Cloud Federations in the Sky: Formation Game and Mechanism

Fanchao Meng

University of Delaware

fcmeng@cis.udel.edu

October 1, 2015

Overview

In The Past ...

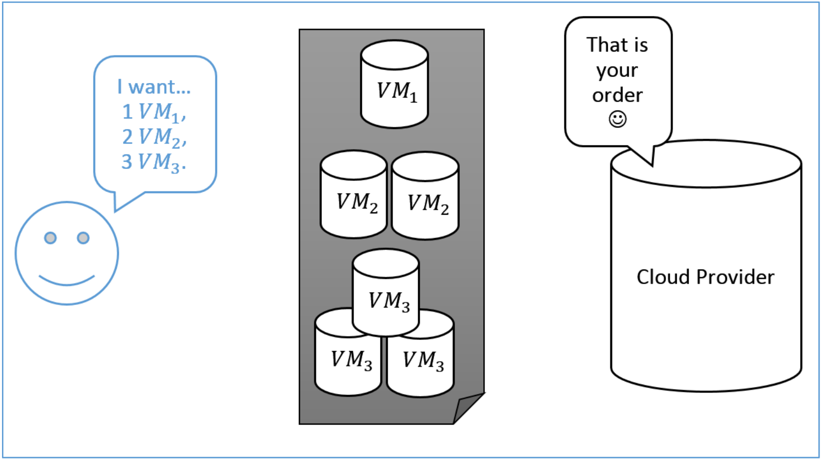

Figure 1: Single cloud provider provisions services.

Things Happen ...

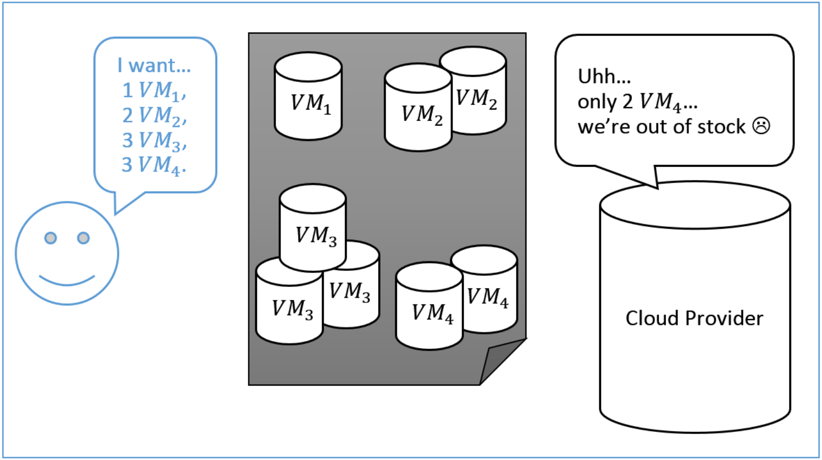

Figure 2: Providers could become powerless when facing increasing demands.

Sometimes Even Worse ...

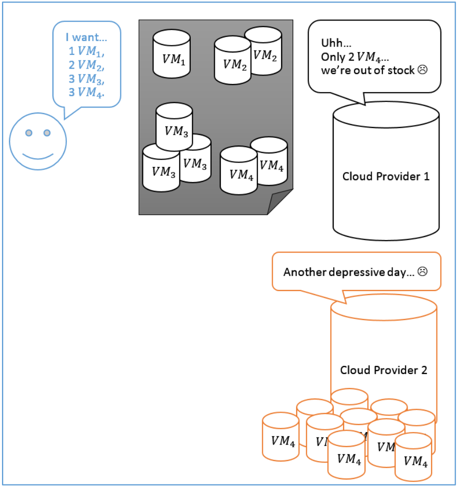

Figure 3: Things can go further down!

Federation Saved Them!

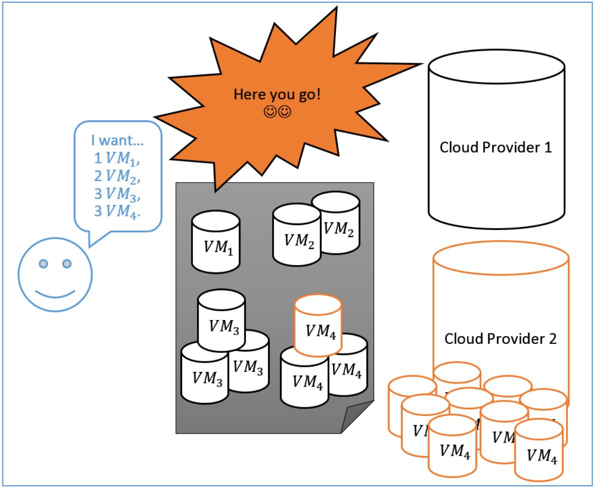

Figure 4: Cooperation makes life easier!

How It Works

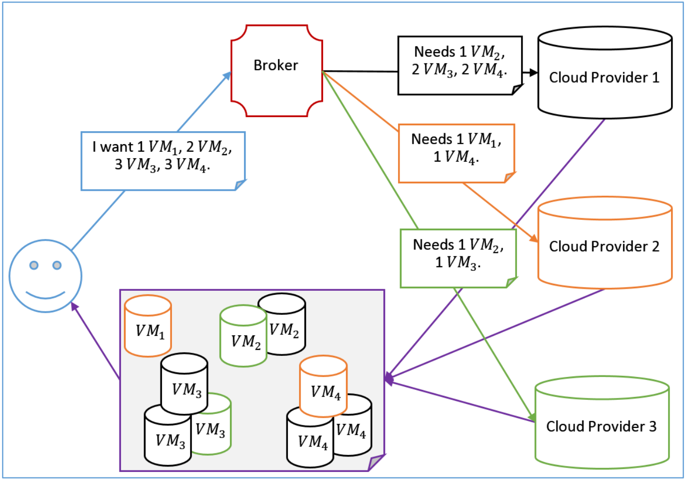

Figure 5: How the cloud federation works.

What We Want ...

Everybody wins! In other words, the user gets what she needs, and the participating providers obtain maximized profits.

Terminology

$\mathcal{I} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_m\}$ A set of cloud providers.

$\mathcal{F} \subseteq \mathcal{I}$ A cloud federation is a set of cloud providers.

$\mathcal{VM} = \{VM_1, \ldots, VM_n\}$ The providers offer n types of VMs.

$VM_j = \langle w_j^c, w_j^m, w_j^s \rangle$ Each VM has cores, memory and storage.

$\mathcal{C}_i : \langle N_i, M_i, S_i \rangle$ Each provider has upper bounds of cores, memory and storage.

c_{ij} Each provision of VM_j from a provider C_i has a cost.

$\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$ A user request sent to a broker.

r_j The amount of VM_j requested by the user.

$\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ Prices set of VMs by the broker.

p_j The price of VM_j set by the broker.

x_{ij} The amount of VM_j provided by provider $\mathcal{C}_i$.

IP-CFPM Problem

Given the context, $\langle \mathcal{I}, \mathcal{VM}, \mathcal{P}, \mathcal{R} \rangle$, find **x** to maximize

$$\sum_{\mathcal{C}_i \in \mathcal{F}} \sum_{j=1}^{n} x_{ij}(p_j - c_{ij}) \tag{1}$$

subject to

$$\sum_{j=1}^{n} w_j^c x_{ij} \leq N_i, \qquad (\forall \mathcal{C}_i \in \mathcal{F}) \tag{2}$$

$$\sum_{j=1}^{n} w_j^m x_{ij} \leq M_i, \qquad (\forall \mathcal{C}_i \in \mathcal{F}) \tag{3}$$

$$\sum_{j=1}^{n} w_j^s x_{ij} \leq S_i, \qquad (\forall \mathcal{C}_i \in \mathcal{F}) \tag{4}$$

IP-CFPM Problem (Cont.)

$$\sum_{\mathcal{C}_i \in \mathcal{F}} x_{ij} = r_j, \qquad (\forall j = 1, \ldots, n) \tag{5}$$

$$\sum_{j=1}^{n} x_{ij} \geq 1, \qquad (\forall \mathcal{C}_i \in \mathcal{F}) \tag{6}$$

$$x_{ij} > 0, \textit{and is integer} \qquad (\forall \mathcal{C}_i \in \mathcal{F} \textit{and} \, \forall j = 1, \ldots, n) \tag{7}$$

Cloud Federation Game

Definition 1: Cloud Federation Game [1]

A Cloud Federation Game, $\langle \mathcal{I}, v \rangle$, is of the **coalitional form** with a **transferable utility** allowing **side payments** to be made among the players, where v is the **characteristic function**.

Definition 2: Coalition [2]

A coalition, $\mathcal{S}$, is defined to be a subset of $\mathcal{I}$, the set of all coalitions is denoted by $2^{\mathcal{I}}$. An empty subset is called the **empty coalition**, and $\mathcal{I}$ itself is called the **grand coalition**.

Definition 3: Characteristic Function [2]

A characteristic function v is a real-valued function defined on the set $2^{\mathcal{I}}$ such that $v : 2^{\mathcal{I}} \to \mathbb{R}^+$, and satisfying

1. $v(\emptyset) = 0$, and
2. (**superadditivity**) if $\mathcal{S}$ and $\mathcal{T}$ are disjoint coalitions $(\mathcal{S} \cap \mathcal{T}) = \emptyset$, then $v(\mathcal{S}) + v(\mathcal{T}) \leq v(\mathcal{S} \cup \mathcal{T})$.

Cloud Federation Game (Cont.)

Definition 4: Characteristic Function in Cloud Federation Game

The characteristic function in the cloud federation game is defined as

$$v(\mathcal{F}) = \begin{cases} 0 & \text{if } |\mathcal{F}| = 0 \text{ or IP-CFPM is not feasible,} \\ P & \text{if } |\mathcal{F}| > 0 \text{ and IP-CFPM is feasible,} \end{cases}$$

where P is the total profit obtained by the given federation.

Cloud Federation Game (Cont.)

Definition 4: Characteristic Function in Cloud Federation Game

The characteristic function in the cloud federation game is defined as

$$v(\mathcal{F}) = \begin{cases} 0 & \text{if } |\mathcal{F}| = 0 \text{ or IP-CFPM is not feasible,} \\ P & \text{if } |\mathcal{F}| > 0 \text{ and IP-CFPM is feasible,} \end{cases}$$

where P is the total profit obtained by the given federation.

The total profit is the output of the objective function in the IP-CFPM problem which we are about to maximize.

$$P = \sum_{\mathcal{C}_i \in \mathcal{F}} \sum_{j=1}^{n} x_{ij}(p_j - c_{ij})$$

Some Explanations

Transferable Utility & Side Payment

Transferable utility means that within a coalition each player has the option to give any amount of money to any other player, or even simply destroy money.[3] Side payment means a payment made to a party or parties to induce them to join an agreement.

Superadditivity

It means that the value of two disjoint coalitions is at least as great when they work together as when they work apart.[2]

Fairness & Stability

General Principles

A cloud federation should satisfy two main properties, fairness and stability. Fairness means that the profit obtained by the federation should be divided fairly among the participants. Stability means that the participants should not have incentives to leave the federation.

Fairness

Purpose

A cloud provider that contributes more resources in all the possible federations in which it participates should receive higher profit regardless of resource allocation in the selected federation.

Definition 5: Banzhaf Value

The Banzhaf value of a given provider $\mathcal{C}_i$ in the cloud federation game $\langle \mathcal{I}, v \rangle$ is defined as

$$\beta_{\mathcal{C}_i}(\mathcal{I}) = \frac{1}{2^{m-1}} \sum_{\mathcal{F} \subseteq \mathcal{I} \setminus \{\mathcal{C}_i\}} [v(\mathcal{F} \cup \{\mathcal{C}_i\}) - v(\mathcal{F})] \tag{8}$$

The normalized Banzhaf value is defined as

$$\mathcal{B}_{\mathcal{C}_i}(\mathcal{I}) = \frac{\beta_{\mathcal{C}_i}(\mathcal{I})}{\sum_{\mathcal{C}_j \in \mathcal{I}} \beta_{\mathcal{C}_j}(\mathcal{I})} \tag{9}$$

Fairness (Cont.)

Definition 6: Payoff

The payoff of a given member provider $\mathcal{C}_i$ in the grand federation is

$$\psi_{\mathcal{C}_i}(\mathcal{I}) = \mathcal{B}_{\mathcal{C}_i}(\mathcal{I})v(\mathcal{I}) \quad (10)$$

A payoff vector $\Psi(\mathcal{I}) = (\psi_{\mathcal{C}_1}(\mathcal{I}), \ldots, \psi_{\mathcal{C}_m}(\mathcal{I}))$ gives the payoff division for the grand federation.

The payoff of a given member provider $\mathcal{C}_i$ in a federation is

$$\psi_{\mathcal{C}_i}(\mathcal{F}) = \frac{\psi_{\mathcal{C}_i}(\mathcal{I})}{\sum_{\forall \mathcal{C}_j \in \mathcal{F}} \psi_{\mathcal{C}_j}(\mathcal{I})} v(\mathcal{F}) \quad (11)$$

Computing the Banzhaf value for a game with a large number of players is NP-hard.

Fairness (Cont.)

The Banzhaf value means ...

A player's power is proportional to the number of times that player is a critical player. In other words, **an important player is important everywhere.**

Fairness (Cont.)

The Banzhaf value means ...

A player's power is proportional to the number of times that player is a critical player. In other words, **an important player is important everywhere.**

Figure 6: An important player is important everywhere.

Stability

Definition 7: Imputation [1] [2]

An imputation is a payoff vector $\Psi(\mathcal{I}) = (\psi_{\mathcal{C}_1}(\mathcal{I}), \ldots, \psi_{\mathcal{C}_m}(\mathcal{I}))$ satisfying:

Individually rational: $\psi_{\mathcal{C}_i}(\mathcal{I}) \geq v(\{\mathcal{C}_i\}), \forall \mathcal{C}_i \in \mathcal{I}$, *and*

Group rational: $\sum_{\mathcal{C}_i \in \mathcal{I}} \psi_{\mathcal{C}_i}(\mathcal{I}) = v(\mathcal{I})$

Stability

Definition 7: Imputation [1] [2]

An imputation is a payoff vector $\Psi(\mathcal{I}) = (\psi_{\mathcal{C}_1}(\mathcal{I}), \ldots, \psi_{\mathcal{C}_m}(\mathcal{I}))$ satisfying:

Individually rational: $\psi_{\mathcal{C}_i}(\mathcal{I}) \geq v(\{\mathcal{C}_i\}), \forall \mathcal{C}_i \in \mathcal{I},$ *and*

Group rational: $\sum_{\mathcal{C}_i \in \mathcal{I}} \psi_{\mathcal{C}_i}(\mathcal{I}) = v(\mathcal{I})$

The individually rational means ...

No player could be expected to agree to receive less than that player could obtain acting alone.

Stability

Definition 7: Imputation [1] [2]

An imputation is a payoff vector $\Psi(\mathcal{I}) = (\psi_{\mathcal{C}_1}(\mathcal{I}), \ldots, \psi_{\mathcal{C}_m}(\mathcal{I}))$ satisfying:

Individually rational: $\psi_{\mathcal{C}_i}(\mathcal{I}) \geq v(\{\mathcal{C}_i\}), \forall \mathcal{C}_i \in \mathcal{I},$ *and*

Group rational: $\sum_{\mathcal{C}_i \in \mathcal{I}} \psi_{\mathcal{C}_i}(\mathcal{I}) = v(\mathcal{I})$

The individually rational means ...

No player could be expected to agree to receive less than that player could obtain acting alone.

The group rational means ...

The entire profit of the grand federation is divided among all players.

Stability (Cont.)

Definition 8: Core

The core is a set of imputations satisfying

$$\sum_{\mathcal{C}_i \in \mathcal{F}} \psi_{\mathcal{C}_i}(\mathcal{I}) \geq v(\mathcal{F}), \forall \mathcal{F} \subseteq \mathcal{I}$$

Stability (Cont.)

Definition 8: Core

The core is a set of imputations satisfying

$$\sum_{\mathcal{C}_i \in \mathcal{F}} \psi_{\mathcal{C}_i}(\mathcal{I}) \geq v(\mathcal{F}), \forall \mathcal{F} \subseteq \mathcal{I}$$

The core means ...

If the core is not empty, then there exists no federation can make any group of participants obtain more profits than that they can obtain from the grand federation.

Life Is Not Perfect...

Unfortunately ...

The core could be empty, which means the grand federation may not be the best way (or to say not stable) for providers to maximize the profit.

Life Is Not Perfect...

Unfortunately ...

The core could be empty, which means the grand federation may not be the best way (or to say not stable) for providers to maximize the profit.

Question ...

Then who is the best?

Federation Formation Framework

Definition 9: Federation Structure

A federation structure is a partition of the set of cloud providers $\mathcal{I}$ such that each pair of subsets in the structure is disjoint. We denote by Π the set of all federation structures.

Federation Formation Framework

Definition 9: Federation Structure

A federation structure is a partition of the set of cloud providers $\mathcal{I}$ such that each pair of subsets in the structure is disjoint. We denote by Π the set of all federation structures.

Bad News ...

The total number of federation structures is the mth Bell number, $B(x) = e^{e^x - 1}$. Finding the optimal federation structure is a NP-Complete problem.

Federation Formation Framework

Definition 10: Hedonic Game

A hedonic game is a tuple $\langle \mathcal{I}, \succeq \rangle$, where $\succeq_i$ is a reflexive, complete, and transitive preference relation defined on Π_i for player i, and Π_i is the set of subsets in $\mathcal{I}$ containing player i.

Federation Formation Framework

Definition 10: Hedonic Game

A hedonic game is a tuple $\langle \mathcal{I}, \succeq \rangle$, where $\succeq_i$ is a reflexive, complete, and transitive preference relation defined on Π_i for player i, and Π_i is the set of subsets in $\mathcal{I}$ containing player i.

A hedonic Game means ...

If $A \succeq_i B$, then player i prefers coalition B as much as coalition A.

Federation Formation Framework (Cont.)

Definition 11: Cloud Federation Formation Game

If the preference relation over federations is defined for each provider, then a federation formation game is defined as a hedonic game

Federation Formation Framework (Cont.)

Definition 11: Cloud Federation Formation Game

If the preference relation over federations is defined for each provider, then a federation formation game is defined as a hedonic game

Definition 12: Preference Relation Over Federations

For all $\mathcal{C}_i \in \mathcal{I}$ and for all $\mathcal{F}, \mathcal{F}^{'} \in \Pi_i$, we define $\succeq_i$ as

$$\mathcal{F} \succeq_i \mathcal{F}^{'} \iff v(\mathcal{F}) \geq v(\mathcal{F}^{'})$$

Federation Formation Framework (Cont.)

Definition 11: Cloud Federation Formation Game

If the preference relation over federations is defined for each provider, then a federation formation game is defined as a hedonic game

Definition 12: Preference Relation Over Federations

For all $\mathcal{C}_i \in \mathcal{I}$ and for all $\mathcal{F}, \mathcal{F}^{'} \in \Pi_i$, we define $\succeq_i$ as

$$\mathcal{F} \succeq_i \mathcal{F}^{'} \iff v(\mathcal{F}) \geq v(\mathcal{F}^{'})$$

This preference means ...

A cloud provider prefers the federation that gives the higher profit.

Federation Formation Framework (Cont.)

Definition 13: Merge Comparison $\gg_m$

The merge comparison $\gg_m$ is defined as

$$\begin{aligned} &\{\mathcal{F} \cup \mathcal{F}^{'}\} \gg_m \{\mathcal{F}, \mathcal{F}^{'}\} \iff \\ &\{\forall \mathcal{C}_i \in \mathcal{F}; \{\mathcal{F} \cup \mathcal{F}^{'}\} \succ_i \mathcal{F} \wedge \forall \mathcal{C}_j \in \mathcal{F}^{'}; \{\mathcal{F} \cup \mathcal{F}^{'}\} \succ_j \mathcal{F}^{'}\} \end{aligned} \tag{12}$$

Federation Formation Framework (Cont.)

Definition 13: Merge Comparison $\gg_m$

The merge comparison $\gg_m$ is defined as

$$\begin{aligned}&\{\mathcal{F} \cup \mathcal{F}^{'}\} \gg_m \{\mathcal{F}, \mathcal{F}^{'}\} \iff \\ &\{\forall \mathcal{C}_i \in \mathcal{F}; \{\mathcal{F} \cup \mathcal{F}^{'}\} \succ_i \mathcal{F} \land \forall \mathcal{C}_j \in \mathcal{F}^{'}; \{\mathcal{F} \cup \mathcal{F}^{'}\} \succ_j \mathcal{F}^{'}\}\end{aligned} \tag{12}$$

Definition 14: Split Comparison $\gg_s$

The split comparison $\gg_s$ is defined as

$$\begin{aligned}&\{\mathcal{F}, \mathcal{F}^{'}\} \gg_s \{\mathcal{F} \cup \mathcal{F}^{'}\} \iff \\ &\{\forall \mathcal{C}_i \in \mathcal{F}; \mathcal{F} \succ_i \{\mathcal{F} \cup \mathcal{F}^{'}\} \lor \forall \mathcal{C}_j \in \mathcal{F}^{'}; \mathcal{F}^{'} \succ_j \{\mathcal{F} \cup \mathcal{F}^{'}\}\}\end{aligned} \tag{13}$$

Federation Formation Framework (Cont.)

Two Rules

Merge Rule: For any pair of federation $\mathcal{F}$ and $\mathcal{F}^{'}$:

$$\{\mathcal{F} \cup \mathcal{F}^{'}\} \gg_m \{\mathcal{F}, \mathcal{F}^{'}\} \implies \textit{Merge}\ \mathcal{F}\ \textit{and}\ \mathcal{F}^{'}$$

Split Rule: For any federation $\{\mathcal{F} \cup \mathcal{F}^{'}\}$:

$$\{\mathcal{F}, \mathcal{F}^{'}\} \gg_s \{\mathcal{F} \cup \mathcal{F}^{'}\} \implies \textit{Split}\ \{\mathcal{F} \cup \mathcal{F}^{'}\}$$

Estimated Banzhaf Value

Definition 15: Estimated Banzhaf Value

By iteratively applying the merge and split rules, some of the possible federations are checked and their values are calculated. Then the estimated Banzhaf value of $\mathcal{C}_i$ is defined as:

$$E_{\mathcal{C}_i}(\mathcal{I}) = \frac{1}{\lambda} \sum_{\substack{\mathcal{F} \subseteq \mathcal{I} \setminus \{\mathcal{C}_i\} \\ \mathcal{F} \in \mathcal{V} \\ \mathcal{F} \cup \mathcal{C}_i \in \mathcal{V}}} [v(\mathcal{F} \cup \mathcal{C}_i) - v(\mathcal{F})] \tag{14}$$

where $\mathcal{V}$ is the set of all checked federations, and λ is the total number of checked federations containing $\mathcal{C}_i$, i.e. $\lambda = 2^{m-1} - \alpha$, where α is the number of non-checked federations.

Estimated Banzhaf Value (Cont.)

Definition 16: Normalized Estimated Banzhaf Value

The normalized estimated Banzhaf value is defined as:

$$\mathcal{E}_{\mathcal{C}_i}(\mathcal{I}) = \frac{E_{\mathcal{C}_i}(\mathcal{I})}{\sum_{\mathcal{C}_j \in \mathcal{I}} E_{\mathcal{C}_j}(\mathcal{I})} \tag{15}$$

Individual Federation Stability

Definition 17: Individual Federation Stability

Since only one federation is needed eventually to fulfill a user request, then w.r.t. the stability we can only consider one federation instead of a federation structure. Therefore, a federation $\mathcal{F}$ is individually federation stable if there is no member $\mathcal{C}_i \in \mathcal{F}$ such that $\forall j \in \mathcal{F}, \mathcal{F} \backslash \{\mathcal{C}_i\} \succeq_j \mathcal{F}$.

Cloud Federation Formation Mechanism (CFFM)

Algorithm 1. Cloud Federation Formation Mechanism

Input: Request $\mathcal{R}$
$\mathcal{V} = \emptyset$
$\mathcal{FS} = \{\{\mathcal{C}_1\}, \ldots, \{\mathcal{C}_m\}\}$
for all $\mathcal{F}_i \in \mathcal{FS}$ **do**
    $v(\mathcal{F}_i) =$ Solve IP-CFPM($\mathcal{F}_i$)
    $\mathcal{V} = \mathcal{V} \cup \mathcal{F}_i$
repeat
    MergeFederations();
    SplitFederation();
until No split happens
Find $\mathcal{F}_k = \arg\max_{\mathcal{F}_i \in \mathcal{FS}}\{v(\mathcal{F}_i)\}$
for all $\mathcal{C}_i \in \mathcal{F}_k$ **do**
    Calculate $\psi_{\mathcal{C}_i}(\mathcal{F}_k)$ based on $\mathcal{V}$
    $\mathcal{F}_k$ allocates and provides the requested VM instances.

Cloud Federation Formation Mechanism (CFFM)

Algorithm 2. MergeFederations()

repeat

  Select two non-checked federations $\mathcal{F}_i, \mathcal{F}_j \in \mathcal{FS}$

  $v(\mathcal{F}_i \cup \mathcal{F}_j) =$ Solve IP-CFPM$(\mathcal{F}_i \cup \mathcal{F}_j)$

  $\mathcal{V} = \mathcal{V} \cup \{\mathcal{F}_i \cup \mathcal{F}_j\}$

  if $\mathcal{F}_i \cup \mathcal{F}_j \gg_m \{\mathcal{F}_i, \mathcal{F}_j\}$ **then**

    $\mathcal{F}_i \leftarrow \mathcal{F}_i \cup \mathcal{F}_j$

    $\mathcal{F}_j \leftarrow \emptyset$ $\{\mathcal{F}_j$ is removed from $\mathcal{FS}\}$

until No merge happens

Cloud Federation Formation Mechanism (CFFM)

Algorithm 3. SplitFederation()

```
for all 𝓕_i ∈ 𝓕𝓢 where |𝓕_i| > 1 do
    for all partitions {𝓕_j, 𝓕_k} of 𝓕_i,
  where 𝓕_i = 𝓕_j ∪ 𝓕_k, 𝓕_j ∩ 𝓕_k = ∅ do
        v(𝓕_j) = Solve IP-CFPM(𝓕_j)
        v(𝓕_k) = Solve IP-CFPM(𝓕_k)
        𝒱 = 𝒱 ∪ 𝓕_j
        𝒱 = 𝒱 ∪ 𝓕_k
        if {𝓕_j, 𝓕_k} ≫_s 𝓕_i then
            𝓕_i ← 𝓕_j
            𝓕𝓢 = 𝓕𝓢 ∪ 𝓕_k
            break
```

CFFM Properties

Theorem 1

CFFM converges to a federation structure composed of disjoint federations of cloud providers.

CFFM Properties

Theorem 1

CFFM converges to a federation structure composed of disjoint federations of cloud providers.

This is guaranteed by the merge-and-split iteration, i.e. no further merges or splits could exist because no more preferred federation could be found.

CFFM Properties (Cont.)

Theorem 2

CFFM produces an individually stable federation.

CFFM Properties (Cont.)

Theorem 2

CFFM produces an individually stable federation.

This is guaranteed by the split procedure, i.e. the split procedure will check all possible splits to see if any provider's leaving would benefit it more, and eventually no future splits could be beneficial.

CFFM Time Complexity

In general ...

The time complexity of CFFM is determined by the number of merge and split operations and the size of the sub-partitions.

CFFM Time Complexity

In general ...

The time complexity of CFFM is determined by the number of merge and split operations and the size of the sub-partitions.

The merge operation ...

In the worst case, the total number of merges is in $O(m^3)$.

CFFM Time Complexity

In general ...

The time complexity of CFFM is determined by the number of merge and split operations and the size of the sub-partitions.

The merge operation ...

In the worst case, the total number of merges is in $O(m^3)$.

The split operation ...

In the worst case, splitting a federation $\mathcal{F}$ is in $O(2^{|\mathcal{F}|})$, which involves finding all the possible partitions of size 2.

Experimental Setup

Virtual Machines

The experiments use VMs offered by Amazon EC2, four different types of VMs are used. The tech specifications are listed below.

	Small	Medium	Large	Extralarge
	VM_1	VM_2	VM_3	VM_4
w_j^c (1.6GHz CPU)	1	2	4	8
w_j^m (GB Memory)	1.7	3.75	7.5	15
w_j^s (TB Storage)	0.22	0.48	0.98	1.99
p_j (price)	0.12	0.24	0.48	0.96

Figure 7: Four types of VMs are used.

Experimental Setup (Cont.)

Parameters

The parameters of the experiments are listed below.

Param.	Description	Value(s)
m	Number of cloud providers	8
n	Number of VM types	4
N_i	Number of cores	[512, 1536]
M_i	Memory (GB)	[870, 2610]
S_i	Storage (TB)	[112, 338]
c_j	VM cost vector (4)	Based on Amazon Regions
p_j	VM price vector (4)	Based on Microsoft Azure

Figure 8: Paramters.

Experimental Setup (Cont.)

Methods

Control Method 1: Optimal Cloud Federation Mechanism (OCFM) which finds the optimal solution to the federation formation problem. It enumerates all the possible federations and solves IP-CFPM optimally for each of these federations.

Control Method 2: The Random Cloud Federation Mechanism (RCFM) mechanism selects several cloud providers randomly and forms a federation.

Implementation: IBM ILOG Concert Technology APIs in C++ is used to solve IP-CFPM associated with the proposed mechanism. Its engine is CPLEX Optimizer to solve integer programming problems.

Experimental Results

Total Profit

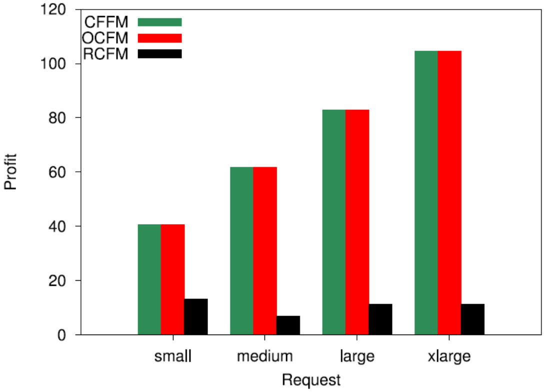

Figure 9: CFFM is very close to the optimal solution.

Experimental Results (Cont.)

Average Size of Federation

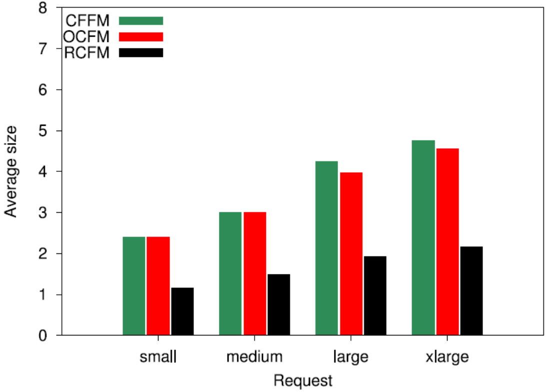

Figure 10: CFFM is close to the optimal solution.

Experimental Results (Cont.)

Individual Profit

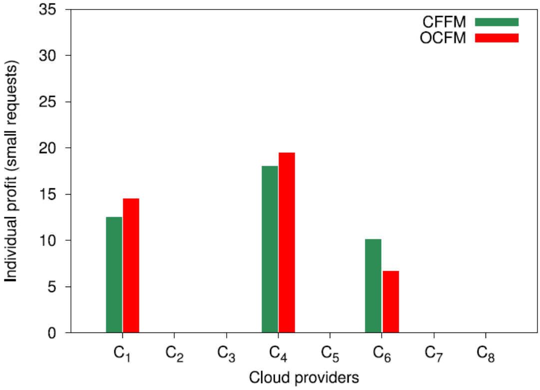

Figure 11: Small Requests.

Experimental Results (Cont.)

Individual Profit (Cont.)

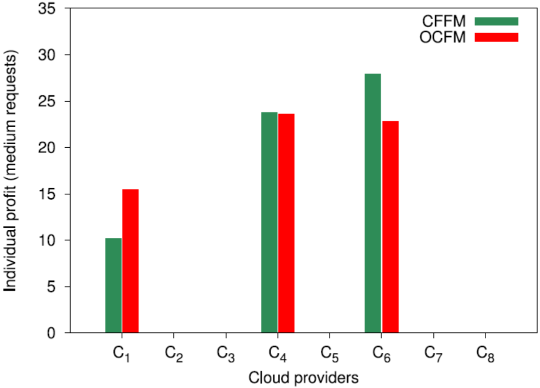

Figure 12: Medium Requests.

Experimental Results (Cont.)

Individual Profit (Cont.)

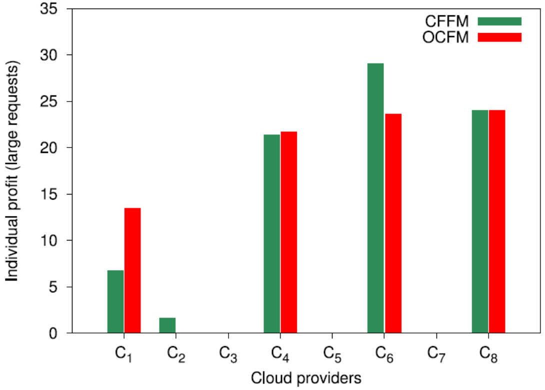

Figure 13: Large Requests.

Experimental Results (Cont.)

Individual Profit (Cont.)

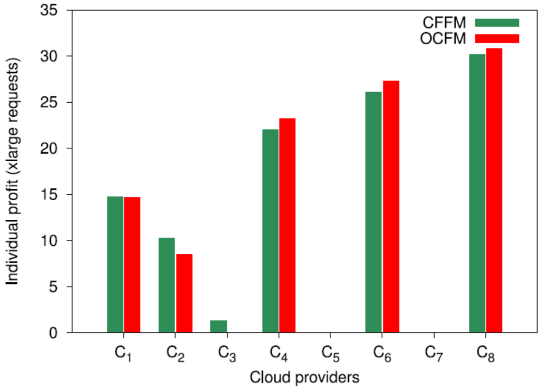

Figure 14: Extra-Large Requests.

Experimental Results (Cont.)

Participation of Providers

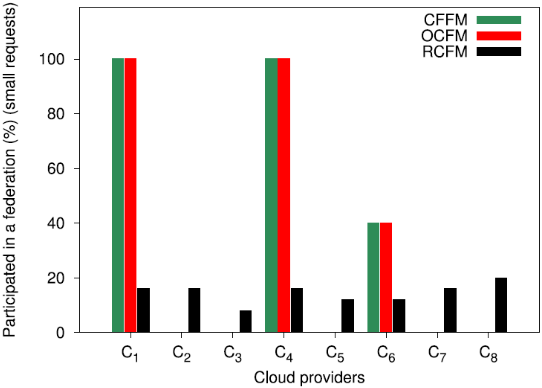

Figure 15: Small Requests.

Experimental Results (Cont.)

Participation of Providers (Cont.)

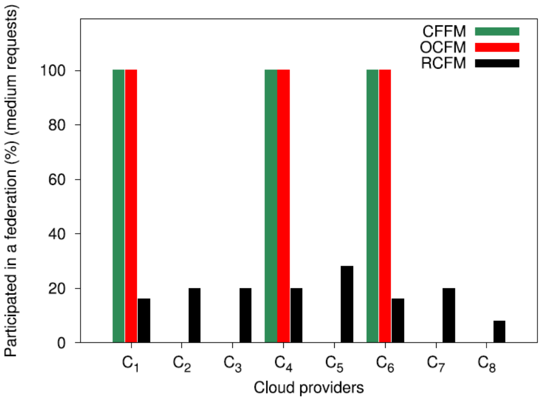

Figure 16: Medium Requests.

Experimental Results (Cont.)

Participation of Providers (Cont.)

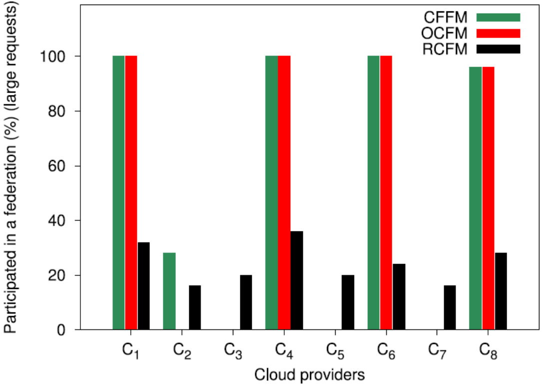

Figure 17: Large Requests.

Experimental Results (Cont.)

Participation of Providers (Cont.)

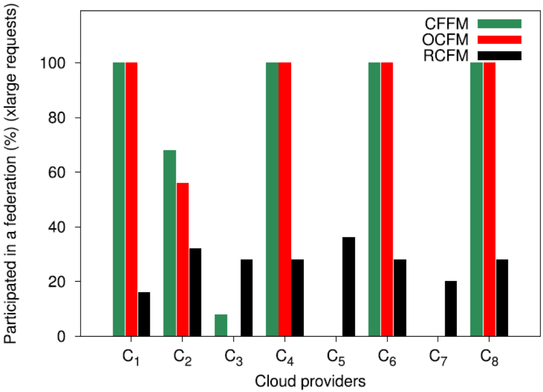

Figure 18: Extra-Large Requests.

Experimental Results (Cont.)

Provision Distribution

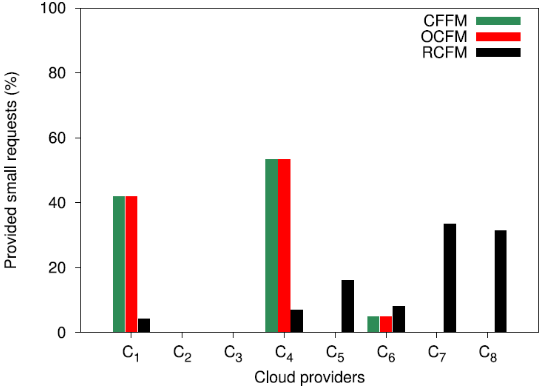

Figure 19: Small Requests.

Experimental Results (Cont.)

Provision Distribution (Cont.)

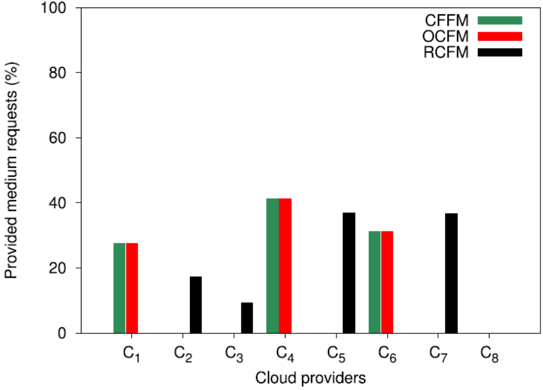

Figure 20: Medium Requests.

Experimental Results (Cont.)

Provision Distribution (Cont.)

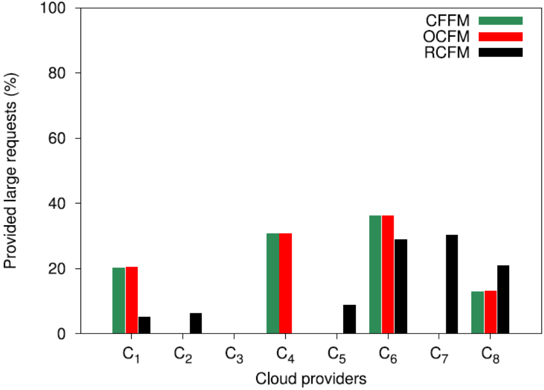

Figure 21: Large Requests.

Experimental Results (Cont.)

Provision Distribution (Cont.)

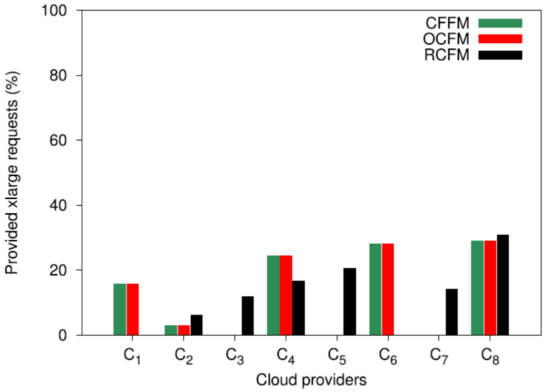

Figure 22: Extra Large Requests.

Experimental Results (Cont.)

Conclusion

The proposed CFFM is able to form stable federations with total profit very close to the optimal profit. Moreover, the running time is reasonable to find such a solution.

Future Work & Questions

Future Work

Take data privacy into account (may have achieved some progress though).

Future Work & Questions

Future Work

Take data privacy into account (may have achieved some progress though).

Questions?

More details on how IP-CFPM is tackled?
Also applicable for SaaS and PaaS?

Future Work & Questions

Future Work

Take data privacy into account (may have achieved some progress though).

Questions?

More details on how IP-CFPM is tackled?
Also applicable for SaaS and PaaS?
Anything else?

References I

[1] Mashayekhy, Lena, Mahyar Movahed Nejad, and Daniel Grosu.
"Cloud federations in the sky: Formation game and mechanism"
Cloud Computing, IEEE Transactions on 3.1 (2015): 14-27.

[2] Ferguson, T. S.
"Game theory (2nd/ed)"
Mathematics Department, UCLA 2014 (2014).

[3] Myerson, Roger B.
"Game Theory: Analysis of Conflict"
Cambridge, MA: Harvard UP, 1991. Print.