

# Live Virtual Machine Migration with Bandwidth Dynamic Assignment

Yiqiu Fang<sup>1, a</sup>, Fangzhengqing Liao<sup>1, \*b</sup> and Junwei Ge<sup>1, c</sup>

<sup>1</sup>School of Computer, Chongqing University of Posts and Telecommunication, Chongqing 400065, China.

<sup>a</sup>994607515@qq.com, <sup>\*</sup>Corresponding author Email: <sup>b</sup>405512135@qq.com, <sup>c</sup>gejw@cqupt.edu.cn

**Keywords:** virtual machine, dynamic migration, bandwidth dynamic allocation, dirty page rate, cloud computing.

**Abstract.** In this paper, we present an algorithm that based on the probability prediction of the dirty page's Pre-Copy algorithm, bring in a prediction algorithm of the bandwidth required for the transmission of the dirty pages. Mainly in the case of mutations of dirty page, Analysis of the probability of occurrence of the dirty page in the process of migration, Obtain the adjusted bandwidth coefficient, dynamically allocates the bandwidth required for the current transmission, and reserved for later transmission bandwidth, so that the transmission process will not have the use of bandwidth imbalance or the bandwidth cannot make full use of, so that each time from the current virtual machine to reach the directory virtual machine transmission speed can be consistent as much as possible. Experiments show that the algorithm can reasonably use limited bandwidth resources, reduce iteration time and downtime, and improve the dynamic migration performance of virtual machines in the case of mutations.

## 1. Introduction

Cloud computing environment to provide users with safe and reliable computing and storage services, known as the next generation computing model. It uses virtualization technology to manage shared resources, by creating multiple virtual machines to load different services to isolate the execution environment to meet the needs of different users of the application; and can be adaptive allocation and scheduling of various physical resources, so efficient and efficient management of cloud computing platform resources and services [1]. Among them, the virtual machine dynamic migration technology [2] is without affecting the service under the premise of the service from the source host to the destination host to continue to run to achieve dynamic load balancing, online system maintenance purposes, is to achieve the importance of green resource management Means [3]. However, there is a small amount of downtime in the virtual machine migration process that can cause a brief disruption to the service, partially affecting the quality of service and the user experience; On the other hand, virtual machine migration also occupies additional CPU, network bandwidth, memory and other physical resources, but also to some extent affect the cloud computing environment in other services. Therefore, to improve the virtual machine migration performance, is to maintain efficient green resource management core, but also virtualization research hotspot.

The key to dynamic migration of virtual machines is to efficiently copy and transfer virtual machine memory from the source host to the destination host. The existing mainstream virtualization platforms such as Xen and VMware generally use the pre-copy algorithm for virtual machine dynamic migration, and there are many successful Applications such as Amazon, IBM and other cloud computing service platform [4 -6]. The pre-copy algorithm is divided into three stages: initialization, iterative copy and down-copy. The multi-round iteration will send the virtual machine memory state from the source host to the destination host. After the memory is synchronized, the destination host will continue to run the virtual machine [7]. This approach can shorten the virtual machine migration time and improve migration performance. However, in practical applications, virtual machines run different services using physical resources such as network bandwidth and dirty memory generated by business operations and other factors will greatly affect the migration performance. For example, when a virtual machine running I / O-intensive business, the memory

modification rate is too large, in the copy of the need to copy a large number of dirty pages, increase downtime, affecting the user access to virtual machine applications. On the other hand, when the available physical network bandwidth is less than the virtual machine dirty page transmission, the pre-copy algorithm in the iterative transmission process will be forced to stop, resulting in memory synchronization convergence problem [8].

For the pre-copy algorithm in the case of high load performance is not ideal, so this design algorithm for the case of higher load memory migration iterative copy of the stage, the required bandwidth for each copy of the forecast. Balance the size of the use of the bandwidth, make full use of limited bandwidth resources, to reduce the iterative copy time, while reducing the time to reduce the purpose of downtime.

## **2. Related work**

Virtual machine existing dynamic migration technology can be divided into pre-copy and post-copy two methods [9], based on the post-copy algorithm dynamic migration technology is to synchronize the memory synchronization to the VM on the destination host after the resumption of operation. When the virtual machine is moved in addition to memory other than the state of the device synchronization between the two hosts, and in the destination host to start the virtual machine, and then use the on-demand way to achieve both ends of the virtual machine memory synchronization. Although the mechanism can ensure that each memory page is copied at least once, fundamentally avoid redundant data on the network bandwidth occupied, but when the virtual machine running on the destination host, and access to the page is not synchronized, the mechanism will make the virtual machine frequently interrupted to synchronize memory pages, resulting in virtual machine performance degradation and increased memory application execution latency. At the same time, it cannot meet the dynamic migration transparency requirements.

As a mainstream platform in the field of virtualization, open source virtual machine management system Xen uses the default network bandwidth transmission allocation and incremental network bandwidth allocation in the process of dynamic migration using the pre-copy algorithm. Among them, the default bandwidth allocation mode is the current round of iterations need to send dirty pages into the internal definition of the cache, when the need to send the memory page is stored or the cache is filled, through the native device driver or device model using security hardware Interface will cache the memory page all the transmission to the physical network card, by the physical network card to select the sending time. In the actual application process, by the actual network bandwidth and operational characteristics of the business, when the physical bandwidth usage is large, it may lead to a large number of cache congestion in the physical network at the queue, the convergence process of the convergence problem, and also there will be sending memory and business operations to seize the bandwidth of the phenomenon, coupled with the pre-copy algorithm by its own characteristics of the constraints, affecting the performance of mobile services and operational services, and even lead to migration failure. The incremental bandwidth allocation mode is to initialize the first-round transmission bandwidth to a constant. The subsequent rounds of iterations send bandwidth is a constant amount based on the dirty page rate generated in the last iteration, but the maximum use bandwidth cannot exceed the system-defined constant level. This model is limited by the internal implementation mechanism, and the network (including the 100 megahertz) environment is invalid. In the gigabit network (including Gigabit) environment, when the physical bandwidth usage is small, it will cause the physical bandwidth resources to be used inadequately, limiting the virtual machine to enhance the performance of the transfer, according to the single round dirty page rate distribution bandwidth, for dealing with business bursts and data jitter is unreasonable.

### 3. Algorithm Design

#### 3.1 Algorithm Thinking.

The dynamic migration process of the virtual machine based on the pre-copy mechanism includes three aspects: the migration initialization phase, the iterative copy phase and the synchronization phase of the virtual machine. The most obvious of the virtual machine migration performance is the iterative copy stage. The iterative copy is a method of using the iterative copy. The virtual machine memory is migrated from the source host to the destination host. Set the initialization time for the  $T_{init}$  initialization phase to complete the iterative copy phase. The shorter the iteration copy time, the shorter the time for business applications and migration of competing resources, and the less the impact of service quality. If the number of iterations is  $n$ , the first round of migration iterations need to send all the memory pages of the migrated virtual machine from the source host to the destination host; From the second round to the  $n-1$  round iteration, it is necessary to judge and calculate the number of virtual machine memory pages to be sent according to the number of memory pages generated in the previous round. Therefore, when the rate of the virtual machine memory is changed, the amount of memory that is sent is related to the time of the iteration, the longer the iteration, the more the amount of spoofing, and the size of the iterative send bandwidth directly affects the time of each iteration. Assuming the size of the virtual machine to be migrated to  $M_{memory}$ , the number of memory pages sent per round is  $M_{iter_i}$ , the bandwidth used to transmit the memory space is  $B_{iter_i}$ , and the time for each iteration is  $T_{iter_i}$ . According to the pre-copy algorithm, Round iterations need to send all the memory of the virtual machine being migrated, knowing that the last round only needs to send the number of memory pages. Machine copy time refers to the  $n$ -round copy of the virtual machine in the process of pause, the  $n$ th round of the stolen pages copied to the destination host, while the memory and device synchronization. Set the downtime copy time for  $T_{down}$ , memory and equipment synchronization time for  $T_{dev}$ . Then we can get the following three formulas:

$$T_{iter_i} \begin{cases} \frac{M_{memory}}{B_{iter_i}}, & i = 1 \\ \frac{M_{iter_i}}{B_{iter_i}}, & 1 < i < n \end{cases} \quad (1)$$

$$T_{down} + T_{dev} = \frac{M_{iter_n}}{B_{iter_n}} + T_{dev} \quad (2)$$

$$T_{total} = T_{init} + \sum_{i=1}^{n-1} T_{iter_i} + T_{down} + T_{dev} \quad (3)$$

It can be seen that when the memory needs to be transmitted, the size of the network bandwidth directly affects the time of each iteration migration; the reasonable allocation of network bandwidth during the downtime will also affect the virtual machine downtime. So, we need to allocate bandwidth to each iteration, and at the same time, give the iterations a reasonable reserve bandwidth to ensure that the iteration is stable and safe to run.

#### 3.2 Data Jitter Processing.

Through the previous analysis, in the case of the need to migrate the total memory and the limited bandwidth resources, if the physical network bandwidth is allocated unreasonable, it will increase the subsequent rounds of the iteration time, the iteration time is bound to cause the next round The number of active pages sent increases, resulting in a vicious cycle, but also may increase the number of active pages in the iteration process. The increase in the number of active entries in the last iteration will cause the downtime to be too long and the efficiency of the dynamic migration of the

virtual machine is greatly reduced. Therefore, it is necessary to allocate bandwidth for each iteration, and to reserve bandwidth for the next iteration.

In order to allocate bandwidth, you need to know the next round of iterative copy of the business bandwidth usage, because the dirty page rate is an objective reflection of business characteristics, so the dirty page rate as an important basis, taking into account when the dirty page rate suddenly Large or smaller will cause the business to reserve bandwidth fluctuations, therefore, the introduction of the coefficient of variation a, b, the current dirty page rate and the previous average dirty page rate for comparison, a is equal to the current dirty page rate divided by the previous average dirty page with the current dirty page rate, b is equal to the previous average dirty page divided by the previous average dirty page with the current dirty page rate. Not only consider the business used to run the current bandwidth and virtual machine migration process of historical data rate reflects the business characteristics, while considering the amount of data jitter in the process of business operation, it is also possible to obtain the real-time service characteristics and the bandwidth usage of the current iterative service, therefore, we use equation (4) to calculate the amount of traffic used in the kth iteration process.

$$B_{bw} = a \times U_{dirty_k} + b \times avg\left(\sum_{i=1}^{k-1} U_{dirty_i}\right) \quad (4)$$

$U_{dirty}$  for the historical data rate, when the bandwidth coefficient a greater than b, said the virtual machine on the increase in business, compared to the previous k-1 times the bandwidth usage, the bandwidth needs of the larger demand, focusing on the use of the current value as a reference; When the bandwidth coefficient a is equal to b, it indicates that the traffic volume of the virtual machine is stable and the network bandwidth usage is averaged; When the bandwidth coefficient a is less than b, it indicates that the traffic volume of the virtual machine is reduced, during the first i iteration, the network bandwidth usage should be compared with the current round, and historical data should be used as a reference.

The bandwidth adjustment coefficients a and b not only affect the reserved bandwidth values, but also affect the feasibility of the optimization algorithm as well as the migration performance and business service quality. It can be seen from the experience that the coefficient can be determined by the ratio of the current physical bandwidth usage and the total amount of physical bandwidth before each round of iteration. Dynamic adjustment of the value of a, b can truly reflect the current use of physical network resources, timely and effective according to the virtual machine to be moved the page rate predicted the current round of the process of business needs of the physical bandwidth.

### 3.3 Algorithm.

In this section, we present a dynamic allocation algorithm based on pre-copy bandwidth. In the initialization phase, one is to obtain the current physical machine connected to the network information, calculate the physical idle network bandwidth; But according to the number of iterations to determine the bandwidth allocation rules. When the first round of the copy, you need to copy all the memory page, so the root of the first stage of the network state to send the amount of bandwidth can be sent, when in the downtime copy stage, the virtual machine for the downtime, no response to the business, so all the free bandwidth for the transfer of memory pages to ensure that the shortest period of time the memory page transfer is completed; when the number of other iterated copy of the wheel, enter the second stage.

The iterative run phase calculates the reserved bandwidth and corrects the reserved bandwidth in two steps. Step 1: Set the bandwidth calculation according to the physical network information obtained from the initialization phase a, b, and use the data page to calculate the amount of business reserved bandwidth, in order to avoid data jitter on the calculation of the effect of bandwidth calculation, using the formula (4) to calculate the second step to reserve bandwidth correction, the first step, although the optimization of data jitter on the calculation of the impact of bandwidth calculation, but the business changes caused by changes in the price of the page, may result in the

calculated business bandwidth is greater than the current idle bandwidth, if this happens, you need to correct the amount of bandwidth reserved for the service. The specific method is to introduce the change coefficient  $c$ ,  $c$  is the ratio of the current dirty page rate to the previous time dirty page rate, and use the formula (5) to allocate the existing free bandwidth. Step 3, calculate the bandwidth of the iterative transmission. According to the results of the second step, calculate the current round of iterative transmission using the bandwidth, and then enter the page to iterate to send. A detailed algorithm is shown in Fig. 1.

$$B\_bw = F\_bw \times (B\_bw / (B\_bw + c \times B\_bw)) \quad (5)$$

```

Input: Iteration rounds n Output: The current allocation bandwidth E_bw
1 if (n==1) // First round iteration
2 Then return E_bw //Initialize iteration to send bandwidth E_bw
3 else //Total bandwidth of the physical network T_bw, Use bandwidth U_bw
4 //Calculate physical idle bandwidth F_bw=T_bw-U_bw
5 if(n==last_iter) // The last round of iterations
6 E_bw=F_bw
7 return E_bw
8 End if
9 Calculate the average of the n-1 rounds before the page D_avg
10 Get the dirty times for this round U_dirty[n]
11 Get the dirty times for this round a = U_dirty[n]/D_avg+ U_dirty[n] b = D_avg/D_avg+ U_dirty[n]
14 B_bw = a× U_dirty[n]+b×D_avg // Calculate the current bandwidth reserved for the service
15 if(B_bw> F_bw) // Fixed service reserved bandwidth
16 B_bw = F_bw × (B_bw/(B_bw + c × B_bw))
17 End if
18 Return E_bw = F_bw-B_bw

```

Fig. 1. Bandwidth Dynamic Allocation (BDA)

#### 4. Experiment and Result Analysis

In order to verify the performance of the bandwidth dynamic allocation algorithm (abbreviated as BDA) based on the Pre-copy algorithm based on probabilistic prediction, it is compared with the traditional Pre-Copy (PCA) and Pre-copy algorithm (abbreviated as PPC) for comparative analysis.

##### 4.1 Lab Environment.

The lab uses three machines configured as Inter (R) Core (TM) i5-3210M 2.5 GHz, 8GB memory, 500GB hard drive as a hardware platform, and colleagues use Ubuntu1.0 as the operating system, Xen-4.1.0 as a virtualization platform. The migration does not use a dedicated network, shares the same network with the physical host, and the physical host is connected to the same local area network through the Fast Ethernet switch.

##### 4.2 Pressure Test.

First of all, the improved BDA mechanism for stress testing, test the virtual machine memory size of 128 ~ 2048MB, mainly tested the downtime and the total migration time.

###### 4.2.1 Downtime.

As shown in Figure 1, when the memory is 128MB and 256MB, the three kinds of policy mechanism downtime is almost the same, starting from 518MB, this paper BDA optimization mechanism can be better according to the traffic adjustment of the iterative bandwidth, the iterative efficiency, In the shutdown copy bandwidth for the maximum free bandwidth, effectively reducing the downtime. Different memory downtime comparison shown in Figure 1.

###### 4.2.2 Total Migration Time.

As shown in Figure 2, at 128 ~ 256MB, the total migration time of the three algorithms is not much difference. Starting from 512MB, BDA reflects the advantages of bandwidth adaptive modules in migration, and significantly reduces the total time of migration compared to PPC and PCA.

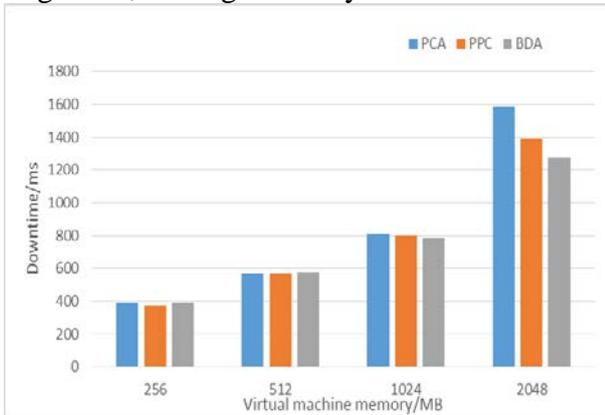


Fig. 2. Comparison of downtime in different memory

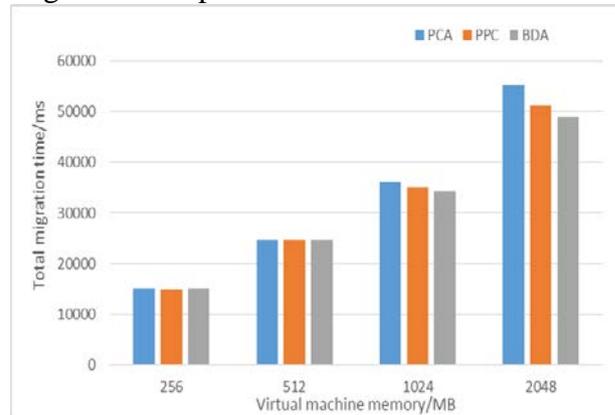


Fig. 3. Comparison of total migration time in different memory

### 4.3 Application Testing

The experiment uses three kinds of workloads to test, the first is the idle system (IS), in addition to the necessary system procedures, no additional applications, used to simulate the low load case iterative migration. The second is the higher load system (KCS), that is, in the system compiler Linux kernel, to simulate the higher dirty page rate of the application environment. The third is intermittent kernel compilation (IKCS), through the intermittent compiler kernel to simulate the dirty page rate of sudden increase or decrease the experimental virtual machine memory size is 1024MB.

#### 4.3.1 Downtime.

The BDA strategy proposed in this paper has been reduced compared with PCA and PPC. In the idle application environment, BDA has not significantly reduced the downtime, under this condition, the virtual machine only runs the system application without any other application, has little effect on the downtime. In the case of more intensive memory, because the transmission bandwidth will automatically adjust the transmission bandwidth according to business conditions, and in the shutdown copy stage, use all the free bandwidth to copy, shorten the downtime. As shown in Figure 4, the BDA was 9.85% lower than the PPC at KCS load; the BDA was reduced by 1.66% compared to the PPC at IKCS load. In the case of more intensive memory, can better shorten the downtime.

#### 4.3.2 Total Migration Time.

As shown in Figure 5, PAA has significantly reduced the total time of migration compared to PCA and PPC. In the case of idle, PAA does not significantly reduce the migration time, but there is a small increase in time. However, in the case of high dirty page rate and dirty page rate change, the optimized algorithm obviously plays the role of shortening the total time of migration. In the KCS load, PAA compared to P PC reduced by 9.51%; in IKCS, PAA compared to P PC reduced by 1.47%.

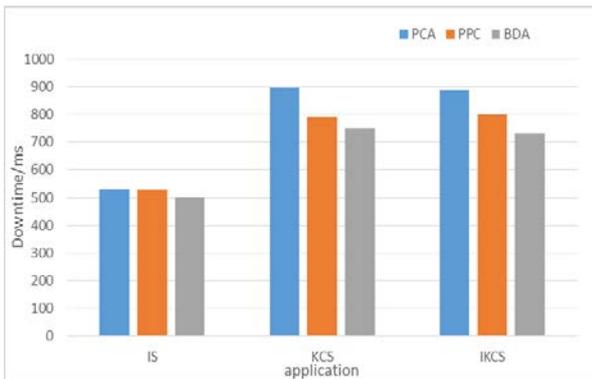


Fig. 4. Comparison of downtime in different application

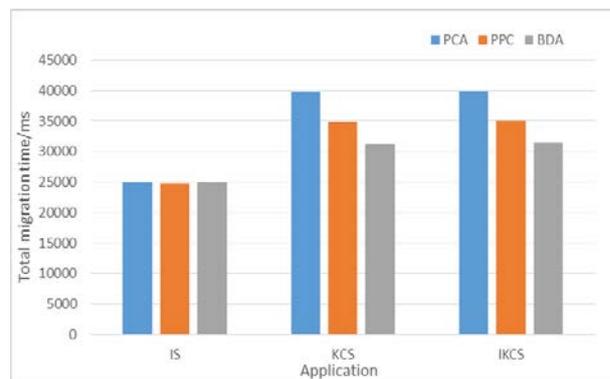


Fig. 5. Comparison of total migration time in different application

## 5. Summary

In this paper, a Pre-Copy method based on probability prediction is proposed, and the experiment is carried out on the Xen platform. The results show that the total migration time and the downtime and the number of iterations are reduced. Future work will optimize the migration bandwidth allocation algorithm for complex network environments and how to improve the reliability of probabilistic prediction.

## References

- [1]. Nathuji R, Schwan K. Virtual power: Coordinated power management in virtualized enterprise systems//Proceedings of ACM Symposium on Operating Systems Principles (SOSP'07). Stevenson, USA, 2007: 265-278.
- [2]. Clark C, Fraser K, Hand S, et al. Live migration of virtual machines//Proceedings of the 2nd Symposium on Networked Systems Design and Implementation NSDI'05. Boston, USA, 2005: 273-286.
- [3]. Wood T P, Shenoy P A, Venkataramani A, Yousif M. Black-box and gray-box strategies for virtual machine migration//Proceedings of the 4th USENIX Symposium on Networked Systems Design and Implementation (NSDI '07). Cambridge, USA, 2007: 229-242.
- [4]. Barham P, Dragovic B, Fraser K, et al. Xen and the art of virtualization//Proceeding of the 19th ACM Symptom on Operating Systems Principles (SOSP'2003), 2003: 164-177.
- [5]. Nelson M, Lim B-H, Hutchins G. Fast transparent migration for virtual machines//Proceedings of the Annual Conference on USENIX Annual Technical Conference. Anaheim, CA: USENIX Association, 2005: 15-35.
- [6]. Nagarajan A B, Mueller F, Engelmann C, Scott S L. Proactive fault tolerance for HPC with xen virtualization//Proceedings of ACM Annual International Conference on Supercomputing(ICS '07). Seattle, USA, 2007: 23-32.
- [7]. Sohan Rice R, Moore A, Hopper A W A. Predicting the performance of virtual machine migration//Proceedings of the Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS). Florida, USA, 2010: 37-46.
- [8]. Chen Yang, Huai Jin-Peng, Hu Chun-Ming. Live migration of virtual machines based on hybrid memory copy approach. Chinese Journal of Computers, 2011, 34(12): 2278 -2291(in Chinese).
- [9]. Hines MR, Deshpande U, Gopalan K. Post-copy live migration of virtual machines. SIGOPS Operation System Review, 2009, 43(3): 14-26.