

Improving (pre-copy) Algorithm in Terms of Time and Productivity

Dr. Ahmad Saker Ahmad*
Haider Khalil**

(Received 18 / 11 / 2019. Accepted 9 / 2 / 2020)

□ ABSTRACT □

Live migration is one of the most important features offered by virtual environments. It is defined as the transfer of the virtual machine from one physical server to another server without interrupting the service of the applications on which it operates. The pre-copy algorithm is one of the most important algorithms that implement live migration. Where the principle of this algorithm depends on the repeated transfer of the dirty page during the live migration between the source and the target, so that the frequency stops at a certain threshold.

The large change in the memory pages will increase the number of pages sent over the network, increasing the total amount of transmitted memory, thus increasing the network consumption rate, total migration time and downtime, which are the basic criteria for evaluating performance during the implementation of live migration. This algorithm has a problem (page re-send problem), which is to move the same pages repeatedly in each iteration, which increases the amount of data sent over the network and increases the total time of migration of the beard, which increases the total number of pages transferred memory much more than the actual size For memory which greatly increases the load on the network. The purpose of this research was to improve the total time of live migration and improve network throughput through the development of the pre-copy algorithm. The modified algorithm lingers before sending variable pages, categorizing these pages into two groups, sending pages that change slightly in the current iteration, and highly variable pages are postponed to the last iteration, after setting a threshold that defines each group. Keeping pages that change frequently until the last occurrence reduces the unnecessary transfer of the same pages and reduces the total number of pages transferred and thus reduces the total time of live migration compared to the previous algorithm

We performed experiments using Centos 7 and XEN-type hypervisor and to analyze the results we used the NMON Performance Analyzer.

Keywords: live migration, virtual machine, hypervisor, pre-copy .

*Professor, Department of System and Computing Network Engineering, Faculty of Informatics, Engineering, Tishreen University, Lattakia, Syria.

**Postgraduate Student (Phd.), Department of System and Computing Network Engineering, Faculty of Informatics, Engineering, Tishreen University, Lattakia, Syria.

تحسين خوارزمية النسخ المسبق (Pre-Copy) من حيث الزمن و الإنتاجية

د. أحمد صقر أحمد*

حيدر خليل**

(تاريخ الإيداع 18 / 11 / 2019. قُبِلَ للنشر في 9 / 2 / 2020)

□ ملخص □

تعتبر الهجرة الحية (live migration) من أهم السمات التي تقدمها البيئات الافتراضية، وتعرف على أنها عملية نقل الآلة الافتراضية من مخدم فيزيائي إلى مخدم فيزيائي آخر دون انقطاع لخدمة التطبيقات التي تعمل عليها، و بزمن توقف صغير جدا يقترب من الصفر. إن خوارزمية النسخ المسبق (pre-copy) هي أول و أهم الخوارزميات التي تتفد الهجرة الحية، حيث يعتمد مبدأ عمل هذه الخوارزمية على النقل المتكرر لصفحات الذاكرة المتغيرة (dirty page) أثناء تطبيق الهجرة الحية بين المصدر و الهدف، بحيث يتوقف التكرار عند عتبة معينة .

إن التغير الكبير في صفحات الذاكرة سوف يزيد من عدد الصفحات المرسلة عبر الشبكة مما يزيد من حجم الذاكرة الكلي المنقول، وبالتالي سيزداد معدل استهلاك الشبكة و الزمن الكلي للهجرة و زمن التوقف. تعاني هذه الخوارزمية من مشكلة (page re-send problem)، و هي نقل الصفحات نفسها مرارا في كل دور مما يزيد من كمية البيانات المرسلة عبر الشبكة و بالتالي يزداد الزمن الكلي للهجرة للحية و يصبح مجموع صفحات الذاكرة المنقولة أكبر بكثير من الحجم الفعلي للذاكرة مما يزيد الحمل بشكل كبير على الشبكة . لذلك كان الغرض من هذا البحث تحسين الزمن الكلي للهجرة الحية و تحسين انتاجية الشبكة (throughput) من خلال تطوير خوارزمية النسخ المسبق، حيث تقوم الخوارزمية المعدلة بالترتيب قبل إرسال الصفحات المتغيرة، و تصنيف هذه الصفحات في مجموعتين و ارسال الصفحات التي تتغير بشكل قليل في الدور الحالي و الصفحات المتغيرة بشكل كبير يتم تأجيل إرسالها الى الدور الأخير، و ذلك بعد تحديد عتبة تحدد كل مجموعة. يؤدي الاحتفاظ بالصفحات التي تتغير بشكل متكرر حتى آخر تكرار إلى تقليل النقل غير الضروري لنفس الصفحات و تقليل العدد الكلي للصفحات المنقولة و بالتالي تقليل الزمن الكلي للهجرة الحية مقارنة مع الخوارزمية السابقة.

قمنا بتنفيذ التجارب باستخدام نظام التشغيل Centos 7 و hypervisor من نوع XEN و لتحليل النتائج استخدمنا محلل الأداء NMON. أظهرت النتائج أن الخوارزمية المعدلة قد حسنت من الزمن الكلي للهجرة الحية كما حسنت من معدل استهلاك الشبكة.

الكلمات المفتاحية: الهجرة الحية، الآلة الافتراضية، مدير الآلة الافتراضية، خوارزمية النسخ المسبق.

* أستاذ - قسم النظم والشبكات الحاسوبية - كلية الهندسة المعلوماتية - جامعة تشرين - اللاذقية - سورية.

** طالب دراسات عليا (دكتوراه) - قسم النظم والشبكات الحاسوبية - كلية الهندسة المعلوماتية - جامعة تشرين - اللاذقية - سورية.

مقدمة:

تتطور الحوسبة السحابية بشكل سريع وذلك لما قدمت من مرونة في الحصول على الخدمات المختلفة و حرية التوسع و توفير الطاقة و المال، الأمر الذي جعلها هدفا للبحث و السعي الدائم من قبل الباحثين لتطوير مكوناتها بهدف الحصول على أفضل أداء ممكن. تعد البيئات الافتراضية (virtualization) البنية الأساسية و المكون الأهم من مكونات الحوسبة السحابية و ذلك لما تقدمه من الاستخدام الأمثل للموارد من خلال تحميل أنظمة تشغيل متعددة على جهاز مادي واحد مع إمكانية العزل الكامل بين هذه الأنظمة لتعمل بشكل مستقل [1]. تقوم الحوسبة السحابية بتوزيع مهام و خدمات الحوسبة على مجموعة كبيرة من الآلات الافتراضية (virtual machines). عندما يزداد الحمل على أحد المخدمات الفيزيائية نتيجة تزايد الطلب على الخدمات التي تم تفعيلها على آلتها الافتراضية يجب نقل عدد معين من آلتها الافتراضية إلى مخدم فيزيائي آخر دون انقطاع الخدمة التي تقدمها التطبيقات على هذه الآلات. إن عملية نقل الآلة الافتراضية VM من مخدم الى اخر دون انقطاع خدمات التطبيقات التي تعمل عليها تسمى بالهجرة الحية للآلة الافتراضية . في الماضي كانت عملية نقل الآلة الافتراضية بين مخدمين تتطلب إيقاف الآلة الافتراضية على المخدم المصدر ومن ثم حجز الموارد من ذاكرة و وحدة معالجة على المخدم الهدف و بعد ذلك يتم النقل.

إن ترحيل الآلة الافتراضية من مخدم إلى آخر يتضمن نقل صفحات الذاكرة و حالة المعالج و مجاري الدخل و الخرج و طلبات المستخدم من المخدم المصدر الى المخدم الهدف [1] . تعتبر خوارزمية النسخ المسبق من أول و أهم الخوارزميات المستخدمة لتنفيذ الهجرة الحية، تعتمد هذه الخوارزمية في مبدأ عملها على تكرار نقل صفحات الذاكرة المتغيرة (dirty page) بين المصدر و الهدف بحيث يتوقف التكرار عند عتبة معينة ، [2] بالتالي إن التغيير الكبير في صفحات الذاكرة سوف يزيد من عدد الصفحات المرسله عبر الشبكة مما يزيد من حجم الذاكرة الكلي و بالتالي يزداد الزمن الكلي للهجرة و حجم البيانات الإضافية المرسله عبر الشبكة مما يزيد من معدل استهلاكها، لذلك كان لا بد من تحسين خوارزمية النسخ المسبق بحيث يتم تقليل النقل غير الضروري لنفس الصفحات عدد كبير من المرات و تقليل العدد الكلي للصفحات المنقولة و بالتالي تقليل الزمن الكلي للهجرة الحية.

أهمية البحث و أهدافه:

تعتبر الهجرة الحية من أهم الميزات التي تقدمها البيئات الافتراضية في مراكز البيانات الافتراضية ، وذلك لما تقدمه من ميزات لمدراء هذه المراكز من مرونة في عمليات الصيانة و موازنة الحمل على المخدمات و الحفاظ على البيانات من الكوارث عن طريق ترحيل الآلة الافتراضية الى أماكن آمنة ، لذلك كان الهدف من هذا البحث تحسين الزمن الكلي للهجرة الحية و تقليل الحمل على الشبكة أثناء تطبيق الهجرة الحية من خلال تحسين خوارزمية النسخ المسبق (pre-copy) و تطبيق الخوارزمية المحسنة على بيئة النمط الافتراضي الجزئي كمدبر للآلة الافتراضية [3].

طرائق البحث و مواد:

قمنا بتحسين خوارزمية النسخ المسبق (pre-copy) من خلال التعديل على عملها في الطورين الأول و الثاني، حيث نقوم في كل دور بكتابة جميع الصفحات المتغيرة في ملف اسمه change_f، ومن ثم يتم حساب عدد المرات التي تتطلب فيها صفحات الذاكرة، و نقوم بتسجيل هذه الأعداد في مصفوفة X[i] ، ثم نقوم بالإعتماد على هذه المصفوفة

بتقسيم الصفحات المتغيرة الى مجموعتين A_1 , A_2 ، حيث تضم المجموعة الأولى A_1 جميع الصفحات التي عدد تكرارها أصغر من عتبة معينة TH_1 ، بينما تحوي المجموعة A_2 جميع الصفحات التي عدد تكرارها أكبر أو تساوي هذه العتبة ، و تطلب ذلك دراسة عميقة لفهم المحاكاة الافتراضية للذاكرة المحجوزة للآلة الافتراضية وهذا ما سنوضحه في الفقرة التالية. تم تسمية الخوارزمية المحسنة $AD-pre-copy$.

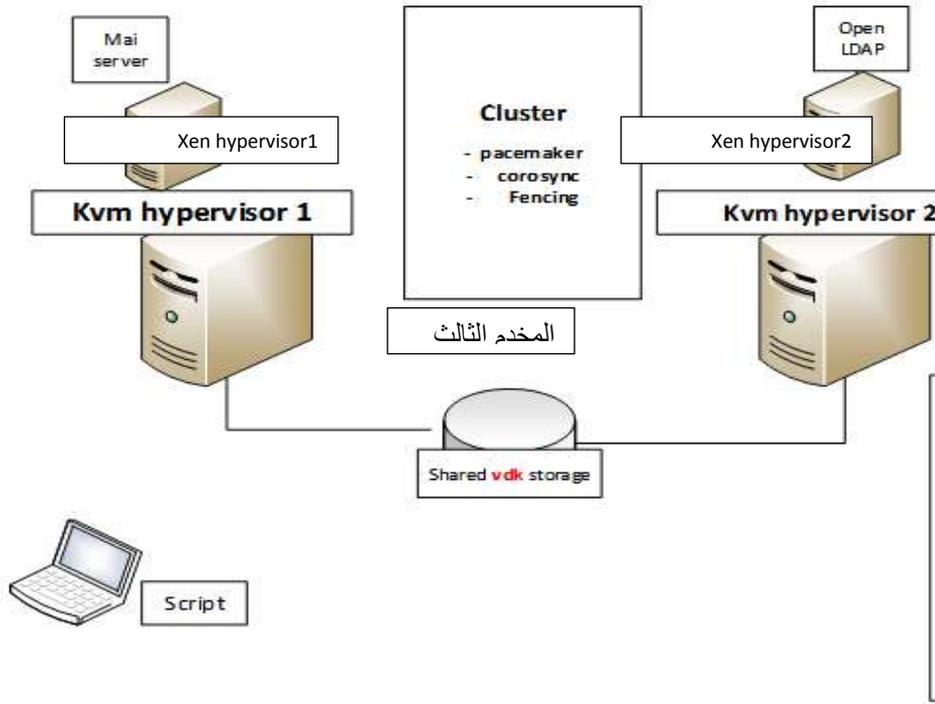
الشكل (1) يوضح البيئة العملية للتجربة ، حيث استخدمنا في تجربتنا ثلاث أجهزة مادية كمخدمات فيزيائية كل جهاز مزود بمعالج $core i3$ و ذواكر $4 GB$. تم تنصيب نسخة $centos 7$ على كلا المخدمين و تنصيب البيئة الافتراضية Xen لتعمل ك $hypervisor$ من النمط الافتراضي الجزئي على كلا المخدمين و استخدام المخدم الثالث كجهاز تخزين مشترك للقرص الافتراضي الخاص بالآلات الافتراضية، كما وضعنا كلا المخدمين في عنقود. قمنا بتجهيز آلة افتراضية بحجم ذاكرة $2 GB$ على أحد المخدمين تقدم خدمة البريد الإلكتروني و طبقنا الهجرة الحية على هذه الآلة و قمنا بترحيلها إلى المخدم الآخر. طبقنا حمل على الآلة الافتراضية أثناء ترحيلها من خلال تغيير عدد الزبائن طالبي خدمة البريد الإلكتروني وذلك بهدف إجراء تغيير على صفحات الذاكرة أثناء الهجرة الحية، حيث أن تغيير صفحات الذاكرة أثناء نقل الآلة الافتراضية يعتبر من أهم العوامل في زيادة الزمن الكلي للهجرة، تم تغيير حجم الملف المرسل وذلك بهدف إشغال الشبكة أثناء تطبيق الهجرة الحية. قمنا بتشغيل محلل الأداء $nmon$ على المخدم المصدر للحصول على النتائج و تحليلها. تم تطبيق خوارزمية النسخ المسبق ($pre-copy$) و خوارزمتنا المحسنة $AD-pre-copy$ على بيئة العمل السابق لتجهيز الآلة الافتراضية من $Xen hypervisor1$ الى $Xen hypervisor2$ و مقارنة النتائج تم كتابة $script$ يقوم بإرسال رسائل الكترونية وفق الكود التالي:

For I in 1 to n do

For j in 1 to n do

echo . | mail -r test\$i.user@Xenlab.local -a file -s "This is test" [test\\$j.user@Xenlab.local](mailto:test$j.user@Xenlab.local)

حيث أن n : عدد المستخدمين و $file$ الملف متغير الحجم ، و هي قيم يتم تغييرها للحصول على تجارب متعددة. فمثلا من أجل $n=8$ و حجم ملف 6 ميغابايت سيكون عدد الرسائل الإلكترونية على الشبكة $64 (8 * 8)$ رسالة بحسب حلقة for المتداخلة و حجم البيانات المنقولة على الشبكة هو $384 (6*64) MB$.

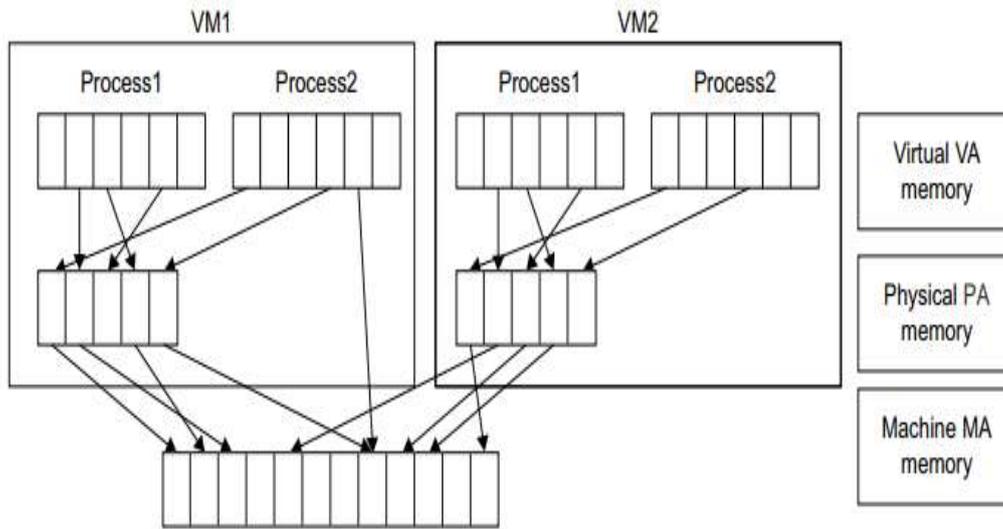


الشكل (1) بيئة عمل التجربة

المحاكاة الافتراضية للذاكرة (memory virtualization) [4]:

تعتمد معظم أنظمة التشغيل الحديثة في عملية إدارة الذاكرة على وحدة إدارة الذاكرة Memory Management Unit (MMU) و ذلك لتحويل العناوين المنطقية التي ينتجها المعالج الى عناوين فيزيائية بالإعتماد على جدول الصفحات paging table ، و بما أن جدول الصفحات مخزن في الذاكرة الفيزيائية فإنه كلما أراد المعالج الحصول على عنوان معين من الذاكرة فإن الوصول الى هذا العنوان يحتاج الى مرورين على الذاكرة الاولى على جدول الصفحات لتحديد موقع العنوان في الذاكرة و الثاني الى مكان العنوان نفسه في الذاكرة، مما يسبب اعباء اضافية الأمر الذي دعى الى ابتكار عتاد مادي يدعى tlp الذي يقوم باختصار المراحل السابقة الى مرحلة واحدة. إن إدارة الذاكرة في البيئات الافتراضية تعتمد على نظام التشغيل الضيف و على مدير الآلات الافتراضية معا، حيث ان تحويل عنوان منطقي الى عنوان فيزيائي على ذاكرة المضيف يتم وفق مرحلتين، المرحلة الأولى تحويل العناوين المنطقية الى عناوين فيزيائية لنظام تشغيل الضيف (GPA: physical memory address in guest OS) ، و المرحلة الثانية تحويل العناوين الفيزيائية لنظام التشغيل الضيف على عناوين فيزيائية حقيقية لنظام تشغيل المضيف (HPA: physical memory address of the host machine) .

إن نظام تشغيل الضيف هو المسؤول عن عملية تحويل العناوين المنطقية للآلة الافتراضية الى عناوين فيزيائية للنظام الضيف بالإعتماد على جداول الصفحات لنظام تشغيل الضيف و مدير الالة الافتراضية هو المسؤول عن المرحلة الثانية التي تتضمن تحويل العناوين الفيزيائية للنظام الضيف الى عناوين فيزيائية حقيقية على النظام المضيف بالإعتماد على جدول صفحات (shadow page table) كما يبين الشكل (2) .



الشكل (2) مراحل عنوانة الذاكرة في البيئة الافتراضية

يقوم مدير الآلة الافتراضية بتشكيل جدول صفحات يسمى (shadow page table) يحوي جميع جداول الصفحات الخاصة بأنظمة تشغيل الضيوف. و بما أن الإجرائية الواحدة في أنظمة التشغيل الحديثة يمكن أن تملك أكثر من جدول صفحات الأمر الذي يسرع من عملية امتلاء جدول الصفحات (shadow page table) مما يزيد استهلاك الذاكرة بشكل كبير و يؤثر على الاداء بشكل سلبي. استخدمت المعالجات الحديثة التي تدعم تقنية ال virtualization عتاد مادي إضافي يدعى TLP و الذي يقوم باختصار مراحل تحويل العناوين السابقة الى مرحلة واحدة، حيث يقوم بتحويل العناوين المنطقية لنظام تشغيل الضيف مباشرة الى عناوين فيزيائية حقيقية على نظام تشغيل المضيف تحت إشراف مدير الآلة الافتراضية الذي يحرص بشكل دائم على تسجيل تغيرات العناوين المنطقية على نظام التشغيل الضيف و تحديثها في جدول الصفحات (shadow page table). إن تطوير خوارزمية النسخ المسبق و الوصول إلى الصفحات المتغيرة في الذاكرة يتطلب الفهم الدقيق لجدول صفحات (shadow page table) و العتاد المادي TLP.

الهجرة الحية (live migration):

هي ترحيل الآلة الافتراضية من مخدم فيزيائي إلى مخدم فيزيائي آخر دون إيقافها عن العمل مع ضمان استمرارية الخدمة التي تقدمها بزمن توقف صغير جدا يقترب من الصفر، وذلك بهدف موازنة الحمل بين المخدمات الفيزيائية أو توفير الطاقة أو من أجل إجراء عمليات صيانة للمخدم الفيزيائي المصدر [5]. إن ترحيل الآلة الافتراضية يتضمن نقل كل من صفحات الذاكرة و حالة المعالج و عمليات الدخل و الخرج و الطلبات من المخدم المصدر إلى الهدف. يعتبر مدير الآلة الافتراضية (hypervisor) المسؤول الرئيسي عن عملية الهجرة الحية، لذلك يجب أن يحقق الشروط التالية أثناء تطبيق الهجرة الحية [6] [7]:

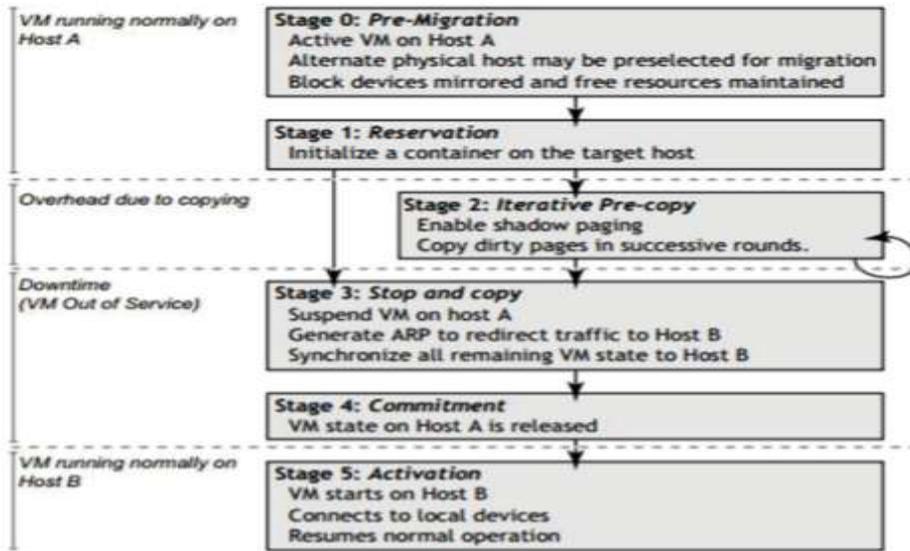
- 1- استمرارية الخدمة: إن تطبيق الهجرة الحية يجب ألا يسبب تدهور في أداء التطبيقات التي تعمل على الآلة الافتراضية.
- 2- الإستهلاك الأمثل للموارد : يجب ألا تستهلك الموارد بشكل كبير أثناء تطبيق عملية الترحيل.

- 3- **التنبؤ:** يجب أن يكون من الممكن التنبؤ بزمان الهجرة الكلي و زمن التوقف و الموارد التي سيتم استهلاكها على المخدم الهدف من ذاكرة و وحدة معالجة و عرض الحزمة على الشبكة.
- 4- **الشفافية:** يجب أن تكون عملية الترحيل شفافة لكل من التطبيقات التي تعمل على الآلة الافتراضية و مستخدمي هذه التطبيقات.
- تعتبر خوارزمية النسخ المسبق (pre-copy) أول و أهم الخوارزميات لتنفيذ الهجرة الحية و في ما يلي سيتم شرح آلية عمل هذه الخوارزمية.

خوارزمية النسخ المسبق pre-copy:

تقوم هذه الخوارزمية بتنفيذ الهجرة الحية و فق المراحل التالية [8]:

- 1- **مرحلة التحضير:** يتم حجز الموارد اللازمة لعمل الآلة الافتراضية على المضيف الهدف.
 - 2- **مرحلة النسخ المتكرر:** يقوم hypervisor بنسخ جميع صفحات الذاكرة من المصدر إلى الوجهة بينما لا تزال الآلة الافتراضية تعمل على المصدر، إذا تغيرت صفحات الذاكرة (dirty pages) أثناء هذه العملية سيتم إعادة إرسالها و تكرار العملية من 2 إلى n-1 مرة حيث شرط التوقف عن الإرسال و الانتقال الى الطور التالي هو عدد تكرارات 29 أو حجم الصفحات المتغيرة في الإرسال السابق 256 KB و هي قيم افتراضية للخوارزمية [8].
 - 3- **مرحلة التوقف و النسخ:** يتم إيقاف الآلة الافتراضية VM على المصدر و يتم نقل جميع الصفحات المتبقية و سجلات المعالج الى الهدف و من ثم استئناف عمل ال VM على الهدف.
- الشكل (3) يوضح مراحل تنفيذ خوارزمية النسخ المسبق pre-copy .



الشكل (3) مراحل تنفيذ خوارزمية النسخ المسبق Pre-copy

يتم قياس أداء خوارزميات الهجرة الحية من خلال أربعة معايير [9] [10]:

- 1- **عدد الصفحات المنقولة:** هو إجمالي عدد الصفحات المنقولة أثناء الترحيل. للحصول على أفضل أداء يجب أن تكون هذه القيمة اقل ما يمكن، كما يجب أن تكون مساوية للعدد الإجمالي لصفحات الآلة الافتراضية التي يتم

ترحيلها، ولكن في خوارزمية النسخ المسبق pre-copy هو دائما أكثر بسبب النقل المتكرر لصفحات الذاكرة المتغيرة خلال أدوار متعددة. يتم تعريف إجمالي الصفحات المنقولة V_{mig} على أنها العدد الإجمالي للصفحات في جميع التكرارات n و يعطى بالعلاقة التالية :

$$V_{mig} = \sum_{i=0}^n V_i \quad (1)$$

حيث V_i هو عدد الصفحات المنقولة في التكرار الواحد و n هو العدد الإجمالي للتكرار.

2- **الزمن الكلي للترحيل:** هو الوقت المستغرق في نقل الآلة الافتراضية بالكامل من المصدر إلى الهدف. ويجب أن يكون هذا الزمن أقل ما يمكن ويعطى بالعلاقة التالية:

$$T_{mig} = \sum_{i=0}^n T_i \quad (2)$$

حيث T_i هو الزمن المستغرق لإنجاز التكرار i .

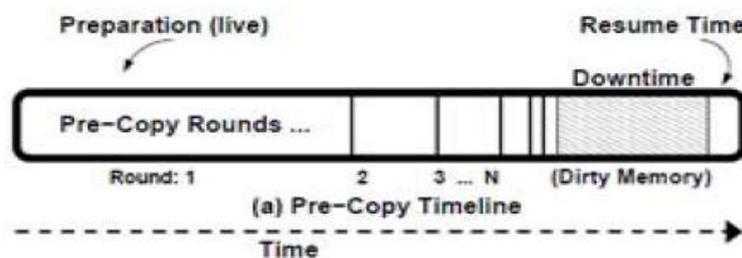
3- **زمن التوقف:** إنه الوقت المستغرق في عملية الترحيل لإيقاف الآلة الافتراضية عند المصدر واستئنافها في العمل على المضيف الهدف. يؤثر هذا الزمن بشكل مباشر على توفر الخدمة ، حيث تعتمد قيم هذا الزمن على الصفحات المتبقية في التكرار الأخير.

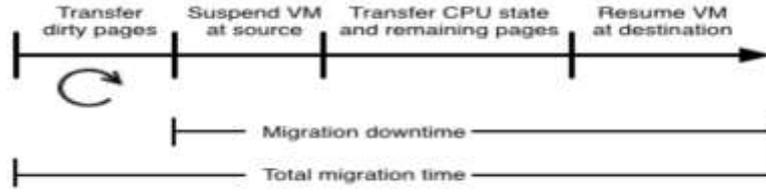
يتم قياس زمن التوقف باعتباره الوقت الذي يستغرقه تكرار عملية الترحيل الأخيرة ، أو يمكن حسابه من خلال حساب زمن انقطاع الخدمة.

4- **Overhead:** هي البيانات الإضافية التي يتم نقلها أثناء الترحيل والتي يتم تعريفها على أنها حجم صفحات الذاكرة المرسله خلال التكرارات على حجم الصفحات الحقيقي للآلة الافتراضية و يعطى بالعلاقة:

$$R_d = \frac{V_{mig}}{V_{mem}} \quad (3)$$

حيث أن V_{mig} هو الحجم الكلي لصفحات الذاكرة التي تم نقلها خلال الترحيل و V_{mem} هو حجم الذاكرة الفعلي للآلة الافتراضية. يجب أن يكون ال overhead أقل ما يمكن للحصول على الأداء الأفضل. إن الاستهلاك الزائد لوحدة المعالجة المركزية ، والتغير السريع في صفحات الذاكرة أثناء الترحيل ، والاستهلاك الكبير لعرض النطاق الترددي للشبكة ، وإجمالي عدد مرات التكرار هي بارامترات تؤثر على أداء الهجرة الحية للآلة الافتراضية ويمكن دراستها بهدف تحسين أداء الترحيل. الشكل (4) يوضح الخط الزمني للهجرة الحية و الزمن الكلي وزمن التوقف أثناء الترحيل.





الشكل (4) الخط الزمني للهجرة الحية و الزمن الكلي و زمن التوقف أثناء الترحيل

من أهم المشاكل التي تعاني منها هذه الخوارزمية [11]:

- **مشكلة معدل النقل (transfer rate problem)**: إن الصفحات المتغيرة (dirtied page) تزداد بمعدل أسرع من معدل نقلها عبر الشبكة مما يستهلك الشبكة بشكل كبير، الأمر الذي يؤثر على استمرارية الخدمة و يزيد من زمن التوقف، و قد يتسبب في قطع اتصال الزبون.

- **مشكلة إعادة إرسال الصفحات (page re-send problem)**: إن إرسال الصفحات المتغيرة في الادوار من 2 الى n-1 قد يؤدي الى نقل الصفحات نفسها عدد كبير من المرات مما يزيد من كمية البيانات المرسله عبر الشبكة و بالتالي يزداد الزمن الكلي للهجرة للحية و يصبح مجموع صفحات الذاكرة المنقولة أكبر بكثير من الحجم الفعلي للذاكرة مما يزيد الحمل بشكل كبير على الشبكة .

وبالتالي سنحاول ايجاد خوارزمية تقوم بالتريث قبل إرسال الصفحات المتغيرة و تصنيف هذه الصفحات في مجموعتين و إرسال الصفحات التي تتغير بشكل قليل في الدور الحالي و الصفحات المتغيرة بشكل كبير يتم تأجيل إرسالها الى الدور الأخير و ذلك بعد تحديد عتبة تحدد كل مجموعة.

خوارزمية pre-copy المعدلة :

قمنا بإجراء تعديلات على خوارزمية ال pre-copy في المرحلتين الأولى و الثانية لحل المشاكل الأساسية التي تعاني منها هذه الخوارزمية و هي **transfer rate problem** و **page re-send problem**.

المرحلة الأولى: في هذه المرحلة يتم إرسال أجزاء الذاكرة التي لم يطرأ عليها أي تغيير و بالتالي نكون قد قللنا حجم الذاكرة المرسله في المرحلة الأولى مقارنة مع خوارزمية ال pre-copy مما يقلل من الزمن الكلي للهجرة الحية.

المرحلة الثانية: في التكرارات من 2 و حتى n-1 تم تقليل معدل نقل الصفحات المتغيرة في كل دور من خلال تتبع عدد مرات تكرار الصفحات المتغيرة في الذاكرة ،حيث نقوم في كل دور بكتابة جميع الصفحات المتغيرة في ملف اسمه change_f ومن ثم يتم حساب عدد المرات التي تطلب فيها صفحات الذاكرة و نقوم بتسجيل هذه الأعداد في مصفوفة $X[i]$ ، ثم نقوم بالإعتماد على هذه المصفوفة بتقسيم الصفحات المتغيرة الى مجموعتين $A1$ $A2$ ، حيث تضم المجموعة الأولى $A1$ جميع الصفحات التي عدد تكرارها أصغر من عتبة معينة $TH1$ بينما تحوي المجموعة $A2$ جميع الصفحات التي عدد تكرارها أكبر أو تساوي هذه العتبة .يتم حساب هذه العتبة في كل دور وفق العلاقة التالية:

$$TH1 = [(\max[\text{page modification rate}] + \min[\text{page modification rate}]) \div 2]$$

يتم حساب THi في كل دور بالإعتماد على المصفوفة $X[i]$.ويتم التوقف و الانتقال الى المرحلة الأخيرة عندما يكون عدد الصفحات المتغيرة في التكرار الأخير أكبر بمره و نصف من عدد الصفحات المتغيرة في التكرار السابق ، و هي قيمة تم ملاحظتها من خلال التجارب بعد أن أصبحت عملية حساب المتوسط غير مجدية وذلك لأن تكرارات صفحات

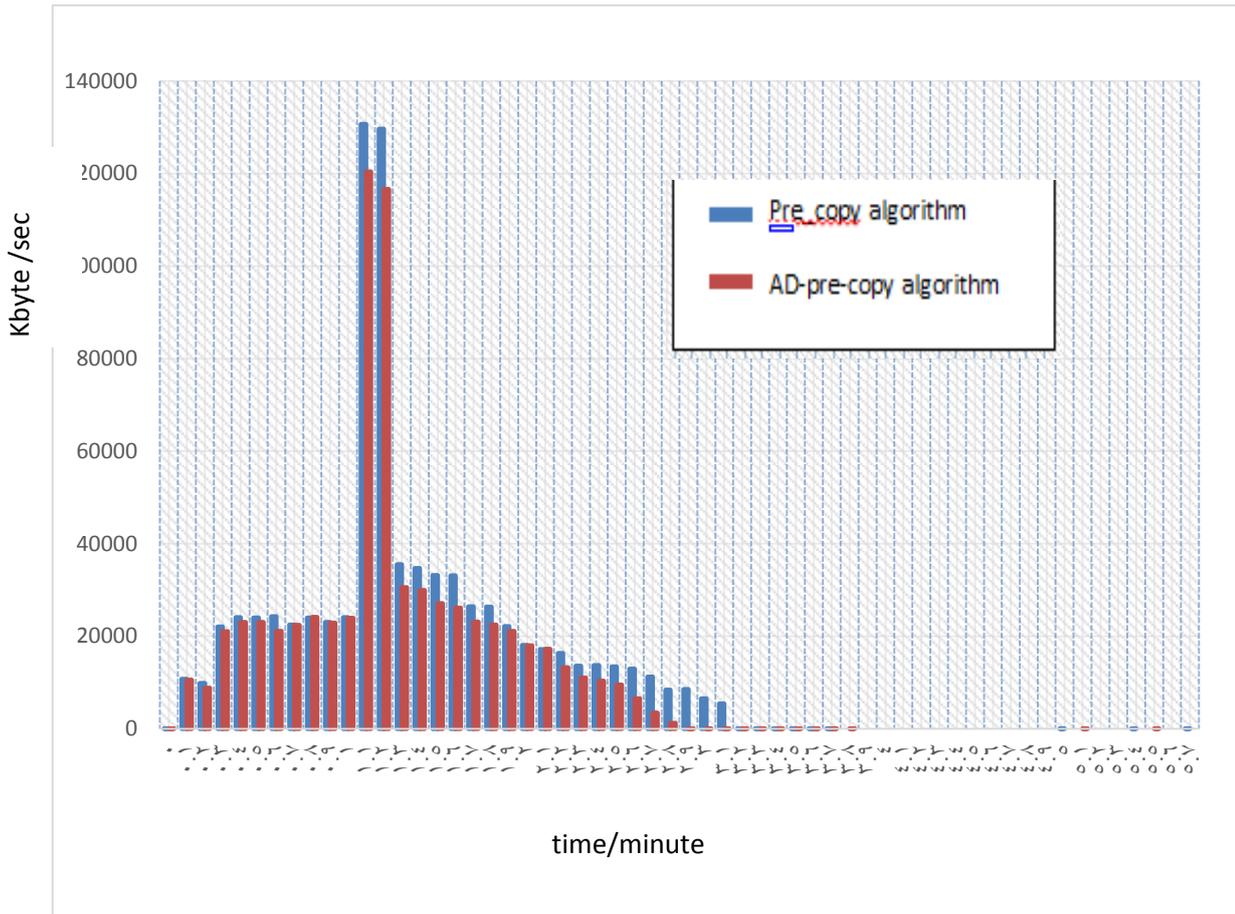
الذاكرة عند مرحلة معينة أصبحت متقاربة . في حال لم يتحقق الشرط يتم التوقف عند التكرار 29 و هو الشرط الافتراضي للتوقف في خوارزمية ال pre-copy. يؤدي الاحتفاظ بالصفحات التي تتغير بشكل متكرر حتى آخر تكرار إلى تقليل النقل غير الضروري لنفس الصفحات مرارًا و تقليل العدد الكلي للصفحات المنقولة و بالتالي تقليل الزمن الكلي للهجرة الحية مقارنة مع الخوارزمية السابقة.

المرحلة الثالثة :في هذه المرحلة يتم نقل جميع الصفحات المتبقية و سجلات المعالج الى الهدف و من ثم استئناف عمل ال VM على الهدف.

النتائج والمناقشة:

- السيناريو الأول من أجل $n=8$ و حجم الملف المرسل 6 MB :
عندما تأخذ n القيمة 8 سيكون عدد الرسائل الإلكترونية 64 رسالة و كل رسالة تنقل حجم ملف 6 MB و بالتالي حجم البيانات المنقولة عبر الشبكة $384=6*64$ MB ، إن زيادة n بمقدار صغير سيزيد حجم البيانات المنقولة عبر الشبكة بشكل كبير كما هو مبين ضمن حلقتي for . قمنا بتنفيذ ال script مرتين في أول التجربة و في الدقيقة الأولى عند تنفيذ الهجرة الحية وذلك بهدف إجراء تعديلات كبيرة على الذاكرة أثناء تطبيق الهجرة الحية .الشكل (5) يبين حجم البيانات المنقولة عبر الشبكة أثناء ترحيل الآلة الافتراضية التي تقدم خدمة البريد الإلكتروني من hyper visor 1 إلى hyper visor 2 وذلك عند 64 طلب (8*8) و حجم ملف 6 MB .

يمثل المحور الأفقي الزمن حيث يعرض حجم البيانات المنقولة كل 6 ثانية بعد أخذ المتوسط الحسابي للفترات السابقة و المحور العمودي حجم البيانات المنقولة (صفحات الذاكرة ،طلبات الزبائن) خلال الترحيل. يمثل المنحني ذو اللون الأحمر خوارزمية ال AD-pre-copy و المنحني ذو اللون الأزرق يمثل خوارزمية ال pre-copy الأساسية . قمنا بتنفيذ أمر الهجرة في الدقيقة 1، و تم حساب الزمن الكلي للهجرة من لحظة تنفيذ الأمر و حتى تشغيل البيئة الافتراضية على المخدم الهدف ،و يمكن حسابها من المخطط من لحظة التنفيذ و حتى توقف نقل البيانات إلى الشبكة عند المخدم المصدر ،حيث أظهرت النتائج أن الزمن الكلي للهجرة أثناء تطبيق الخوارزمية المعدلة 2 min في حال كان 2.3 min عند تطبيق خوارزمية ال pre-copy الأساسية .نلاحظ أن معدل استهلاك الشبكة قد زاد بشكل كبير أثناء تطبيق الهجرة الحية و يظهر هذا الاستهلاك في ذروته بين الفترتين الزمنيتين 1.1 min و 1.4 min و ذلك لان خوارزمية النسخ المسبق تقوم في الطور الاول بنقل كامل الذاكرة للآلة الافتراضية و من ثم تقوم بإرسال صفحات الذاكرة المتغيرة في حين الخوارزمية المعدلة تنقل في الطور الأول الذاكرة غير المتغيرة فقط . يظهر المخطط أن الخوارزمية المعدلة قد خفضت من حجم صفحات الذاكرة المنقولة خلال عملية الترحيل، و بالتالي تم تقليل حجم البيانات الإضافية المنقولة و تخفيض الزمن الكلي للهجرة الحية.



الشكل (5) حجم البيانات المنقولة عبر الشبكة أثناء ترحيل الآلة الافتراضية مع 64 طلب و حجم ملف 6 MB

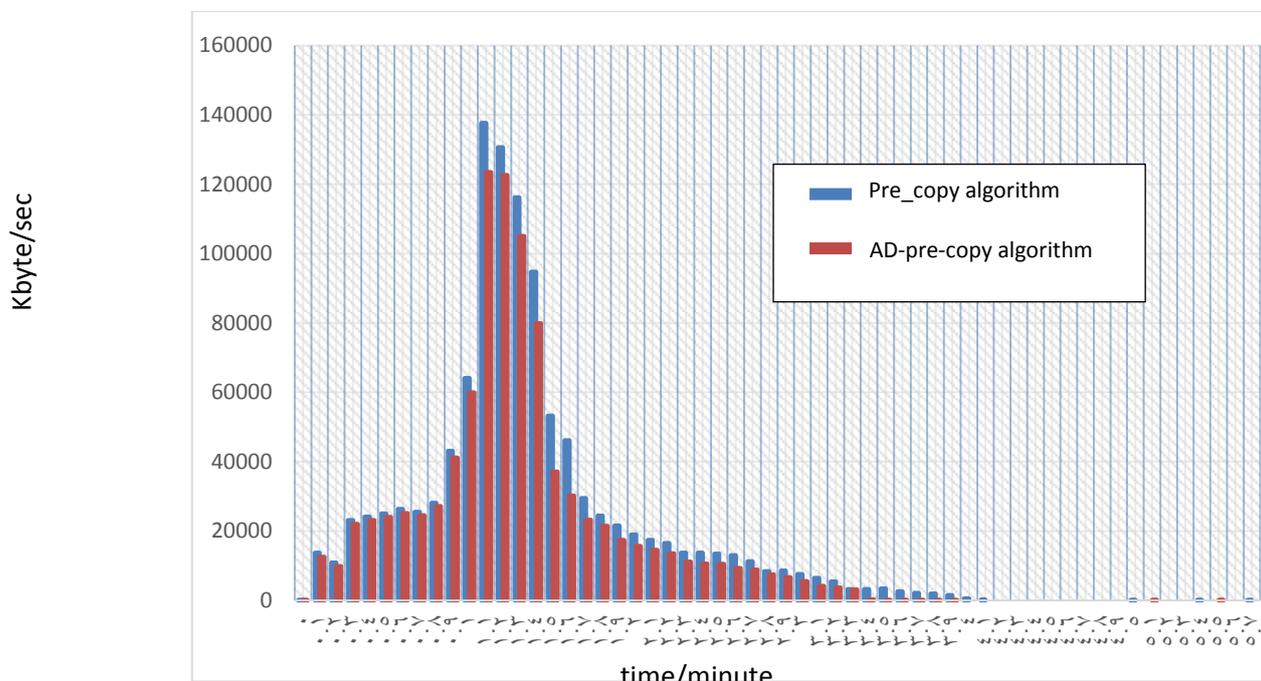
- السيناريو الثاني $n=12$ و حجم الملف 6 MB:

عندما تأخذ n القيمة 12 سيكون عدد الرسائل الإلكترونية 144 رسالة و كل رسالة تنقل حجم ملف 6 MB و بالتالي حجم البيانات المنقولة عبر الشبكة $864 = 6 * 144$ MB.

الشكل (6) يبين حجم البيانات المنقولة عبر الشبكة أثناء ترحيل الآلة الافتراضية التي تقدم خدمة البريد الإلكتروني من hypervisor 1 إلى hypervisor 2 وذلك عند 144 طلب ($12 * 12$) و حجم ملف 6 MB. قمنا بتنفيذ أمر الترحيل في الدقيقة 1.

حيث أظهرت النتائج أن الزمن الكلي للهجرة أثناء تطبيق الخوارزمية المعدلة 2.3 min في حال كان 2.75 min عند تطبيق خوارزمية ال pre-copy الرئيسية. نلاحظ أن معدل استهلاك الشبكة قد زاد بشكل كبير أثناء تطبيق الهجرة الحية و يظهر هذا الإستهلاك في ذروته بين الفترتين الزمنيتين 1.1 min و 1.5 min و ذلك لان خوارزمية النسخ المسبق تقوم في الطور الاول بنقل كامل الذاكرة للآلة الافتراضية و من ثم تقوم بإرسال التغييرات في صفحات الذاكرة، يظهر المخطط أن الخوارزمية المعدلة قد قللت من حجم صفحات الذاكرة المنقولة خلال عملية الترحيل و بالتالي قللت حجم البيانات الإضافية المنقولة و خفضت الزمن الكلي للهجرة الحية ، وبمقارنة النتائج مع السيناريو السابق نلاحظ ازدياد فعالية الخوارزمية المعدلة كلما زاد حجم الصفحات المتغيرة في الذاكرة. فعندما كان عدد الطلبات 64 طلب و

حجم الملف 6 MB نلاحظ تحسن الزمن الكلي للهجرة الحية بمقدار 18 ثانية في حين نلاحظ تحسن هذا الزمن بمقدار 27 ثانية عندما كان عدد الطلبات 144 و حجم الملف 6 MB.



الشكل (6) حجم البيانات المنقولة عبر الشبكة أثناء ترحيل الآلة الافتراضية مع 144 طلب و حجم ملف 6 MB

الاستنتاجات والتوصيات:

إن استخدام خوارزمتنا المعدلة AD-pre-copy قد قلل من حجم صفحات الذاكرة المنقولة خلال عملية الترحيل و بالتالي قلل حجم البيانات الإضافية المنقولة عبر الشبكة مما أدى إلى تحسين الزمن الكلي للهجرة الحية .لاحظنا من خلال السيناريو الأول و الثاني زيادة فعالية الخوارزمية المعدلة و ذلك كلما زاد حجم الصفحات المتغيرة في الذاكرة.

References:

- [1] PANKAJDEEP KAUR AND ANITA RANI CSE DEPARTMENT GNDU, RC, JALANDHAR, *Virtual Machine Migration in Cloud Computing* , International Journal of Grid Distribution Computing. Vol. 8, No.5, 2015, pp.337-342.
- [2] ERIK GUSTAFSSON, *Optimizing Total Migration Time in Virtual Machine Live Migration*, Examensrbete 30 hp Mars 2016.
- [3] AHMAD SAKER AHMAD ,HAIDER KHALIL, *Improve the total migration time of live migration in virtualization data centers* ,k,ISSN:2079-3081,vol.39,NO.3,2018.
- [4] Pankaj Lathar, Yudhvir Singh, Girish Kumar Sharma, *Memory Virtualization Technique for Efficient Access of Data Resources in Cloud Environment*, International Journal of Computer Applications (0975 – 8887) Volume 99– No.10, August 2014.
- [5] Umesh Bellur and Prof. Purushottam Kulkarni, Department of Computer Science and Engineering Indian Institute of Technology Bombay, *Performance Modeling of Virtual Machine Live Migration*, March 13, 2015.

- [6] JANNARM AND T. KAEWKIRIYA, *Framework of Dynamic Resource Allocation System for Virtual Machine in Virtualization System*, International Journal of Computer Theory and Engineering, Vol. 8, No. 4, August 2017.
- [7] JANNARM AND T. KAEWKIRIYA, *Framework of Dynamic Resource Allocation System for Virtual Machine in Virtualization System*, International Journal of Computer Theory and Engineering, Vol. 8, No. 4, August 2016.
- [8] KASSEM KABALAN ,HAIDER KHALIL,*Performance evaluation of virtual machine manager by using DRBD as a shared storage of virtual disk*,ISSN:2079-3081,vol.39,NO.3,2017.
- [9]MOSTAFANOSHY,ABDELHAMEEDIBRAHIM,HESHAMARAFATALI,*Optimization of live virtual machine migration in cloud computing: A survey and future direction* Journal of Network and Computer Applications, Volume 110, 15 May 2018 .
- [10] MONCHAI BUNYAKITANON,MENGYUAN PENG, *Performance Measurement of Live Migration Algorithms*, Master Thesis Electrical Engineering September 2014.
- [11] SARASWATHI AT A, KALAASHRI.Y.RA B, DR.S.PADMAVATHI C1, *Dynamic Resource Allocation Scheme in Cloud Computing*, Procedia Computer Science 47 30 – 36 , 2015.
- [12] G. SUNITHA REKHA, *A Study On Virtualization And Virtual Machines*, International Journal of Engineering Science Invention (IJESI), Volume 7 Issue 5 Ver. III , May 2018.