УДК 004.9

Nataliya Boyko
Lviv Polytechnic National University
ISN deparment

CLOUD SERVICES AND THEIR SELECTION ALGORITHM

Описано архітектуру хмарних сервісів та основні технології для алгоритму вибору сервісів (послуг). Аналізовано хмарні обчислення є актуальними для різного виду програмного забезпечення та розподілених обчислень, які ґрунтуються на доступі до Інтернету, що означає, що користувачі можуть отримати доступ до сховищ і програм з віддалених серверів за допомогою web-браузерів чи інших стаціонарних або мобільних терміналів. Описуються ресурси стаціонарних та мобільних терміналів, які є обмеженими, саме тому хмарні обчислення є правильним інструментом, який забезпечить термінали різного роду потужними ресурсами для досягнення складних сервісів. Розглянуто архітектуру хмарних сервісів та ключові алгоритми вибору сервісів з адаптивними виконаннями і мінімальними витратами. Запропонована архітектура хмарних сервісів є небезпідставною, а наведені алгоритми вибору необхідного сервісу є доступними, можуть змінювати масштаб та є водночас адаптивними до різних типів сервісних і клієнтських середовищ.

Ключові слова: система, інформаційний потік, інформаційні процеси, "хмарні обчислення", корпоративні інформаційні системи, інформаційна технологія.

This article describes the architecture of cloud services and key technologies for algorithm selection services. Cloud computing is relevant to various types of software and distributed computing based on Internet access, which means that users can get an access to applications and storage from remote servers using web browsers or other stationary or mobile terminals. Resources of stationary and mobile terminals are limited, which is why cloud computing is the right tool that will provide different kinds of terminals with powerful resources to achieve complex services. In the article considered the cloud services architecture and key algorithms for choosing services of adaptive performance and minimal costs. Architecture cloud services is arguable, as proposed algorithms choose the desired service is available, and can change scale, it is adaptable to different types of service and client environments, at the same time.

Keywords: system, information flow, information processes, "cloud computing," corporate information systems, information technology.

Introduction

Quick advances in cloud computing gradually changes the mode of distribution and maintenance of software and applications. In addition, hardware manufacturers suffer losses on conversion of information products, people may need only basic devices to connect to the Internet and for infinitely powerful resources and services from remote servers. This is, as a huge challenge, as an opportunity to make distributed computing more mobile. Cloud computing is actually a cloud service to users and customers, which has the following characteristics: dynamism, lightness, low cost, reliability and confidentiality. Cloud services are increasingly important in terms of costs and benefits. In the future there will be many services available from service or cloud computing, and the choice of optimal service is the main problem, this is our motivation and contribution into this project.

Related works

T. V Raman [1] presented a model of use / delivery, where application logic, along with the data network is in the clouds. Cloud computing [2] is similar calculations "on demand", as the software is similar

to the service or the Internet as a platform. Cloud computing does not provide local and remote access to various types of software and applications. Random practical application of cloud computing are services for handling with the documents (Google Docs, Adobe Buzzword, Photoshop Express, etc.) and computing in enterprises (Salesforce.com, IBM Blue Cloud, Amazon S3 storage cloud, EC3, etc..). Cloud computing [3] based on decades of research in the field of virtualization of distributed computing, utility computing, networking and software services. In the document [4] presented by cloud service integration and expansion of the distributed system architecture. Infrastructure based on cloud [5], includes cloud storage services, cloud data services and cloud computing services. Clouds are different from grids [6], so that they evolved from Web 2.0, while a grid has isolated of cooperation simple calculations. There are three types of displays using clouds: large procurement proposal centers with many hardware with good natural communications, and companies with large IT departments, who will build their own clouds (such as Intel, Boeing, and eBay). Cloud services are expanded concept of cloud computing. Cloud service focuses on server-side service mode and method of selecting service - for computing and data from the cloud.

The aim of the study is to determine the characteristics of cloud services for mobile computing and analysis of algorithms of their choice.

Architecture of cloud service

Cloud services for mobile computing services include: architectural services methods and procedures (forward request / process / selection / return), which uses both server and client side. Cloud service as service-oriented architecture, can be divided into hierarchical model, based on its types provided to clients. We discuss some aspects of customer service and parts.

Hierarchical model of cloud service

Cloud service provides all kinds of services from infrastructure hardware to data, software, applications and storage. Hierarchical cloud service model can be represented in Fig. 1.

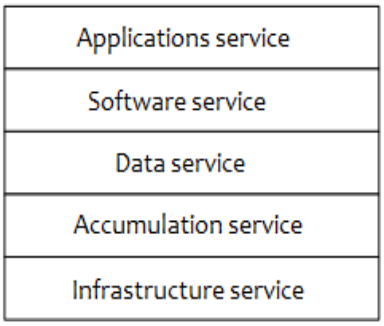

Fig. 1. The hierarchical model of cloud service

Cloud services can be seen as an ecosystem, consisting of devevopers and consumers, according to its life cycle from emergence up to the application, and users can be divided into 4 types according to Fig. 2. This is hierarchical structure, the shape of which means that users with different numbers scale, bottom-up, usually numbers tend to increase.

Granularity Service (graininess)

When there are many services in cloud, they must be organized in a hierarchical structure to reduce the cost of the search. Fig. 3 shows us that the root service may consist of subservices. This also means, that cloud services can have several partial replications in distributed nodes, that require certain features cloud-based service. Locality and replication may reduce the cost of communication and ensure the reliability of recovery. Service granularity (graininess) varies as a pair of low-grade and high degree of intensity. Low-service small grain may rise to high granularity, as well as integrated average grain service may increase to service of large or extra large granularity that satisfies an integrative task or project. Fig. 3, puff numbers marked nodes can be atomic services that are used to achieve higher service levels and higher capacity.

Atomic services can be all kinds of service units such as hardware, software, platform, storage, data, components, and business, etc. that may be spread wide coverage areas or clusters specified some remote servers that belong or operated by different companies or virtual organizations or in owners collaboration or competitive contracts, policies, laws or regulations, governed by economic or other kinds of benefits.

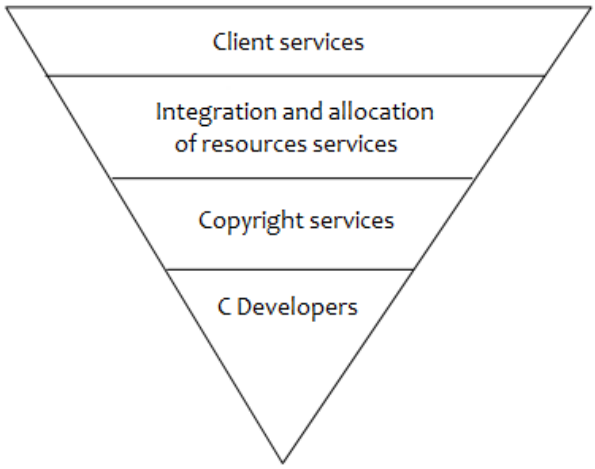

Fig. 2. The hierarchical structure of the cloud ecosystem

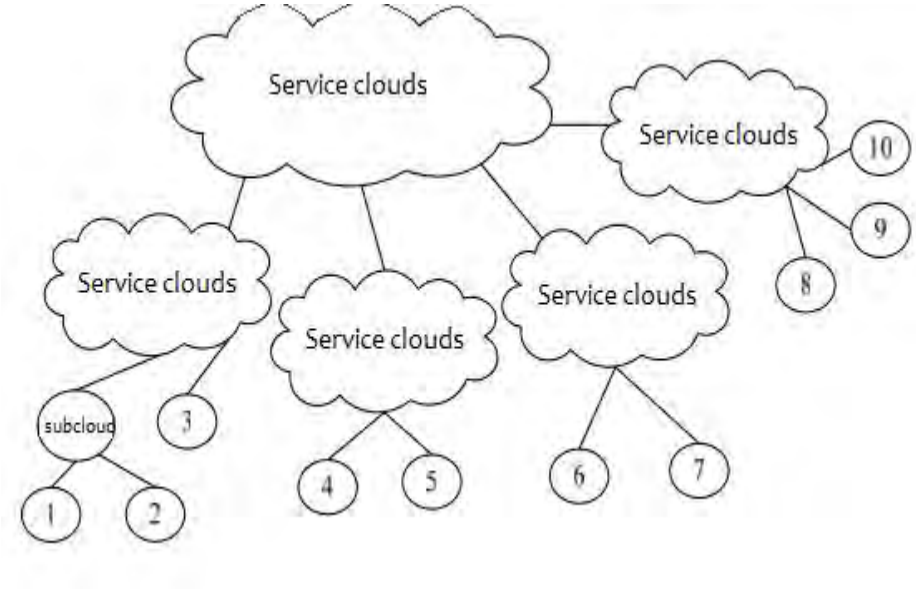

Fig. 3. Grainy cloud service model

Selection

When users get access the cloud service, they will choose the services that best meet their needs. Cloud service delivery mechanism has to evaluate custom preferences by selecting the desired context and design service that meets certain requirements.

Users opportiunities

In the mobile computing user capacity is limited. For example, mobile devices have small screen, limited storage space, poor ability to process dynamic energy and limited mobility. Cloud services must meet all of these requirements to the result of his work was effective, responsive and accessible.

The relationship between clients and services

The main goal of cloud services is to provide customers convenient and low-cost service. The relationship between cloud and client service are obvious. Customers can access the cloud services using the service proxy, a proxy service can be one or more, as shown in Fig. 4 and Fig. 5.

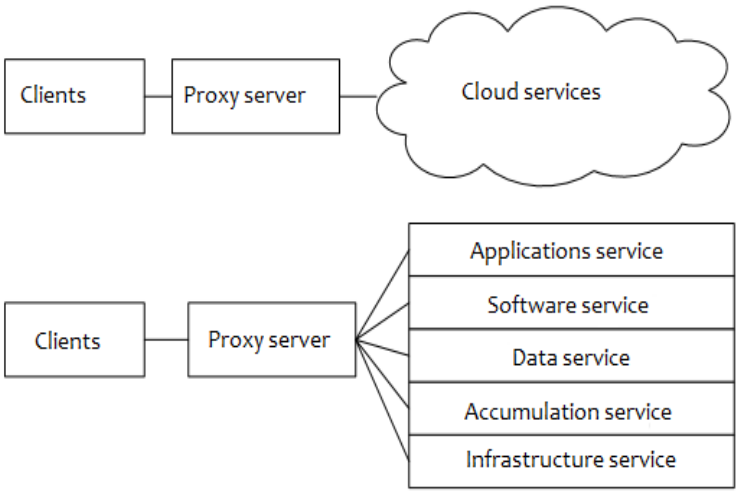

Fig. 4. Access to the cloud through a simple proxy

Fig. 4. Illustrates all clients access to the cloud service through a proxy server, which is responsive only to small scale services, which can easily lead to errors. Fig. 5. we describe user access to the cloud service with many proxy servers of various types, locations, etc. The last ratio is adaptive to large clouds of scale and is the best in reliability, load balancing, stability, etc.

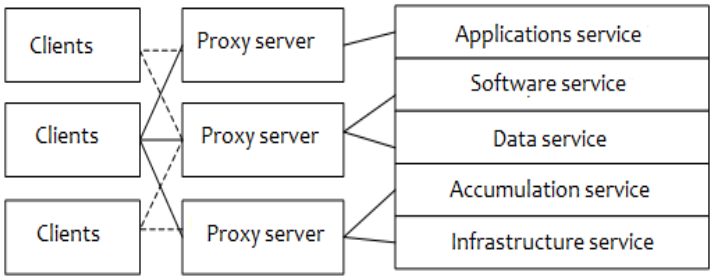

Fig. 5. Access to the cloud through several proxies

Conduct mode service

Like service-oriented structure, there are three roles in cloud services: check-in service, discovery service and proxy service as shown in Fig. 6. Before the service becomes available, it should be registered. When a client sends a request to the service, the service should be open by using a proxy server. When a proxy server is needed customer service, it responds to the request of the client and performs certain functions or components.

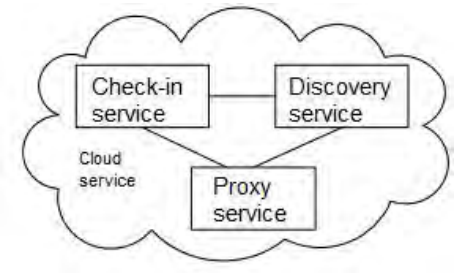

Fig. 6. Daily cloud service behavior

Here there are many constituent units, they are: service registration, service proxy, service discovery, based on the requirements for configuration provider, and the requirements of the cloud. Multiple proxy services can interact with each other to build a network overlay for cooperation or competition. This will involve complex and interesting tasks, similar to humanity organization and social behavior that can be discussed later by game theory, evolutionary algorithms, etc.

Search service algorithm

When customers get access the cloud service via a proxy server, many servers will be available for them. Choice algorithm of services will determine the cost of services and their benefits. Cloud services has such service attributes as price per unit of service, distance, time response, traffic and places to save, etc. When a client sends a request service, he does not know his attributes, unlike the proxy server that has all the information. Client shall only make a request for a particular purpose, such as minimum cost or maximum profit, then proxy server assess attributes relevant service and choose the best for customer service. If there are multiple proxy servers, the principle of selecting the best service remains unchanged.

The procedure for selecting the service can be illustrated as follows. When a client sends a request for a service through a proxy server, proxy server has the following algorithm. First, the proxy selects a list of all the relevant services from the cloud using the service discovery capabilities and assesses the state of these services, remove unavailable services and their elements, and keeps a list of all available services. Second, the proxy server chooses from a list of available services based on user preferences and objectives, such as the maximum benefit, the best performance or minimal cost. Third, proxy directs client request to your service provider, in turn, provider performs certain programs to return the results of computations required or service. If only selected provider can't succeed to return the client required service, proxy server will be notified and will choose the next best provider to do it again. If attempts to return exceeds a certain threshold (eg, 5 times, or 16 minutes), the proxy server returns an error to the client. Finally, the client must send a service request again not in any order or voluntarily.

Selection list of available services

When a customer sends request to the proxy server, it asks service discovery to get list of existing services and the associated degree of availability of service. Accessibility tools can include a many aspects, of service and as S_i, we denote as A_{i1}, A_{i2}, A_{i3},... A_{ij},..., A_{im}, where m – is the number oft tribute for every service. Also we use W_{ij} to refer importance of every service attribute weight, where $j = 1, 2, 3,..., m$. Then the availability of service can be characterized as A_i, described by equation (1). Service with attribute A_i have to be more than the threshold θ, then the service is available and it is added to the list of available services. Later proxy server need to choose the most optimal service from the list of available services. If there is a large number of services, the proxy can choose several of the best services available as a set of services.

$$A_i = (\sum_{j=1..m} W_{ij} A_{ij}) - q \quad (1)$$

Algorithm of election list of available services representations in Fig. 7.

```
// Algorithm 1
AvailableServiceListSelection (Si, Wij, Aij where i = 1..n, j = 1..m; float Oo; int k)
// Si  - means, that service is in the cloud
// Aij - means, j attribute of service i
// Wij - means, every service attribute weight
// Oo  - means, availability threshold
// k   - means, the expected number of service, initialized number 0
{
    As = { };
    for_each Si in the cloud by service metadata do
    {
        calculate Ai by equation (1);
        if (Ai >= 0) then
        {
            As = AsU{(Si,Ai)};
        }
    }
    sort(As) by Ai descendent;
    if (As >= k and k > 0) then
    {
        return Ask which top k items of As;
    }
    return As;
}
```

Fig. 7. List of available tools election algorithm

Selection of service with a maximum use and minimal costs

Given that, there are *k* avaible services from systems with cloud services, that are selected by previous algorithm, the next step is to choose an optimized set from a set of available services. Algorithm of selection service with the greatest benefit can be described as follows. It is assumed that the service *i* has *m* subbenefits G_{i1}, G_{i2}, G_{i3},..., G_{im}, and importance of weight can be mark as α_{i1}, α_{i2}, α_{i3},..., α_{im}. Then benefit G_i can be calculated using the formula (2).

$$G_i = \sum_{j=1..m} a_{ij} G_{ij} \quad (2)$$

Proxy service expects to benefit from the list of available services, service and receives the maximum benefit. Minimum cost selection algorithm can be described as follows. Given that the service *i* has *m* -subcosts C_{i1}, G_{i2}, C_{i3},..., C_{im}, and importance of weight β_{i1}, β_{i2}, β_{i3},..., β_{im}. Then the minimum cost C_i can be calculated using the formula (3).

$$C_i = \sum_{j=1..m} b_{ij} C_{ij} \quad (3)$$

Proxy calculates the costs of services from the list of available services and then gets service at minimal cost.

Considering the proportion *r*, which corresponds to the maximum benefit, and *(1-r)*, which meets the minimum costs, where $r \in [0,1]$. If $r = 1$, it is the algorithm of maximum usefulness, if $r = 0$, the algorithm is equal to the reverse algorithm of minimum cost. There proxy server have to select grid of the most useful services to improve service, designated as G_{ci}.

Compromise maximizing benefits and minimizing costs is presented in Figure 8.

```
// Algorithm 2
OptimizedServiceSelection (Si, aij, Gij, bij,Cij where i = 1..n, j = 1..m; float r)
// Si       - means, that service is on the cloud
// Aij      - means, j attribute of service i
// Gij      - means, j link of service i
// Cij      - means, j maintenance costs of service i
// aij, bij - means, take the value 1 or value set provider
// r        - means, that the percentage increase from 0 to 1
{
    Os = { };
    for_each Si in the cloud by the available service set As do
    {
        calculate Gi by equation (2);
        calculate Ci by equation (3);
    }
    select maximum of Gi;
    select minimum of Ci;
    calculate Gci by equation (4);
    Os = sort {(Si, Gci) i = 1,2,...};
    select the service k with maximized Gci;
    do
    {
        execute the service k;
        if (success) then
        {
            return result to the client;
        }
        else
        {
            Os = Os - {(Sk, Gck)};
            select the next neares service k with Gci to be executed;
        }
    } while (Os <> OO and (not timeout));
    if (timeout) then
    {
        return failure to the client;
    }
}
```

Fig. 8. Optimized algorithm of service selection

Analysis of the two-step selection algorithm

We can define algorithms that have been described above, as two- stage selection services algorithm, as the first step is to identify the list of avaible services, and the second – optimization of selecting of service, taking into account the maximum benefit and minimum cost. We mostly appreciate and analyze the performance of two-stage algorithmby such factors as availability and scalability. Other factors of performance such as reliability and security more concern to applications. First we are talking about - the availability. It is assumed that the condition of these algorithms is independent of services and types of customers, whether they are services of various types, so-called “rich” clients as users PCs, laptops, or “poor” as owners of PDA devices, mobile phones and so on, the two-step algorithm is accessible and responsive to them.

Second, we discuss scalability. With algorithm of service selection, proxy can choose the best available service that meets the user’s query. Even if the number of client requests is huge, proxy server perform the same calculation method to choose the best service to the customer. Therefore, this algorithm is reliable and scaled. Also, proxy servers can communicate with each other in cloud services that will help reduce the computation time and increase efficiency, but increase time communication and increase traffic. Results of range of services selection can be cached or recorded in the profiles for improved performance later related processes for selection.

Conclusion

This article cloud service contains more expanded content than just cloud computing. Also it was proposed cloud service architecture and a two-stage selection algorithm service. Typically, users (clients) validate access to the cloud service using a proxy server to select the service with the lowest cost and greatest benefit, such as greater speed of response, greater efficiency or greater convenience. The algorithm is affordable, scaled and adaptable to different kinds of clients and service environments. This algorithm can be applied also for cloud computing in the future, because the number of available services that you can choose is big.

Raman, T. V. Cloud computing and equal access for all. Proceedings of the 2008 international cross disciplinary conference on Web accessibility (W4A), Apr. 2008, 1–4. DOI= http://doi.acm.org/10.1145/1368044.1368046. 2. Brian Hayes. Cloud computing. Communications of the ACM, July 2008, vol. 51, no. 7, 9–11. 3. Vouk, Mladen A. Cloud computing - Issues, research and implementations. Proceedings of the International Conference on Information Technology Interfaces (ITI’08), June 2008, 31–40. 4. Philip K. McKinley, Farshad A. Samimi, Jonathan K. Shapiro, and Chiping Tang. Service Clouds: A Distributed Infrastructure for Constructing Autonomic Communication Services. Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC’06), Sept.2006, 341–348. 5. Robert L. Grossman, Yunhong Gu, Michael Sabala, and Wanzhi Zhang. Compute and storage clouds using wide area high performance networks. Future Generation Computer Systems, vol. 25, no. 2, 179–183. 6. Milojicic Dejan. Cloud computing: Interview with Russ Daniels and Franco Travostino. IEEE Internet Computing, vol. 12, no. 5, 7–9, 1048.