How Hard Can It Be?

Understanding the Complexity of
Replica Aware Virtual Cluster Embeddings

Carlo Fuerst (TU Berlin, Germany), Maciek Pacut (University of Wroclaw, Poland)
Paolo Costa (Microsoft Research, UK), Stefan Schmid (TU Berlin & T-Labs, Germany)
Today’s Cloud Computing
Today’s Cloud Computing

Figure 1: Percentiles (1-25-50-75-99th) for intra-cloud network bandwidth observed by past studies.

Source: Ballani et al. [1] in Sigcomm’11
Today’s Cloud Computing

“Hadoop traces from Facebook show that, on average, transferring data between successive stages accounts for 33% of the running times of jobs with reduce phases”

Figure 1: Percentiles (1-25-50-75-99th) for intra-cloud network bandwidth observed by past studies.

Source: Ballani et al. [1] in Sigcomm’11

Source: Chowdhury et al. [2] in Sigcomm’11
Today’s Cloud Computing

“Hadoop traces from Facebook show that, on average, transferring data between successive stages accounts for 33% of the running times of jobs with reduce phases”

Figure 1: Percentiles (1-25-50-75-99th) for intra-cloud network bandwidth observed by past studies.

Source: Ballani et al. [1] in Sigcomm’11

Source: Chowdhury et al. [2] in Sigcomm’11

Costs for the tenats become unpredictable
Proposed Solutions: Virtual Clusters

\[ V_1 \quad V_2 \]
Proposed Solutions: Virtual Clusters

$V_1 \ ? \ V_2$
Proposed Solutions: Virtual Clusters

Remove the uncertainty by specifying the bandwidth connecting the VMs.
Proposed Solutions: Virtual Clusters

• Introduced by Ballani et al. [1]
• Provides absolute guarantees on VMs and network performance
• Specified by two parameters:
  • N the number of VMs
  • B the available bandwidth between VMs.
Embedding

$\mathbf{v}_1$  $\mathbf{v}_2$
Embedding

\[ v_1 \rightarrow v_2 \]
Embedding
Embedding
Embedding
Embedding
Embedding
Virtual Cluster Embedding Problem

• Subproblem of the NP-hard virtual network embedding problem
• Good heuristics available
  • Ballani et al. [1] in Sigcomm’11
  • Xie et al. [3] in Sigcomm’12
Virtual Cluster Embedding Problem

• Subproblem of the NP-hard virtual network embedding problem
• Good heuristics available
  • Ballani et al. [1] in Sigcomm’11
  • Xie et al. [3] in Sigcomm’12

but...
Virtual Cluster Embedding Problem

• Subproblem of the NP-hard virtual network embedding problem
• Good heuristics available
  • Ballani et al. [1] in Sigcomm’11
  • Xie et al. [3] in Sigcomm’12

but...

The virtual cluster embedding problem is not NP-hard.[4]
Can the problem be solved efficiently with additional properties?
Cloud Application: Batch processing

Example: MapReduce

1. Input is given by a set of atomic chunks
Cloud Application: Batch processing

Example: MapReduce
1. Input is given by a set of atomic chunks
2. Every chunk is processed by a map task
Cloud Application: Batch processing

Example: MapReduce
1. Input is given by a set of atomic chunks
2. Every chunk is processed by a map task
Cloud Application: Batch processing

Example: MapReduce
1. Input is given by a set of atomic chunks
2. Every chunk is processed by a map task
Cloud Application: Batch processing

Example: MapReduce

1. Input is given by a set of atomic chunks
2. Every chunk is processed by a map task
3. The output of the map task is transferred to reduce tasks (shuffle)
Cloud Application: Batch processing

Example: MapReduce
1. Input is given by a set of atomic chunks
2. Every chunk is processed by a map task
3. The output of the map task is transferred to reduce tasks (shuffle)
Cloud Application: Batch processing

Example: MapReduce
1. Input is given by a set of atomic chunks
2. Every chunk is processed by a map task
3. The output of the map task is transferred to reduce tasks (shuffle)
4. Reduce tasks are executed
Cloud Application: Batch processing

Example: MapReduce

1. Input is given by a set of atomic chunks
2. Every chunk is processed by a map task
3. The output of the map task is transferred to reduce tasks (shuffle)
4. Reduce tasks are executed
5. Once all reduce tasks finished there is an aggregated output
Cloud Application: Batch processing

Example: MapReduce

1. Input is given by a set of atomic chunks
2. Every chunk is processed by a map task
3. The output of the map task is transferred to reduce tasks (shuffle)
4. Reduce tasks are executed
5. Once all reduce tasks finished there is an aggregated output
Shortcoming of Virtual Clusters
Shortcoming of Virtual Clusters
Shortcoming of Virtual Clusters
Shortcoming of Virtual Clusters

Virtual Clusters provide a guarantee for the shuffle phase, but not for the transfer of chunks.
Basic Problem
Basic Solution
Problem Decomposition

The basic problem can be extended with:

• VM interconnect (NI)
Problem Decomposition
Problem Decomposition

\[ c^2 c_1 v^2 v_1 \]
Problem Decomposition

The basic problem can be extended with:

• VM interconnect (NI)
• Replica Selection (RS)
Problem Decomposition

\[ c_2 c_1 \]

\[ v_2 v_1 \]
Problem Decomposition

\[ c_1 c_1 c_2 c_2 v_1 v_2 \]
Problem Decomposition

The basic problem can be extended with:

• VM interconnect (NI)
• Replica Selection (RS)
• Multiple Assignment (MA)
Problem Decomposition
Problem Decomposition
Problem Decomposition

The basic problem can be extended with:

• VM interconnect (NI)
• Replica Selection (RS)
• Multiple Assignment (MA)
• Free placement of VMs (FP)
Problem Decomposition
Problem Decomposition
Problem Decomposition

The basic problem can be extended with:

• VM interconnect (NI)
• Replica Selection (RS)
• Multiple Assignment (MA)
• Free placement of VMs (FP)
• Bandwidth Constraints (BW)
Problem Decomposition
What is in the Paper?

• Trivial problem identification
What is in the Paper?
What is in the Paper?

- Trivial problem identification
- Matching based algorithms
What is in the Paper?
What is in the Paper?
What is in the Paper?

• Trivial problem identification
• Matching based algorithms
• Flow based algorithm
What is in the Paper?
What is in the Paper?
What is in the Paper?

• Trivial problem identification
• Matching based algorithms
• Flow based algorithm
• Hardness results
What is in the Paper?
Everything but Replicas (MA + NI + FP + BW)
Everything but Replicas (MA + NI + FP + BW)
Dynamic Programming

• Create physical topology annotations in a bottom-up manner
• Start at the servers
• For each amount $n$ of VMs in $\{0, \ldots, N\}$
  • Set cost[$n$] to $\infty$ if $n$ exceeds the servers capacity
  • Set cost[$n$] to the bandwidth costs of placing $n$ VMs at the server
Dynamic Programming

- Max 1 VM per server
- 2 Chunks per VM
Dynamic Programming

• Max 1 VM per server
• 2 Chunks per VM
Dynamic Programming

• Create physical topology annotations in a bottom-up manner
• Start at the servers

• For each amount n of VMs in \{0,...,N\}
  • Set cost[n] to $\infty$ if n exceeds the servers capacity
  • Set cost[n] to the bandwidth costs of placing n VMs at the server

• For each switch and each amount of VMs in \{0,...,N\}
  • Set cost[n] to the sum of the cheapest combination of the children and add the costs for the bandwidth on the uplink
Dynamic Programming

- Max 1 VM per server
- 2 Chunks per VM
Dynamic Programming

- Max 1 VM per server
- 2 Chunks per VM
Runtimes

- Intel(R) Xeon(R) CPU L5420 @ 2.50GHz with (single threaded)
- 512 MB
- openjdk-7
- Max 4 VMs per Server
- 3 Chunks per VM
Which problems can be solved like this?
Which problems can be solved like this?
What is in the Paper?
Summary

• Virtual clusters provide dedicated resource guarantees

• Datalocality can be incorporated into the virtual cluster abstraction

• Problem decomposition into five properties
  • NP-hardness proofs for some property combinations
  • Algorithms for all other property combinations
References


[2] Chowdhury et al. „Managing Data Transfers in Computer Clusters with Orchestra.“ The ACM SIGCOMM Conference on Data Communication (SIGCOMM'11)
