Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Data locality and replica aware virtual cluster embeddings

Carlo Fuerst [a], Maciej Pacut [b,*], Stefan Schmid [c,*]

[a] TU Berlin, Germany
[b] University of Wroclaw, Poland
[c] Aalborg University, Denmark

ARTICLE INFO

Article history:
Received 1 June 2016
Received in revised form 23 May 2017
Accepted 28 June 2017
Available online xxxx
Communicated by P. Krysta

Keywords:
Virtual network embeddings
Flow algorithms
NP hardness

ABSTRACT

Virtualized datacenters offer great flexibilities in terms of resource allocation. In particular, by decoupling applications from the constraints of the underlying infrastructure, virtualization supports an optimized mapping of virtual machines as well as their interconnecting network (the so-called *virtual cluster*) to their physical counterparts: a graph embedding problem.

However, existing virtual cluster embedding algorithms often ignore a crucial dimension of the problem, namely *data locality*: the input to a cloud application such as MapReduce is typically stored in a distributed, and sometimes redundant, file system. Since moving data is costly, an embedding algorithm should be data locality aware, and allocate computational resources close to the data; in case of redundant storage, the algorithm should also optimize the *replica selection*.

This paper initiates the algorithmic study of data locality aware virtual cluster embeddings on datacenter topologies. We show that despite the multiple degrees of freedom in terms of embedding, replica selection and assignment, many problems can be solved efficiently. We also highlight the limitations of such optimizations, by presenting several NP-hardness proofs; interestingly, our hardness results also hold in uncapacitated networks of small diameter.

1. Introduction

Distributed cloud applications, such as batch-processing applications or scale-out databases, generate a significant amount of network traffic [25]. For instance, MapReduce consists of a network intensive shuffle phase, where data is transferred from the mappers to the reducers. In order to ensure a predictable application performance, especially in shared cloud environments, it is important to provide isolation and bandwidth guarantees between the virtual machines [37], e.g., by making explicit network reservations [5]. Accordingly, modern batch-processing applications provide the abstraction of entire *virtual networks* [25], defining both the virtual machines as well as their interconnecting network. The most prominent virtual network abstraction is the *virtual cluster* [5,16,30,34].

Virtualized datacenters offer great flexibilities on where these virtual networks can be instantiated or *embedded*. In order to maximize the resource utilization in the datacenter, it is in principle desirable to map the virtual machines of a given virtual network as close as possible in the underlying physical network, as this minimizes communication costs (respectively, bandwidth reservations) [5,16,30,34].

* Corresponding authors.
E-mail address: stefan.schmid@tu-berlin.de (S. Schmid).

http://dx.doi.org/10.1016/j.tcs.2017.06.025
0304-3975/

Please cite this article in press as: C. Fuerst et al., Data locality and replica aware virtual cluster embeddings, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.06.025

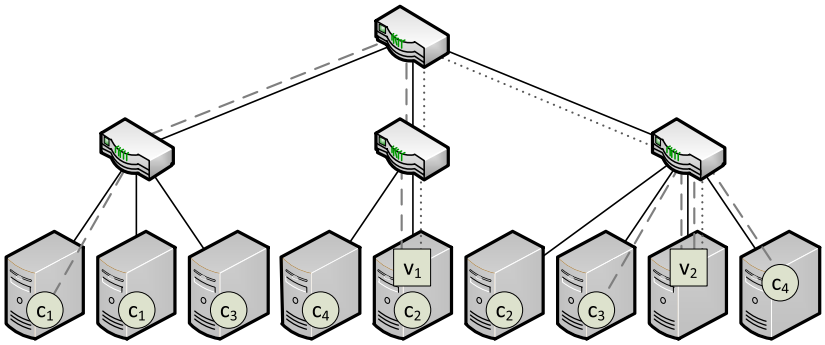

Fig. 1. Overview: a 9-server datacenter storing $\tau = 4$ different chunk types $\{c_1, \ldots, c_4\}$ (depicted as *circles*). The chunk replicas need to be selected and assigned to the two virtual machines v_1 and v_2; the virtual machines are depicted as *squares*, and the network connecting them to chunks (at bandwidth b_1) is *dashed*. In addition, the virtual machines are inter-connected among each other at bandwidth b_2 (*dotted*). The objective of the embedding algorithm is to minimize the overall bandwidth allocation (sum of *dashed* and *dotted* lines).

However, existing systems often ignore a crucial dimension of the virtual network embedding problem: the fact that the input data for a cloud application, consisting of atomic *chunks*, is typically distributed across different servers and stored in a distributed file system [6,18,32]. In order to properly minimize communication costs, an embedding algorithm should hence also be *data locality aware* [4,23,36], and allocate (or *embed*) computational resources close to the to be processed data. Moreover, in case of redundant storage (batch processing applications often provide a 3-fold redundancy [32]), an algorithm should also be aware of, and exploit, *replica selection* flexibilities.

1.1. Our contributions

This paper initiates the formal study of data-locality and replica aware virtual network embedding problems in datacenters. In particular, we decompose the general optimization problem into its fundamental aspects, such as assignment of chunks, replica selection, and flexible virtual machine placement, and answer questions such as:

1. Which chunks to assign to which virtual machine?
2. How to exploit redundancy and select good replicas?
3. How to efficiently embed virtual machines and their inter-connecting network?
4. Can the chunk assignment, replica selection and virtual machine embedding problems be jointly optimized, in polynomial time?

We draw a complete picture of the problem space: We show that even problem variants exhibiting multiple degrees of freedom in terms of replica selection and embedding, can be solved optimally in polynomial time, and we present several efficient algorithms accordingly. However, we also prove limitations in terms of computational tractability, by providing reductions from 3-D matching and Boolean satisfiability (SAT). Interestingly, while it is well-known that (unsplittable) multi-commodity flow problems are NP-hard in capacitated networks, our hardness results also hold in *uncapacitated* networks; moreover, we show that NP-hard problems already arise in small-diameter networks (as they are widely used today [2]).

1.2. Organization

Section 2 introduces our formal model in detail. Algorithms are presented in Section 3 and hardness results are presented in Section 4. Section 5 takes a deeper look at the NP-hardness variants. After discussing related work in Section 6, we conclude our work in Section 7.

2. Model

To get started, and before introducing our formal model and its constituent parts in detail, we will discuss the practical motivation. Fig. 1 gives an overview of our model.

2.1. Background and practical motivation

Our model is motivated by batch-processing applications such as MapReduce. Such applications use multiple virtual machines to process data, often redundantly stored in a distributed file system implemented by multiple servers [4,10]. Datacenter networks are typically organized as fat-trees, with servers are located at the tree leaves and inner nodes being switches or routers. Given the amount of multiplexing over the mesh of links and the availability of multi-path routing protocol, e.g. ECMP, the redundant links can be considered as a single aggregate link for bandwidth reservations [5,16,30,34].

Please cite this article in press as: C. Fuerst et al., Data locality and replica aware virtual cluster embeddings, Theoret. Comput. Sci. (2017), http://dx.doi.org/10.1016/j.tcs.2017.06.025