Olga Wenge, Dieter Schuller, Ulrich Lampe, Melanie Siebenhaar and Ralf Steinmetz: Composition of Cloud Collaborations under Consideration of Non-Functional Attributes. In: Proceedings of the International Conference on Service-oriented Computing (ICSOC), November 2014.

Composition of Cloud Collaborations under Consideration of Non-Functional Attributes

Olga Wenge, Dieter Schuller, Ulrich Lampe,
Melanie Siebenhaar, and Ralf Steinmetz

TU Darmstadt, KOM, Rundeturmstr. 10, 64283 Darmstadt, Germany
{firstName.lastName}@KOM.tu-darmstadt.de

Abstract. Cloud markets promise virtually unlimited resource supplies. Some providers set up distributed data centers at different geographical locations and jurisdictions and may not always be able to offer effectual physical capacity to serve large customers in one location. A solution is cloud collaborations, where multiple providers unite to conjointly offer capacities. Both Quality of Service and security properties of such collaborations will be determined by the "weakest link in the chain", therefore resulting in a trade-off between monetary aggregates, cumulative capacity and non-functional attributes of a collaboration. Based on our previous research, we examine in our paper efficient composition of cloud collaborations from the brokers perspective, considering Quality of Service and security requirements of cloud providers and users. We propose a Mixed Integer Programming-based heuristic approach CCCP-HEU.COM with deterministic and stochastic variants and provide its quantitative evaluation in comparison with our prior optimal approach.

Keywords: cloud computing, collaboration, QoS, security, cloud broker

1 Introduction

Cloud markets promise unlimited resource supplies, standardized commodities and proper services in a scalable, pay-as-you-go fashion [1]. Some providers set up distributed data centers at different geographical locations and jurisdictions and may not always be able to offer effectual physical capacity to serve large customers in one location. A solution is cloud collaborations within cloud markets, i. e., the cooperation of multiple providers to aggregate their resources and conjointly satisfy users demands. Supposably, such cloud collaborations have both Quality of Service (QoS) and security impacts. As a user may potentially be served by any provider within a collaboration, the aggregated non-functional service attributes (e. g., availability, security protection level, data center location) will be determined by the "weakest link in the chain", i. e., by the provider with the lowest guarantees. Consideration of country- and industry-specific data protection laws and regulations is another concern by building cloud collaborations, as providers can act in different jurisdictions (the European Union, Canada, Singapore, or the United States), where data privacy laws differ [4].

Based on our previous research [5], we examine the Cloud Collaboration Composition Problem (CCCP) with a focus on a broker, who aims to maximize his/her profit through the composition of cloud collaborations from a set of providers and assignment of users to these collaborations. In that assignment, QoS and security requirements, i. e., non-functional attributes, are to be considered and fulfilled. This work extends the previously introduced exact optimization solution approach with a heuristic approach that improves the computational time in the context of cloud markets.

The remainder of this paper is structured as follows: In Section 2, we briefly describe the problem and the formal optimization model, we discussed in our position paper [5]. Section 3 introduces a heuristic approach CCCP-HEU.COM with deterministic and stochastic variants, which is quantitatively evaluated and compared with the previous results. Section 4 concludes the paper.

2 Cloud Collaboration Composition Problem

In our work, we take the perspective of a broker, who acts within a cloud market and unites cloud providers to build cloud collaborations and provides assignment of cloud users to these collaborations. So, the cloud market consists of a set of providers $P = \{1, 2, \ldots, P^{\#}\}$ and a set of users $U = \{1, 2, \ldots, U^{\#}\}$. We define resource demand of each user $u \in U$ as $RD_u \in \mathbb{R}^+$ units, for which he/she is willing to pay a total of $M_u^+ \in \mathbb{R}^+$ monetary units. Resource supply of each cloud provider $p \in P$ is defined as $RS_p \in \mathbb{R}^+$ units at a total cost of $M_p^- \in \mathbb{R}^+$.

We define QoS and security constraints as *non-functional* constraints and distinguish two sets of quantitative $A = \{1, 2, \ldots, A^{\#}\}$ and qualitative $\hat{A} = \{1, 2, \ldots, \hat{A}^{\#}\}$ non-functional attributes. Quantitative attributes represent numerical properties, e. g., availability. Qualitative attributes depict nominal properties, e. g., applied security policies. The providers make certain guarantees with respect to the non-functional attributes. For each quantitative attribute $a \in A$, the value guaranteed by provider $p \in P$ is denoted as $AG_{p,a} \in \mathbb{R}$. For each qualitative attribute $\hat{a} \in \hat{A}$, the corresponding information is given by $\hat{AG}_{p,\hat{a}} \in \{0, 1\}$. The users specify certain requirements concerning their non-functional attributes. With respect to each quantitative attribute $a \in A$, the value required by user $u \in U$ is denoted as $AR_{u,a} \in \mathbb{R}$. Likewise, $\hat{AR}_{u,\hat{a}} \in \{0, 1\}$ denotes the requirement for each qualitative attribute $\hat{a} \in \hat{A}$, i. e., indicates whether this attribute is mandatory or not.

Based on these notations, the CCCP can be represented as an optimization model, as shown in Model 1. We define $x_{u,c}$ and $y_{p,c}$ as the *main* decision variables in the model (cf. Equation 11). They are binary and indicate whether user u or provider p, are assigned to collaboration c or not. We introduce $y'_{p,c}$ as auxiliary decision variables, which are binary as well and indicate the non-assignment of a provider p to a collaboration c. Furthermore, $z_{a,c}$ and $\hat{z}_{\hat{a},c}$ are defined as real and binary, respectively, and represent the cumulative value of the non-functional property a or $\hat{a}$, respectively, for collaboration c (cf. Equation 12).

The monetary objective function for a broker consists in profit maximization,

Model 1 Cloud Collaboration Composition Problem

$$\text{Max. } Pr(x, y, y', z, \hat{z}) = \sum_{u \in U, c \in C} x_{u,c} \times M_u^+ - \sum_{p \in P, c \in C} y_{p,c} \times M_p^- \tag{1}$$

such that

$$\sum_{c \in C} x_{u,c} \leq 1 \quad \forall u \in U \tag{2}$$

$$\sum_{c \in C} y_{p,c} \leq 1 \quad \forall p \in P \tag{3}$$

$$y_{p,c} + y'_{p,c} = 1 \quad \forall p \in P, \forall c \in C \tag{4}$$

$$\sum_{u \in U} x_{u,c} \times RD_u \leq \sum_{p \in P} y_{p,c} \times RS_p \quad \forall c \in C \tag{5}$$

$$z_{a,c} \leq y_{p,c} \times AG_{p,a} + y'_{p,c} \times \max_{p \in P}(AG_{p,a}) \quad \forall p \in P, \forall c \in C, \forall a \in A \tag{6}$$

$$\hat{z}_{\hat{a},c} \leq y_{p,c} \times \hat{AG}_{p,\hat{a}} + y'_{p,c} \quad \forall p \in P, \forall c \in C, \forall \hat{a} \in \hat{A} \tag{7}$$

$$z_{a,c} \geq x_{u,c} \times AR_{u,a} \quad \forall u \in U, \forall c \in C, \forall a \in A \tag{8}$$

$$\hat{z}_{\hat{a},c} \geq x_{u,c} \times \hat{AR}_{u,\hat{a}} \quad \forall u \in U, \forall c \in C, \forall \hat{a} \in \hat{A} \tag{9}$$

$$C = \{1, 2, \ldots, \min(P^{\#}, U^{\#})\} \tag{10}$$

$$x_{u,c} \in \{0,1\} \quad \forall u \in U, \forall c \in C \qquad y_{p,c} \in \{0,1\} \quad \forall p \in P, \forall c \in C \tag{11}$$

$$y'_{p,c} \in \{0,1\} \quad \forall p \in P, \forall c \in C \qquad z_{a,c} \in \mathbb{R} \quad \forall a \in A, \forall c \in C \qquad \hat{z}_{\hat{a},c} \in \{0,1\} \quad \forall \hat{a} \in \hat{A}, \forall c \in C \tag{12}$$

i. e., maximization of the difference between the revenue from the served cloud users and the spending on the used cloud providers (cf. Equation 1).

Equations 2 and 3 make sure that each user and provider are assigned only to one collaboration simultaneously. Equation 4 determines the inverse variable $y'_{p,c}$ for each decision variable $y_{p,c}$ (cf. Equations 6 and 7). These equations determine the cumulative non-functional values for quantitative and qualitative attributes and are formulated such that quantitative properties are given by the "worst" value among all providers in a certain collaboration. Equation 5 prevents the resource demand from exceeding the resource supply. Equations 8 and 9 make sure that users can only be assigned to collaborations with sufficient non-functional guarantees. Equation 10 defines a set of potential cloud collaborations, its cardinality is given by the number of users or providers, whichever is lower.

3 Heuristic Optimization Approach CCCP-HEU.KOM

In our previous research [5], we implemented the described model and evaluated the optimal approach CCCP-EXA.KOM in order to obtain an exact (i. e., profit maximal) solution. We used a Mixed Integer Program (MIP) and a branch-and-bound optimization algorithms [2]. The evaluation results indicated that the computation time of the proposed CCCP exact solution grows in dependence on the number of market participants and in the worst case it is exponential, thus indicating the need for development of heuristic approaches. In the following, we propose a heuristic optimization approach CCCP-HEU.KOM with the improved computation time. Our CCCP-HEU.KOM approach is based on the Divide-and-Conquer principle, i. e., we recursively breaking down the CCCP problem into sub-problems and combine the solutions of sub-problems to provide a solution to the original problem [3]. It consists of four components (sub-problems):

1. ASSIGN: Assignment of cloud users to cloud providers
2. COLLAB: Building of cloud collaborations
3. RCHECK: Checking of resource constraints
4. COMPOSE: Composition of cloud collaborations

ASSIGN: Assignment of users to providers. In this step, the assignment of users to providers will be performed with respect to the fulfillment of NFAs - non-functional requirements of users and non-functional guarantees of providers, as shown in Algorithm 2. The algorithm starts with two empty lists: $assign.P_p$ - a list of all assigned users $u \in U$ of a provider p, and $\hat{P}$ - a list of all providers who can satisfy at least one user. Non-functional guarantees (quantitative AG and qualitative $\hat{AG}$) of each provider will be compared with non-functional requirements (quantitative AR and qualitative $\hat{AR}$) of each user; if a provider p can fulfill the requirements of a user u (or has even better guarantees), then this user u will be added to provider's p list $assign.P_p$ (lines 5-8). Providers who cannot fulfill requirements of any user will be deleted (line 9). Users who cannot be served by any provider will be not added to the lists; thus, the number of users and providers will be reduced. At the end, a set $\hat{P}$ of NFAs-valid assignments (provider - users) is built with respect to the defined NFAs. Resource demand/supply constraints are not considered in this step.

COLLAB: Building of collaborations. In this step, we build cloud collaborations $\hat{C}$, i. e., we bring together providers, who can serve the same users. Thereby, Equations 6 and 7 are to be considered, i. e., the aggregated NFAs of collaborative providers will be defined by the worst ones. The set of valid collaborations is the intersecting set of $\hat{P}$. Applying of the intersection can be examined in two ways: determinictic and stochastic. By the deterministic approach (Algorithm 3), the complete set $\hat{P}$ will be searched through: all permutations of users $\hat{u} \in \hat{U}$ from the $assign.P_{\hat{p}}$ lists will be compared (lines 7-12). Thus, we have $\hat{P}^{\#} * 2^{\hat{U}^{\#}}$ possibilities (single provider sets and empty sets are exclusive), that leads in the worst case to asymptotical exponential runtime for $\hat{U}$, namely $O(\hat{P} * 2^{\hat{U}^{\#}})$. By the stochastic approach, we generate a random subset from the set $\hat{P}$ (Algorithm 4), where not all permutations are considered. The replacement of the Input ($\hat{P}$) of Algorithm 3 by the subset generation improves the algorithm and leads to asymptotical polynomial runtime.

RCHECK: Checking of resource constraints. In this step, we check resource constraints (as defined in Model 1). As shown in Algorithm 5, firstly, the quotients $Q_{\hat{u}} = M^{+}_{\hat{u}}/RD_{\hat{u}}$ (willingness to pay for a resource unit) will be calculated for all users from the provider-users assignments list $\hat{P}$. These quotients are then will be sorted in the descending order with respect to our objective function, namely, profit maximization (lines 5-9). So, the users with the best willingness to pay will be considered first.

COMPOSE: Composition of cloud collaborations. In this step, the best composition of cloud collaborations will be selected. As only one collaboration is allowed for providers and users simultaneously, the duplicates of them must be eliminated. So, the cloud collaborations with the same collaborative partners will be examined and the best constellation with respect to the maximum profit for a broker will be selected. The selected collaborations build then the complete solution of CCCP - $CCCPsol$. As shown in Algorithm 6, each collaboration $c \in C$ produces a certain profit PR_c. To provide an optimal solution, mostly profitable collaborations must be selected to fulfill the objective function. We apply here again the greedy principle and go through all collaborations. In lines (3-7) the collaborations that include the same collaborative partners will be compared - and the collaboration with the best profit $CCCPbest$ will be added to the complete solution $CCCPsol$. So, the composition of cloud collaborations occurs in a polynomial time.

3.1 Evaluation

To assess the required computation time of CCCP-HEU.KOM for different problem sizes and compare it with the exact optimization approach CCCP-EXA.KOM we provided before in [5] we prototypically implemented our heuristc approach in Java and used the same set up for our evaluation (JavaILP and IBM ILOG

Algorithm 2 Assignment

Input: set of providers $P = \{1, 2, \dots, P^{\#}\}$; set of users $U = \{1, 2, \dots, U^{\#}\}$
Output: set NFAs-valid provider-users assignments $\hat{P}$
$\hat{P} = \varnothing$; $assign.P_p = \varnothing$
for all $p \in P$ **do**
    for all $u \in U$ **do**
        if $AG_p \geq AR_u$ and $\hat{AG_p} \geq \hat{AR_u}$ **then** ▷ check the NFAs fulfillment
            $assign.P_p = assign.P_p + u$ ▷ assign user u to provider p
            if $assign.P_p = \varnothing$ **then** delete p
                $\hat{P} = \hat{P} + P_n(assign.P_p)$
            end if
        end if
    end for
end for

Algorithm 3 Building of collaborations (Full set)

Input: set $\hat{P}$ ▷ set of NFAs-valid provider-users assignments
Output: set of collaborations $\hat{C}$
$\hat{C} = \varnothing$
for all $\hat{p} \in \hat{P}$ **do**
    intersect $assign.P_{\hat{p}}$ with $assign.P_{\hat{p}+1}$ ▷ check shared users in assignment lists
    if intersect $\neq \varnothing$ **then**
        $users_{\hat{p},\hat{p}+1} = intersect(assign.P_{\hat{p}}/assign.P_{\hat{p}+1})$
        $\hat{C} = \hat{C} + \hat{c}_{\hat{p},\hat{p}+1}(users_{\hat{p},\hat{p}+1})$ ▷ build collaboration
        ... ▷ go through all permutations of users u
        $AG_{\hat{c}} = min(AG_{\hat{p}})$ and $\hat{AG_{\hat{c}}} = min(\hat{AG_{\hat{p}}})$ ▷ aggregated NFAs are
        ▷ determined by the worst ones
    end if
end for

Algorithmus 4 Building of collaborations (Random sub-set)

Input: set $\hat{P}$ ▷ set of NFAs-valid provider-users assignments
Output: subsets of $\hat{P}$
size=$\hat{P}.length$; $\hat{P}.subset = \varnothing$;
for $size < counter$ **do**
    for all $\hat{p}$ **do**
        subset = generate random subset from $\{1...size\}$
        $\hat{P}.subset = \hat{P}.subset + subset$
    end for
    counter=counter+1
end for

CPLEX framework). We regard *computation time* as the *dependent* variable of our evaluation. As *independent* variables, we include again the number of considered users and providers, i. e., $U^{\#}$ and $P^{\#}$. Each specific combination of $U^{\#}$ and $P^{\#}$ results in a *test case*. For each test case, we created 100 specific CCCP in-

Algorithmus 5 Checking of resources constraints

Input: $\hat{C}$
Output: set of built collaborations C with valid resources demand/supply
$\hat{RD}_{\hat{c}} = 0$ ▷ resource demand for collaboration $\hat{c}$
for all $\hat{c}_{\hat{p},\hat{p}+1} \in \hat{C}$ **do**
  for all $\hat{u} \in (users_{\hat{p},\hat{p}+1})$ **do** ▷ all users in the collaboration $\hat{c}_{\hat{p},\hat{p}+1}$
    calculate $Q_{\hat{u}} = M^{+}_{\hat{u}} / RD_{\hat{u}}$ ▷ quotients Q - willingness
    ▷ to pay for a resource unit
  end for
  sort $\hat{u}$ descending according to $Q_{\hat{u}}$ ▷ sorted list $\hat{U}_{\hat{c}}$
  for all $\hat{u} \in \hat{U}_{\hat{c}}$ **do**
    if $RS_{\hat{c}_{\hat{p},\hat{p}+1}} = RS_{\hat{p}} + RS_{\hat{p}+1} > RD_{\hat{u}}$ **then**
      $\hat{RD}_{\hat{c}} = \hat{RD}_{\hat{c}} + \hat{RD}_{\hat{u}}$
      if $RS_{\hat{c}} = RD_{\hat{u}}$ **then** stop ▷ maximum supply reached
      end if
    end if
  end for
end for

Algorithmus 6 Composition of cloud collaborations

Input: set of collaborations C
Output: solution of CCCP - $CCCPsol$
$CCCPsol = \varnothing$; ▷ complete solution
for all $c \in C$ **do**
  if $c_n \cap c_{n+1} \neq \varnothing$ **then** ▷ intersect set of c_n and c_{n+1} not empty
    $CCCPbest$ = insert c with $maxPR(c_n, c_{n+1})$ ▷ insert the collaboration
    ▷ with the best profit
  else
    $CCCPbest = c_n$
    $CCCPsol = CCCPsol + CCCPbest$
  end if
end for

stances with the according dimensions and used the same parameters. The results of our evaluation, i. e., the observed ratio of solved instances and the ratio of the mean computation times in comparison to the CCCP-EXA.KOM approach, are summarized in Table 1. As can be clearly seen, the mean computation times are drasticaly improved, and even the test case (12,18) by CCCP-HEUfull.COM (a heuristic with the full set COLLAB component) takes only 3.46% of the previosly computation time used by the exact approach. This variant shows rather optimal ratio of solving instances in all test cases. CCCP-HEUsub.COM (a heuristic with the sub-set COLLAB component) has better computation times, but the ratio of the solved instances (from 100 problem instances) goes already down with the test case (8,8). It explains also drastical improvement in CCCP-HEUsub.COM computation times for test cases (8,12)-(12,18), as not all solution will be examined - only in the randomly generated sub-sets.

Table 1. Evaluation results of CCCP-HEUfull.KOM and CCCP-HEUsub.KOM

Test case $P^{\#}$, $U^{\#}$	Ratio of solved instances HEUfull / HEUsub	Ratio of mean computation times HEUfull / HEUsub
$4, 4$	100% / 89,79%	0.94% / 0.50%
$4, 6$	98.23% / 81.78%	1.57% / 0.99%
$6, 6$	96.56% / 78.19%	1.87% / 1.13%
$6, 9$	92.47% / 67.77%	2.45% / 1.22%
$8, 8$	92.33% / 66.81%	2.62% / 1.45%
$8, 12$	87.34% / 63.93%	2.85% / 0.60%
$10, 10$	87.26% / 54.87%	3.30% / 0.45%
$10, 15$	85.20% / 54.84%	3.37% / 0.56%
$12, 12$	88.30% / 45.16%	3.40% / 0.40%
$12, 18$	82.52% / 49.96%	3.46% / 0.23%

4 Conclusions

While cloud markets promise virtually unlimited resources, the physical infrastructure of cloud providers is actually limited and they may not be able to serve the demands of large customers. A possible solution is cloud collaborations, where multiple providers join forces to conjointly serve customers. In this work, we introduced the corresponding *Cloud Collaboration Composition Problem* with our new heuristic optimization approach CCCP-HEU.KOM, as a complement to our prior exact optimisation approach. Our evaluation results indicated drastic improvement in the computation times, but showed also that the proposed heuristic optimization approach CCCP-HEU.KOM is still rather limited and needs further improvements, as a broker acts under rigid time constraints. In our future work, we aim at the development of heuristic approches with metaheuristics and dynamic changes. In addition, we plan to extend the proposed model with more complex non-functional constraints.

References

1. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. Future Generation Computer Systems 25(6), 599–616 (2009)
2. Hillier, F., Lieberman, G.: Introduction to Operations Research. McGraw-Hill (2005)
3. Jonson, D.S.: A Brief History of Np-completeness. In: Documenta Mathematica (2012)
4. Wenge, O., Lampe, U., Müller, A., Schaarschmidt, R.: Data Privacy in Cloud Computing An Empirical Study in the Financial Industry. In: 20th Americas Conference on Information Systems (AMCIS) (2014)
5. Wenge, O., Lampe, U., Steinmetz, R.: Qos- and Security-Aware Composition of Cloud Collaborations. In: 4th International Conference on Cloud Computing and Services Science (CLOSER). pp. 578–583 (2014)