

## تحسين كفاءة التجميع الديناميكي للآلات الافتراضية في مراكز البيانات السحابية

د. رضوان صالح دندة\*

د. قاسم قبلان\*\*

حيان رجب\*\*\*

(تاريخ الإيداع 10 / 8 / 2020. قُبِلَ للنشر في 7 / 1 / 2021)

### □ ملخص □

يعتبر الاستخدام غير الكفاء للموارد واحداً من أهم الأسباب التي تقف وراء الاستهلاك الضخم للطاقة في مراكز البيانات السحابية. ولحل هذه المشكلة قدم الباحثون تقنية التجميع الديناميكي للآلات الافتراضية، والتي تهدف إلى تجميع الآلات الافتراضية على أقل عدد ممكن من المضيفات (الآلات الفيزيائية). وعلى أية حال، فإن التجميع المكثف للآلات الافتراضية يؤدي إلى زيادة عدد الآلات الافتراضية المهجرة، وإلى مضيفات تعاني من الحمل الزائد. وهذا ما ينعكس بدوره على جودة الخدمة للتطبيقات العاملة ضمن الآلة الافتراضية. لذلك من المهم أن تتم الموازنة بين ضمان جودة الخدمة وتوفير الطاقة.

نقدم في هذا البحث خوارزمية التجميع الديناميكي المتكيفة الاستباقية ( Proactive Adaptive Dynamic Consolidation ) والتي تقوم بتخفيض كمية الطاقة المستهلكة مع المحافظة على مستويات الأداء المطلوبة في مركز البيانات السحابي.

قمنا بإجراء تقييم تجريبي باستخدام أداة المحاكاة Cloudsim لاختبار فعالية الخوارزمية المقدمة مع مسارات حمل من العالم الحقيقي ومقارنة أدائها مع مجموعة من الأعمال السابقة المقدمة في هذا المجال والتي تشمل الخوارزميات التالية: LR/MMT/SM/MFPED، LR/MMT/SM/Shi-AC، LR/MMT/PA/RUA، LR/MMT/SM/MBFD، ESS وقد أظهرت النتائج بأن الخوارزمية المقترحة تتفوق على هذه الخوارزميات من حيث استهلاك الطاقة وضمان جودة الخدمة، وعدد الآلات الافتراضية المهجرة.

**الكلمات المفتاحية:** مراكز البيانات السحابية ، كفاءة الطاقة ، اتفاقيات مستوى الخدمة.

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

[radwan.dandah@tishreen.edu.sy](mailto:radwan.dandah@tishreen.edu.sy)

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

[kasem.kabalan@tishreen.edu.sy](mailto:kasem.kabalan@tishreen.edu.sy)

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

[hayyanmohammadrajab@tishreen.edu.sy](mailto:hayyanmohammadrajab@tishreen.edu.sy)

## Improving the Efficiency of Dynamic Virtual Machines Consolidation in Cloud Data Centers

Dr. Radwan Saleh Dandah<sup>\*</sup>  
Dr. Kasem Kabalan<sup>\*\*</sup>  
Hayyan Rajab<sup>\*\*\*</sup>

(Received 10 / 8 / 2020. Accepted 7 / 1 / 2021)

### □ ABSTRACT □

Inefficient resource usage is one of the greatest causes of high energy consumption in cloud data centers. To solve the problem, researchers presented Dynamic consolidation of virtual machines technique, which aim to consolidate VMs in minimum number of hosts (physical machines). However, aggressive consolidation of VMs leads to an increase in the number of migrated VMs, and hosts suffering from Overloading. This in turn affects the quality of service (QoS) of the applications running in the VMs. So, it is important to balance between guaranteeing Quality of Service (QoS) and saving energy.

In this research, we introduce Proactive Adaptive Dynamic Consolidation Algorithm to reduce energy consumption while maintaining the required performance levels in a cloud data center.

We have performed experimental evaluation on Cloudsim toolkit to validate the effectiveness of the proposed algorithm with real-world workloads, and compare its performance against a collection of many previous works in this field which includes the following algorithms: LR/MMT/SM/MBFD, LR/MMT/PA/RUA, LR/MMT/SM/Shi-AC, LR/MMT/SM/MFPED, EES. The simulation results showed that the suggested algorithm outperforms these algorithms regarding energy consumption, QoS guarantees, and the number of VM migrations

**Keywords:** Cloud Datacenters, energy efficiency, service level agreement

---

<sup>\*</sup> Professor, Department of Systems and Computer Networks, Faculty of Information Engineering, Tishreen University, Lattakia, Syria. radwan.dandah@tishreen.edu.sy

<sup>\*\*</sup> Assistant Professorr, , Department of Systems and Computer Networks, Faculty of Information Engineering, Tishreen University, Lattakia, Syria. kasem.kabalan@tishreen.edu.sy

<sup>\*\*\*</sup> Postgraduate Student (Ph.d.), Department of Systems and Computer Networks, Faculty of Information Engineering, Tishreen University, Lattakia, Syria.  
hayyanmohammadrajab@tishreen.edu.sy

**مقدمة:**

ظهرت الحوسبة السحابية (Cloud Computing) خلال السنوات الماضية كحقل جديد في مجال تكنولوجيا المعلومات، ووفقاً للتعريف الذي قدّمه المعهد الوطني للمعايير والتكنولوجيا (National Institute of standard and Technology) تعتبر الحوسبة السحابية نموذجاً يمكن المستخدمين من النفاذ حسب الطلب ومن أي مكان على شبكة الإنترنت إلى تجمّع مشترك من الموارد الحاسوبية والتي يمكن توفيرها بسرعة ومرونة وبحد أدنى من الجهد الإداري والتفاعلي مع مزود الخدمة [1].

تستخدم مراكز البيانات تقنية المحاكاة الافتراضية لتمكين وجود أكثر من آلة افتراضية (Virtual Machine) على نفس الآلة الفيزيائية. حيث أن هذه القدرة على التخصيص تحقق الاستثمار الأمثل للموارد الفيزيائية المتوفرة وهو ما يحقق عائدات أعلى لمزود الخدمة [2].

قاد التوسع الكبير في بناء مراكز البيانات حول العالم إلى زيادة كبيرة في الطاقة الكهربائية المستهلكة، وقد بيّنت دراسة منشورة مؤخراً إلى إن مراكز البيانات قد استهلكت حوالي 205 تيرا وات ساعي خلال العام 2018 وهو ما يمثل حوالي 1% من الاستهلاك العالمي للكهرباء وزيادة بنسبة 6% مقارنة بالعام 2010 [3]. وهذا يعني ارتفاع كلفة النفقات التشغيلية بالنسبة إلى مزودي الخدمة السحابية. وكنتيجة لما سبق تعتبر قضية توفير الطاقة واحدة من أكثر القضايا التي تم البحث فيها في مجال الحوسبة السحابية.

تشير الأبحاث في هذا المجال إلى أن السبب الرئيسي في هذا الاستهلاك الضخم لا يعود فقط للنمو في أعداد وأحجام مراكز البيانات، وإنما كنتيجة للاستثمار غير الفعّال للموارد الموجودة ضمنها [4]. كما وجد الباحثون أن متوسط نسبة استخدام وحدة المعالجة المركزية كان أقل من 50% في العديد من مراكز البيانات التي تمت دراسته [5, 6]. علاوة على ذلك، فإن المضيفات الخاملة التي لا تستضيف أية آلات افتراضية تستهلك تقريباً حوالي 70% من ذروة استهلاكها (عندما تكون محمّلة بشكل كامل)، وهذا ما سنناقشه بالتفصيل لاحقاً في هذا البحث. لذلك فإن تواجد المضيفات في حالة حمل منخفض هو أمر غير مرغوب من وجهة نظر استهلاك الطاقة.

من جهة أخرى وكنتيجة للأحمال المتغيرة في التطبيقات المشغّلة على الآلات الافتراضية، فإن بقاء المضيفات في حالة حمل زائد (Overloaded) يقود إلى عدم حصول الآلات الافتراضية على الموارد الكافية لحاجتها، مما ينعكس بشكل سلبي على الأداء ويؤدي إلى خرق اتفاقيات مستوى الخدمة (Service Level Agreement (SLA)) المتفق عليها بين مزود ومستهلك الخدمة مما يترتب خسائر مالية على مشغل مركز البيانات [7].

يعتبر التجميع الديناميكي للآلات الافتراضية إحدى الطرق الفعّالة المستخدمة لتخفيض استهلاك الطاقة مع المحافظة بنفس الوقت على متطلبات جودة الخدمة باستخدام تقنية الهجرة الحية (live migration) حيث تقوم بتجميع الآلات الافتراضية على أقل عدد ممكن من المضيفات وفقاً لمتطلباتها الحالية من الموارد دون التوقف عن العمل. يتم حل تخفيض مشكلة استهلاك الطاقة عن طريق إيقاف المضيفات الخاملة أو منخفضة الاستخدام، في حين يتم المحافظة على متطلبات الأداء عن طريق تهجير الآلات الافتراضية من المضيفات ذات الحمل الزائد إلى مضيفات في وضع استخدام طبيعي [8]. على أية حال يمكن أن يقودنا التجميع المكثف للآلات الافتراضية إلى حدوث اختراقات لاتفاقيات مستوى الخدمة نتيجة زيادة متطلبات التطبيقات من الموارد مع مرور الزمن.

بناء على كل ما سبق تعتبر الإدارة الفعّالة للموارد قضية جوهرية في البيئات السحابية، كما أنها تضع مزودي الخدمة أمام تحدّي الموازنة ما بين العمل على توفير الطاقة دون أن ينعكس ذلك سلباً على الأداء.

## أهمية البحث وأهدافه:

تعتبر الموازنة بين استهلاك الطاقة وجودة الخدمة إحدى أبرز المشاكل التي يواجهها مزودو الخدمة في مراكز البيانات السحابية، وتكمن أهمية البحث في اقتراح خوارزمية تعمل على توفير الطاقة دون الإخلال بمتطلبات جودة الخدمة، وذلك بالاعتماد على تقنيات التنبؤ بالسلاسل الزمنية بهدف تلبية ليس فقط المتطلبات الحالية وإنما المتطلبات المستقبلية للآلات الافتراضية من الموارد لتفادي تهجيرها في وقت لاحق إلى مخدمات أخرى تُلبّي هذه المتطلبات. كما أنها تأخذ بعين الاعتبار عدم التجانس في مواصفات المضيفات الموجودة في مركز البيانات والذي ينعكس بدوره على كمية الطاقة المستهلكة مقابل الأداء الذي تقدّمه.

يهدف هذا البحث إلى دراسة الخوارزمية المقترحة، ومقارنة أدائها مع العديد من الخوارزميات المقدمة سابقاً، وذلك وفقاً لمجموعة من المعايير المعتمدة في هذا المجال.

## طرائق البحث ومواده:

لتحقيق هدف البحث، تمّ اتباع المنهجية التالية:

✓ دراسة نظرية لبيئة مركز البيانات السحابي والنماذج الرياضية المستخدمة في حساب كمية استهلاك الطاقة ومعايير خرق اتفاقيات جودة الخدمة.

✓ دراسة لمجموعة من الأعمال السابقة المقدمة في هذا المجال مع تحليل نقاط قوتها وضعفها بهدف تصميم خوارزمية قوية قادرة على تحقيق أهداف البحث.

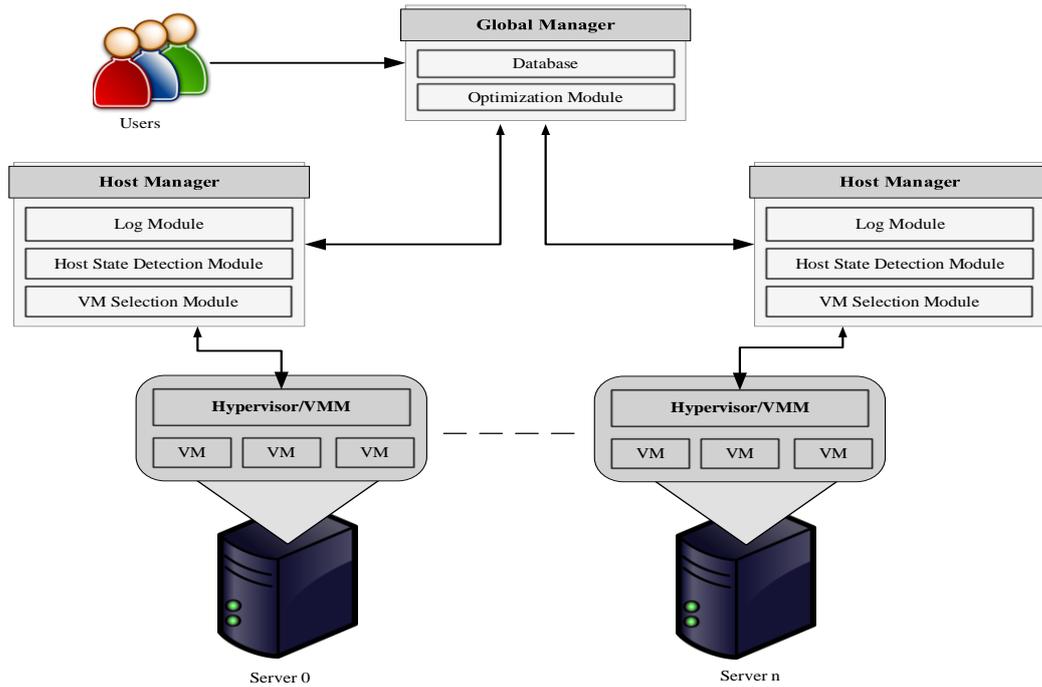
✓ اعتماد المحاكاة الحاسوبية للتحقق من أداء الخوارزمية المقترحة، ثمّ القيام بمناقشة النتائج، وصياغة الاستنتاجات.

### 1- الدراسة النظرية:

#### 1-1 نموذج النظام:

يتألف مركز البيانات من  $N$  عقدة فيزيائية غير متجانسة. كل عقدة تتميز بقدرة معالجة مقاسة بوحدة مليون تعليمة في الثانية ((Millions Instructions Per Second (MIPS)، حجم ذاكرة RAM، عرض حزمة شبكة. ونفترض عدم وجود أقراص محلية في العقد الفيزيائية، وإنما يتم استخدام تخزين شبكي لتمكين عملية الهجرة الحية للآلات الافتراضية [9]. في هذه البيئة نحن لا نملك معلومات مسبقة عن حمل التطبيقات والوقت الذي تتواجد فيه  $M$  آلة افتراضية تم طلبها من قبل العديد من المستخدمين والتي تتصف بمواصفات محددة من حيث قدرة المعالجة وحجم الذاكرة RAM وعرض الحزمة. يتم تأسيس اتفاقية مستوى الخدمة (SLA) بين المستخدم ومشغل مركز البيانات بحيث يترتب على المشغل غرامة في حال خرق هذه الاتفاقية، حيث يحدث هذا الخرق عندما لا تستطیع العقدة الفيزيائية تلبية متطلبات الآلات الافتراضية الموجودة عليها من الموارد.

يوضح الشكل (1)[10] نموذج النظام المستخدم والذي يتكون من جزأين هما المدير العام (the global manager) ومدير المضيف (host manager). يوجد مدير المضيف في كل عقدة فيزيائية ويقوم بالتفاعل مع كل من مراقب الآلة الافتراضية (VM Monitor (VMM) المسؤول عن عملية التغيير الفعلي لحجم الآلة الافتراضية بالإضافة للقيام بعملية التهجير، والمدير العام الذي يتواجد على عقدة رئيسية (master node) ويقوم بالحصول على المعلومات من قبل مدراء المضيفات وهو المسؤول عن تحسين توضع الآلات الافتراضية بناء على المعلومات التي جمعها.



الشكل (1) نموذج النظام

يتألف مدير المضيف من الوحدات التالية

- وحدة السجل Log Module: وهي الوحدة المسؤولة عن مراقبة وتسجيل قيمة الحمل الموجود على المضيف.
  - وحدة تحديد حالة المضيف Host State Detection Module: وهي المسؤولة عن تحديد فيما إذا كان المضيف محملاً بشكل زائد (overloaded) أو في وضع حمل طبيعي (normal) بناءً على المعلومات الموجودة في وحدة السجل.
  - وحدة اختيار الآلات الافتراضية VM Selection Module: وهي المسؤولة عن تحديد الآلات الافتراضية التي يجب تهجيرها وفقاً لاستراتيجية الاختيار المتبعة.
- يتألف المدير العام من الوحدات التالية:
- وحدة قاعدة البيانات Database Module: والتي تقوم بتخزين المعلومات عن المضيفات الموجودة في مركز البيانات بما فيها نسبة استخدام الموارد الموجودة في كل مضيف.
  - وحدة تحسين التوضع Placement Optimization Module: وهي الوحدة المسؤولة عن إطلاق عملية تهجير الآلات الافتراضية بهدف تحسين توزيعها وتلبية احتياجاتها من الموارد.

### 1-2 نموذج الطاقة:

تشكل الطاقة المستهلكة من قبل المخدمات حوالي 43% من إجمالي الطاقة المستهلكة في مركز البيانات [11]، وتشير الدراسة [12] إلى وجود علاقة خطية بين الطاقة المستهلكة و نسبة استخدام وحدة المعالجة المركزية. وقد تم قياس كمية الطاقة المستهلكة نسبة إلى حمل وحدة المعالجة المركزية في البحث [13] حيث تم استخدام مخدمين للقيام بعملية القياس:

- HP ProLiant G4 بالمواصفات: معالج ثنائي النوى بسرعة 1.86 GHZ، وذاكرة وصول عشوائية 4 G
- HP ProLiant G5 بالمواصفات: معالج ثنائي النوى بسرعة 2.66 GHZ، وذاكرة وصول عشوائية 4 G

يظهر الجدول (1) كمية استهلاك الطاقة مقدره بالواط (watt) لكلا المخدمين مع مستويات مختلفة لحمل وحدة المعالجة المركزية.

الجدول (1) : الطاقة الكهربائية المستهلكة مع وجود أحمال مختلفة على المخدم [13]

Server Type	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProLiant G4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant G5	93.7	97	101	105	110	116	121	125	129	133	135

تشير النتائج إلى أن المخدم الخامل يستهلك بشكل تقريبي حوالي 70% من كمية الطاقة التي يستهلكها عندما يكون محملاً بشكل كامل. وبناء عليه يتم تعريف الطاقة المستهلكة كتابع لاستخدام وحدة المعالجة المركزية كما توضح لنا العلاقة (1) [14]:

$$P(u) = k \cdot P_{max} + (1 - k) \cdot P_{max} \cdot u = P_{max} \cdot (0.7 + 0.3 \cdot u) \quad (1)$$

حيث  $P_{max}$  يُمثل كمية الطاقة المستهلكة في حال حمل وحدة معالجة مركزية 100%،  $k$  هو معامل الطاقة المستهلكة من قبل المخدم الخامل،  $u$  هو نسبة استخدام وحدة المعالجة المركزية. ولأن هذه النسبة تتغير عبر الزمن نتيجة لتغير الحمل فإننا نعتبرها كتابع للزمن  $u(t)$  وعليه تكون الكمية الكلية للطاقة المستهلكة من قبل المخدم معرفة وفق العلاقة (2) [14]:

$$E = \int_{t_0}^{t_1} P(u(t)) dt \quad (2)$$

وعليه فإنه يتم تحديد الطاقة التي يستهلكها المخدم كتابع لنسبة استخدام وحدة المعالجة المركزية. وكنتيجه لما سبق فإنه ولتخفيض الطاقة المستهلكة في مركز البيانات، تقوم معظم الأبحاث في هذا المجال بالتركيز على وحدة المعالجة المركزية كمورد عند مناقشة خوارزميات قضايا التجميع الديناميكي للآلات الافتراضية.

### 1-3- كلفة تهجير الآلات الافتراضية:

للهجرة الحية تأثيرها السلبي على أداء التطبيقات المشغلة ضمن الآلة الافتراضية. في [15] تم إجراء دراسة تجريبية لتقييم هذا التأثير ومحاولة نمذجته. وقد وجد الباحثون أن انخفاض الأداء ووقت التوقف يعتمد على سلوك التطبيق الذي يتم تشغيله، بمعنى عدد صفحات الذاكرة التي يقوم التطبيق بتحديثها أثناء تنفيذه. على أية حال فإنه من أجل صف تطبيقات الوب فإن انخفاض الأداء الوسطي بما فيه زمن التوقف يمكن تخمينه تقديرياً بحوالي 10% من استخدام وحدة المعالجة المركزية. وهذا يعني أن كل تهجير قد يسبب خرقاً لاتفاقية مستوى الخدمة، لذلك فإن هناك حاجة لتخفيض عدد التهجيرات.

حيث يعتمد الزمن اللازم للتهجير على كمية الذاكرة الكلية المستخدمة من قبل الآلة الافتراضية وعلى عرض الحزمة المتوفر. لذلك وبناء على كل ما سبق نستطيع تعريف انخفاض الأداء الذي تتعرض له الآلة الافتراضية وفق العلاقة [13](3):

$$U_{d_j} = 0.1 \cdot \int_{t_0}^{t_0+T_{m_j}} u_j(t) dt, T_{m_j} = \frac{M_j}{B_j} \quad (3)$$

حيث:  $U_{d_j}$  هي انخفاض الأداء الكلي للآلة الافتراضية المهجرة  $VM_j$ ،  $t_0$  يمثل الزمن الذي بدأ فيه التهجير،  $T_{m_j}$  يمثل الزمن المستغرق لإنهاء التهجير،  $u_j(t)$  هو نسبة استخدام وحدة المعالجة المركزية من قبل  $VM_j$ ،  $M_j$  هو كمية الذاكرة المستخدمة بواسطة الآلة الافتراضية،  $B_j$  هو عرض الحزمة المتوفرة.

**1-4 مقاييس اختراق اتفاقيات مستوى الخدمة:**

إن تلبية متطلبات جودة الخدمة مهمة جداً في بيئة الحوسبة السحابية. حيث يتم تعريف هذه المتطلبات وفق اتفاقيات جودة خدمة (SLAs) والتي من الممكن أن تحدد من خلال أقل إنتاجية مقدمة أو زمن استجابة أعظمي. ولكن هذه المقاييس ترتبط بسلوك التطبيقات لذلك من الضروري تعريف معيار مستقل عن الحمل والذي يمكن استخدامه لتقييم مدى الالتزام باتفاقيات جودة الخدمة المقدمة لأي آلة افتراضية موجودة في بيئة الحوسبة السحابية. لذلك ووفقاً للتجارب العملية تم افتراض أن الاتفاقيات تكون محققة عندما يتم توفير 100% من كمية الموارد المطلوبة من الآلة الافتراضية لتلبية احتياجات التطبيقات المشغلة عليها في أي لحظة. وهذا الشيء يمكن أن لا يتحقق نتيجة لعاملين [13]:

- زمن خرق اتفاقيات مستوى الخدمة في المضيفات النشطة ( SLA violation Time per Active Host ) ((SLATAH)): وهي نسبة الوقت الذي يكون فيه المضيف النشط في حالة تحميل كامل (أي نسبة استخدام وحدة المعالجة المركزية 100%) وهو في هذه الحالة لن يستطيع تلبية أي متطلبات إضافية للآلة الافتراضية التي تطلب كمية جديدة من مورد وحدة المعالجة المركزية ويعطى هذا المعيار وفق العلاقة (4):

$$SLATAH = \frac{1}{N} \sum_{i=1}^N \frac{T_{s_i}}{T_{a_i}} \quad (4)$$

حيث  $N$  هي عدد المضيفات،  $T_{s_i}$  هو الزمن الكلي الذي يكون فيه المضيف  $i$  في حال استخدام كُلي لوحد المعالجة المركزية،  $T_{a_i}$  هو الزمن الكلي الذي يكون فيه المضيف  $i$  في حالة نشاط (أي يستضيف آلات افتراضية).

- انخفاض الأداء نتيجة عمليات التهجير ((Performance Degradation due to Migrations (PDM)) والذي تحدده العلاقة (5):

$$PDM = \frac{1}{M} \sum_{j=1}^M \frac{C_{d_j}}{C_{r_j}} \quad (5)$$

حيث  $M$  هو عدد الآلات الافتراضية،  $C_{d_j}$  هو تقدير لانخفاض الأداء للآلة الافتراضية  $j$  بسبب التهجير،  $C_{r_j}$  هو السعة الكلية التي تطلبها الآلة الافتراضية خلال حياتها.

يتم دمج المعيارين السابقين في معيار واحد يسمى خرق اتفاقيات مستوى الخدمة (SLA Violation) ( SLAV ) ويعطى وفق المعادلة (6):

$$SLAV = SLATAH * PDM \quad (6)$$

**1-5 خوارزميات التجميع الديناميكي للآلات الافتراضية:**

عند تصميم الخوارزمية يجب أن تتم الإجابة على مجموعة من الأسئلة تحدد بمجملها كيفية إدارة الموارد في بيئة الحوسبة السحابية [16]:

1. متى يمكن اعتبار المضيف مُحملاً بشكل زائد؟ وهذا يقودنا إلى ضرورة تحديد عتبة نسبة استخدام عليا مسموح بها، يجب عند تجاوزها القيام بعملية تهجير للآلات الافتراضية من هذا المضيف إلى مضيفات أخرى. وهذه العتبة إما أن تكون ساكنة (Static Threshold) وذات قيمة ثابتة لكل أنواع المضيفات [14]، أو يتم تحديد عتبة ثابتة لكل نوع بناء على معيار نسبة الأداء إلى الطاقة ((Performance-to-Power Ratio (PPR)) والذي يمثل عدد العمليات التي يستطيع المضيف إنجازها خلال فترة زمنية معينة مقابل متوسط الطاقة المستهلكة في نفس الفترة، وهو ما يمثل مدى كفاءة استهلاك الطاقة على هذا المضيف [17]. في حين قامت بعض الدراسات بتقديم عتبات متكيفة بعضها يعتمد على الأساليب الإحصائية التي تحلل البيانات السابقة لحمل المضيف مثل وسيط الانحراف

المطلق ((Median Absolute Deviation (MAD)) والمدى الربيعي ((interquartile range (IQR))، في حين اعتمد بعضها الآخر على تقنيات التنبؤ بالسلاسل الزمنية مثل الانحدار المحلي ((Local Regression (LR)) والانحدار المحلي المتين ((Local Regression Robust (LRR)) للتنبؤ فيما إذا كانت متطلبات الآلات الافتراضية من الموارد سوف تتجاوز السعة الكلية للموارد المتوفرة في المضيف [13].

2. ماهي الآلات الافتراضية التي يتم اختيارها للتهجير؟ وتبرز في هذا المجال سياسة أقل زمن للتهجير (The Minimum Migration Time (MMT)) والتي تعتمد على حجم الذاكرة المخصصة للآلة الافتراضية وعرض الحزمة المتوفر في المضيف [13].

3. متى يمكننا اعتبار المضيف ذا حمل خفيف؟ وهنا يتوجب علينا نقل جميع الآلات الافتراضية من هذا المضيف ومن ثم وضعه في حالة سبات أو حتى إيقافه عن العمل. تعتمد بعض الخوارزميات على اختيار مضيف واحد للدخول في حالة السبات وهي السياسة المتبعة في [13] حيث يتم اختيار المضيف الذي لديه أقل نسبة استخدام لوحدة المعالجة المركزية ((Simple Method (SM))، أو المضيف الذي لديه أقل قيمة لحاصل قسمة الطاقة المستهلكة على عدد الآلات الافتراضية التي يستضيفها ((Power-Aware (PA)) [18]. في حين تعتمد خوارزميات أخرى على تحديد عتبة دنيا لنسبة استخدام وحدة المعالجة المركزية واعتبار أي مضيف لديه نسبة استخدام أقل من هذه العتبة في حالة حمل منخفض، وهذه العتبة إما أن تكون ساكنة كما في [14] أو ديناميكية ويتم تحديد قيمتها بناء على عدد المضيفات من نفس النوع، قيمة عتبتها العليا، المجموع الكلي لاستخدام وحدة المعالجة المركزية كما في [17].

4. ماهي الطريقة المتبعة في تحديد المضيفات التي سوف نقوم بتهجير الآلات الافتراضية إليها؟ وتمثل هذه الطريقة سياسة توضع الآلات الافتراضية التي تتبعها الخوارزمية. وفي هذا المجال تبرز لدينا :

✓ سياسة خوارزمية المناسب الأفضل المعدلة التنازلية ((Modified Best Fit Decreasing (MBFD))، والتي تعتمد على ترتيب الآلات الافتراضية تنازلياً وفقاً لنسبة استخدام وحدة المعالجة المركزية ومن ثم تخصيص الآلة الافتراضية للمضيف الذي تسبب له أقل زيادة في كمية الطاقة المستهلكة [14].

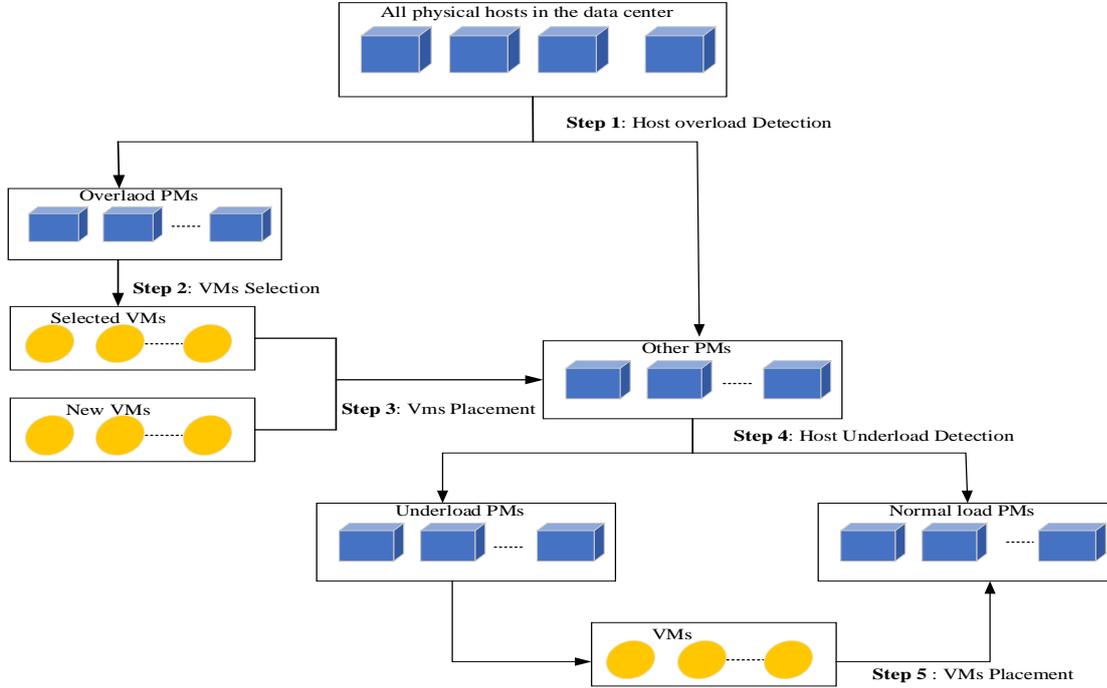
✓ السياسة المدركة للاستخدام المتبقي ((Remaining Utilization-Aware (RUA)) والتي تقوم على حجز نسبة ثابتة (17%) من سعة وحدة المعالجة المركزية في كل مضيف لا يتم تخصيصها لأي آلة افتراضية لتجنب وقوع المضيف في وضع الحمل الزائد مع تغير الحمل في المستقبل، ومن ثم محاولة نقل الآلة الافتراضية إلى المضيف الذي يلبي احتياجاتها من مورد وحدة المعالجة المركزية مع تحقيق أكبر قدر ممكن من عدالة التخصيص بين الآلات الافتراضية الموجودة عليه [18].

✓ سياسة السعة المطلقة ((Shi-AC (Absolute Capacity)) والتي تعتمد على ترتيب المضيفات تنازلياً حسب قدرة وحدة المعالجة المركزية فيها، ومن ثم تخصيص الآلة الافتراضية لأول مضيف يلبي احتياجاتها [19].

✓ سياسة المناسب المتوسط الفعالة بالنسبة للطاقة المتناقصة (Medium-Fit Power-Efficient (Decreasing (MFPE)) والتي تعتمد على ترتيب الآلات الافتراضية بشكل تنازلي حسب نسبة استخدام وحدة المعالجة، ومن ثم تخصيصها للمضيف الذي تكون القيمة المطلقة للفرق ما بين نسبة استخدام وحدة المعالجة المركزية وقيمة ثابتة (60%) أقل ما يمكن، وفي حال التساوي بين مضيفين يتم اختيار المضيف الأكثر كفاءة بالنسبة للطاقة ((Power Efficiency)) والذي يتم تحديده بناتج قسمة السعة المطلقة لوحدة المعالجة المركزية على القيمة العظمى للطاقة المستهلكة والمعرف وفق العلاقة (7)[20]:

$$PE = CPU_{total}/Power_{max} \quad (7)$$

يوضح الشكل (2) [21] الخطوات التفصيلية التي تتبعها خوارزميات التجميع الديناميكي والتي يتم تنفيذها بشكل دوري.



الشكل(2) : مخطط خوارزمية التجميع الديناميكي للآلات الافتراضية

## 2- الخوارزمية المقترحة:

بناء على دراستنا المرجعية وتحليلنا لأداء الخوارزميات المقدمة في مجال التجميع الديناميكي للآلات الافتراضية قمنا بتصميم خوارزمية جديدة باسم خوارزمية التجميع الديناميكي المتكيفة الاستباقية ( Proactive Adaptive Dynamic Consolidation) وفيما يلي شرح للسياسات التي تتبعها الخوارزمية المقترحة والتي يتم تشغيلها بشكل دوري (كل 5 دقائق) في محاولة لتخفيض استهلاك الطاقة مع الحفاظ على جودة الخدمة في مراكز البيانات السحابية:

### 1-2 سياسة تحديد المضيفات ذات الحمل الزائد :

يعتبر تحديد الحمل الزائد في المضيفات الخطوة الأولى في خوارزمية التجميع الديناميكي للآلات الافتراضية. بالنسبة لخوارزمتنا المقدمة فإن اعتبار المضيف في حال حمل زائد يكون نتيجة أحد احتمالين:

- أن تتجاوز نسبة استخدام وحدة المعالجة المركزية عتبة تم تحديدها مسبقاً لكل نوع من المضيفات والتي تعبر عن أعلى قيمة للأداء مقابل الطاقة المستهلكة بشكل مشابه للمنهج المستخدم في [17] والذي يعتمد على معيار نسبة الأداء إلى الطاقة<sup>1</sup>، وبهذه الطريقة نحافظ على المضيفات في وضع تشغيل مثالي.

- أن تتجاوز القيمة المستقبلية لنسبة استخدام وحدة المعالجة المركزية في المضيف القيمة 1 (100%). وهنا نحن نتنبأ بأن المضيف لن يستطيع في المستقبل تلبية احتياجات الآلات الافتراضية لذلك نقوم بتهجير الآلات

<sup>1</sup> يمكن الحصول على قيمة نسبة الأداء مقابل الطاقة (PPR) لمختلف أنواع المخدمات ، من الرابط التالي التابع لمؤسسة الأداء والتقييم الأساسية [https://www.spec.org/power\\_ssi2008/results](https://www.spec.org/power_ssi2008/results)

الافتراضية مسبقاً. وقد قمنا باختيار تقنية الانحدار الذاتي والمتوسطات المتحركة التكاملية (Autoregressive Integrated Moving Average (ARIMA) للتنبؤ؛ إذ أن هذه التقنية قد أثبتت كفاءة كبيرة في مجال التنبؤ بالأحمال المستقبلية في البيئات السحابية، حيث أظهرت الدراسة [22] أن دقة التنبؤ تصل إلى نسبة 91%. بناء على القيم الموجودة في وحدة السجل والتي تقوم بتسجيل نسبة استخدام وحدة المعالجة المركزية في المضيف خلال فواصل زمنية متساوية يمكن أن يصبح نموذج ARIMA قابلاً للتطبيق.

باستخدام هذه الطريقة نضمن بأن المضيف سوف يعمل في الوقت الحالي بأفضل مستوى من الأداء مقابل الطاقة، وسيكون في أسوأ الأحوال قادراً على تلبية جميع المتطلبات المستقبلية لآلاته الافتراضية (يتجنب خرق اتفاقيات مستوى الخدمة) ريثما يتم تشغيل الخوازمية من جديد وإعادةه إلى وضع التشغيل المثالي. يبين الشكل (3) الشيفرة المزيفة لهذه السياسة.

Host Overload Detection Policy	
<b>Input:</b>	hostList
<b>Output:</b>	overloadedHostList
1	overloadedHostList ← NULL;
2	<b>foreach</b> host in hostList <b>do</b>
3	Utilization ← host.getUtilizationOfCpu();
4	$T_u$ ← host.getLevelOfObservedMaxPPR();
5	data[] ← host.getUtilizationHistory();
6	futureUtilization ← getHostFutureUtilization(data);
7	<b>if</b> ((Utilization ≥ $T_u$ ) <b>or</b> (futureUtilization ≥ 1)) <b>then</b>
8	overloadedHostList.add(host);
9	<b>end if</b>
10	<b>end foreach</b>
11	<b>return</b> overloadedHostList;

الشكل(3) الشيفرة المزيفة لسياسة خوازمية PADC في تحديد المضيفات ذات الحمل الزائد [الباحث]

## 2-2 سياسة اختيار الآلات الافتراضية:

تتضمن الخطوة التالية في الخوارزمية تحديد الآلات الافتراضية التي سوف يتم نقلها من المضيف لإعادةه إلى الوضع الطبيعي. يبين الشكل(4) الخطوات المتبعة في تحديد لائحة الآلات الافتراضية التي سوف يتم تهجيرها. حيث نقوم في كل مرة باختيار إحدى الآلات الافتراضية ومن ثم نقوم بفحص حالة المضيف فيما إذا لا زال في وضع حمل زائد. حيث يعتمد اختيارنا لهذه الآلة على المعيار التالي:

$$\text{Metric}_{vm} = \frac{vm_{\text{CurrentRequestedCPU}} - vm_{\text{CurrentAllocatedCPU}}}{vm_{\text{migrationTime}}} \quad (8)$$

وهو يمثل الفرق بين حجم مورد وحدة المعالجة المركزية المطلوب حالياً من قبل الآلة الافتراضية والحجم المخصص لها من قبل المضيف مقسوماً على الزمن اللازم لتهجير هذه الآلة. وكلما زادت قيمة هذا المعيار كلما كانت الآلة الافتراضية مرشحة للاختيار ويعود السبب في ذلك إلى أننا نبحث عن الآلة الافتراضية الأكثر حاجة للموارد والتي تشكل ضغطاً على المضيف وفي نفس الوقت لا تتطلب زمناً طويلاً للتهجير.

VM Selection Policy	
<b>Input:</b>	overLoadedHostList
<b>Output:</b>	vmsToMigrateList
1	vmsToMigrateList $\leftarrow$ NULL;
2	<b>foreach</b> host in overLoadedHostList <b>do</b>
3	<b>while</b> (host.isOverLoaded == TRUE) <b>do</b>
4	vmList $\leftarrow$ host.getVmList();
5	maxMetric $\leftarrow$ MIN;
6	vmToMigrate $\leftarrow$ NULL;
7	<b>foreach</b> vm in vmList <b>do</b>
8	metric $\leftarrow$ calculateMetric(vm);
9	<b>if</b> (metric > maxMetric) <b>then</b>
10	maxMetric $\leftarrow$ metric;
11	vmToMigrate $\leftarrow$ vm;
12	<b>end if</b>
13	<b>end foreach</b>
14	<b>if</b> (vmToMigrate $\neq$ NULL) <b>then</b>
15	vmsToMigrateList.add(vmToMigrate);
16	Host.removeVm(vm);
17	<b>end if</b>
18	<b>end while</b>
19	<b>end foreach</b>
20	<b>return</b> vmsToMigrateList;

الشكل (4) الشيفرة المزيقة لسياسة خوارزمية PADC في اختيار الآلات الافتراضية [الباحث]

## 2-3: سياسة توضع الآلات الافتراضية:

يتم تطبيق هذه السياسة الموضحة في الشكل (5) وفقاً لخوارزمية التجميع الديناميكي في مرحلتين: الأولى عند نقل الآلات الافتراضية من المضيفات ذات الحمل الزائد، والثانية عند محاولة تفريغ المضيفات ذات الحمل المنخفض من آلتها الافتراضية. ويكمن الفرق بين المرحلتين في المضيفات التي يتم استثنائها من اختيار التوضع الجديد لهذه الآلات. حيث يتم في المرحلة الأولى استثناء المضيفات ذات الحمل المرتفع، وفي الثانية يتم استثناء المضيفات الموجودة في حالة الإيقاف عن العمل أيضاً. أما طريقة اختيار المضيف الجديد فهي متماثلة في كلا المرحلتين. تعتبر هذه السياسة نسخة معدلة من خوارزمية المناسب الأول (First Fit)، حيث يتم في البداية ترتيب المضيفات وفق الترتيب التنازلي لكفاءة الطاقة والمعرفة وفق العلاقة (7) السابقة، وفي حال المساواة نقوم بمقارنة نسبة استخدام وحدة المعالجة المركزية (تتازلياً أيضاً)، كما نقوم بترتيب الآلات الافتراضية تصاعدياً وفق ما يسمى بفجوة وحدة المعالجة المركزية (CPU gap) والذي يمثل الفرق مابين السعة المستخدمة والسعة الكلية بدلاً من ترتيبها تنازلياً حسب نسبة الاستخدام ويعود السبب في ذلك إلى أن الآلات الافتراضية غير متجانسة (ذات ساعات مختلفة لوحدة المعالجة المركزية) وأن تكون آلة افتراضية ذات نسبة استخدام أعلى لا يعني بالضرورة أن تكون متطلباتها الإضافية المحتملة من المورد أقل قيمة من آلة افتراضية ذات نسبة استخدام أقل. ومن ثم نقوم بتخصيص الآلة الافتراضية لأول مضيف

يحقق متطلبات هذه الآلة من مورد وحدة المعالجة المركزية، وفي نفس الوقت لا يعاني من الحمل الزائد نتيجة لعملية التخصيص. وهذا يعني بأن قيمة نسبة استخدام وحدة المعالجة المركزية لن تتجاوز عتبة الأداء الأفضل مقابل الطاقة، ولن تجعله في المستقبل في حالة تحميل كامل (100%). تضمن هذه الطريقة بأن يتم إيصال المضيفات الأعلى نسبة استخدام إلى عتبات الأداء الأفضل مقابل الطاقة بأسرع وقت، بالإضافة إلى أن حجم الموارد الإضافية التي يمكن أن تطلبها الآلة أقل ما يمكن وهو ما يقلل من احتمال دخول المضيف في حالة الحمل الزائد.

#### VM Placement Policy

**Input:** hostList,excludedHosts,VmList

**Output:** mapping

```

1 mapping ← NULL;
2 hostList ← sort(hostList,PE,U,'Des');
3 VmList ← sort(vmList,CPUGap, 'ASC');
4 foreach vm in vmList do
5     allocatedHost ← NULL;
6     foreach host in hostList do
7         if excludedHosts.contains(host) then
8             continue;
9         end if
10        if (host.isSuitable(vm) and host.isNotOverLoadedAfterAllocation(vm)) then
11            allocatedHost ← host;
12            break;
13        end if
14    end foreach
15    if (allocatedHost ≠ Null ) then
16        mapping.add(vm,allocatedHost);
17    end if
18 end foreach
19 return mapping;
```

الشكل(5) الشيفرة المزيفة لسياسة خوارزمية PADC في تحسين توزيع الآلات الافتراضية [الباحث]

#### 4-2 سياسة تحديد المضيفات ذات الحمل المنخفض:

يتم تطبيق هذه السياسة إثر القيام بتوزيع الآلات الافتراضية المهجرة من المضيفات ذات الحمل الزائد، حيث يظهر الشكل (6) الخطوات المتبعة في هذه السياسة.

Host Underload Detection Policy	
<b>Input:</b>	hostList,excludedHosts,highLoadFlag
<b>Output:</b>	underLoadHostList
1	underLoadHostList ← NULL;
2	candinateHostList ← NULL;
3	<b>foreach</b> host in hostList <b>do</b>
4	<b>if</b> excludedHosts.contains(host) <b>then</b>
5	<b>continue</b> ;
6	<b>else</b>
7	candinateHostList.add(host);
8	<b>end if</b>
9	<b>end foreach</b>
10	<b>If</b> (highLoadFlag == TRUE) <b>then</b>
11	underLoadHost ← selectHostWithMinimumMetric(candinateHostList);
12	underLoadHostList.add(underLoadHost);
12	<b>else</b>
13	Utilization <sub>ave</sub> ← calculateAverageUtilization(hostList);
14	Utilization <sub>stv</sub> ← calculateStandardDeviationUtilization(hostList);
15	T <sub>1</sub> ← Utilization <sub>ave</sub> - Utilization <sub>stv</sub> ;
16	<b>foreach</b> host in candinateHostList <b>do</b>
17	Utilization ← host.getUtilizationOfCpu();
18	<b>if</b> (Utilization < T <sub>1</sub> ) <b>then</b>
19	underLoadHostList.add(host);
20	<b>end if</b>
21	<b>end foreach</b>
22	<b>end if</b>
23	<b>return</b> underLoadHostList;

الشكل (6) الشيفرة المزيفة لسياسة خوارزمية PADC في تحديد المضيفات ذات الحمل المنخفض [الباحث]

تقوم الخوارزمية بالبحث عن المضيفات ذات الحمل المنخفض مستثنية من ذلك المضيفات ذات الحمل المرتفع والمضيفات المطفأة لتفريغها من الآلات الافتراضية ومن ثم وضعها في حالة إيقاف بهدف توفير الطاقة. في هذه المرحلة تكتسب خوارزمتنا صفتها التكيفية حيث أن طريقة تحديد هذه المضيفات تعتمد بشكل كبير على المرحلة السابقة (تهجير الآلات الافتراضية من المضيفات ذات الحمل الزائد). يتم النظر إلى مخطط التوضع الذي تم الوصول إليه وفي حالة تضمنه مضيفاً كان في حالة إيقاف عن العمل وتم تشغيله من أجل استضافة آلة افتراضية مهجرة فهذا يعني بأن النظام في حالة حمل كبير (HighLoadFlag == TRUE)، وهنا يتم تطبيق منهج بسيط يعتمد على البحث عن المضيف الذي يحتوي على أقل عدد من الآلات الافتراضية ذات نسبة الاستخدام المنخفضة والتي من المحتمل أن نجد لها توضعاً يلبي متطلباتها حيث يتم استخدام المعيار التالي:

$$Host_{metric} = \sqrt{VmList_{averageUtilization}^2 + VmList_{sizeRatio}^2} \quad (9)$$

حيث يمثل  $VmList_{averageUtilization}$  متوسط نسبة الاستخدام للآلات الافتراضية الموجودة على المضيف، و  $VmList_{sizeRatio}$  نسبة عدد هذه الآلات إلى عدد الآلات الكلي الموجود في مركز البيانات.

أما في حالة كان مخطط التوزيع لا يحتوي حواسيب مطفأة سابقاً، فهذا يعني أن النظام في حالة حمل طبيعي وهنا نقوم بتحديد عتبة بناء على متوسط نسبة الاستخدام على جميع المضيفات والانحراف المعياري لهذه النسب (السطر 15) وفي حالة كانت نسبة الاستخدام أقل منها يتم اعتباره في حالة حمل منخفض وإضافته إلى اللائحة. وهنا يجب أن نشير إلى أنه في حال الفشل في إيجاد توضع جديد لأي من الآلات الافتراضية المهجرة من مضيف منخفض الحمل يتم إلغاء العملية كاملة ويبقى المضيف في حالة نشاط، أما إذا نجحنا بإيجاد توضع جديد لكل الآلات فإننا نستكمل عملية التهجير ومن ثم نضع المضيف في وضع الإيقاف.

### 3- بيئة المحاكاة ومعايير الأداء:

نستخدم في هذا البحث أداة المحاكاة cloudsims وهي واحدة من أشهر الأدوات المستخدمة في محاكاة البيئات السحابية، وقد قمنا بإجراء التجربة عن طريق محاكاة مركز بيانات غير متجانس يحتوي على 800 مضيف تنقسم مناصفة بين نوعين من المضيفات، بالإضافة إلى أكثر من ألف آلة افتراضية مع أربع أنواع منها تحاكي تلك المستخدمة في Amazon EC2<sup>2</sup>. يوضح الجدول (2) مواصفات كل من المضيفات والآلات الافتراضية

الجدول (2) أنواع المضيفات والآلات الافتراضية [13]

Name	CPU (MIPS)	Cores	Memory	Bandwidth	Level of Max PPR
Host Type 1	1.86 GHz	2	4 GB	1 Gbit/s	90%
Host Type 2	2.66 GHz	2	4 GB	1 Gbit/s	90%
VM Type 1	2.5 GHz	1	870 MB	100 Mbit/s	-
VM Type 2	2.0 GHz	1	1740 MB	100 Mbit/s	-
VM Type 3	1.0 GHz	1	1740 MB	100 Mbit/s	-
VM Type 4	0.5 GHz	1	613 MB	100 Mbit/s	-

وقد قمنا ببناء التجارب باستخدام مسارات حمل حقيقية تم اختيارها من مشروع CoMon وهو نظام مراقبة تابع لمختبرات PlanetLab [23]، حيث تم تجميع البيانات الخاصة بنسبة استخدام وحدة المعالجة المركزية لآلاف الآلات الافتراضية الموزعة على مخدمات موجودة في أكثر من 500 موقع حول العالم كل خمس دقائق كما هو موضح في الجدول (3).

الجدول (3) مسارات حمل PlanetLab

Date	Number of VMs	Mean(%)	St.dev(%)
2011/03/03	1052	12.31%	17.09%
2011/03/06	898	11.44%	16.83%
2011/03/09	1061	10.70%	15.57%
2011/03/22	1516	9.26%	12.78%
2011/03/25	1078	10.56%	14.14%
2011/04/03	1463	12.39%	16.55%
2011/04/09	1358	11.12%	15.09%
2011/04/11	1233	11.56%	15.07%
2011/04/12	1054	11.54%	15.15%
2011/04/20	1033	10.43%	15.21%

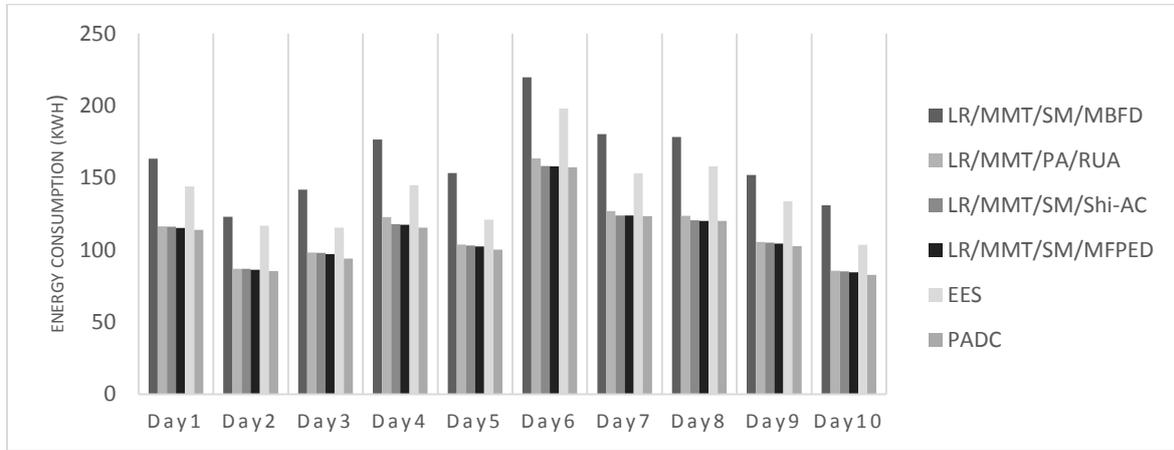
<https://aws.amazon.com/ec2/instance-types/><sup>2</sup>

سنقوم بمقارنة أداء خوارزمتنا المقترحة (PADC) مع كل من خوارزمية : LR/MMT/SM/MBFD [12] وهي الخوارزمية الافتراضية المستخدمة في أداة المحاكاة cloudsimsim [18]، LR/MMT/PA/RUA، [19] LR/MMT/SM/Shi-AC، [20] LR/MMT/SM/MFPED، [21] EES من حيث استهلاك الطاقة، عدد التهجير، زمن خرق اتفاقيات مستوى الخدمة في المضيفات النشطة (SLATAH)، انخفاض الأداء نتيجة عمليات التهجير (PDM)، خرق اتفاقيات مستوى الخدمة (SLAV)، بالإضافة إلى استخدام معيار يجمع بين استهلاك الطاقة وخرق اتفاقيات مستوى الخدمة تم تقديمه في [13]:

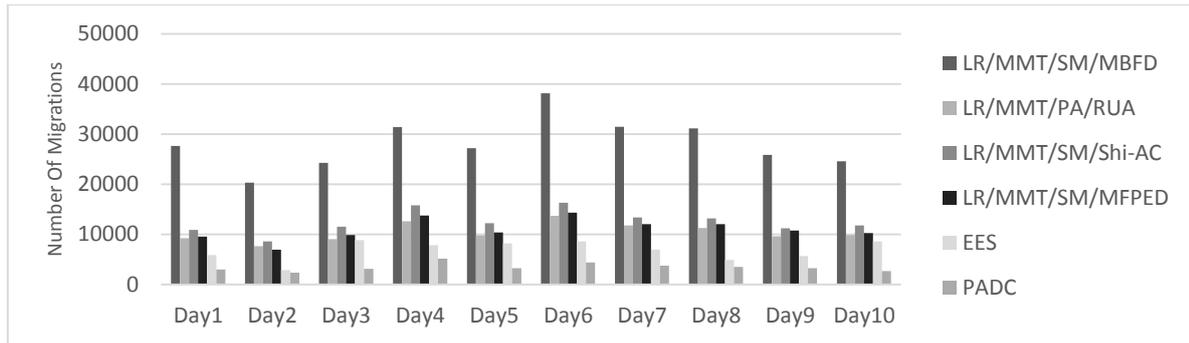
$$ESV = EC * SLAV \quad (10)$$

### النتائج والمناقشة:

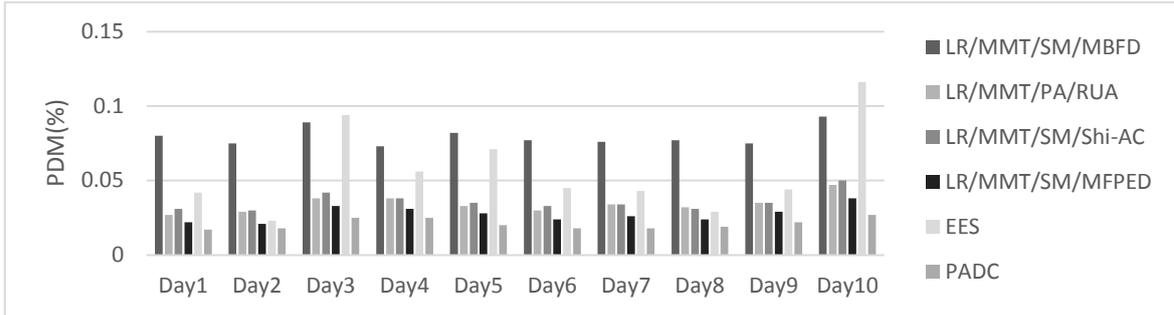
تظهر الأشكال من (7) حتى (12) نتائج المحاكاة لمختلف الخوارزميات وفقاً لمعايير الأداء التي قمنا باختيارها من أجل كل مسار حمل.



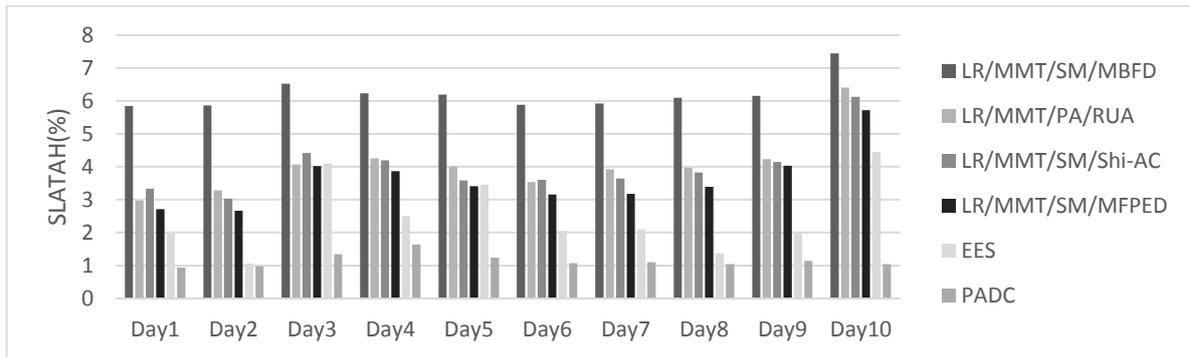
الشكل (7) مقارنة الخوارزميات من حيث استهلاك الطاقة



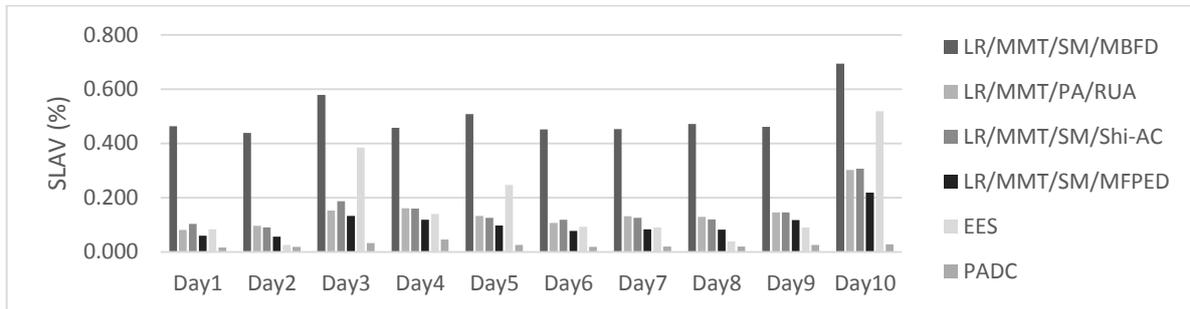
الشكل (8) مقارنة الخوارزميات من حيث عدد التهجير



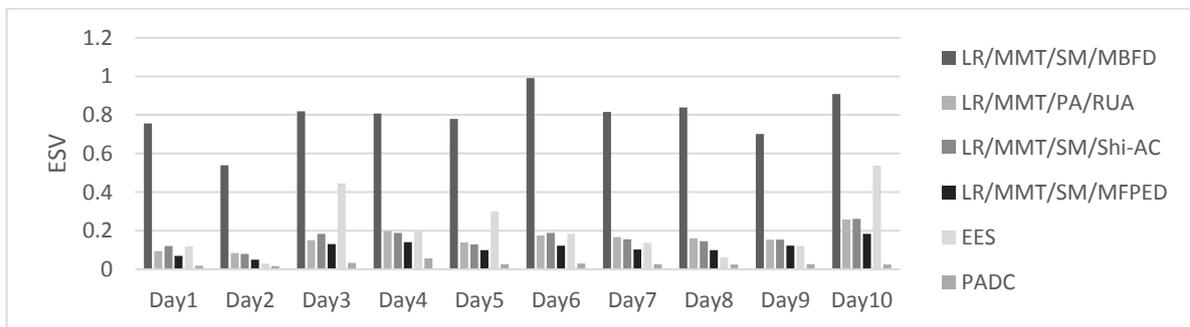
الشكل(9) مقارنة الخوارزميات من حيث انخفاض الأداء نتيجة عمليات التهجير



الشكل(10) مقارنة الخوارزميات من حيث زمن خرق اتفاقيات مستوى الخدمة في المضيفات النشطة



الشكل(11) مقارنة الخوارزميات من حيث خرق اتفاقيات مستوى الخدمة



الشكل(12) مقارنة الخوارزميات وفق معيار ESV

تظهر النتائج أن أداء الخوارزميات يختلف من يوم إلى آخر تبعاً للحمل المطبق على مركز البيانات والذي يتغير نتيجة لاختلاف عدد الآلات الافتراضية التي يطلبها المستخدمون والتطبيقات التي يتم تشغيلها عليها. ونلاحظ بأنه وباستثناء

خوارزميتي LR/MMT/SM/MBFD و EES تظهر بقية الخوارزميات أداءً متقارباً من حيث استهلاك الطاقة مع أفضلية بسيطة لخوارزمتنا المقترحة (الفرق بين خوارزمتنا وأقرب الخوارزميات LR/MMT/SM/MFPED بحدود الواحد في المئة فقط). وقد حققت الخوارزمية هذا المستوى من استهلاك الطاقة بالرغم من استخدامها عدداً أكبر من المضيفات حيث أظهرت التجارب أن عدد المضيفات التي تم إطفائها من قبل خوارزمتنا هو الأقل مقارنةً ببقية الخوارزميات إلا أن سياسة التوضع الخاصة بها وطريقة اختيارها للمضيفات تعتمد على إعطاء الأولوية للمضيفات الأكثر كفاءة في استهلاك الطاقة ومحاولة الوصول بها إلى عتبة أفضل قيمة للأداء مقابل الطاقة المستهلكة. وفقاً للشكل (8) فإن عدد التهجيريات التي قامت بها خوارزمتنا هو الأقل من بين جميع الخوارزميات وأقرب خوارزمية وهي EES قامت وسطياً بضعف هذا العدد، وهذا يعني أن الآلات الافتراضية لم تُهجر بكثافة في مرحلة التهجير من المضيفات ذات الحمل المرتفع حيث نلاحظ أن معيار زمن خرق اتفاقيات مستوى الخدمة في المضيفات النشطة (SLATAH) في أغلب الأحيان بحدود 1% (أقل بنسبة 45% وسطياً من خوارزمية EES)، وهذا ما يؤكد دقة تقنية ARIMA في التنبؤ بالمتطلبات المستقبلية من الموارد وتوقعها على تقنية LR والعتبات الساكنة في تحديد المضيفات ذات الحمل المرتفع.

يوضح الشكل (9) التحسين الكبير الذي قدمته الخوارزمية في معيار انخفاض الأداء نتيجة التهجير (PDM) (أقل بنسبة 25% وسطياً من خوارزمية LR/MMT/SM/MFPED) وهذا يعود إلى العديد من العوامل مثل: سياسة الخوارزمية في اختيار المضيفات ذات الحمل المنخفض (التي تستضيف أقل عدد من الآلات الافتراضية ذات نسبة الاستخدام المنخفضة)، العدد المنخفض من التهجيريات، كفاءة سياسة الاختيار المستخدمة للتهجير من حيث قيامها بالبحث عن الآلات الافتراضية الأكثر حاجة للموارد والأقل زمناً للتهجير.

كنتيجة للتفوق الذي أحرزته الخوارزمية في مجال جودة الخدمة مع أدائها الجيد من حيث استهلاك الطاقة فإنه من الطبيعي أن تتجاوز بمراحل بقية الخوارزميات من حيث المعيار المشترك لاستهلاك الطاقة وخرق اتفاقيات مستوى الخدمة (ESV) حيث أن القيمة الوسيطة أقل بنسبة 75% من خوارزمية LR/MMT/SM/MFPED.

### الاستنتاجات والتوصيات:

لقد تمكننا بحثنا من تقديم حل فعال لمشكلة الموازنة ما بين توفير الطاقة وجودة الخدمة، حيث تم الوصول إلى مستويات مقبولة من استهلاك الطاقة مع المحافظة قدر الإمكان على اتفاقيات مستوى الخدمة القائمة بين مزودي ومستخدمي الخدمة السحابية. كما أظهرت النتائج التي تم توصل إليها الدور الكبير الذي يمكن أن تلعبه تقنيات التنبؤ بالسلاسل الزمنية في تحسين كفاءة التجميع الديناميكي للآلات الافتراضية عن طريق التنبؤ بالمتطلبات المستقبلية للآلات الافتراضية ومحاولة تلبينها مسبقاً وهذا ما يخفف من احتمال تهجيرها لاحقاً، كما أن مراعاة عدم التجانس الموجود في المضيفات يمكننا من استثمار الموارد بشكل فعال من حيث الأداء واستهلاك الطاقة. بالنسبة للأعمال المستقبلية فإنها سوف تتمحور حول اختبار فعالية الخوارزمية مع مسارات حمل أخرى غير مسارات PlanetLab، بالإضافة إلى العمل على تصميم نموذج تنبؤي فعال وأكثر دقة من النماذج الموجودة. كذلك سوف نحاول تطوير الخوارزمية بحيث تشمل موارد متعددة وليس فقط وحدة المعالجة المركزية.

## References:

- [1] P. G. Mell and T. J. N. S. P. Grance. *The NIST Definition of Cloud Computing*. 2011.
- [2] S. He, L. Guo, M. Ghanem, and Y. Guo. *Improving resource utilisation in the cloud environment using multivariate probabilistic models*. In 2012 IEEE Fifth International Conference on Cloud Computing, 2012, 574-581.
- [3] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. J. S. Koomey. *Recalibrating global data center energy-use estimates*. Science, vol. 367, no. 6481, 2020, 984-986 .
- [4] L. Salimian, F. S. Esfahani, and M.-H. J. C. Nadimi-Shahraki. An adaptive fuzzy threshold-based approach for energy and performance efficient consolidation of virtual machines. Computing, vol. 98, no. 6, 2016, 641- 660
- [5] C. Reiss, A. Tumanov, G. R. Ganger ,R. H. Katz, and M. A. Kozuch. *Heterogeneity and dynamicity of clouds at scale: Google trace analysis*. In Proceedings of the Third ACM Symposium on Cloud Computing, 2012,1-13.
- [6] Y. Cheng, Z. Chai, and A. Anwar. *Characterizing co-located datacenter workloads: An alibaba case study*. In Proceedings of the 9th Asia-Pacific Workshop on Systems, 2018,1-3.
- [7] L. Tomás and J. J. I. T. o. C. C. Tordsson. *An autonomic approach to risk-aware data center overbooking*. IEEE Transactions on Cloud Computing, vol. 2, no. 3, 2014, 292-305.
- [8] Z. Cao and S. J. T. J. o. S. Dong. *An energy-aware heuristic framework for virtual machine consolidation in Cloud computing*. The Journal of Supercomputing, vol. 69, no. 1, 2014,429-451.
- [9] H. Liu, H. Jin, X. Liao, C. Yu, C.-Z. J. I. T. o. P. Xu. *Live virtual machine migration via asynchronous replication and state synchronization*. IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 12, 2011, 1986-1999. .
- [10] X. Zhou, K. Li, C. Liu, and K. J. I. A. Li. *An experience-based scheme for energy-SLA balance in cloud data centers*. IEEE Access, vol. 7, 2019, 23500-23513 .
- [11] A. Shehabi et al. *United states data center energy usage report*. Lawrence Berkeley National Laboratory, Berkeley, California, vol. 4, 2016.
- [12] R. Basmadjian, N. Ali, F. Niedermeier, H. De Meer, and G. Giuliani. *A methodology to predict the power consumption of servers in data centres*. In Proceedings of the 2nd international conference on energy-efficient computing and networking, 2011, 1-10.
- [13] A. Beloglazov, R. J. C .Buyya, C. Practice. *Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers*. Concurrency and Computation: Practice and Experience, vol. 24, no. 13, 2012,1397-1420 .
- [14] A. Beloglazov, J. Abawajy, and R. J. F. g. c. s. Buyya. *Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing*. Future generation computer systems, vol. 28, no. 5, 2012, 755-768.
- [15] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya. *Cost of virtual machine live migration in clouds: A performance evaluation*. In IEEE International Conference on Cloud Computing, 2009, 254-265.
- [16] Z. Usmani and S. J. P. C. S. Singh. *A survey of virtual machine placement techniques in a cloud data center*. Procedia Computer Science, vol. 78, 2016, 491-498.
- [17] Y. Saadi and S. J. S. C. El Kafhali. *Energy-efficient strategy for virtual machine consolidation in cloud environment* . Soft Computing, 2020, 1-15.
- [18] G. Han ,W. Que, G. Jia, and L. J. S. Shu. *An efficient virtual machine consolidation scheme for multimedia cloud computing*. Sensors, vol. 16, no. 2, 2016, 246 .

- [19] Z. A. Mann, M. J. C. Szabó, C. Practice, and Experience. *Which is the best algorithm for virtual machine placement optimization?*. *Concurrency and Computation: Practice and Experience*, vol. 29, no. 10, 2017 .
- [20] F. F. Moges and S. L. J. J. o. C. C. Abebe. *Energy-aware VM placement algorithms for the OpenStack Neat consolidation framework*. *Journal of Cloud Computing*, vol. 8, no. 1, 2019, 2
- [21] Y. Liu, X. Sun, W. Wei, and W. J. I. A. Jing. *Enhancing energy-efficient and QoS dynamic virtual machine consolidation method in cloud environment*. *IEEE Access*, vol. 6, 2018, 31224-31235 .
- [22] R. N. Calheiros, E. Masoumi, R. Ranjan, and R. J. I. T. o. C. C. Buyya. *Workload prediction using ARIMA model and its impact on cloud applications QoS*. *IEEE Transactions on Cloud Computing*, vol. 3, no. 4, 2014, 449-458 .
- [23] K. Park and V. S. J. A. S. O. S. R. Pai. *CoMon: a mostly-scalable monitoring system for PlanetLab*. *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, 2006, 65-74 .