On User Capacity Optimization Strategy of Server Computing Based Desktop Cloud

CHEN Ningjiang[1,*], XU Bin[1,2]
[1] College of Computer, Electronic, and Information, Guangxi University, Nanning 530004, China

HU Dandan[1], WAN Yimin[1]
[2] China United Network Communications Corporation Limited, Guangxi Branch, Nanning 530022, China

***Abstract*—Cloud computing is an emerging technology. On the basis of intensive resource management and sharing, it provides on-demand resources to reduce costs and optimizes the IT services. Though the desktop cloud platform can allocate resource flexibly, its efficiency is restricted by the limited system resource. Firstly, the features of SBC mode in the desktop cloud platform are analyzed. Secondly, the SBC mode-based user Capacity evaluation strategy is introduced in order to formulate the calculation principle of the maximum number of users. Based on the maximum number of users, a user capacity decision-making strategy-User Capacity Strategy algorithm is introduced in order to solve the optimal user capacity problem in the desktop cloud platform. It has been applied to the telecom applications, and the experiment results show that it can efficiently determine the optimal user Capacity and improve the performance effectively.**

Keywords-desktop cloud; user capacity; performance optimization

I. INTRODUCTION

The mobile application has many characteristics, such as, it has a centralized system, it is fully functional, it has to deal with massive data, and it has a large-scale number of users, etc. How to improve user satisfaction, the QoS of business and reduce the maintenance workload are big challenges for telecom operators. The emerging desktop cloud technology[1-4] provides a practical solution, because it can provide the function of desktop unify and resource sharing. It can separate the user physical terminal and logic desk effectively, which makes it easier to centrally deploy the work-related logic desktop environment, so as to achieve the goal of unified management and control, environment protection, TCO reduction, improvement of the usage and maintenance experience. SBC and VDI are two major modes of desktop cloud platform, which different in the occupancy way of operating system resources. The feature of SBC (Server-Based Computing) mode is that many users share the same operating system, it is suitable for the application scenario that business is unified and user scale is large. While the feature of VDI (Virtual Desktop Infrastructure) mode [5] is that each user has a standalone operating system, it is suitable for application scenario that users have different customized requirements, such as OA and business hall of operators. At present, SBC mode-based desktop cloud platform is a relatively mature infrastructure. It is used in many fields, such as telecom, banks, hospitals and retail industries, which require large-scale client deployment, have relatively simple application, and have high requirement of safety and centralized management. In order to improve the QoS of SBC mode, there are still some key problems to be solved. For example, if user applications compete with operating system services for resources, the user satisfaction may decrease. Taking application requirements in telecom field into consideration, this paper studies the user performance optimization strategy on the SBC mode-based desktop cloud platform, then User Capacity Strategy algorithm is proposed. It has been testified in actual business applications, and the results show that it could decide the optimal user capacity, so that the performance of backend server can be improved.

Project supported by the Natural Science Foundation of China(No. 61063012), Guangxi Natural Science Fund (No. 2012GXNSFAA053222), Guangxi university talents support project(No. [2011]40), Guangxi provincial scientific research projects([2010]10).
Corresponding author: chnj@gxu.edu.cn (Chen Ningjiang).

II. RELATED WORK

Recently, there are plenty of researches on the desktop cloud technology research. Paper [6] presents a four layer logic-based overall infrastructure of private desktop cloud, and designs a security infrastructure of private desktop cloud from six aspects, such as device security, user authentication and authorization, and border security. Paper [7] analyzes the desktop cloud architecture in terms of network, and proposes a network building solution. Paper [8] studies the reliability optimization strategy of desktop cloud. It analyzes the characteristics of virtual resources, establishes a resource reliability evaluation model, and proposes an optimal reliability-oriented resource scheduling algorithm based on this model. This algorithm can provide an effective support for QoS improvement of virtual desktop services. Paper [9] proposes a performance optimization solution of desktop cloud platform. As there's large redundant data in the virtual desktop storage, this solution uses data de-duplication to reduce storage space of virtual desktop infrastructure, and uses the local disk caching and solid state disk in the shared storage pool to optimize the startup performance of virtual machines. Paper [10] introduces a business important level to optimize the resource scheduling algorithm of hypervisor, and improve the virtual resource scheduling strategy, so as to enhance the user experience. Paper [11] proposes an interactive classification method of cloud workload to optimize the cloud resource. All these related researches show that there're many researches on the overall performance, safety, reliability, and network construction of desktop cloud architecture, but the researches on user capacity of desktop

cloud platforms are rare. Taking into the actual business requirements into consideration, we study the user capacity assessment strategies of the desktop cloud platforms, and introduce a user capacity decision-making algorithm. Comparison with the above work, our contribution is that system capacity is taking into consideration to optimize the desktop cloud platforms, a user satisfaction evaluation mechanism is introduced to make the user capacity decision-making algorithm more suitable for the actual business usage scenarios.

III. The Evaluation and Optimization Strategy of User Capacity

A. *The calculation principle of sustainable user capacity*

In the SBC mode, N high-performance servers are deployed at the backend, each of which has its own operating system. Each server has its own number of users which is decided by the server performance. The number of terminal users is a key factor that influences the performance of the desktop cloud platform. If there are too many users access the system simultaneously, server performance will decrease dramatically as well as the user satisfaction. The resources of one server is limited, so this paper discusses the main factors that affect the performance of the backend server, including the consumptions of basic resources and operational resources, system redundancy, etc. As the resources of desktop cloud platform are limited, the tradeoff between the system resources and application requirements must be made and the balance point must be found to maximize the resource utilization.

The maximum number of sustainable users can be described as the following function:

F: (U, I, R, T)→MU, where MU is the maximum number of sustainable users in normal operation environment; U is a unit of resource which represents the system resources that a user requires in the logging in and running operations; I is the initial resources which represents the basic resources to maintain system normal operation; R is the redundancy rate, it refers to the redundancy(%) that can maintain system operation in abnormal circumstances; T is the total amount of backend server resources. If the system is running in a circumstance which exceeds the amount of redundancy, it shall be treated as an abnormal circumstance. The relationship between the parameters is shown in Figure.1.

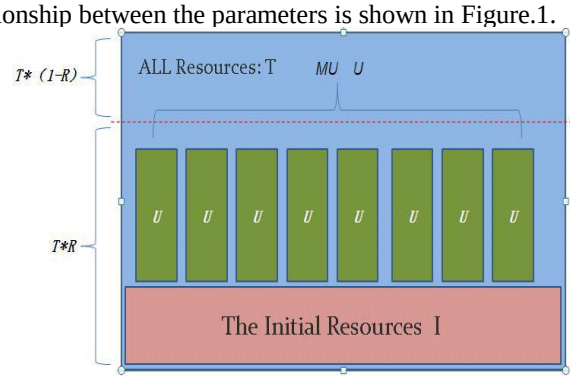

Figure.1: Relationship between the parameters

This paper assumes that if many users log in, the required resources increase linearly with increase of the number of users. To ensure the system can function well in abnormal circumstances, system resources should have a certain amount of redundancy. In the meanwhile, a certain initial resource is needed even when there is no user, so the maximum available resource can be described as R*T-I, divided by U, the maximum number of sustainable users can be calculated as:

$$MU = (R*T - I) / U \quad (1)$$

As can be seen from formula (1), MU varies inversely with U and I, so there're two ways to increase MU: to reduce U by lowering the resources for running applications; or to reduce I by lower the resources for keeping the system function well. Since U varies greatly in different business types, it's hard to find a general solution. Therefore this paper doesn't discuss the optimization strategies of U, it focuses on optimizing I.

B. *The user capacity decision-making algorithm*

In practical applications, the maximum user capacity may not be equal as the number of optimized sustainable user capacity, because user awareness has to be considered. Assuming the maximum number of sustainable user is known, we design a user capacity decision-making algorithm. It continuously decreases the number of users until the users are satisfied with the performance, and this number of users is the optimized user capacity.

US is the user satisfied value, it can only be one of [0, 1, 2], which means "unsatisfactory", "available" and "satisfied" respectively. RT is the user satisfied value of response time; T_v is the user satisfied value of response time threshold; R is the redundancy; Num is the number of test users. Let the user set be A={a_1,a_2, ... ,a_N}, satisfaction set be S= {us_1,us_2, ... ,us_N}, TestTime is the time interval of testing(minutes).The user capacity decision-making algorithm is as follows:

(1) According to basic hardware configuration of backend server, and the calculation principle of sustainable user capacity in chapter A, the maximum sustainable user capacity for a single backend server, Num, can be calculated. Let Num users log in the same backend server simultaneously, and initialize their satisfaction value. (2) Keep them running the application for TestTime, and collect each user's response time RT. If every RT is no bigger than the response time threshold T_y, the satisfaction value is 2, it mean "satisfied". (3) Even if one RT is bigger than T_y, reduce one user, that is, Num=Num-1, until every user's RT is no bigger than T_y. (4) Finally, to check whether the percentage of performance is lower than system redundancy or not. If it is, then output Num, which is the optimal user capacity. The pseudo code of the user capacity decision-making algorithm is described as follows:

Input: The maximum number of users *Num*,
initial testing time *InitialTime*;

```
Output: The Optimal number of users OptimalUser
Step 1:
            OptimalUser = 0;            // initialize optimalUser
      TestTime = InitialTime;
      for(i=0;i<Num;i++) us[i] = 0;
Step 2:
      While(CheckSatisfy(us,Num)!=0)
       {
          CreatThread(Num, TestTime);
                                    i = Num;
                        for(j=0; j<Num; j++) us[ j ] = 0;
Step 3:
                  while (i){
                        RT[i-1] = GetRT(i-1);
                              if (RT[i-1] <= Tv )
                                    us[ i ] = 2;
                        else  goto Step 4;
                                    i = i -1;
                  }
            if ( i = = 0 && the percentage of background server
                              performance < R )
                                    goto Step 5;
Step 4:
                              Num = Num -1 ;
      }
Step 5:  OptimalUser = Num;
```

In the above algorithm, we introduce user satisfaction as an indicator and use it as a metric to determine the optimal number of sustainable users. What's more, the test time interval is introduced and it can be adjusted to check the satisfaction value, and it has a positive effect to calculate the optimal number of users.

IV. EXPERIMENTS AND ANALYSIS

We conducted the experiments on desktop cloud platform of the customer service call center in Guangxi Unicom. The customer service call center adopts a concentrated way, and it has 300 seats. Its desktop cloud platform is in SBC mode, and the front layer uses the thin client TC of Huawei, this desktop cloud platform contains 15 PC servers (Huawei ATAE blade servers with CPU 4*2C*2.33GHz and Windows 2003). The desktop cloud platform has been used in the customer service call center to do call business for 1 years; the number of user connection for each backend server TC is 15. When the number of user connection is bigger than 15, the performance will sharply decline and the customer service staff reflects that the client browser responses slowly, even crash down, seriously affecting the quality of service.

In this context, we conducted the experiments to test the performance of our algorithm, which contains two steps. Step 1, calculating the maximum number of sustainable user for each server. According to formula (1), in order to optimize the initial resource I, we have to adopt the following operations: optimizing the AD Group Policy of Windows 2003 to adjust the maximum number of connections; each user has and only has one session; the time interval of session is limited(15 minutes) in order to limit the number of users.

After the above optimized operations, we take average unit resource U, and average initial resource I as calculation value. Taking CPU utilization as an example, we get a set of data after a period of testing, T=100, U=2.6, I=1.3, and redundancy=70%, which is equal to the general redundancy of telecom application, according to formula (1), the maximum number of users MU= (100 * 70% - 1.3)÷2.6 ≈26.

The target of the second step is to get the optimal user capacity. After step 1, we can get the maximum number of sustainable user, which is equal to 26. According to the user capacity decision-making algorithm, we began to decrease the number of application threads, and keep this connection number for 60 minutes, and observe the variants (CPU, Memory) of performance. Figure 2 shows that when the number of users descends to 22, the average rate of CPU utilization goes below 70%, which suits the requirements of the telecommunication applications. Figure 3 also shows that at the same time, the available memory is adequate which is bigger than 16GB.

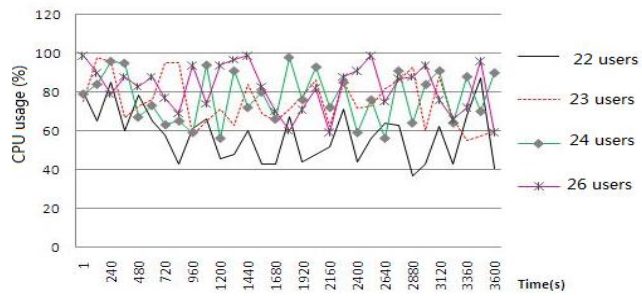

Figure2. The variants of CPU Utilization

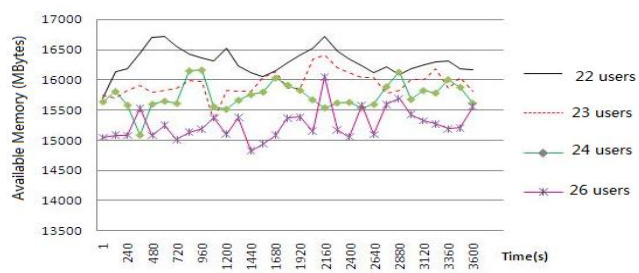

Figure3. The variants of memory

The relationship between the number of clients and the CPU utilization is shown in Table 1.

Table 1 The comparison of performance and user satisfaction

Agents	Average CPU utilization	Percentage of user time	Percentage of privilege time	Percentage of interrupting time	Satisfaction Evaluation
22	57.79%	30.19%	27.64%	12.52%	Normal
23	78.56%	31.28%	47.38%	30.15%	Normal
24	74.48%	29.90%	44.57%	27.17%	Normal
26	85.66%	28.90%	56.68%	38.14%	Slow

When the number of users is bigger than 22, CPU utilization will increase to over 70%, the maximum will be over 90%, so 22 is balance point of performance. When TC

is equal to 22, the average CPU utilization rate is 55.79%, the privileged time of CPU is less than 30%, and less than the user time of CPU, the average interrupt time of CPU is 12.52%, and the state of CPU consumption is normal. So we can conclude that the optimal user capacity is 22. In addition, it should be pointed out that, the reason CPU utilization rate of 23 agents is bigger than that of 24 is that in the 24 agents test, the average queue length of processor is longer, process scheduling can lead to an increase in CPU privilege time. The satisfaction survey to actual service staffs (see the last column in table 1), also proves that the experimental data is consistent with the actual effect.

In order to further verify this conclusion, we continue to observe the reading and writing performance of disk with 22 agents. The experimental data is shown in Figure 4.

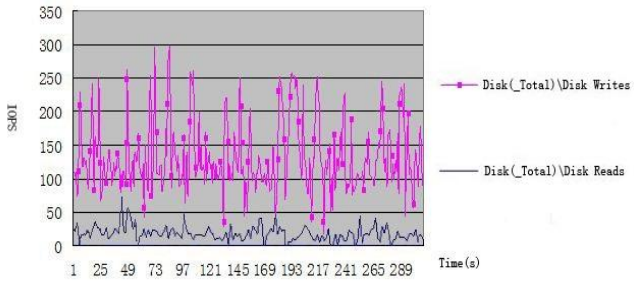

Figure.4: the reading and writing performance of disk with 22 users

It can be seen that, in the writing operation, 125 IO operations per second on average, and 300 IO operations as the most; in the reading operation, 23 IO operations per second. And the disk for this server is SATA disk 15,000 rpm, with a capacity of 640G. The maximum IOPS in theory is 300 IO operations, and the experimental data shows that the reading and writing performance are both lower than 70%

The results show in the SBC-mode, the optimal number of sustainable users of each server is 22 in the Guangxi Unicom Custom Service Call Center. With the guidance of the user evaluation strategy, we optimized backend servers' initial resources in the customer service call center, and it improves the performance profoundly. The maximum number of sustainable user is bigger than the actual number of users before optimization. In addition, by using the user capacity decision-making algorithm, we conducted the optimal number of users testing. Combining the CPU performance test data and the agent satisfaction, we concluded the optimal number of users is 22, which increases the actual number of sustainable users a lot. In this context, all agents are using the same software to do business operation, so it can be regarded as a single virtual resource pool. But in practical use, clients often have different needs for resources, these users can be classified into different virtual resource pool according to their needs. Thus, the optimal number of such users in the virtual resource pool can be derived, according to user capacity decision-making algorithm and the characteristic of resources that users are using.

V. CONCLUSION

In this paper, we analyze the problem of optimal user capacity in the SBC-mode desktop cloud platform, and introduce a user satisfaction-based optimization strategy. The main idea is that performance optimization of backend server is considered firstly, and then take the resource consumption of one user as a unit, to calculate the maximum number of sustainable users in a backend server. On the basis of the maximum number, the numbers of users who login and execute at the same time are decreased, and considering user satisfaction values, we build a user capacity optimization strategy. This policy is applied to a customer service call center of a telecom operator. Experiments show that by using the strategy, the user capacity can be effectively assessed. The SBC mode-based desktop cloud platform can be widely used in the field of telecommunication. In the future, we will research further in the following two aspects: this strategy only focus on a certain scene and the similar kind of business, it's unable to meet the requirement of different scenes and different business operations in the same desktop cloud platform. So it needs to be improved to take into consideration the situation that different types of business have different resource consumptions. Then we continue to explore the relationship between the user capacity and scheduling of virtual desktop on backend servers.

REFERENCES

[1] IBM, Virtual Infrastructure Access Service Product,http://www-03.ibm.com/industries/education/us/detail/solution/H353182K73375S31.

[2] Citrix Corporation, .Citrix Application Delivery Infrastructure.

[3] Microsoft Corporation, .Windows Terminal Services.

[4] VMWare EMC, .http://www.vmware.com.

[5] Ding Ding, He Jin. Strategy of Resource Scheduling and Management in Integrated Desktop Cloud Architecture, Computer Systems & Applications, 2012.21(4):31-35.

[6] Wang Xiaolin, Zhou Kai, Zhang Binbin, Yang Liang, Luo Yingwei, Li Xiaoming. Interactive Performance Optimization for Desktop Virtualization Environment. Journal of Frontiers of Computer Science and Technology, 2012,6(4):289-300.

[7] Zhao Rui, MaoLiang. The Application of Desktop Virtualization Technology to Business System. Designing Techniques of Posts and Telecommunications Magazine Office, 2010(8): 21-24.

[8] Shi Shu, Xiang Shen, Yongxin Zhu, Tian Huang, Shunqing Yan and Shiming Li, Prototyping Efficient Desktop-as-a-Service for FPGA Based Cloud Computing Architecture, IEEE Fifth International Conference on Cloud Computing, pp. 24-29 June 2012

[9] Fu Yinjin, Xiao Nong, Liu Fang, and BaoXianQiang. Deduplication Based Storage Optimization Technique for Virtual Desktop. Journal of Computer Research and Development, 2012.49(s1): 124-129.

[10] Kirk Beaty, Andrzej Kochut, Hidayatullsh Shaikh. Desktop to Cloud Transformation Planning, IEEE Workshop on System Management Techniques, Processes, and Services (SMPTS 2009) , pp. 23-29, 2009.

[11] Kochut, Kirk A. Beaty, Hidayatullah Shaikh, Dennis G. Shea. Desktop workload study with implications for desktop cloud resource optimization. The 24th IEEE International Symposium on Parallel and Distributed Processing, pp.1-8, 2010.