QoS-aware trust establishment for cloud federation

Usama Ahmed
usamaahmed@hotmail.com
The Communication and Information Research Center (CIRC)
Sultan Qaboos University
P.O Box 17,
P.C 123, Muscat
Sultanate of Oman

Asma Al-Saidi
aalsaidi@squ.edu.om
The Communication and Information Research Center (CIRC)
Sultan Qaboos University
P.O Box 17,
P.C 123, Muscat
Sultanate of Oman

Ioan Petri
petriI@cardiff.ac.uk
School of Engineering,
Cardiff University,
Cardiff CF24 3AA, U.K

Omer F. Rana
RanaOF@cardiff.ac.uk
School of Computer Science
Cardiff University,
Cardiff CF10 3AT, UK

Abstract—Cloud federation enables inter-layer resource exchanges among multiple, heterogeneous Cloud Service Providers (CSPs). This paper proposes a Quality of Service (QoS) aware trust model for effective resource allocation in response to the various user requests within the Clouds4Coordination (C4C) federation system. This QoS mainly comprises of nine parameters combined into three categories: (i) node profile, (ii) reliability, and (iii) competence. Numerical values for these parameters are computed every 't' seconds for each cloud provider. All values measured over an interval Δt are further processed by the proposed model to evaluate the utility associated with a provider (referred to as a discipline in the presented case study). The decision about interacting with a discipline in a collaborative project is based on this utility value. The systems architecture, evaluation methodology, proposed model and experimental evaluation on a practical test bed is outlined. The proposed QoS-aware trust evaluation mechanism allows selection of the most useful (based on a utility value) providers. The proposed approach can be used to support federation of cloud services across a number of different application domains.

Index Terms—QoS, composite services, cloud, federation, trust.

I. Introduction

CLOUD federation also known as "cloud-of-clouds" provides services that involve aggregation of capabilities from multiple Cloud Service Providers (CSPs) [1]. Federated services help a CSP to deal with unanticipated changes in resource requirements by acquiring the same resource from other CSPs (often in a dynamic manner, i.e. the interaction pattern between CSPs may not be known apriori). In federated clouds, a Service Level Agreement (SLA) is signed between a user and its parent CSP, which in turn may have leased resource(s) or service(s) from various other providers that are a part of the federation [2]. Typically a permanent coalition formed between well-known CSPs having similar infrastructures, well acquainted due to continuous interactions with each other, offer limited benefit when application requirements/ demands change over time [3]. However, a cloud federation offers benefits in terms of scalability and potential for diversity in service composition in a dynamic manner [4]. Cloud providers contributing to a federation are not constrained by the resource limits of their peer CSPs and they have an option to choose from resource pools owned and shared by multiple peers. Regardless of this, CSPs may be reluctant to take part in a federation, typically owing to the absence of trust in other providers in the federation [5].

Trust establishment in cloud computing has always remained a foremost apprehension of users, despite the problems of access control and single sign-on support, Denial of Service (DoS) attacks etc. [6]. Cloud consumers fear the loss of control on their data and applications offloaded to a CSP [7-9]. Cloud federation takes this challenge of trust establishment one step further, by presenting a scenario of delegated trust due to connectivity of multiple service providers [5]. A chain of providers is now introduced, with a user having limited visibility on the set of providers involved in realizing a particular service. The performance of a federated service is always reflective of the performance of its multiple providers including a home CSP [10]. A home CSP can provide evidence to a customer for establishing trust in its own behavior. However, convincing the customer for trusting non-transparent sub-providers is quite challenging. In such a scenario of delegated trust, we suggest that establishing trust among CSPs in a federation is a unique problem and must be adequately addressed. An efficient solution to this problem will result in a trusted and reliable federation.

Although significant research already exists about establishing trust in cloud providers, parts of this literature is focused on trust between a cloud consumer and cloud providers. Recent literature for trust evaluation in conventional cloud computing [11-20], multi-clouds [21-26] and federated clouds [27] are all representative of the customer-to-cloud trust perspective. Limited coverage exists for trust management in cloud federation because it is assumed that all cloud computing models, including federation, share the same set of trust requirements. A federation must establish trust among its participating CSPs before sharing resources [28] as required by the cloud service user. This property of federation dictates the need for an adaptive model to establish trust among the participants of such federation, instead of relying on individual trust levels of user-facing CSPs [29].

Although precise decision making requires multiple sources of trust information [30], policy or service level agreement (SLA) based trust or feedback/recommendation based trust is not a feasible option in federations due to complexity and the likelihood of collusion. Instead, trust evaluation is based on capturing performance metrics that reflect real time behavior of service providers. Therefore, this work utilizes a QoS-aware trust evaluation mechanism so that participating CSPs can be assessed over a commonly defined feature space. For this purpose, QoS metric mainly comprise of three categories i.e. *node profile*, *reliability* and *competence*. These in turn comprise

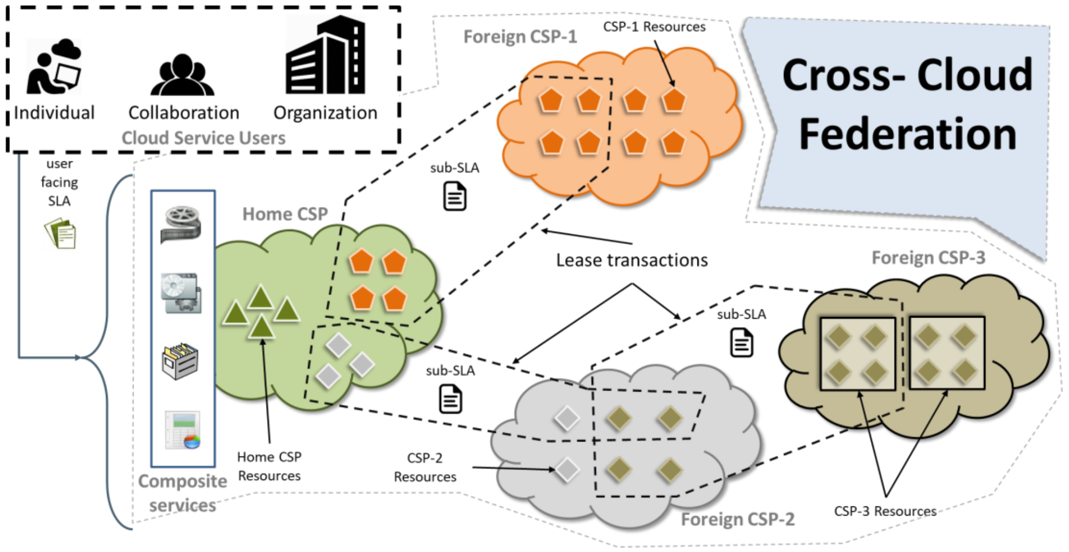


Figure 1: A typical cloud federation scenario

of nine service parameters as defined in later sections and numerical values of these parameters are computed every *'t'* seconds for each cloud provider in a federation. All values measured over an interval Δt are further processed by the proposed model to evaluate the utility value of a discipline. This approach tends to be optimistic for CSPs that are new entrants in the cloud market to compete with more mature service provider having long term experiences in service delivery. Moreover, the proposed approach has been validated on a federation testbed hosted on the Google Cloud Platform (GCP) utilizing real life workload from a construction project [31] instead of hypothetical datasets. This paper is divided into six further sections, the key concepts and related research is presented in section II. Section III presents the system architecture supporting the proposed research. Section IV presents the methodology and underlying details of the proposed QoS model respectively. Section V presents the experimentation, validation and results followed by a concluding discussion in section VI.

II. Basic Concepts & Related Work

This section describes the basic concepts of trust evaluation and cloud federation to support multi-disciplinary construction projects. Related work that focuses on cloud federation and trust evaluation to support such federation is also outlined.

A. *Cloud federation*

Cloud federation or bridging [1, 4] involves dynamic sharing of resources among CSPs. Service composition among heterogeneous participants of the federation occurs in layers [5, 10]. In a cloud federation, a request generated by a home CSP (against a specific SLA) to lease a resource is referred to as a transaction. All other exchanges for resources that originate as a part of this transaction can be termed as its sub-transactions. Generally, a (sub)-transaction is enacted across multiple stages, with the first being a resource discovery request, followed by resource matchmaking and eventually establishing a relationship between the CSPs [4].

Considering service as a method of representing, performing and delivering a specific task, a federated service is an accumulation of various sub-services or service components or resources from various providers. A federated service scenario is illustrated in Figure 1 having four CSPs. A home CSP is providing various SLA bound services to consumers (i.e. Individual and Enterprise) while other CSPs are shown as foreign CSPs. Each CSP owns a set of distinguished virtualized resources. Home CSP has one set of additional resources leased from *CSP-1*. Another set of resources is leased from *CSP-2*, which in turn has leased a part of these resources from *CSP-3* thus forming a hierarchy of resource exchanges.

All CSPs joining the federation may have heterogeneous infrastructures. A CSP could be a service provider with a large and sustained user community, or a new market entrant with limited service delivery experience. Each relationship within the federation is governed by rules and agreements, which essentially need to be a subset of the SLA signed by the consumer with its parent CSP [4, 5], thereby making the home CSP exclusively accountable for service delivery to the customer.

B. *Trust evaluation*

Conventionally, trust in an object has always been used to *measure* the extent to which that object will behave as expected. The notion of trust is mostly considered in the perspective of "performance", "security" and "privacy" parameters when it comes to distributed and multi-agent systems. In such cases, trust evaluation utilizes various indicator values of these parameters collected from multiple sources i.e. human perception, behavior and interaction with the system. In general, trust sources can be classified into three categories based on (*i*) *Recommendations*, either direct or transitive, provided to a potential user by others based on their own experience (*ii*) *Verification of contract* signed between the user and the provider to estimate variation from the defined thresholds in policy, and (*iii*) *Attribute assessment* to verify the capabilities and competencies of cloud providers. Although these sources are

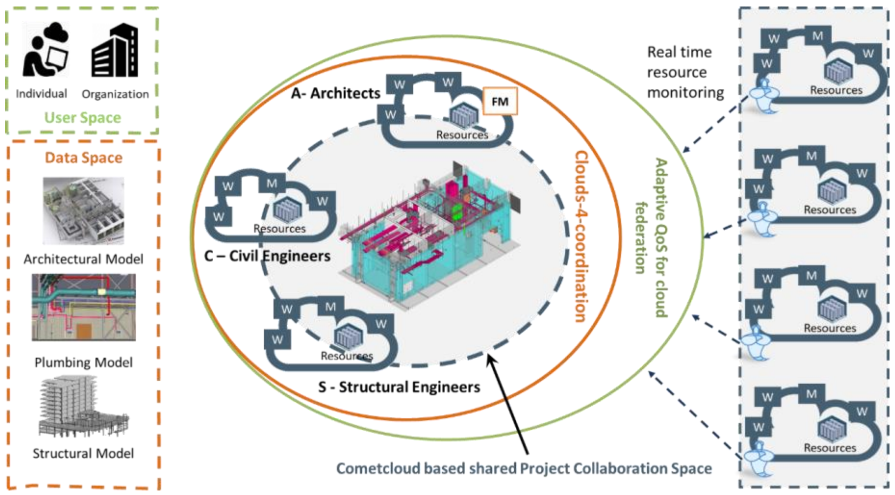


Figure 2: Abstract illustration of proposed system

context dependent, multi-source trust evaluation is always best suited for use in dynamic environments. In a cloud federation, however, trust evaluation based on policy verification is not suitable due to heterogeneity in its participants' infrastructures, services. Moreover, the only concern in a federation is cloud-to-cloud trust establishment, and does not require direct engagement and feedback from an end user [29]. Literature focusing on trust establishment in a cloud federation often utilizes the same user-to-cloud trust perspective with a variety of factors, such as behavior [32] or pricing [33] to characterize relationships within a federation.

Recent work also makes use of cloud interactions as a way to evaluate trust/reputation within a federation. A coalitional graph game, called "trust-aware cloud federation formation game" is proposed in [34] to support cooperation among cloud providers in a federation. The proposed approach considers a specific case of Map/Reduce programs while considering reputation among the participating cloud providers to achieve maximum profit for their participation. A cloud provider rates another cloud provider based on its direct interaction, as a basis for a local rating. However, the proposed mechanism does not take false feedback and other security vulnerabilities in recommendation based trust mechanisms into consideration.

Other work has proposed a lightweight algorithm [35] based on ratings of cloud providers using prior interactions. The proposed mechanism is an extension of the Trust Network Analysis with Subjective Logic (TNA-SL) algorithm [36] for cloud environments – utilizing recommendation and feedback ratings in the federation. This approach has an inherent risk of being susceptible to malicious feedback and collusion attacks.

Another work [37] has proposed a mechanism of Aggregated Capability Assessment (*AgCA*) for composite services in cloud federation. The authors have proposed a mechanism to evaluate the trustworthiness of a service as a factor of trust and dependency for all participants involved in offering the service. Trust evaluation has been carried out using subjective belief theory in which the trust score is an auditor's opinion regarding the security capability of a cloud provider. This is assessed based on the Cloud Security Alliance methodology (utilizing Consensus Assessment Initiative Questionnaire (CAIQ) and the Cloud Maturity Matrix (CMM)). Cloud providers can then engage in various service compositions and are only eligible to take part in a service if the trust of the entire service cluster remains above the given threshold. The approach however is only based on security-focused static audit information that only gets updated if there is any change in the provider infrastructure.

C. *Cloud federation for construction disciplines*

Cloud bridging or federation [38] is a complex case of dynamically sharing services/resources among different cloud providers. Service composition among heterogeneous participants of the federation may happen at any layer of cloud service delivery model [39]. Various cloud bridging solutions like IBMs Cast Iron Cloud Integration tools [40] are present in market, and are used in applications development in various environments. Cast Iron integrates with multiple other products and systems from IBM and various other vendors, such as Salesforces and SAP CRM. This helps in integrating locally placed systems with private and public Cloud environments [41].

Many other bridging solutions such as Oracle Cloud Machine which can connect to an external data center while being deployed on local infrastructure at customer premises are also available. Other similar work includes Munkley et al. [42], who have created information synchronization systems using Revit Server and internal storage server that allows customers to view a read-only version of the Revit (core) model. A client/server approach has been developed by Boeykens et al. [43] which provides an event based communication mechanism between embedded components of Building Information Modeling (BIM) authoring packages.

In relation to establishment of trust in cloud based construction projects, the rapid sharing of data also raises the issue of data confidence – acknowledged more commonly in the AEC industry through the use of 'issue status' for physical documents (where documents are provided status equal to what they can be used for reliably and thus what the issuing party assumes responsibility and/or liability for). This is motivated in particular by governmental objective of achieving full cooperative BIM across the AEC industry (with all project and asset information, paperwork and data being electronic). In [44], this aspect has been addressed by developing a "perceived trust model" for project collaboration in the C4C system, particularly for determining whether a new participating system can be integrated into a federation with existing ones. This approach has used cloud audit information to minimize the risk of include an unreliable system into a cloud federation. The audit information is retrieved from Cloud Security Alliance (CSA) repository in the form of Consensus Assessment Initiative Questionnaire (CAIQ), which is static self-assessment process for cloud providers. Another work proposed in [45] extends this approach to complement the static perceived trust with competence of the cloud provider to evaluate a Cooperation Value Estimate (CoVE) of cloud provider's ability to participate in a federation. However, the competence is not evaluated in real time and is measured as a factor of performance from previous projects that a particular cloud provider has been a part of.

Keeping in view the requirements of trust within the construction industry, the real time performance evidence along with the capability and competence assessment should be utilized to determine the trustworthiness of participants in the federation. Our proposed work utilizes a QoS-aware trust evaluation mechanism so that participating CSPs can not only be assessed over a commonly defined feature space but also provide irrefutable evidence of their performance available through a centralized Trusted Third Party broker. In this way, it is always ensured that the proposed trust mechanism follows the concept of EigenTrust. Among a set of '*n*' cloud providers, there are always a small number *'m' (m < n)* of pre-trust nodes in the network to bootstrap the federation. As these nodes increase, the impact of malicious nodes decreases.

III. System Architecture

A. Clouds4Coordination (C4C) overview

Construction projects are a complex set of activities involving a wide range of professions from multiple organizations. These organizations collaborate throughout the project life-cycle and generate variable amounts and types of data for the project. In this way, the federation acts as a mean to support collaboration among several disciplines i.e. suppliers, designers and cost consultants etc. for carrying out design and construction tasks.

For a construction project, any of the disciplines may join C4C by sharing its data located at a cloud based data center. A fundamental challenge in a C4C-enabled project is to create and manage the federation space through a master-slave cloud computing environment called CometCloud. Discipline specific service/ cloud providers can then use this federation to join/leave any time during the course of a project. This joining/leaving during the lifetime of the project occurs without any initial assessment (of their QoS). In the C4C system, each participating site must maintain a CometCloud deployment usually with a minimum of one master (agent) and multiple workers. The C4C master accepts project related tasks/ events from other participating disciplines and its workers execute these tasks and report results to the master node. Each master keeps a local copy of the project model made up of Industry Foundation Classes (IFC) objects i.e. an engineering and construction sector data format.

All disciplines within the C4C system communicate using a coordinating mechanism based on propagating events within the federation space. This mechanism is used to notify all disciplines about any new projects or changes within existing projects. Whenever any notification related to a project is disseminated, every master retrieves and updates the model and creates a new copy of the project on its local cloud. The entire federation is owned and managed by a discipline that acts as a "Federation Manager" (*FedMgr*), which is also responsible for retrieving the up-to-date version of the project.

When a new discipline needs to be added to the federation, trust values for the new discipline are evaluated before inducting this discipline to the project. Once a discipline qualifies for the given criteria, and is added to a project, the broker keeps track of its performance during the course of the project.

B. System entities and their roles

A number of different entities that constitute the C4C based cloud federation system are illustrated in Figure 2. These entities may appear in different roles (given a different context) and interact with each other influenced by certain rules that govern the federation.

- *AEC Organization / Discipline / Cloud Service Providers* – In the context of a C4C system, every AEC organization or discipline is the owner of multiple artifacts generated for the purpose of a project. The generated artifacts or data may be stored within a private cloud, or resources from multiple CSPs may be used in some cases. This is achieved by merging data sets from multiple parts of a BIM model (referred to as a "data space"). Any discipline represented by a CSP may appear in the federation in any of the two roles: either as a FedMgr or the discipline provider.

- *Cloud Service User / Project owner* – A Cloud Service User (CSU) is the owner of a project and responsible for: (i) identifying the QoS requirements of the project; (ii) managing the operations within the AEC project; (iii) managing interactions among various entities involved in an AEC project. This may include individuals or groups of users that use externally hosted services. A CSU is connected to the FedMgr through its service without any intervention from the cloud federation broker.

- *Cloud Federation Broker* – Service interactions between different AEC organizations participating in the federation are administrated by this Cloud Federation Broker (CFB). The CFB is responsible for evaluating and conveying the QoS utility value of a particular discipline to the participants of the federation before collaborating in a project. The broker-agent "Cloud Genie" is installed on locally managed cloud servers of each successfully registered discipline provider. Broker and its agent are responsible for starting the federation process, on request from *FedMgr*, on all disciplines that take part in the project. If a new discipline or provider is to be included in the project, it has to be registered with a description of its resources.

C. System workflow and interactions

The C4C federation system contains multiple entities interacting with each other to support the entire workflow of the proposed system as illustrated in Figure 3. The federation consists of

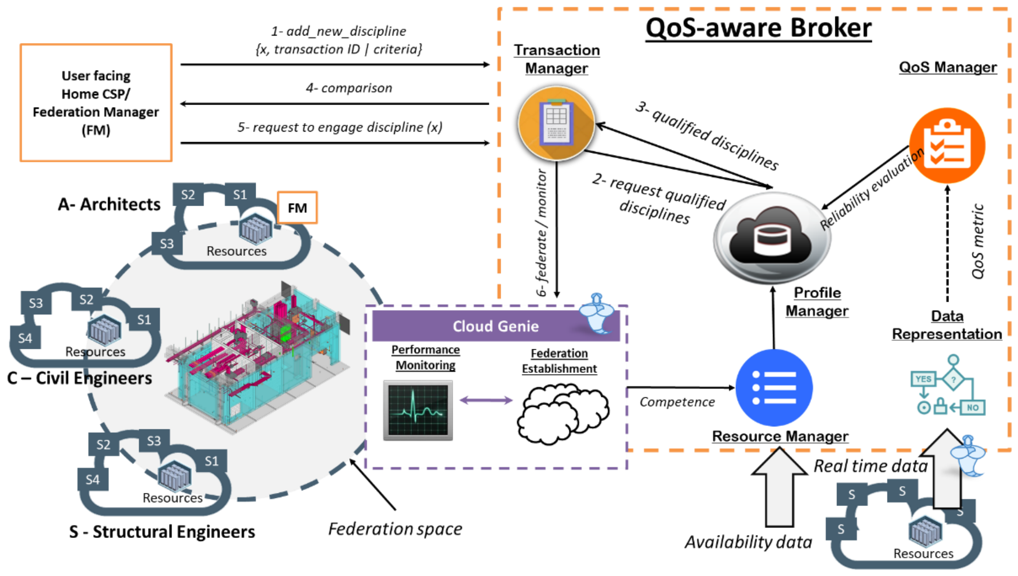


Figure 3: System design and interaction of entities within the federation

multiple disciplines, each offering a set of services to the project. The typical C4C process starts when *FedMgr* engages other disciplines to coordinate a project and adds/removes disciplines as required. However, the proposed architecture extends this mechanism to include a request from *FedMgr* to the CFB and a reply from *CFB* to *FedMgr* for the selection of disciplines, from the list of registered disciplines. This request is initiated by *FedMgr* every time a discipline is to be added to the project.

Each discipline interested in joining the federation has to be registered with the CFB by providing its credentials (name, endpoint, metadata etc.) and its resource capacity (or node profile as in section 3.1.1). After a discipline is a registered participant of the federation, it can participate in a project by leasing or acquiring resources (or capabilities) as and when required. The CFB keeps track of available resources offered by a discipline (reliability in section 3.1.2) with the help of a broker agent present within the discipline's cloud infrastructure. This agent helps the broker to maintain a Federated Resource List within the Profiler, containing a discipline specific list of resources that are either Waiting to Acquire (WTA) or Ready to Lease (RTL). A WTA is a resource demanded by any discipline that is required by the project and has not yet been matched. An RTL is an additional resource offered by a participant that has not yet been matched. Usually, a discipline generating a WTA is a part of an ongoing project and a discipline generating RTL is not a part of a project but announcing its capabilities to be taken onboard. The entire procedure for engaging a new discipline to the project with requisite resources is given as below.

- The *FedMgr* upon receiving a RFR generates a request *add_new_discipline (x, trid, criteria)* and forwards it to the CFB. Here 'x' is any discipline required in the project, *trid* is the project identification number and is null for the first time the federation is created, criteria and threshold is any attribute-value pair for the required resource, including values for node profile, reliability and responsiveness.
- This RFR is directed to the Transaction Manager, which verifies the availability of matching resource(s) and associated providers from the Federated Resource List. Multiple eligible providers for discipline 'x' from the Profiler repository that fulfil the criteria mentioned in the incoming request are forwarded to the QoS Manager.
- The federation manager then generates a Request to Engage (RTE) for that given discipline. A new transaction is generated whenever a RTE message is not associated with any ongoing transaction (i.e. a null trid). On the other hand, if the RTE contains a transaction id, then this resource exchange is termed as a sub-transaction. This entire mechanism from the generation of RFR until the engagement of resource(s) by a RTE is a recurring process for any federation participant.
- Adding new discipline to the federation also engages the 'Cloud Genie' on the requisite discipline to monitor the performance during the course of the project, and update the broker with the performance of the discipline. Every time a project completes, the broker computes the competence of the discipline based on the performance reported by the broker (section **IV.C**).

IV. ADAPTIVE QOS-AWARE TRUST MODEL

Our QoS model involves a number of dynamically invoked services within a cloud federation, keeping in view anticipated performance based on availability, reliability and responsiveness of these services. It is therefore essential to construct a suitable model that can predict performance and stability of a given service. The proposed QoS-aware trust model for a cloud federation environment can be expressed as: (S, R, T, C, U, D), where

- S = service request sent by a *CSU*,
- R = {resources} $\cup$ {services},
- T = time slot such that $T = (T_1, T_2, \ldots, T_n)$,
- $C := R \rightarrow V$ = capacity of each resource/ service given by an integer vector V,
- $U := (R \mid T) \rightarrow [0,1]$ = resource availability at a given time slot T.

- $D := S \rightarrow V$ = resource/ service demand function represented by an integer vector V.

The proposed QoS-aware trust model specifies an anticipated availability U of a service S for any period T_n. This availability is a prediction based on availability of a resource hosting the service during previous time slots T_{n-1} with an objective to distribute tasks efficiently for minimizing task failure and project delays. For this reason, the proposed model makes use of the historical usage pattern of the resource to predict service availability. In order to maintain an up-to-date status of the system and to provide the requested resources promptly, the model analyses three different types of information from each node: a basic node profile, real time information measured at regular intervals known as Reliability, and the non-real time performance information related to previously completed tasks known as Competence. For QoS analysis, the utilized resource/ service usage information is gathered using the nine different parameters as given in Table 1.

Table 1 QoS operators for trust prediction

	Parameters	**Type**	**Category**
I_1	CPU frequency	One time	Node Profile
I_2	Memory Size		
I_3	Storage Capacity		
I_4	Average Network Bandwidth	Recurring (real time)	Reliability
I_5	Average CPU utilization		
I_6	Average memory utilization		
I_7	Average storage utilization		
I_8	Average task success ratio	Recurring (non real time)	Competence
I_9	Average performance		

A. Node Profile

Node profile is the capacity of a cloud node in terms of available resources and can be measured using direct evidence from the broker agent. I_1 to I_3 denotes the capacity of each cloud in terms of resources e.g. as given in Table 2.

Table 2: Example of Node Profile

Cloud	**CPU**	**memory size**	**storage capacity**
N_1	1.90 GHz	3.0 GB	80 GB
N_2	2.40 GHz	5.0 GB	60 GB
N_3	2.33 GHz	6.0 GB	60 GB

B. Reliability

I_4 to I_7 are four reliability parameters obtained by real time monitoring of data using a broker agent. These values are a result of real-time evaluation of cloud resources within a given time window 'Δt'. For example, given a resource R that is performing 'n' tasks within time frame Δt, then

$$I_4(\Delta t) = \sum_{i=1}^{g} B(i) \Big/ n, \quad I_5(\Delta t) = \sum_{i=1}^{g} C(i) \Big/ n, \quad I_6(\Delta t) = \sum_{i=1}^{g} M(i) \Big/ n, \quad I_7(\Delta t) = \sum_{i=1}^{g} D(i) \Big/ n \qquad (1)$$

where $B(i)$ is the network bandwidth, $C(i)$ is CPU utilization, $M(i)$ is the memory utilization and D(i) is the hard disk utilization all measured at the ith time instance, for any cloud service provider.

C. Competence

Competence of a discipline is obtained by measuring two parameters in the context of tasks assigned to it at any given time. The first I_8 is the ratio of successfully completed tasks and the second I_9 is the performance of a given provider. Competence is then calculated as a product of these two parameters.

1) Average task success ratio

Average task success ratio, I_8 can be given as

$$I_8(\Delta t) = P(\Delta t) \Big/ P(\Delta t) + Q(\Delta t) \qquad (2)$$

where $P(\Delta t)$ and $Q(\Delta t)$ are the number of successful and unsuccessful tasks respectively. In order to acquire these values, a traditional mechanism is to let the broker agents on the running sites (subscribers) report when a task has stopped (due to completion or failure). However, this result reporting mechanism based on subscriber sites can bring the problem of fraudulent reports and site collusion. To overcome this limitation within a C4C federation, which is actually a collaborative paradigm, the running tasks are reported to the discipline which initiated the task (i.e. the publisher). Based on this mechanism of C4C federation, in which there is one discipline publishing the task (update or fetch) and at least one discipline subscribing to the request, four different cases for reporting results can be specified:

i. When the subscriber and the publisher report a task as successfully completed.

ii. When neither the discipline nor the *FedMgr* reports the result, it is considered as unsuccessful.

iii. When the publisher reports as not completed, it is considered as unsuccessful.

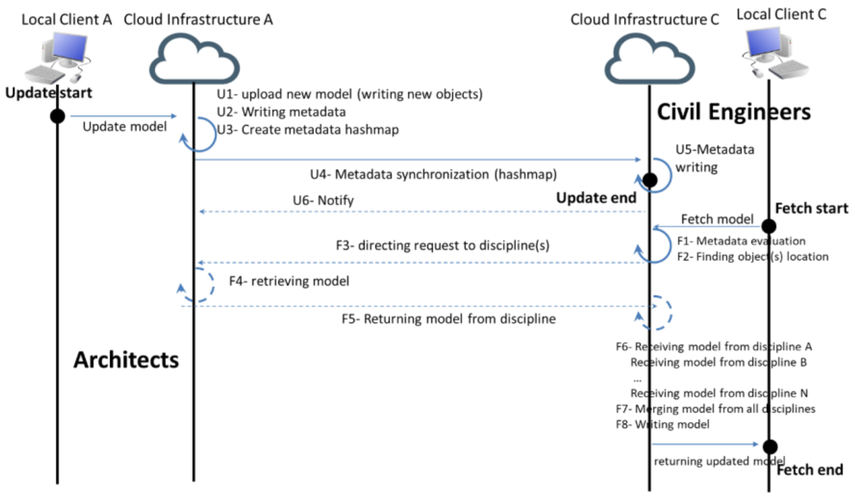

Local Client A
Cloud Infrastructure A
Cloud Infrastructure C
Local Client C
Update start
Update model
U1- upload new model (writing new objects)
U2- Writing metadata
U3- Create metadata hashmap
Civil Engineers
U4- Metadata synchronization (hashmap)
U5-Metadata writing
U6- Notify
Update end
Fetch start
Fetch model
F1- Metadata evaluation
F2- Finding object(s) location
F3- directing request to discipline(s)
F4- retrieving model
F5- Returning model from discipline
Architects
F6- Receiving model from discipline A
Receiving model from discipline B
...
Receiving model from discipline N
F7- Merging model from all disciplines
F8- Writing model
returning updated model
Fetch end

Figure 4: Perceived competence metric and workflow

iv. When the subscriber discipline reports the result as completed, and the publisher fails to reports it, it is considered as successful.

2) Average performance

Since the system can process different types of tasks, response time should be measured for a task with a fixed data size, and performance should be calculated for batch processing tasks with variable lengths of data. The proposed C4C system is a batch processing system with variable model sizes, therefore task based competence for individual disciplines is calculated. This calculation for each discipline is based on the number of objects assigned to it in the form of a BIM model, and the time taken by the discipline to process them.

To support this form of competence, each site is said to have a local C4C environment, which enables other sites to interact with it. In the workflow presented in Figure 3, discipline *A* creates the C4C project which is formed of Industry Foundation Class (IFC) objects stored at a local client *A*. Once a model is complete (or has been successfully updated), it has to be updated in the shared federation space so that other disciplines may work on it as per their role in the project. For example, considering discipline *A (Architects)* have finalized a building schematic as required by the customer, it should now be updated so that the the civil engineering discipline *C* is able to use it. All sites participating in the project are notified about the new project being created (based on role(s) in the project).

The update process starts when the local client A pushes the model data into the cloud. The cloud owned by discipline A writes the new objects and their metadata to disk. Afterwards a metadata HashMap is created and advertised in the shared federation space. All sites receive the update notification and update the metadata for the C4C project. This process of update is represented in Figure 4 by Update Start and Update Finish events. The time taken from U1 to U6 is called the update time and measured as:

$$update\_time = writing\_time + synchronization\_time \quad (3)$$

This process is similar in case of more than two sites participating in the project. For the lifetime of a project, whenever a discipline *N* updates a model with a newer version '*v'* in the shared space, a similar process is undertaken. This results in a similar round of notifications and metadata propagated to interested site(s). All sites eligible for version '*v*' updates their metadata with this new version so that they may fetch that version at any later stage if required.

During the project lifetime, a participating discipline may want to retrieve an updated version of the model. This process of retrieving the model is represented as a *Fetch* process. Figure 4 shows a civil engineering discipline *C* interested to work on the model updated by discipline *A* recently. The fetch process from discipline *C* is represented to occur between ***'Fetch Start'*** and ***'Fetch End'*** in Figure 4. This includes (*i)* finding the location of all objects marked as updated in the metadata, (*ii)* requesting the discipline(s) for updated model(s), (*iii)* receiving these model(s), (*iv)* merging all the different updates and *(v)* writing the model to the shared space.

Once a merged model has been created for a given discipline, it is returned to the local client for that discipline for working and updates. A model fetched at the local client is afterwards updated with any changes and may again be updated to the shared space following the update model process. The total time it takes for fetch process starting from F1-F8 in Figure 4 is denoted by the fetch time. Writing *(w)*, synchronizing *(s)* and fetch *(f)* times are utilized as a measure of performance for any given discipline as:

$$performance(n) = \frac{number\_of\_IFC\_objects\_processed}{time\_taken} \quad (4)$$

D. QoS Evaluation

In order to evaluate the QoS of any given cloud provider, the reliability parameters (I_4 to I_7) are first derived and normalized (to enable comparison across a common scale). In order to normalize the values of these parameters over any time window Δt_j, given *n* cloud resources and a total group of measurement samples $X = \{x_1, x_2, \cdots, x_z, \cdots, x_n\}$. For *n* resources, we obtain a characteristic matrix as follows.

$$X(\Delta t_j) = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ & & \ddots & \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix} \quad (5)$$

Given $1 \le z \le n$ with $x_z = x_{z1}, x_{z2}, \cdots, x_{zm}$ and $1 \le k \le m$ with $x_k = x_{1k}, x_{2k}, \cdots, x_{nk}$ any row operator $x_{zk} \in I_k$ is normalized into a range of [0.01, 0.99] obtaining a new matrix $\square\left(\Delta t_j\right) = \left(r_z\right)_{n\times 1} = \left(r_{zk}\right)_{n\times m}$ as given below:

$$\square(\Delta t_j) = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1m} \\ r_{21} & r_{22} & \cdots & r_{2m} \\ & & \ddots & \\ r_{n1} & r_{n2} & \cdots & r_{nm} \end{pmatrix} \quad (6)$$

Considering $\square\,(\Delta t_j)$ as one sample of data at the j_{th} time-stamp window Δt_j, each row operator r_z can given by

$$r_z = \sum_{k=1}^{m} r_{zk} \Big/ m \quad (7)$$

The reliability of a discipline provider composed of real-time trust parameters I_4 to I_7 is then the average resource utilization. The best node is then defined as a utility function having these three types of service operators i.e. node profile, reliability and competence as:

$$\rho_i = (\omega_N N_i + \omega_R R_i + \omega_C C_i) \quad (8)$$

where ρ is the utility of interaction i.e. the trust associated with provide node i that belongs to a specific discipline between {0,1}. An incoming *RFR* requesting a discipline for a project contains the same weighted *criteria* as per user requirement for these three parameters. The matching providers are then ranked in order of their utility of interaction, and the results are returned to the Federation Manager. The selection with the largest value is then invoked by the broker to take part in the project, as defined in section III.C.

From a computational perspective, the project collaboration framework is dynamically created at runtime, where disciplines join or leave based on a trust assessment. Each discipline also has multiple processes that carry out task executions on locally available resources. A process starts when a client process requests an update (changes one or more IFC object(s)) and terminates when the update is observed at another discipline. This requires an object to be transferred from the client's local machine to the remote discipline that has requested a view or update. The overheads of the framework are measured with an aggregated time-to-complete (ATTC) metric that depends on the number of IFC objects being executed, the number of simultaneous client requests that need to access the federated model and the number of disciplines that are part of the federation. The overheads of trust evaluation are however negligible as compared to these overheads of the framework.

V. Experimentation

This section describes the implementation details along with the experimental evaluation of the proposed approach. Data acquisition and system settings are also described in this section to elaborate experimentation.

A. *Experimental setup*

The proposed model has been implemented using a Trusted Third Party (TTP) QoS-aware broker and its agent built in Python. They are integrated with the C4C federated cloud system that is based on the CometCloud [46] federation framework. The broker agent resides within the cloud providers' infrastructure and makes use of Linux system level commands to obtain data from C4C nodes/ disciplines. The entire C4C federation along with the broker is hosted at the Google Cloud Platform (GCP) in multiple geographical region as described in Table 3. All nodes have similar specifications i.e. the effect of node profile is not considered as of now.

Table 3: System specification for experimentation

Type	OS	Compute	Memory	Storage	Region (no. of nodes)
C4C Nodes	Ubuntu 16.04.5 LTS	1 vCPU	3.75 GB	10 GB	europe-west2-a (4) us-east1-b (3) asia-east1-a (3) europe-west4-a (2)
QoS-aware Broker	------ same as above ------				asia-south1-a (1)

Each discipline provider has numerous values assigned to it as defined in Table 1. For the sake of simplicity, only three disciplines ('A' as architect, 'C' as civil engineer and 'S' as structural engineer) are considered in these experiments for which the 'Node profile' parameters are stored once initially. Each category has four candidate providers, for which the parameters for the 'Reliability' metric are observed and stored in real time at the QoS-aware broker for each provider. The 'competence' metric is retrieved and stored every time a workload is finished executing in the federation.

The proposed research makes use of actual workloads of varying sizes i.e. {300, 689, 956, 3442, 5342, 8940} KBs obtained from a highway construction project [31]. Every workload (as project *'P')* has a number of update and fetch tasks carried out by each discipline from time to time. The project is considered complete when each discipline has at least: (*i) made one update to the project* and (*ii) fetched one version of the final model from the shared space*. In order to ensure coherence among the performance metrics associated with disciplines, the size of update that each discipline makes to the project is identical. These nodes are setup and connected to the broker, enabling live monitoring at the broker with the following considerations.

1. Data associated with metrics I_4 to I_7 are collected over time and stored in the broker repository.
2. The operating system running on the node has minimal/ negligible CPU and memory footprint with limited network traffic sent and received across the federation.
3. CPU, memory, network and disk utilization are frequently varied using hypothetical workload generated by open source tools. Two different cases are considered with utilization load varied with thresholds set to: (i) ≤ 25% and (ii) ≤ 50% for each node.

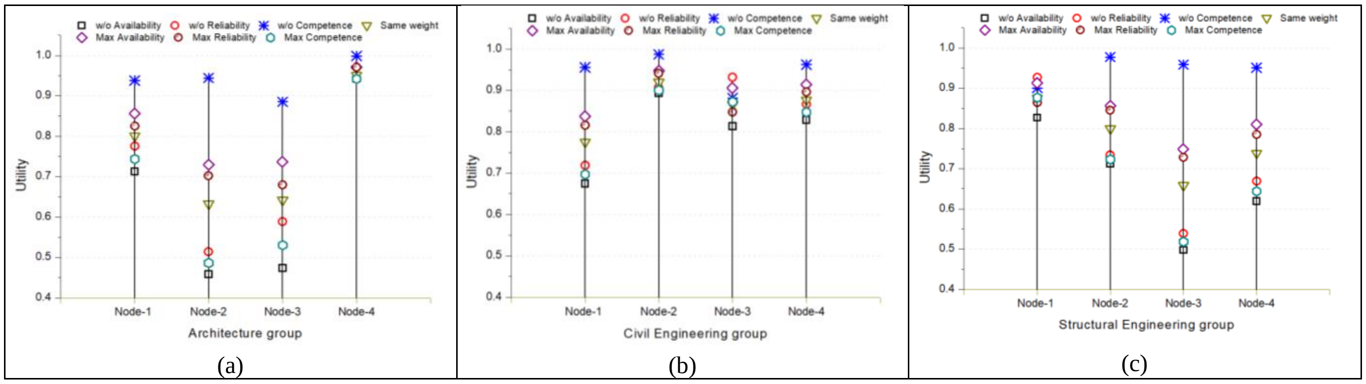


Figure 5: All data set with max utilization below 25% for experiment 1

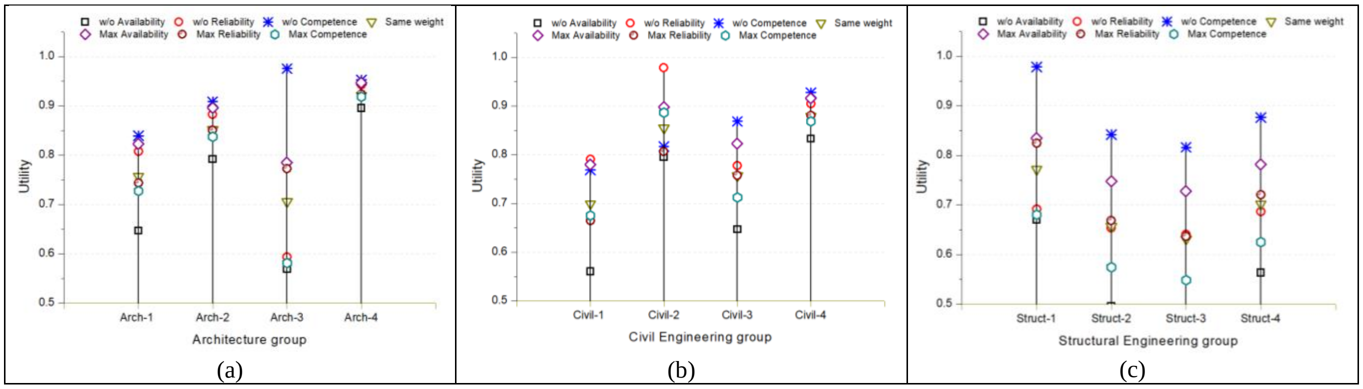


Figure 6: All data set with max utilization below 50% for experiment 1

After the federation has been setup, all users connects to their respective disciplines and performs their desired update and fetch operations as defined at the start of section V.

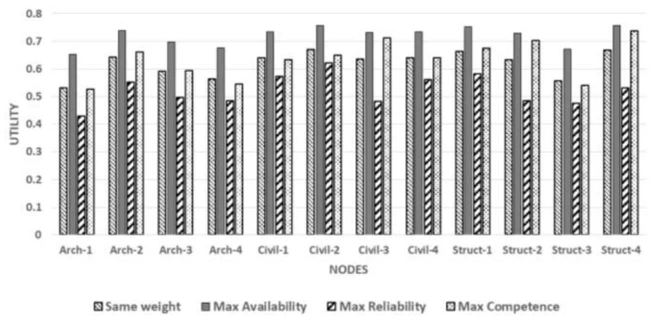
B. Experiment-1

The aim of this experiment is to collect QoS metrics data without considering any specific time slot, achieved by continuously monitoring each node. The data is gathered for 4 days including data for performing a series of projects involving all discipline providers and construction workloads. Since ρ has the maximum value of 1, the weight (ω) is set to 0.33 for each operator i.e. node profile, reliability and competence.

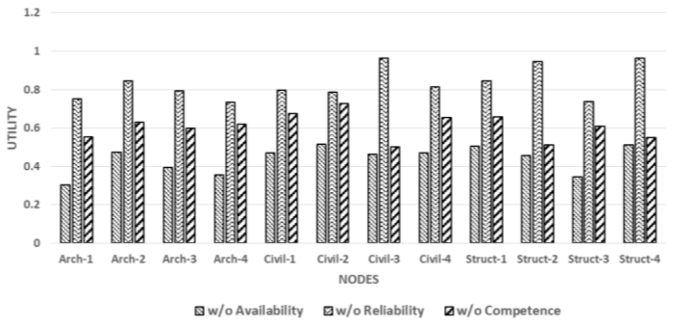
Error! Reference source not found. represents the results of service selection with varying weight assignments considering the complete dataset when the utilization of a node is kept equal to or below 25%. **Error! Reference source not found.** represents the results of service selection considering the complete dataset when utilization of a node is kept equal to or below 50%. In the case of varying weight assignment, ρ is iteratively calculated with alternate weights for node profile, reliability and competence. As shown in **Error! Reference source not found.** and **Error! Reference source not found.**, participants from various service groups are compared for their utility on the basis of varying weight assignment to different service parameters. At first a specific case is considered where each service parameter is iteratively assigned a zero weight (referred to as w/o availability, w/o reliability and w/o competence in the graphs) to nullify its effect on the service selection. Afterwards, maximum weights are iteratively assigned to any single service parameter starting with node profile (referred to as max availability in graphs), whereas reliability and competence share the remaining weight in this case. Similarly, in the second and third iterations maximum weight is assigned to reliability (referred to as max reliability in graphs) and to the competence (referred to as max competence in graphs) respectively leaving the remaining parameters with lesser weights. The service groups identified for the same weight assignment are considered in Figure 9. The same process can be used to identify the set of disciplines for any project, based on the weight associated with each discipline.

C. Experiment-II

This experiment is based on the observation that each node cannot maintain a consistent usage pattern over a period of time. Hence the current data set may reflect the potential usage pattern in a better way. This experiment uses QoS data for each node gathered for the last 24 hours. All other parameters remain the same as in Experiment 1. Figure 7 shows the results of service selection with varying weight assignments considering this recent dataset when the utilization of a node is kept equal to or below 25%. Similarly, Figure 8 represents the results of service selection when utilization of a node is kept equal to or below 50%. In Figure 7 (b) and Figure 8 (b) represents a case when each service parameter (node profile, reliability or competence) is iteratively assigned a zero weight to nullify its effect on the utility value and hence on service selection.

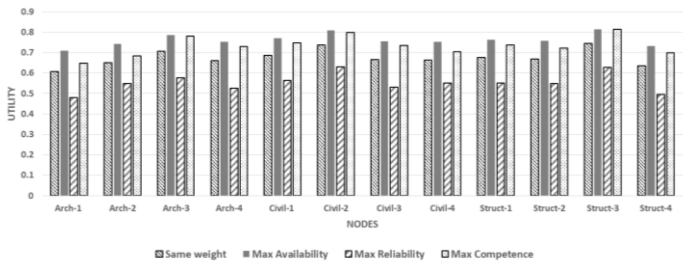


(a)

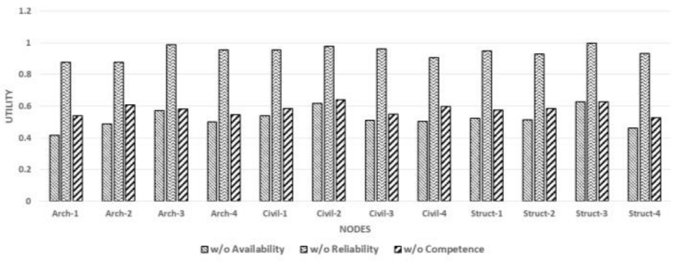


(b)

Figure 7: Recent data set with max utilization kept below 25%



(a)



(b)

Figure 8: Recent data set with max utilization kept below 50%

D. Experiment III

The aim of this experiment is to use the outcomes of the previous experiments to identify the *best* set of cloud providers within a C4C federation. Three different sets of nodes are identified i.e. one set of nodes selected on the basis of the proposed QoS criteria from previous experiments as shown in Figure 9, one makes use of Reliability, the other focuses on Competence achieved during establishing the federation. Figure 10 and Figure 11 show the two sets of nodes identified on the basis of reliability and competence respectively. These three sets of nodes are then used in C4C projects for executing the same workload and project completion times are observed for each project. Figure 12 shows the overall project completion times for all selection criteria. It can be observed from Figure 12 that the proposed QoS selection criteria performs the same for both recent and older datasets. Since Reliability is the only variable that is time dependent, selection based only on Reliability has performance degradation for the older data set.

Discipline	All Dataset		Recent Dataset	
	≤ 25% utilization	≤ 50% utilization	≤ 25% utilization	≤ 50% utilization
Architecture	Arch-4	Arch-4	Arch-2	Arch-3
Civil Engineering	Civil-2	Civil-4	Civil-2	Civil-2
Structural Engineering	Struct-1	Struct-1	Struct-4	Struct-3

Figure 9: Service groups for project execution in case of same weight assignment with proposed approach

Discipline	All Dataset		Recent Dataset	
	≤ 25% utilization	≤ 50% utilization	≤ 25% utilization	≤ 50% utilization
Architecture	Arch-4	Arch-3	Arch-2	Arch-2
Civil Engineering	Civil-2	Civil-4	Civil-2	Civil-2
Structural Engineering	Struct-2	Struct-1	Struct-1	Struct-3

Figure 10: Service groups for project execution based only on best reliability

Discipline	All Dataset		Recent Dataset	
	≤ 25% utilization	≤ 50% utilization	≤ 25% utilization	≤ 50% utilization
Architecture	Arch-4	Arch-4	Arch-2	Arch-3
Civil Engineering	Civil-3	Civil-2	Civil-3	Civil-2
Structural Engineering	Struct-1	Struct-1	Struct-4	Struct-3

Figure 11: Service groups for project execution based only on best competence

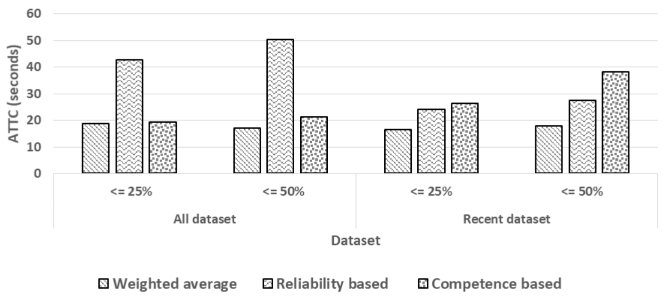


Figure 12: Project completion time (seconds) for various selection criteria

VI. CONCLUSION

This paper has presented a QoS-aware trust establishment mechanism for C4C-based federated cloud system. The proposed mechanism makes use of profiles of participating nodes, Reliabilty of their usage patterns over a period of time and Competence based on their positive performance. The selection mechanism ranks nodes according to user requirements and their role in the project. We have validated our approach using experiments carried out on the C4C system using a BIM dataset. The top ranked nodes selected for each discipline through the proposed selection are then integrated in different projects for the validation of accuracy and comparison to other selection mechanisms i.e. only based on performance and usage pattern. In future, we propose to extend this work by having dedicated use of in-premises high profile cloud platforms (local cluster) like IBM or Azure etc. Moreover, we propose to extend this work by introducing more parameters specifically related to user perceived aspects of trust.

VII. ACKNOWLEDGEMENT

This work has been funded by the Deanship of Research at Sultan Qaboos University (Grant IG/DVC/CIRC/18/02).

The system used in this work was developed in the “Cloud4Coordination” project (EPSRC/InnovateUK), and hosted at School of Engineering at Cardiff University, U.K.

REFERENCES

[1] N. Grozev and R. Buyya, "Inter-Cloud architectures and application brokering: taxonomy and survey," *Software: Practice and Experience*, vol. 44, pp. 369-390, Mar 2014.

[2] R. Buyya, R. Ranjan, and R. N. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *International Conference on Algorithms and Architectures for Parallel Processing*, 2010, pp. 13-31.

[3] V. Massimo, B. Ivona, and T. Francesco, *Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice*. Hershey, PA, USA: IGI Global, 2012.

[4] A. Celesti, F. Tusa, M. Villari, and A. Puliafito, "How to enhance cloud architectures to enable cross-federation," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, 2010, pp. 337-345.

[5] K. Bernsmed, M. G. Jaatun, P. H. Meland, and A. Undheim, "Thunder in the Clouds: Security challenges and solutions for federated Clouds," presented at the 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom '12), 2012.

[6] Y. Chen, V. Paxson, and R. H. Katz, "What’s new about cloud computing security," *University of California, Berkeley Report No. UCB/EECS-2010-5 January*, vol. 20, pp. 2010-5, 2010.

[7] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, *et al.*, "Controlling data in the cloud: outsourcing computation without outsourcing control," presented at the Proceedings of the 2009 ACM workshop on Cloud computing security, Chicago, Illinois, USA, 2009.

[8] K. Julisch and M. Hall, "Security and Control in the Cloud," *Information Security Journal: A Global Perspective*, vol. 19, pp. 299-309, Nov 2010.

[9] S. Pearson and A. Benameur, "Privacy, security and trust issues arising from cloud computing," in *IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom '10)* 2010, pp. 693-702.

[10] A. Al Falasi, M. A. Serhani, and R. Dssouli, "A model for multi-levels SLA monitoring in federated cloud environment," in *10th International Conference on Ubiquitous Intelligence and Computing and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC '13)*, 2013, pp. 363-370.

[11] M. Mrabet, Y. ben Saied, and L. A. Saidane, "Modeling correlation between QoS attributes for trust computation in cloud computing environments," in *17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2017, pp. 488-497.

[12] C. Mao, R. Lin, C. Xu, and Q. He, "Towards a Trust Prediction Framework for Cloud Services Based on PSO-Driven Neural Network," *IEEE Access*, vol. 5, pp. 2187-2199, Jan 2017.

[13] V. C. Emeakaroha, K. Fatema, L. v. d. Werff, P. Healy, T. Lynn, and J. P. Morrison, "A Trust Label System for Communicating Trust in Cloud Services," *IEEE Transactions on Services Computing*, vol. 10, pp. 689-700, Sept 2017.

[14] T. H. Noor, Q. Z. Sheng, Z. Maamar, and S. Zeadally, "Managing Trust in the Cloud: State of the Art and Research Challenges," *Computer*, vol. 49, pp. 34-45, Feb 2016.

[15] Q. Li, W. Wu, J. Huang, Z. Sun, and M. Feng, "A Novel Trustworthy Framework for Cloud Based Rendering Application," in *IEEE Trustcom/BigDataSE/I SPA* 2016, pp. 1951-1956.

[16] N. Ghosh, S. K. Ghosh, and S. K. Das, "SelCSP: A framework to facilitate selection of cloud service providers," *IEEE transactions on cloud computing*, vol. 3, pp. 66-79, Jan 2015.

[17] S. M. Habib, S. Ries, M. Mühlhäuser, and P. Varikkattu, "Towards a trust management system for cloud computing marketplaces: using CAIQ as a trust information source," *Security and Communication Networks*, vol. 7, pp. 2185-2200, Nov 2014.

[18] S. M. Habib, V. Varadharajan, and M. Muhlhauser, "A trust-aware framework for evaluating security controls of service providers in cloud marketplaces," in *12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom '13)* 2013, pp. 459-468.

[19] S. Rizvi, J. Ryoo, Y. Liu, D. Zazworsky, and A. Cappeta, "A centralized trust model approach for cloud computing," in *23rd Wireless and Optical Communication Conference (WOCC), ,* 2014, pp. 1-6.

[20] S. Rizvi, K. Karpinski, B. Kelly, and T. Walker, "Utilizing Third Party Auditing to Manage Trust in the Cloud," *Procedia Computer Science*, vol. 61, pp. 191-197, Jan 2015.

[21] O. A. Wahab, J. Bentahar, H. Otrok, and A. Mourad, "Towards Trustworthy Multi-Cloud Services Communities: A Trust-based Hedonic Coalitional Game," *IEEE Transactions on Services Computing*, vol. 11, pp. 184-201, Jan 2018.

[22] X. Li, H. Ma, F. Zhou, and W. Yao, "T-broker: A trust-aware service brokering scheme for multiple cloud collaborative services," *IEEE Transactions on Information Forensics and Security*, vol. 10, pp. 1402-1415, Jul 2015.

[23] X. Li, H. Ma, F. Zhou, and X. Gui, "Service operator-aware trust scheme for resource matchmaking across multiple clouds," *IEEE transactions on parallel and distributed systems*, vol. 26, pp. 1419-1429, May 2015.

[24] H. Shen and G. Liu, "An efficient and trustworthy resource sharing platform for collaborative cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, pp. 862-875, 2014.

[25] A. Celestini, A. L. Lafuente, P. Mayer, S. Sebastio, and F. Tiezzi, "Reputation-based cooperation in the clouds," in *IFIP International Conference on Trust Management*, 2014, pp. 213-220.

[26] W. Fan and H. Perros, "A novel trust management framework for multi-cloud environments based on trust service providers," *Knowledge-Based Systems*, vol. 70, pp. 392-406, Nov 2014.

[27] A. Kanwal, R. Masood, and M. A. Shibli, "Evaluation and establishment of trust in cloud federation," in *8th International Conference on Ubiquitous Information Management and Communication*, 2014, p. 12.

[28] K. Bernsmed, M. G. Jaatun, P. H. Meland, and A. Undheim, "Thunder in the Clouds: Security challenges and solutions for federated Clouds," in *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, 2012, pp. 113-120.

[29] U. Ahmed, I. Raza, and S. A. Hussain, "Trust Evaluation in Cross-Cloud Federation: Survey and Requirement Analysis," *ACM Computing Surveys (CSUR)*, vol. 52, p. 19, 2019.

[30] J. Huang and D. M. Nicol, "Trust mechanisms for cloud computing," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 2, p. 9, Dec 2013.

[31] I. Petri, O. F. Rana, T. Beach, and Y. Rezgui, "Performance analysis of multi-institutional data sharing in the Clouds4Coordination system," *Computers & Electrical Engineering*, vol. 58, pp. 227-240, 2017.

[32] S. Pang, Q. Gao, T. Liu, H. He, G. Xu, and K. Liang, "A behavior based trustworthy service composition discovery approach in cloud environment," *IEEE Access*, vol. 7, pp. 56492-56503, 2019.

[33] Q. Wu, M. Zhou, Q. Zhu, and Y. Xia, "VCG auction-based dynamic pricing for multigranularity service composition," *IEEE Transactions on Automation Science and Engineering*, vol. 15, pp. 796-805, 2017.

[34] L. Mashayekhy, M. M. Nejad, and D. Grosu, "A Trust-Aware Mechanism for Cloud Federation Formation," *IEEE Transactions on Cloud Computing*, 2019.

[35] H. Kurdi, A. Alfaries, A. Al-Anazi, S. Alkharji, M. Addegaither, L. Altoaimy, *et al.*, "A lightweight trust management algorithm based on subjective logic for interconnected cloud computing environments," *The Journal of Supercomputing*, vol. 75, pp. 3534-3554, 2019.

[36] A. Jøsang, R. Hayward, and S. Pope, "Trust network analysis with subjective logic," in *Proceedings of the 29th Australasian Computer Science Conference-Volume 48*, 2006, pp. 85-94.

[37] U. Ahmed, I. Raza, O. Rana, and S. A. Hussain, "Aggregated Capability Assessment (AgCA) for CAIQ enabled Cross-cloud Federation," *IEEE Transactions on Services Computing*, 2021.

[38] A. N. Toosi, R. N. Calheiros, and R. Buyya, "Interconnected cloud computing environments: Challenges, taxonomy, and survey," *ACM Computing Surveys (CSUR),* vol. 47, p. 7, Jul 2014.

[39] D. Villegas, N. Bobroff, I. Rodero, J. Delgado, Y. Liu, A. Devarakonda, *et al.*, "Cloud federation in a layered service model," *Journal of Computer and System Sciences,* vol. 78, pp. 1330-1344, Sept 2012.

[40] I. Solution. (2019, Last accessed May 21, 2018). *CastIron*. Available: https://www-01.ibm.com/software/integration/cast-iron-cloud-integration/salesforceintegration/.

[41] P. Mell and T. Grance, "The NIST definition of cloud computing," NIST, USA 800-145, 2011.

[42] J. Munkley, M. Kassem, and N. Dawood, "Synchronous building information model-based collaboration in the cloud: A proposed low cost IT platform and a case study," in *Computing in Civil and Building Engineering (2014)*, ed, 2014, pp. 89-96.

[43] S. Boeykens and N. Koenraad, "Interactive bi-directional BIM model and application linking," in *Proceedings of IASS Annual Symposia*, 2015, pp. 1-11.

[44] U. Ahmed, I. Petri, O. Rana, I. Raza, and S. A. Hussain, "Risk-based Service Selection in Federated Clouds," in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, 2018, pp. 77-82.

[45] U. Ahmed, I. Petri, O. Rana, I. Raza, and S. A. Hussain, "Federating Cloud Systems for Collaborative Construction and Engineering," *IEEE Access,* vol. 8, pp. 79908-79919, 2020.

[46] H. Kim and M. Parashar, "CometCloud: An autonomic cloud engine," *Cloud Computing: Principles and Paradigms,* pp. 275-297, 2011.