Optimized Resource Mapping in Distributed Cloud Systems


Atakan Aral

Thesis Advisor: Asst. Prof. Dr. Tolga Ovatman

Istanbul Technical University – Department of Computer Engineering

July 12, 2016
Outline

1. Introduction
   - Summary of the Term
   - Time Plan

2. Distributed Replica Placement for Edge Computing
   - Facility Location Heuristic
   - Replica Discovery

3. Evaluation and Discussion

4. Overview of the Thesis Study
   - Virtual Machine Mapping
   - Data Replica Mapping
Outline

1. Introduction
   - Summary of the Term
   - Time Plan

2. Distributed Replica Placement for Edge Computing
   - Facility Location Heuristic
   - Replica Discovery

3. Evaluation and Discussion

4. Overview of the Thesis Study
   - Virtual Machine Mapping
   - Data Replica Mapping
Distributed Replica Placement (DRP) Algorithm

- A fully distributed algorithm for data replication in Edge Computing Systems
  - that is based on Facility Location Problem in Operation Research
  - that addresses the trade-off between price (storage and bandwidth) and performance (latency)
  - that is dynamic, online, and incremental.
  - that works with the limited knowledge of the system.

- Benefits include
  - low latency access to centralized data
  - better price–performance ratio than caching
  - reduced network overhead
Evaluation

- Implemented and Simulated in **CloudSim**
- Real workload traces are used
  - The CAIDA Anonymized Internet Traces 2015 Dataset [1M requests]
  - 1998 FIFA World Cup Web Site Requests Dataset [880K requests]
- **Baselines**: Centralized storage, Caching
- **Parameters**: Quantum length, level of expansion
- **Criteria**: Data access latency, data storage and transfer cost, network overhead, false positive hit rate
Journal Publication

- **Network-Aware Embedding of Virtual Machine Clusters onto Federated Cloud Infrastructure**

  *The Journal of Systems and Software, Elsevier* [Impact Factor: 1.767]

  - Revised on **04.06.2016**, accepted for publication on **06.07.2016**
  - Available online with [DOI: 10.1016/j.jss.2016.07.007](http://doi.org/10.1016/j.jss.2016.07.007)

  - Revision includes extended related work, a use case scenario, and evaluation of runtime performance.
1 Introduction
   ■ Summary of the Term
   ■ Time Plan

2 Distributed Replica Placement for Edge Computing
   ■ Facility Location Heuristic
   ■ Replica Discovery

3 Evaluation and Discussion

4 Overview of the Thesis Study
   ■ Virtual Machine Mapping
   ■ Data Replica Mapping

Outline
Introduction
Distributed Replica Placement for Edge Computing
Evaluation and Discussion
Overview of the Thesis Study

Gantt Chart

Atakan Aral
Optimized Resource Mapping in Distributed Cloud Systems
Outline

1. Introduction
   - Summary of the Term
   - Time Plan

2. Distributed Replica Placement for Edge Computing
   - Facility Location Heuristic
   - Replica Discovery

3. Evaluation and Discussion

4. Overview of the Thesis Study
   - Virtual Machine Mapping
   - Data Replica Mapping
Distributed Replica Placement

Motivation

Edge Computing

Pushing the frontier of computing applications, data, and services away from centralized nodes to the logical extremes of a network (e.g. mobile devices, sensors, nano data centers, routers, modems, ...)

- Provides low-latency access to computing resources for code offloading.
- However, many applications still need to access data that is stored centrally
  - Due to the limited storage capacity of the edge entities, economic constraints, availability for offline analysis, simpler maintenance and concurrency control
Distributed Replica Placement

Data Replication

A frequently used technique to improve availability, fault-tolerance, security, and/or access latency.

1. Which data to replicate?
2. When to replicate?
3. Where to place the replica?
4. How to direct requests to replicas?
5. How to keep replicas consistent?
6. How to prefetch data to exploit spatial locality?
Facility Location Heuristic
Solution Idea

Main Idea
Create replicas of the data object on locations where access paths for that object frequently pass through so that future requests from multiple locations can be served from a single replica.

- We assume temporal and geographical locality of reference.
- We also assume that geographical distance is correlated with latency.
Facility Location Heuristic
Temporal and Geographical Locality

Histogram of Request Pair Distances
Facility Location Heuristic

Temporal and Geographical Locality

Percentages of Request Pairs by Maximum Distance
Facility Location Heuristic

Facility Location Problem

Finding a placement of facilities in order to serve the demands of geographically distributed customers with minimum cost (transportation and facility building)

\[
\text{minimize } \sum_{j} f_j \cdot Y_j + \sum_{i} \sum_{j} h_i \cdot d_{ij} \cdot X_{ij}
\]
Facility Location Heuristic

\[
\text{minimize } \sum_{j} f_{j} \cdot Y_{j} + \sum_{i} \sum_{j} h_{i} \cdot d_{ij} \cdot X_{ij}
\]

- Facility opening cost \((f_{j}) \propto\) storage cost for the replica in the next quantum.
  \[f_{j} = \text{unit} \_ \text{price}_j \cdot \text{replica} \_ \text{size} \cdot \text{quantum}\]
- Demand \((h_{i}) \propto\) number of requests received in the previous quantum
  \[h_{i} = \text{num} \_ \text{requests} \cdot \text{replica} \_ \text{size}\]
- Distance \((d_{ij}) \propto\) latency between a VM and a replica
  \[d_{ij} = \text{latency}_{ij} \cdot \lambda\]
Facility Location Heuristic

\[
\text{minimize } \sum_{j} \text{unit\_price}_j \cdot \text{quantum} + \sum_{i} \sum_{j} \text{num\_requests} \cdot \text{latency}_{ij} \cdot \lambda
\]

- \( \lambda \) is the unit conversion factor.
- It represents the expendable unit cost in exchange for a unit decrease in latency per unit demand.
- It can be tuned for different service level objectives in terms of latency.
Facility Location Heuristic

\[
\text{minimize} \sum_{j} \text{unit\_price}_j \cdot \text{quantum} + \sum_{i} \sum_{j} \text{num\_requests}_i \cdot \text{latency}_{ij} \cdot \lambda
\]

- In every quantum, locally evaluate the objective function at the central storage to **Create** replicas in the neighbours.

- In every quantum, locally evaluate the objective function at each replica location for one of the following operations:
  - **Remove** the replica at that location
  - **Duplicate** the replica to one of the neighbour locations
  - **Migrate** the replica to one of the neighbour locations
Outline

1. Introduction
   - Summary of the Term
   - Time Plan

2. Distributed Replica Placement for Edge Computing
   - Facility Location Heuristic
   - Replica Discovery

3. Evaluation and Discussion

4. Overview of the Thesis Study
   - Virtual Machine Mapping
   - Data Replica Mapping
How to notify an edge entity when a closer replica is created?

- VM_CREATE
- TASK_SUBMIT
- DATA_REQUEST
- DATA_RETURN
- LOC_UPDATE
- DATA_REQUEST
- DATA_RETURN
- CREATE_CACHE
- TASK_RETURN
Replica Discovery

1. When a DC decides to deploy a replica in one of its neighbours, it notifies previous requesters.
2. When a new replica is created in a location, its neighbours are notified.
3. When any message is received, DC calculates and stores the latency to the source of the message.
4. When a VM requires a data object, list of previous notifications is searched.
5. If there are multiple candidates, the decision is made based on known latencies.
6. Otherwise, the data is requested from the main storage.
Outline

1 Introduction
   - Summary of the Term
   - Time Plan

2 Distributed Replica Placement for Edge Computing
   - Facility Location Heuristic
   - Replica Discovery

3 Evaluation and Discussion

4 Overview of the Thesis Study
   - Virtual Machine Mapping
   - Data Replica Mapping
Introduction
Distributed Replica Placement for Edge Computing
Evaluation and Discussion
Overview of the Thesis Study

Lambda Parameter

Atakan Aral
Optimized Resource Mapping in Distributed Cloud Systems
Lambda Parameter

![Lambda Parameter Graph](image-url)

- **X-axis**: Lambda Parameter
- **Y-axis**: Cost ($/GB$) on the left and Latency (ms) on the right

The graph shows the relationship between Lambda Parameter and cost per GB as well as average latency. The data is color-coded with blue representing Cost per GB and orange representing Average Latency.
Quantum Duration

Atakan Aral

Optimized Resource Mapping in Distributed Cloud Systems
Quantum Duration
Relative to Centralized Solution

[0.012, 2000]: 28.04% latency decrease 22.12% cost increase  
[0.012, 4000]: 32.99% latency decrease 33.10% cost increase
Outline

1. Introduction
   - Summary of the Term
   - Time Plan

2. Distributed Replica Placement for Edge Computing
   - Facility Location Heuristic
   - Replica Discovery

3. Evaluation and Discussion

4. Overview of the Thesis Study
   - Virtual Machine Mapping
   - Data Replica Mapping
"Optimized Resource Mapping in Distributed Cloud Systems"

- Mapping virtual entities to physical cloud resources at hand
- Resources are distributed and networked
- Decreasing the network latency and optimizing bandwidth utilization so as to improve the performance of cloud services
- Also considering the resource costs to improve price-performance ratio
- Employing approximation algorithms and/or heuristics to solve NP-complete problems dynamically
Use Case Scenario

A Cloud Broker or Mediator

- Has clients who wish to execute their data-intensive tasks with high performance and availability. Clients demands SLOs in terms of average global latency or total execution time.
- Hires cloud resources from a IaaS provider to fulfill client needs.
- Has to minimize its operating costs but also satisfy SLOs

**Input:** VM Cluster to be executed, required data objects, SLOs

**Output:** mapping between VMs and compute servers (usually static), mapping between data objects and storage servers (usually dynamic)
Outline

1. Introduction
   - Summary of the Term
   - Time Plan

2. Distributed Replica Placement for Edge Computing
   - Facility Location Heuristic
   - Replica Discovery

3. Evaluation and Discussion

4. Overview of the Thesis Study
   - Virtual Machine Mapping
   - Data Replica Mapping
Part I: Virtual Machine Mapping

Topology based Mapping (TBM) Algorithm

Map VM Clusters onto the federated cloud infrastructure based on their topology.


Outline

1. Introduction
   - Summary of the Term
   - Time Plan

2. Distributed Replica Placement for Edge Computing
   - Facility Location Heuristic
   - Replica Discovery

3. Evaluation and Discussion

4. Overview of the Thesis Study
   - Virtual Machine Mapping
   - Data Replica Mapping
Part II: Data Replica Mapping

Distributed Replica Placement (DRP) Algorithm

Map replicas to the locations where its access paths frequently pass through.


Thank you for your time.