A Survey on Virtual Machine Migration Techniques in Cloud Computing Infrastructure

Parvin Ahmadi Doval Amiri1, Shayan Zamani Rad2, Faramarz Safi esfahani 3
1Azad University of Babol, Computer Engineering Department, p.ahmadi@khuisf.ac.ir
2Mazandaran University of Science and Technology Computer Engineering and IT Department Babol, Iran, sh.zamani@ustmb.ac.ir
3Azad University of Najaf Abad, Computer Engineering Department, fsafi@iaun.ac.ir

Abstract. Cloud computing has been considered as a new fashion of providing Information Technology services such as computational and storage resources to individual users and organizations. To achieve this purpose, it uses internet in order to give service to users. In fact, cloud is a new technology of distributed computing which is based on virtualization technology. With respect to benefits of virtualization, today, most of data centers use this technology for improvement of task efficiency and increase of their own clients. This technology transforms a physical server to several virtual machines (VMs), and the VMs shares a logical file format. With regard to this property, we can transfer the machines from one place to another place that is called virtual machine migration. High usage environments of VMs migration technique are data centers which their load scale and demands bulk are dynamically changing. Migration is used to satisfy these goals: load balancing, fault tolerance, power management, reducing response time, increasing quality of service, and repairing and maintaining servers. The rest of this paper is organized as follows: In section 2, migration and all kinds of its techniques will be described. Finally in section 3, we will represent important criteria for evaluation of all migration methods and then we will compare and evaluate the described methods based on these criteria.

2- Migration Methods
Migration process divides into two general categories; each migration method is a subcategory of these: 1) non-live migration 2) live migration. In non-live migration a virtual machine completely stops in source, and then its CPU states, memory pages, and disk data (if possible regarding to our policy) will transfer to destination machine. Service providing even during migration process and, pages and CPU states transfer, is
one of its advantages. All migration techniques in this paper are subcategory of live migration. **Error! Reference source not found.** shows migration techniques.

3- **Migration by Using Shared Disk**

In general, process of virtual machines migration is accomplished in a local network. Therefore, source and destination hosts have access to a shared disk. This shared disk which is SAN or NAS, is used for storing of virtual machines’ disks and another information that should be accessible by all hosts. In this situation, when migration process begins, the virtual machine disk is not moved, in contrast information such as CPU states and page memories are moved. As a result, use of shared disk causes the decrease of total migration time.

As regards, virtual machine migration has benefits like decrease of migration time, decrease of machine downtime, decrease of degradation time in service efficiency and decrease of replication time, today, the most of propose migration methods depend on shared disk and in most of them, only the virtual machine memory along with CPU states are translocated.

Hence, the difference between various methods of migration is the manner of memory transfer (transmission) and CPU states of virtual machine. In follow, we will study the methods that use shared disk.
Table 1: Categorizing the methods of Virtual Machines Migration

<table>
<thead>
<tr>
<th>Method</th>
<th>Disadvantages</th>
<th>Advantages</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Copy</td>
<td>• Duplicate transfer overhead</td>
<td>Balance between virtual machine down time and the total migration time</td>
<td>This technique is consist of three steps – Iterative copy, Stop and Copy, Resume. In iterative copy: hypervisor sends the VM page memories to destination host in background. In stop and copy: virtual machine is suspending and the remaining pages with CPU states are sent to destination. In resume: the VM begins to work from the last state before suspension.</td>
</tr>
<tr>
<td>Adaptive Memory Compression</td>
<td></td>
<td>Reducing the compression time and significantly reducing the number of changed pages</td>
<td>The System recognize the number of changed pages, in real time and automatically, set the threshold for the maximum and minimum likelihood</td>
</tr>
<tr>
<td>Improved pre-copy approach by adding a bitmap</td>
<td>The exact recognition of the most frequent changing pages</td>
<td>Compared with pre copy method save 35% of migration time and 34% of transferred data. The replication process is done five times faster than pre copy method</td>
<td>Adding a replication bitmap, the most frequent changing pages are placed on the page bitmap. And those pages are transferred in the last iteration, and therefore it is guaranteed these kind of pages transfer only once</td>
</tr>
<tr>
<td>Optimizing the live migration of virtual machine by CPU scheduling</td>
<td>Specifying proper “e” and the share percentage of each virtual machine from processor time</td>
<td>Reduce 88% down time of applications and the overhead is acceptable</td>
<td>The goal is using reducing the rate of dirty pages in virtual machine. the CPU usage is limited for each virtual machine and prevent timing with weighted algorithms which are too complicated, and consider only one “e” for virtual machine and the specified time for each machine to use the processor .“does not pass the “e”</td>
</tr>
<tr>
<td>CloudSpider</td>
<td>This method minimize the latency for WAN networks with low bandwidth</td>
<td>Minimize the migration latency</td>
<td>Suggests combining virtual machine replication with virtual machine scheduling, in order to live migration of large virtual machines in a wan with low bandwidth</td>
</tr>
<tr>
<td>Improved Live VM Migration using LRU and Splay Tree Algorithm</td>
<td>The main problem is finding defined FAMES, according to the last used time</td>
<td>comparing with preprocessing algorithm there is a 23.67 percent reduction in data during migration time and 11.45 percent reduction in migration time itself</td>
<td>In the preprocessing phase , there is a prediction about the writing set, this algorithm suggest to combine LRU with SPLAY tree, this system could reduce the number of transferred memory pages</td>
</tr>
<tr>
<td>Live Migration of Virtual Machines Based on DPDT</td>
<td>In those systems that changed pages rate is not high, the cost of checking current run changed pages in order to put in bit map causes overhead</td>
<td>It efficiently reduces the total migration time and the down time, which are too time consuming by their nature</td>
<td>In this algorithm there is a delay in dirty pages transmission, and there is an extra bitmap called latency which is used by those pages that change in the last run or in the current run, to be transferred later</td>
</tr>
<tr>
<td>Priority-Based Live Migration of Virtual Machine</td>
<td>Having too many high priority applications, the low priority applications may face starvation and never been processed</td>
<td>Comparing with pre copy algorithm it reduce 57% of malfunction of high priority application services</td>
<td>In this method application which ask for low down time, get a high priority and those that can tolerate longer lose of service get lower priority. During replication time only dirty pages which belong to high priority applications are transferred</td>
</tr>
</tbody>
</table>
3-1- Pre-Copy

This technique is used in most hypervisors in order to migrate virtual machines. Generally, it consists of 3 phases. But, researchers add new components and extensions in it to add extra features. First phase: source machine iteratively send virtual machine data including memory pages and CPU states to destination machine, during this operation it also records virtual machine memory pages changes. After doing this operation several times, and each time changed pages have transferred, second phase starts. In this phase: virtual machine suspends in source and then other pages, which have changed during first phase (due to continuous changes it was impossible to transfer them), in addition to CPU states which are essential for launching (restoring) virtual machine on destination are transmitted. Finally, in third phase: virtual machine starts on destination host from last state before suspension.

Last but not least is to mention less change rate in first phase leads to less downtime, which occur in phase 2 [3].

3-2- Adaptive Compression Memory Pages

For efficiency improvement on migration process, the compression technique is used. The compression causes the increase of accessible bandwidth and considerably the decrease of network traffic due to operation of migration. It should be noted that overload rate due to the operation of memory pages compression should be negligible. If overload is more important than compression benefits, represented method will not have any usage. Algorithm design for implementation of compression is easy, but if maximum rate of compression is needed, this task will be difficult. Therefore, the balance between created overload and rate of compression effect should be maintained. If the rate of memory pages changes is more than the rate of their sending in network, a mass of pages will change with writing on memory. In such circumstances, compressions of this mass of pages causes the increase of migration time. To resolve this problem, the comparative compression is proposed [13]. In this method, first the system distinguishes the extent of changed pages volume instantly. And it determines a limit between minimum and maximum analogy coefficient. With this technique, the extent of compression time will be decreased and then, the number of changed pages will be diminished [3][4][5].

3-3- Improved Pre-Copy Method

In this method, pre-copy improved by adding a bite map that tracks changed pages and marks these pages in it. During page sending process (second step of migration) the pages that are often changes (with high rate) are recorded in bite map and these pages will be sent in last step. Therefore, we will sure that the pages which are updated continually, will be sent only one time. Thus, overload of repetitive pages transfer in pre-copy technique is decreased considerably.

The results show that this method in compare to pre-copy technique decreases 35% of total time of migration and 34% of total transferred data. And repetition process is completed more than 5 - times rate (velocity) and it is finished [3][6].

3-4- CPU Scheduling

In pre-copy method, which is the most fundamental method for live migration of virtual machines, if the rate of changed pages is high, migration process will last long time, because a mass of data should be transferred. In some cases, when the rate of changed pages output in memory is more quick than the rate of pages sending in network, live migration will encounter error. In [7] an algorithm proposed that it is idea was the decrease of CPU frequency of virtual machines is relevant to pages transmission speed in pre-copy step.

In fact, in this method, when the rate of pages change is higher than the rate of their transmission in network with decrease of virtual machine processor speed, the level of created changes in memory are decreased temporary and consequently, changed pages volume decreased, too.

The results show that this method decreases 88% of virtual machine breakdown time and overload level is also acceptable[3][7].

3-5- Cloud Spider

This method has presented as a combination of virtual machine repetition based on scheduling in order to optimize virtual machines live migration and transfer in wide area networks such as the internet. A virtual machine during its own lifetime can schedule in different places. Accordingly, relocation of virtual machines in wide area networks, (that have low bandwidth and high delay time) will complete in a long time. Thus, we can resolve this problem with combination and strategies of virtual machine repetition.

In [9] Cloud Spider method has been proposed which is copied the image of virtual machine throughout infrastructure of the cloud. Then, a replica of virtual machine image is chosen as first version (original). Next, Cloud Spider publishes gradual changes in first version to all remained copies of that image. It should be noted that by this method, delay time in migration is minimized. On other hand, the strategy of repetition location selection is
determined based on elements which are influenced by long-term casts such that the average cost of each strong unit and computation unit in different places of cloud infrastructure will be checked.

One of the faults of this method is necessity of storage of similar copies in different places of cloud that will be negligible with regard to it is advantage (transfer of machine on internet or wide area network). In addition, in order to decrease of repetitive copies volume, de-duplication technique is used.

3-6- LRU and widespread Tree Algorithm

In this method, a framework have been proposed that has a pre-processing phase for predicting of changeable set of memory pages in pre-copy method. This framework causes decrease of transferred data during migration. In other word, the purpose of this method is combination of LRU algorithm and widespread tree.

As you know, LRU algorithm in operating system is replacement of newly used pages. For each page, it considers the time if final use of that page and then the page that isn’t used a long time, will be chosen for transfer. For implementation of this algorithm, a stack and a counter are used. The stack function stores page numbers, furthermore, the newly used page is transferred over stack. Thus, the newly used pages are always located over stack. We can use this idea in process of virtual machine migration. In order to complete described method, widespread tree is used. This tree is a binary search tree. The results show that 23.67% of sent data volumes during migration process and 11.45% of total migration time have been decreased [10][11].

3-7- DPDT

As earlier described, pre-copy algorithm does not work properly while pages are changing rapidly. For resolve this problem, the method of dirty (changed) pages transfer with delay has been presented. In this method, changing pages are sent with delay. Therefore, in offered algorithm [14], a bite map that is called to-delay is added. In this bite map, the pages which have changed in last step will send later. In addition to this bite map, three maps to-send, to-skip and to-fix have been added to the algorithm, too. These four bite maps are added in order to be compared, and correct decision is chosen before sending of determined pages. For example, if to-send equals one and to-delay equals zero, the pages will be sent. In contrast, if to-send equals one and to-delay equals one the page will not be sent immediately, and it will be sent with delay, because algorithm distinguishes the page is changing rapidly and it is immediate sending is useless.

Use of this method will cause the decrease of repetitive sending of dirty (changed) pages, the decrease of sent data volume and total migration time and virtual machine disruption time. The purpose is the decrease of transferred pages during migration with delay algorithm in transfer of dirty pages in order to decrease total time of migration and disruption time[11][14].

3-8- Priority-Based Live Migration

In this method, applications that cannot tolerate delay or failure in the service have more priority than other applications that own high toleration against delay and failure. During virtual machine migration and during memory pages transfer, only pages which belong to applications with high priority will be transferred. In implementation of this method, a shared disk (like SAN or NAS) have been used in order to maintain virtual machine disk and just transfer memory pages and virtual machine CPU states in migration process. This method includes two parts: Part1) it’s called MAVM: has the duty of application management in virtual machine. Part2) it’s responsible for application ranking management in live migration process. The results show that the represented method decreases 57% of service failure in applications with high priority [12][16].
4 - Evaluation Metrics for Migration Methods

In order to evaluate migration methods and techniques following metrics are used [4][12]:

- **Downtime:** Time period in which a specific service is unavailable. This period starts when virtual machine stops at source and ends when it starts up at destination.
- **Disruption Time:** Time period in which respond level decrease and response time increase from user’s point of view. In fact, time period in which service performance drops due to migration process.
- **Total Migration Time:** Time period in which migration process starts until states and information of source and destination machines become fully synchronized. Decreasing transfer file size and compressing data before transfer are methods which decrease this time period.
- **Amount of migrated data:** Amount of transfer data during migration process.
- **Performance overhead:** The overhead due to migration process, which is generally comparison of service throughput during migration and without migration.

Evaluation results of migration methods depicted in Table 2.

<table>
<thead>
<tr>
<th>Method</th>
<th>Downtime</th>
<th>Disruption Time</th>
<th>Total Migration Time</th>
<th>Transferred Workloads</th>
<th>Overhead Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-Copy</td>
<td>S</td>
<td>L</td>
<td>S</td>
<td>H (Iterative Data)</td>
<td>L</td>
</tr>
<tr>
<td>Adaptive Compression</td>
<td>S</td>
<td>L</td>
<td>S</td>
<td>L</td>
<td>H or M (Depends on algorithm complexity)</td>
</tr>
<tr>
<td>Memory Pages</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Improved Pre-Copy</td>
<td>S</td>
<td>L</td>
<td>S(35% Pre-Copy Improvement)</td>
<td>L(34% Pre-Copy Improvement)</td>
<td>L</td>
</tr>
<tr>
<td>CPU Scheduling</td>
<td>VS(88% Pre-Copy Improvement)</td>
<td>M</td>
<td>S</td>
<td>L</td>
<td>M</td>
</tr>
<tr>
<td>Cloud spider</td>
<td>S</td>
<td>UL</td>
<td>H</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>LRU&amp;SPLA Y Tree</td>
<td>-</td>
<td>-</td>
<td>S(11.45% Pre-Copy Improvement)</td>
<td>L(23.67% Pre-Copy Improvement)</td>
<td>M</td>
</tr>
<tr>
<td>DPDT</td>
<td>S (When Pages Change Rate is High)</td>
<td>-</td>
<td>VS</td>
<td>-</td>
<td>L</td>
</tr>
<tr>
<td>Priority Base</td>
<td>VS(57% Pre-Copy Improvement)</td>
<td>L</td>
<td>-</td>
<td>-</td>
<td>M</td>
</tr>
</tbody>
</table>
5- Conclusion

In this paper we studied new and optimized methods of virtual machine migration technique. In pre-copy method that is often used as a primary method for live migration of virtual machines, if memory pages change rate is high, migration process will last long time. In addition, if memory pages change rate is more rapid than pre-copy speed, live migration will be failed. We can conclude that live migration based on CPU scheduling is the most optimized and useful method because most of applications have high writing rate on memory or pre-copy speed is low, because of limitation in network bandwidth. This method causes the efficiency improvement in applications during live migration operation.

References