement scheme called first-fit-decreasing (FFD). The idea is to iteratively place each VM to the first server that can fully satisfy its resource requirement. Meng et al. (2010a) propose a statistical multiplexing method to consolidate VMs with different correlations, e.g., low correlation since one VM needs higher I/O and another one needs more CPU resource. In Meng et al. (2010b), they further investigate how to minimise the inter-VM traffic by exploring the inter-PM communication cost diversity to improve the data centre scalability. Zhang et al. (2012b) proposes virtual machine migration in an over-committed data centre to minimise VM migration cost with the consideration of the data centre network topology. Shrivastava et al. (2011) introduce AppAware which jointly considers the communication dependencies among VMs to minimise the data centre network traffic. Later on, Huang et al. (2012) explore the balance between server energy consumption and network energy consumption and present an application dependency aware VM placement to reduce network energy consumption. Zhang et al. (2012a) study the VM placement problem with the objective of minimising the total inter-VM traffic. Biran et al. (2012) notice that the inter-VM traffic patterns may vary over time and the prediction is extremely difficult and propose an algorithm that is resilient to certain variations. Recently, Cohen et al. (2013) focus on a bandwidth-constrained VM placement optimisation problem to maximising the benefit from the overall communication sent by the VMs to a single designated point in the data centre, e.g., storage area network.

We notice that all these studies discussed above ignore either the resource constraints, e.g., Meng et al. (2010b) or the actual data-centre architectures, e.g., Zhang et al. (2012a), and Huang et al. (2012). Therefore, we are motivated to study the VM placement
problem for inter-VM communication cost minimisation by taking resource constraints, VM traffic shape and data centre architecture into consideration.

3 System model

We consider a data centre with $M$ racks $\mathcal{K} = \{K_1, K_2, \cdots, K_M\}$ and $N$ servers $\mathcal{S} = \{S_1, S_2, \cdots, S_N\}$. The location of servers is described by an $N \times M$ 0-1 matrix $\mathcal{M}^{\text{SK}} = \{M_{ij}^{\text{SK}}\}$, $i = 0, 1, \cdots, N$, $j = 0, 1, \cdots, M$, where

$$M_{ij}^{\text{SK}} = \begin{cases} 1, & \text{if ever } S_i \text{ locates at rack } K_j, \\ 0, & \text{otherwise}. \end{cases}$$

(1)

Let $\mathcal{V} = \{V_1, V_2, \cdots, V_L\}$ be all VMs to be placed in the data centre. A service or an application (e.g., web application, Hadoop, ERP) typically consists of multiple collaborative VMs that communicate with each other. The inter-VM traffic is represented by an $L \times L$ matrix $\mathcal{T} = \{T_{ij}\}$, $i, j = 0, 1, \cdots, L$, where $T_{ij}$ is the pairwise rate between $V_i$ and $V_j$. The communications can be generally categorised into intra-server, inter-server and intra-rack, and inter-rack classes, depending on the locations of the corresponding VMs. Without loss of generality, their communication costs per unit are denoted as $C_\alpha$, $C_\beta$ and $C_\gamma$, respectively, which are determined by the data centre architecture [e.g., tree, VL2 (Greenberg et al., 2009), fat-tree (Leiserson, 1985)] as indicated in Meng et al. (2010b). In general, $C_\alpha < C_\beta < C_\gamma$. Let $C_{ij}$ denote the unit communication cost between any VMs $V_i$ and $V_j$, i.e.,

$$C_{ij} = \begin{cases} C_\alpha, & V_i \text{ and } V_j \text{ on the same PM}, \\ C_\beta, & V_i \text{ and } V_j \text{ on the same rack but different PMs}, \\ C_\gamma, & V_i \text{ and } V_j \text{ on different racks}. \end{cases}$$

We consider a more realistic data-centre resource model that each server has only limited resources (i.e., CPU, memory, hard disk, I/O, etc.) denoted by an $H$-vector: $\mathbb{R} = \{R_1, R_2, \cdots, R_H\}$. The available resources on all servers are described by an $N \times H$ matrix $\mathcal{A} = \{A_{ij}\}$, $i = 1, \cdots, N$, $j = 1, \cdots, H$, where $A_{ij}$ refers to the available capacity of resource $R_j$ on server $S_i$. Similarly, the resource requirements of all VMs are denoted by $\mathcal{Q}^{\text{VR}} = \{Q_{ij}\}$, $i = 1, \cdots, L$, $j = 1, \cdots, H$, where $Q_{ij}$ is the request for resource $R_j$ from VM $V_i$.

Note that $\mathcal{M}^{\text{SK}}$, $C_\alpha$, $C_\beta$, $C_\gamma$ and $\mathcal{A}$ will be determined once a data centre is deployed. Moreover, $\mathcal{T}$ and $\mathcal{Q}^{\text{VR}}$ will be also determined when all VMs in the data centre are fixed.
4 The VM-placement problem

In this section, we analyse the hardness of the VM-placement problem and then develop a mathematical formulation for the optimal solution.

Theorem 1: The virtual machine placement problem under our system model is NP-hard.

Proof: We consider a special case of the problem with the configurations: \( M = 1 \) and \( C = 0 \). In other words, the intra-PM communication incurs no cost and the inter-rack communication does not exist. This is exactly the problem studied in Zhang et al. (2012a), where the impact of data-centre architecture is ignored and the resulting problem is NP-hard proved by reducing the balanced minimum K-cut problem.

4.1 Architectural constraints

To determine the communication cost of any pair of VMs, we define binary variables:

\[
\begin{align*}
    f_{ij}^{VS} &= \begin{cases} 
        1, & \text{if } V_i \text{ and } V_j \text{ placed on the same server,} \\
        0, & \text{otherwise,}
    \end{cases} \\
    f_{ij}^{VK} &= \begin{cases} 
        1, & \text{if } V_i \text{ and } V_j \text{ placed on the same rack,} \\
        0, & \text{otherwise.}
    \end{cases}
\end{align*}
\]

Note that if two VMs locate in the same server, they are definitely within the same rack, i.e., if \( f_{ij}^{VS} = 1 \), then \( f_{ij}^{VK} = 1 \). On the other hand, if two VMs are within the same rack, they could be in the same server or different ones. These can be described by the following constraint:

\[
f_{ij}^{VK} \geq f_{ij}^{VS}, \forall i,j = 1,\ldots,L.
\]  

By similar observation, we conclude that \( f_{ij}^{VK} - f_{ij}^{VS} \) is equal to one if \( V_i \) and \( V_j \) are within the same rack but different servers, and zero otherwise. These lead to the communication cost \( C_{ij} \) to be written as:

\[
C_{ij} = f_{ij}^{VS} C_a + (f_{ij}^{VK} - f_{ij}^{VS}) C_B + (1 - f_{ij}^{VK}) C_C. \tag{3}
\]

4.2 Placement constraints

To describe the VM-PM placement, we define an \( L \times N \) 0-1 matrix \( \mathcal{M}^{IS} = \{ M_{ij}^{IS} \} \), \( i = 1,\ldots,L, \ j = 1,\ldots,N \), where

\[
M_{ij}^{IS} = \begin{cases} 
    1, & \text{if } V_i \text{ is placed on server } S_j, \\
    0, & \text{otherwise.}
\end{cases}
\]  

First of all, each VM must be placed on exactly one PM,
\[
\sum_{j=1}^{N} M_{ij}^{VS} = 1, \ \forall i = 1, \cdots, L.
\] (5)

In addition, recalling that the PM-rack placement is given by \( M^{SK} \), we can derive the VM-rack placement, denoted by an \( L \times M \) matrix \( M^{VK} \), by \( M^{VK} = M^{VS} \times M^{SK} \), or equivalently,

\[
M_{ij}^{VK} = \sum_{k=1}^{N} M_{ik}^{VS} \cdot M_{kj}^{SK}, \ \forall i = 1, \cdots, L, \ \forall j = 1, \cdots, M.
\] (6)

Finally, variables \( f_{ij}^{VS} \) and \( f_{ij}^{VK} \) can be rewritten as respectively:

\[
f_{ij}^{VS} = \sum_{k=1}^{N} M_{ik}^{VS} \cdot M_{kj}^{SK}, \ \forall i, j = 1, \cdots, L
\] (7)

\[
f_{ij}^{VK} = \sum_{k=1}^{M} M_{ik}^{VK} \cdot M_{kj}^{VK}, \ \forall i, j = 1, \cdots, L
\] (8)

4.3 Resource constraints

The final set of constraints are for our capacity-limited resource model. A number of VMs can be consolidated onto a PM only if their resource requirements are all satisfied by the PM, i.e.,

\[
\sum_{k=1}^{L} M_{ik}^{VS} Q_{ij}^{VK} \leq A_{ij}, \ \forall i = 1, \cdots, N, \ \forall j = 1, \cdots, H.
\] (9)

Algorithm 1 The TAF Algorithm

- **Require:** pairwise traffic rate: \( T \)
- **while** at least one VM has not been placed do
  - Let \( V_i \) and \( V_j \) be the PM pair with maximum rate \( T_{ij} \) in the head of \( T \)
  - if both \( V_i \) and \( V_j \) have not been placed then
    - Place \( V_i \) and \( V_j \) to the target server \( S_d \) found by Algorithm 2({\( V_i, V_j \)})
  - if \( S_d \) is null then
    - Place \( V_i \) to \( S_d \) found by Algorithm 2({\( V_i \)})
  - Place \( V_j \) to \( S_d \) found by Algorithm 2({\( V_j \)})
  - end if
- else if only \( V_i \) has already been placed then
  - Place \( V_i \) to \( S_d \) found by Algorithm 2({\( V_i \)})
- else if only \( V_j \) has already been placed then
  - Place \( V_j \) to \( S_d \) found by Algorithm 2({\( V_j \)})
- end if
Our objective is to find an optimal VM placement $\mathcal{M}^{VS}$ that minimises the overall inter-VM communication cost in a data centre. By summarising all the constraints derived above, we formulate the placement problem as an integer programming with quadratic constraints as:

$$\min_{\mathcal{M}}: U = \sum_{i=1}^{L} \sum_{j=1}^{L} T_{ij} C_{ij}$$  \hspace{1cm} (10)$$

s.t.: (2), (3), (5), (6), (7), (8), and (9).

5 The traffic-amount-first algorithm

Due to the NP-hardness of the VM placement problem as formulated in (10), we propose a low-complexity heuristic VM placement algorithm in this section. Our idea comes from the intuition that the more communication traffic can be completed within a server, the less aggregated communication cost would be. Therefore, the VMs with higher traffic rate should be considered with higher priority.

Following this principle, we design our traffic amount first (TAF) algorithm that always tries to place the VM pair with heaviest traffic to the same server without violating the resource constraints (9). The details of TAF are summarised in Algorithms 1 and 2. To find the VM pairs with high traffic rates, we first sort matrix $\mathcal{T}$, in a decreasing order, in a vector $\mathcal{T}$. Then, we check the pairs from $\mathcal{T}$ one by one until all of them are placed (line 2–16 in Algorithm 1).

During each placement iteration, we first pick up PM pair $V_i$ and $V_j$ with the maximum traffic rate (line 3 in Algorithm 1). If both $V_i$ and $V_j$ have not been placed, we take them as a whole and place them on the same server provided that it has enough residual resources to satisfy both of their requirements. We check the whole server set $\mathcal{S}$ to find out the candidate target server set $\mathcal{S}'$ with sufficient resources as shown in Algorithm 2, where $\Delta U$ is denoted as the incremental communication cost incurred by placing the new VMs on a candidate server. Finally, the one from $\mathcal{S}'$ with minimal incremental communication cost will be applied (line 7 in Algorithm 2). Otherwise, if no server has enough residual resources to satisfy both $V_i$ and $V_j$ simultaneously, we have to locate the server with the minimal incremental communication cost for each of them (lines 7 and 8 in Algorithm 1).

Algorithm 2  Find the server with minimal incremental
4 Add \( S \) into the candidate server set \( S' \)
5 \textbf{end for}
6 \textbf{if} \( S' \neq \emptyset \) \textbf{then}
7 \( S' \leftarrow \arg\min_{c \in S'} \Delta U \)
8 \textbf{else}
9 \( S' \leftarrow \text{null} \)
10 \textbf{end if}

In some cases, the host of one VM (\( V_i \) or \( V_j \)) may have been decided (line 10 and 12, respectively, in Algorithm 1) in previous iterations. Following the same principle, we find the best server for the other (\( V_j \) or \( V_i \)) as shown in line 11 and 13, respectively, in Algorithm 1.

Remark: The computation time complexity of TAF algorithm is \( O(n^2) \). Note that the dominant part of the TAF algorithm to the computation complexity is between lines 2 and 16. Up to \( L \) iterations are required in Algorithm 1 to ensure that all VMs are completely placed onto PMs. In each iteration, at most \( N \) servers will be checked in Algorithm 2 to find out an appropriate server with sufficient resources to hold the VMs to be placed. Therefore, in the worst case, totally \( L \times N \) iterations are required.

6 Performance evaluation

To evaluate the performance of the proposed heuristic algorithms, we conduct simulation studies for both small-scale (i.e., with small values of \( L, M \) and \( N \)) and large-scale data centres based on the model described in Section 3. For small-scale data centres, the optimal solutions are obtained by solving (10) using commercial solver Gurobi optimizer (Gurobi Optimization, 2013). For large-scale networks, we compare our TAF algorithm against the FFD algorithm (Ajiro and Tanaka, 2007).

In our simulations, without loss of generality, the resources on each PM are CPU, memory and I/O with capacities of 1,500, 1,500 and 300 units, and the corresponding requests by each VM are generated uniformly and randomly in range (0, 300), (0, 500) and (0, 50), respectively. The traffic rate in matrix \( T \) for each pair of VMs is also randomly generated within 0 and 10. We consider fat-tree architecture (Leiserson, 1985) and therefore communication costs \( C_{\alpha}, C_{\beta} \) and \( C_{\gamma} \) are set as 1, 3 and 5, respectively, according to \cite{1}.

In the following, our experiments show how the exact and heuristic algorithms perform, in terms of aggregated communication cost, as a function of \( L, N, M \) and \( A \). For each setting, we present the average cost obtained from 100 simulation instances.

6.1 On the effect of the number of VMs

We first study how the number of VMs (i.e., \( L \)) affects the communication cost by varying it from 5 to 12 and from and 40 to 70 in small-scale and large-scale data centres, respectively.
From the results shown in Figure 2(a), we notice that the performance of our heuristic algorithm approaches the optimal one. In particular, the communication cost is a nonlinear increasing function of $L$. Such phenomenon is attributed to the following two factors.

1. when the number of VMs is small, most communication can be conducted in an intra-server or intra-rack way such that the aggregated communication cost is low
2. the aggregated communication traffic itself increases with the number of VMs in a scale of $O(L^2)$ as the number of possible connections among $L$ VMs is $L(L - 1) / 2$. We can also see from Figure 2(b) that TAF outperforms FFD under any number of VMs. The high efficiency of our heuristic algorithm is thus validated.

6.2 On the effect of the number of PMs

We also conduct two groups of experiments. In the small-scale data centres, we fix $M = 2$, $L = 10$ and vary the value of $N$ from 2 to 6. In the large-scale data centres, we fix $M = 4$, $L = 60$ and vary the value of $N$ from 20 to 60. We have witnessed the effect of the number of VMs on the aggregated data centre communication cost as shown in Figure 3.

Once again, we notice that TAF outperforms FFD and approaches to the optimal solution under different numbers of PMs. Furthermore, the communication cost shows as a decreasing function of the number of PMs. This is because, under a fixed number of racks, increasing the number of PMs in each rack also increases the probability of intra-rack communications, without sacrificing the intra-server communications. The communication cost is thus reduced.

6.3 On the effect of the number of racks

Then, we would like to see how the number of racks (i.e., $M$) affects the communication cost. In this group of evaluations, we fix the values of $N = 6$, $L = 10$ and vary $M$ from 2 to 6 in both small-scale and large-scale data centres, respectively.

**Figure 2** Total communication costs vs. number of VMs, (a) small-scale data centres (b) large-scale data centres (see online version for colours)
The closeness of TAF to the optimal solution and its advantage over FFD can be observed in Figure 4. We also notice that the communication cost is an increasing function of $M$. This is because, under the same number of PMs, increasing the number of racks will also increase the probability of inter-rack communications, leading to comparatively higher cost.

6.4 On the effect of resources capacities

Next, let us investigate the effect of resource to the communication cost by varying the resource capabilities of CPU, memory and I/O in different sets of experiments. The values of $(L, M, N)$ are fixed as $(10, 3, 6)$ and $(60, 4, 40)$ in small-scale and large-scale data centres, respectively. Meanwhile, we vary the capabilities of CPU, memory and I/O from 500 to 1,500, from 600 to 2,000 and from 60 to 340, and show the evaluation results in Figures 5, 6 and 7, respectively.

From Figure 5, we see that the total communication cost is a decreasing function of CPU capacity in Figure 5(b).
Similar phenomenon can be observed from Figures 6 and 7 as well. This is because each server with richer available resources can host more VMs such that the chances of intra-server and intra-rack communications would be improved. In other words, the aggregated communication cost is decreased.

On the other hand, the performance will finally converge when the capability exceeds a certain value, e.g., 1,000 CPU units in Figure 5. This is because under such condition, the dominant factor to the communication cost becomes the intrinsic pairwise traffic.
rates, not the resource capabilities any more. Furthermore, we can see that our heuristic algorithm can always provide much better solutions than FFD.

6.5 On the effect of VN traffic load

Finally, we present our evaluation results about how the VM traffic load affects the total inter-VM communication cost. Same as in the evaluation on the effect of resource capacities, \((L, M, N)\) are fixed as \((10, 3, 6)\) and \((60, 4, 40)\) in small-scale and large-scale data centres, respectively. In both cases, we vary the traffic load generation range for each pair of VMs from 10 to 50. The evaluation results are shown in Figure 8(a) and Figure 8(b), respectively. From both figures, it can be noticed that the total inter-VM communication cost is a linearly increasing function of the VM traffic load. Higher inter-VM traffic load shall exhibit larger total inter-VM communication cost. Nevertheless, the closeness of our proposed TAF algorithm to the optimal one and its advantage over FFD algorithm can be always observed from Figure 8.

![Figure 8](image)

**Figure 8** Total communication costs vs. VM traffic load, (a) small-scale data centres (b) large-scale data centres (see online version for colours)

7 Conclusions

In this paper, we address the problem of VM placement to minimise the aggregated communication cost within a data centre under the consideration of both architectural and resource constraints. We prove this optimisation problem NP-hard and formally formulate it as an integer programming problem with quadratic constraints. To tackle the high computational complexity of the exact algorithm, we further propose a low-complexity heuristic algorithm TAF. By extensive simulation-based studies, the high efficiency of TAF is validated by the fact that it always approaches to the optimal performance in small-scale instances and outperforms a representative existing algorithm in large-scale instances, under various settings.

As part of our future work, the VM migration approach would be taken into consideration for the similar optimisation problem in a data centre with dynamic service provisioning.
References


