

Load Balancing in Cloud Computing (Compare Among Many Algorithms)

Dr. Nasser Nasser*
Dr. Talal Alaateky**
Orwah Abd Alhameed***

(Received 20 / 5 / 2020. Accepted 20 / 7 / 2020)

□ ABSTRACT □

Load Balancing is essential for efficient operations in distributed environments. As Cloud Computing is growing rapidly and clients are demanding more services and better results, load balancing for the Cloud has become a very interesting and important research area. Many algorithms were suggested to provide efficient mechanisms and algorithms for assigning the client's requests to available Cloud nodes. These approaches aim to enhance the overall performance of the Cloud and provide the user more satisfying and efficient services. In this paper, we investigate the different algorithms proposed to resolve the issue of load balancing and task scheduling in Cloud Computing. We discuss and compare these algorithms to provide an overview of the latest approaches in the field.

Keywords: Cloud Computing, Load Balancing, Task Scheduling, Cloud Storage, Replications

* Associate Professor, Department of Software Engineering, Faculty of Informatics Engineering, Tishreen University, Latakia, Syria.

**Assistant Professor, Department of Networks Engineering, Faculty of Informatics Engineering, Tishreen University, Latakia, Syria.

***Postgraduate Student, Department of Software Engineering, Faculty of Informatics, Engineering, Tishreen University, Latakia, Syria.

موازنة الحمل في الحوسبة السحابية (مقارنة بين مجموعة من الخوارزميات)

د. ناصر ناصر*

د. طلال العاتكي**

عروه عبد الحميد***

(تاريخ الإيداع 20 / 5 / 2020. قُبِلَ للنشر في 20 / 7 / 2020)

□ ملخص □

موازنة الحمل هو أمر حقيقي وملح في العمليات الأساسية للبيئات الموزعة، وحيث أن الحوسبة السحابية أخذت في النمو بشكل سريع مع زيادة متطلبات العملاء لخدمات أكثر ونتائج أفضل، أصبحت موازنة الحمل في السحابة محل بحث مهم ومشوق.

تم التطرق في هذه المقالة لمجموعه من الخوارزميات المقترحة لتأمين آليات فعالة لتوزيع طلبات العملاء على عقد سحابية متوفرة، يساعد هذا النهج على تحسين الأداء العام للسحابة ويقدم للمستخدم خدمات أكثر فعالية وشفافية وبالتالي حل مشكلة موازنة الحمل وجدولة المهام في الحوسبة الشبكية وتمت المقارنة بين هذه الخوارزميات لأخذ نظرة عامه عن أحدث الأطر في هذا المجال واستكشاف أفضل الخوارزميات المسلطة على هذا الموضوع.

الكلمات المفتاحية: الحوسبة السحابية، موازنة الحمل. جدولة المهام، التخزين السحابي، النسخ المتكرر.

* أستاذ مساعد - قسم البرمجيات - كلية الهندسة المعلوماتية - جامعة تشرين - اللاذقية - سورية.

** مدرس - قسم الشبكات والنظم الحاسوبية - كلية الهندسة المعلوماتية - جامعة تشرين - اللاذقية - سورية.

*** طالب دراسات عليا - قسم البرمجيات - كلية الهندسة المعلوماتية - جامعة تشرين - اللاذقية - سورية.

مقدمة:

أصبحت الحوسبة السحابية شائعة جدا في السنوات القليلة الماضية. فهي تقدم طريقة سهلة ومرنة لحفظ واستعادة الملفات والبيانات، كجزء من خدماتها. وخصوصا في ظل إنشاء مجموعات بيانات وملفات ضخمة جدا للمستخدمين المنتشرين حول العالم. تتطلب معالجة هذا الكم الكبير من البيانات تقنيات عديدة لتحسين العمليات وتزويد المستخدمين بمستويات أداء عالي. لذلك من المهم عند البحث في مجال السحابة أن يتم تحسين استغلال المخزون وأداء التحميلات عند المستخدمين. سنتعرض لمشكلة مهمة في هذا المجال ألا وهي موازنة الحمل الديناميكية أو جدولة المهام. هذا وقد تم فحص خوارزميات موازنة الحمل بكثافة على البيئات المختلفة؛ ومع ذلك فإنه مع البيئة السحابية ظهرت بعض التحديات الإضافية والتي يجب ان تتم معالجتها. تتطوي المساعي الرئيسية في الحوسبة السحابية تحت توزيع المهام على العقد السحابية والقيام بمعالجة الطلبات بكفاءة قدر الإمكان [1] مع إمكانية تحمل الأعباء المؤثرة المختلفة مثل عدم التجانس أو التأخيرات الكبيرة للاتصالات تصنف خوارزميات موازنة الحمل الى ديناميكية وثابتة. تعد الخوارزميات الثابتة مناسبة جدا للبيئات المتجانسة والمستقرة وبإمكانها ان تحقق نتائج جيدة في تلك البيئات، ولكنها عادة غير مرنة ولا تستطيع مواكبة التغييرات الديناميكية للمعطيات وقت التنفيذ.

بينما تعد الخوارزميات الديناميكية أكثر مرونة وتأخذ بعين الاعتبار الأنماط المختلفة لمعطيات النظام قبل وأثناء وقت التنفيذ. [2] بإمكان هذه الخوارزميات التكيف مع المتغيرات وتقديم نتائج أفضل من الثابتة في البيئات الغير متجانسة والديناميكية. ومع ذلك فإنه عندما تصبح المعطيات الموزعة أكثر تعقيدا وديناميكية قد تصبح كنتيجة لذلك بعض من هذه الخوارزميات غير مجدية مسببة الكثير من التكاليف أكثر من الضرورة مؤدية بذلك إلى تراجع ملحوظ في أداء الخدمات.

أهمية البحث وأهدافه:

المسألة الرئيسية التي نقدمها في هذه المقالة هي أن الحوسبة السحابية توزع عبء العمل بشكل حيوي عبر عقد متعددة حيث لا يوجد مورد واحد غير مستغل تماما ويعرف باسم مشكلة التحسين. للتغلب على هذه المسألة- الحمل الزائد- تمت مناقشة مجموعة من الخوارزميات ونقارن بينها لمعرفة مدى مواءمة الخوارزمية للمسألة المعطاة مع أفكار عن أعمال مستقبلية.

طرائق البحث ومواده:

وصفنا في القسم الأول التحديات في موازنة الحمل في الحوسبة الشبكية وفي القسم الثاني تم التطرق الى الخوارزميات الثابتة في موازنة الحمل للحوسبة السحابية مع شرح مفصل لخوارزمية MapReduce، وفي القسم الثالث تم التطرق الى خوارزميات موازنة الحمل الديناميكية مع القليل من التفصيل لخوارزمية موازنة الحمل الديناميكية DDFTP، و تمت مناقشة الخوارزميات السابقة وقارنا فيما بينها في القسم الرابع. واختتمنا هذا العمل في القسم الخامس مع أفكار عن عمل مستقبلي حول تحسين موازنة الحمل في الحوسبة السحابية.

لقد تم في هذا البحث دراسة لخوارزميات موازنة الحمل الحالية التي تتناسب بيئات السحابة سنقدم نظرة عامة لهذه الخوارزميات ونناقش خواصها وسنقارن بينها بالاعتماد على (حجم المعطيات المأخوذة بعين الاعتبار، حمل الشبكة بشكل عام، السلاسل الزمنية).

أولاً: التحديات في موازنة الحمل في الحوسبة الشبكية:

يجب أن نتعرف على المشاكل والتحديات المتعلقة بهذا البحث ومدى تأثيرها على كيفية إنجاز الخوارزمية قبل استعراض الأطر الحالية لموازنة الحمل في الحوسبة السحابية ، قمت مناقشة التحديات التي يجب معالجتها عند محاولة اقتراح الحل الأمثل لمشكلة موازنة الحمل في الحوسبة السحابية. يمكن أن تلخص هذه التحديات في النقاط التالية:

- المسافة الجغرافية بين عقد السحابة:

بعض الخوارزميات مصممة لتكون فعالة فقط في الشبكات الداخلية أو عند توضع العقد بالقرب من بعضها حيث تكون التأخيرات الناتجة عن الاتصال ضئيلة. لذلك يعد تصميم خوارزمية موازنة حمل تعمل لعقد موزعة بمسافات كبيرة فيما بينها تحد حقيقي لأنه يجب أن نأخذ بعين الاعتبار عوامل أخرى مثل سرعة الشبكة الرابطة بين العقد والمسافة بين العميل وعقد معالجة المهمة والمسافة بين العقد نفسها المسؤولة عن تقديم الخدمة. إذا هناك حاجة لتطوير طريقة للتحكم بآلية موازنة الحمل بين العقد المتباعدة جغرافياً مع البقاء على إمكانية تجاهل التأخيرات [3].

- التخزين / التكرارات:

لا تأخذ الخوارزميات ذات التكرار الكامل التمثيل الفعال للتخزين بعين الاعتبار لأن نفس البيانات ستخزن بكل العقد وبالتالي ستفرض تكاليف أكبر بسبب زيادة حجم التخزين، بينما تقوم الخوارزميات ذات التكرار الجزئي بحفظ أجزاء من مجموعات البيانات في كل عقدة (بمستوى محدد من التداخل) حسب إمكانيات كل عقدة كالسعة وقوة المعالجة، [4] وهذا بدوره يؤدي لتمثيل أفضل للتخزين. الآن ومع ازدياد تعقيد خوارزميات موازنة الحمل سوف تأخذ بالاعتبار توفر أجزاء مجموعات البيانات عبر العقد المختلفة.

- تعقيد الخوارزمية:

يفضل أن تكون خوارزمية موازنة الحمل أقل تعقيداً في إنجاز العمليات، فكلما زاد تعقيد التنفيذ كلما زادت تعقيدات المعالجة والتي تسبب بدورها مشاكل أداء سلبية. بل وأكثر من ذلك، عندما تتطلب الخوارزميات معلومات أكثر واتصالات أكبر من أجل المراقبة والتحكم، ستتسبب التأخيرات في مشاكل أكثر وانهايار في الفعالية. وبناء عليه يجب أن تصمم خوارزمية موازنة الحمل بأبسط شكل ممكن [5].

- نقطة الفشل:

يجب تصميم الخوارزمية بحيث تتجنب نقطة الفشل أو انهيار النظام، بعض الخوارزميات (المركزية منها) يمكن أن تقدم آليات فعالة لمشكلة موازنة الحمل في ظروف معينه، إلا أنها تعاني من مشكلة المتحكم الوحيد لكل النظام وفي هذه الحالة ينهار كل النظام إذا انهار المتحكم. يجب أن تصمم الخوارزمية لمواجهة هذا التحدي [6]. الخوارزميات الموزعة تقدم حلاً أفضل مع أنها حتى الآن أكثر تعقيداً وتتطلب تنسيق وتحكم أكبر بالوظيفة المطلوبة بشكل صحيح.

ثانياً: الخوارزميات الثابتة لموازنة الحمل:

توزع هذه الخوارزمية المهام على العقد بناء على قابلية العقدة لمعالجة طلبات جديدة، وهذه العملية مبنية على معلومات مسبقة بخواص وإمكانيات العقدة (قوة المعالجة، الذاكر، سعة التخزين، وأداء أحدث اتصال معروف). على الرغم من امتلاكها معلومات مسبقة عن خواص العقد وامكاناتها الا انها لا تأخذ بعين الاعتبار التغيرات الديناميكية الطارئة على هذه الخواص وقت التنفيذ بالإضافة لعدم قدرتها على التكيف لتحميل التغيرات وقت التنفيذ.

اقترح Rakocevic خوارزمية (Central Load Balancing Decision Model) (CLBDM) [7] وهي تحسين لخوارزمية Round Robin [8] (جولة روبن) وهذه الخوارزمية (CLBDM) مبنية على تبديل الجلسة في طبقة

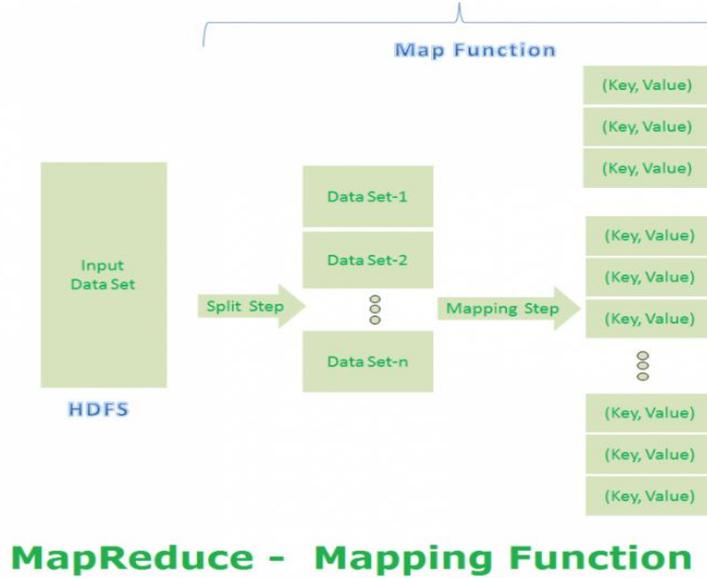
التطبيقات، ففي جولة روبن التي تعد من خوارزميات موازنة الحمل الشهيرة تقوم بإرسال الطلبات الى العقد بأقل عدد من الاتصالات . يكون التحسين في الخوارزمية الجديدة على الشكل التالي : يتم حساب وقت الاتصال بين العميل والعقدة في السحابة فإذا تجاوز هذا الوقت عتبة معينة (threshold) سيحصل مشكلة وبناء عليها تتم مقاطعة الاتصال وسترسل المهمة الى عقدة أخرى لتطبق عليها قواعد خوارزمية جولة روبن .تعمل CLBDM كمشرف آلي ، والفكرة مستوحاة من المشرف البشري.

KUMAR اقترح خوارزمية [9] لتحسين نسخة الخوارزمية في [10] حيث أن كلا الخوارزميتين تستخدم تصريف النمل أو خوارزمية النمل لجمع المعلومات حول عقد السحابة من أجل توزيع المهمة على عقدة محددة. سنصادف مشكلة المزامنة في الخوارزمية [10] فجاءت الخوارزمية المقترحة [9] لتضيف خاصية الانتحار (suicide) على النمل وتحل مشكلة المزامنة، كلا الخوارزميتين تعمل بالشكل التالي:

عندما تبدأ المهمة يقوم النمل وال فيرومون بالبدء بالعمل فيسير النمل باتجاه أمامي بدءاً من العقدة "الرأس" الحركة الأمامية تعني أن النمل يتحرك من عقدة مشغولة تماماً "منقطة" الى العقدة التالية لفحص انشغالها إذا وجد النمل عقدة غير مثقلة سيستمر بالاتجاه الامامي لفحص العقدة التالية فاذا كانت مثقلة سيستعمل الحركة الخلفية للعودة الى العقدة السابقة، خاصية الخوارزمية [9] ستجعل النمل ينتحر حالما يجد العقدة الهدف وهذا بدوره يمنع الحركة الخلفية الغير ضرورية للنمل.

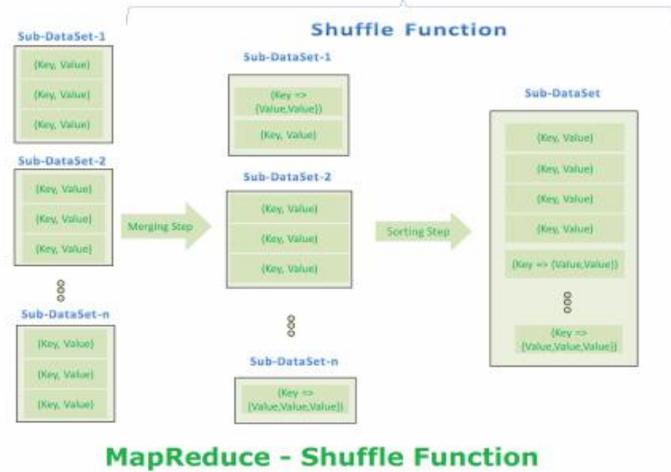
الخوارزمية المقترحة في المقالة [11] إضافة لخوارزمية ال (MapReduce) [12] التي تعد خوارزمية موزعة لمعالجة البيانات مقدمة من كوقل وهي مفيدة الكميات الضخمة من البيانات على التفرع فهي فعالة جدا في البيئات العنقودية و تعتمد هذه الخوارزمية على تقسيم وتشبيت البيانات ومن ثم إعادة تركيبها بشكل يناسب المشكلة المعطاة فهي تقسم المهمة المعطاة الى عدة مهام فرعية ليتم التعامل مع كل مهمة على حدة على التفرع. خوارزمية ال- Map Reduce تستخدم ثلاث خطوات رئيسية: (Map Function·Shuffle function·Reduce function).

Map function: في هذه الخطوة الرئيسية نستخدم خطوتين ثانويتين ضمنا وهما (Splitting, mapping). في الsplitting نقوم بتقسيم مجموعة البيانات الى مجموعات فرعية ، وفي ال mapping نقوم بتنفيذ حسابات معينة ليكون الخرج عبارة عن أزواج المفاتيح والقيم (Key,value) كما في الشكل رقم(1) :



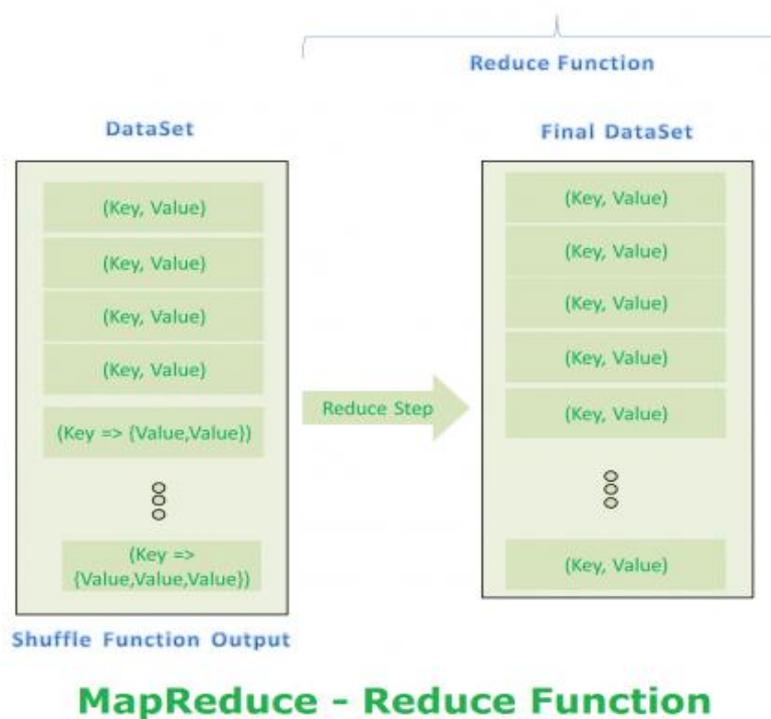
الشكل رقم (1)

Shuffle function: نستخدم هنا أيضا خطوتين ثانويتين وهما (Sorting, Merging) يتم دمج كل الأزواج المشتركة في الخطوة الأولى Merging بنفس المفاتيح ليكون خرج هذه الخطوة على الشكل $(Key, \langle list\ values \rangle)$.
 يتم ترتيب كل القيم السابقة في الخطوة الثانية Sorting حسب اسم المفتاح كما في الشكل رقم (2):



الشكل رقم (2)

Reduce function: يتم في هذه الخطوة انقاص العمليات كما في الشكل رقم (3):



الشكل رقم (3)

قد تبدو مخرجات الخطوة الرئيسية كمخرجات الخطوة الأولى ولكن في الحقيقة هناك اختلاف جوهري فمخرجات المرحلة الأخيرة من أزواج (Key, Value) تمت معالجتها بعدة توابع. وسنتطرق لمثال بسيط لتطبيق نظرية MapReduce يوضح الفرق بين هذه المخرجات. مثال MapReduce (عداد الكلمات):

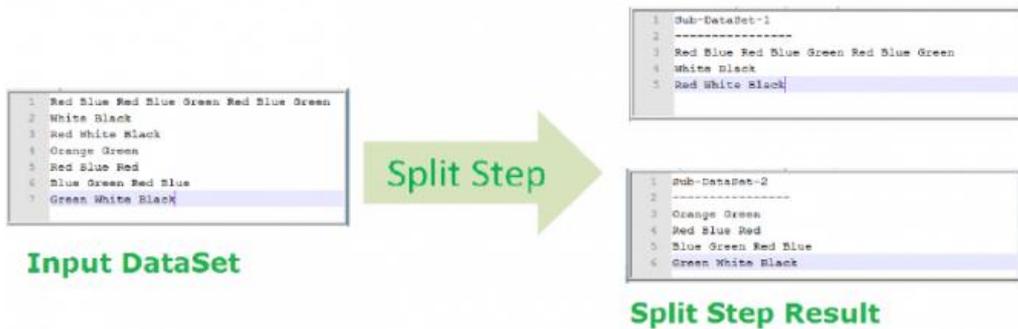
- بيان المشكلة: حساب عدد تكرار كل كلمة في مجموعة البيانات.
 - مجموعة البيانات المدخلة: سنعرض مجموعة بسيطة لشرح النظرية ولكن في الحقيقة نستطيع تطبيق هذه الخوارزمية على مجموعة ضخمة من البيانات وتحقيق نتائج ممتازة.
- نفرض أن مجموعة البيانات المدخلة على الشكل المبين:

```

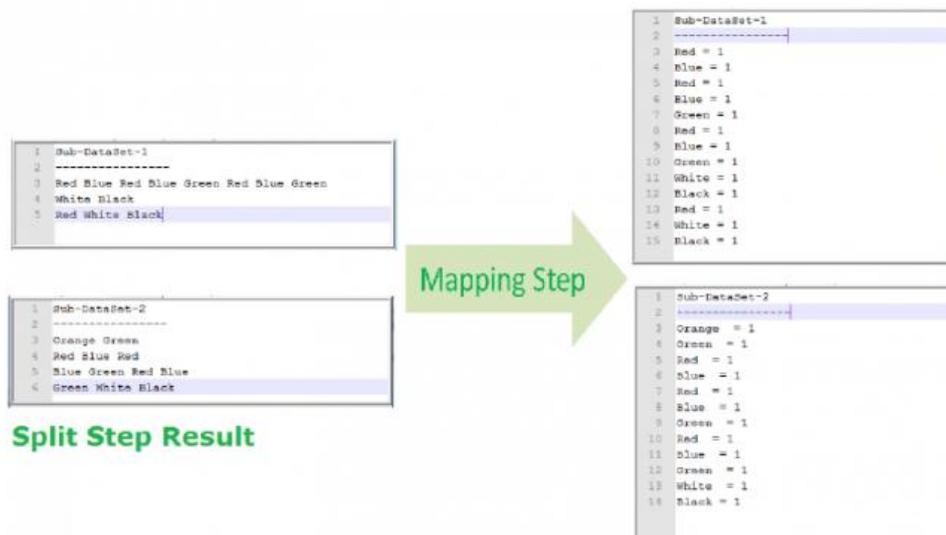
1 Red Blue Red Blue Green Red Blue Green
2 White Black
3 Red White Black
4 Orange Green
5 Red Blue Red
6 Blue Green Red Blue
7 Green White Black
    
```

سنطبق الخطوات بالترتيب المبين سابقا:

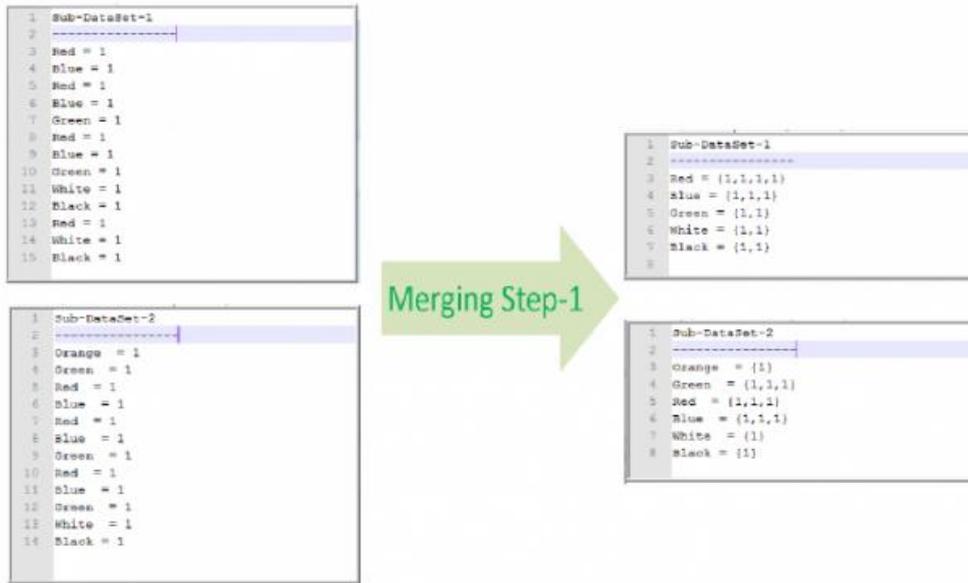
-1 : Map Function (Split Step) A



-1 : Map Function (Mapping Step) B



: Shuffle Function (Merge Step) A -2



: Shuffle Function (Sorting Step) : B-2



: Reduce Function (Reduce Step) -3



بعد الشرح المقدم عن خوارزمية MapReduce يمكن القول أن استخدامها في مشكلة موازنة الحمل يتم عن طريق ال (Part & Compare) Map tasks و ال Reduce tasks results (Group) حيث يتم تقسيم الطلب الى عدة أقسام فرعية (Parting) ويخزن مفتاح كل قسم الى جدول معين نسميه (Hash Key Table)، ومن ثم يتم المقارنة بين هذه الأقسام (Comparing)، وفي النهاية يتم تجميع الأقسام المتشابهة لمعالجتها (Grouping). وحيث أنه سيتم تعيين ومعالجة المهام على التفرع بعد تجزئة المهام كلها (في المرحلة الأولى Mapping) سيؤدي بدوره إلى زيادة الحمل على مهام المرحلة الثانية (Reduce). لذلك فإنه في المقالة المقترحة [11] تم اقتراح إضافة مستوى من موازنة الحمل بين الخطوتين (Mapping & Reducing) مما يؤدي بدوره الى تخفيف الضغط عن المرحلة الثانية، موازنة الحمل بين المرحلتين تهدف إلى تجزئة المهام الضخمة فقط الى مهام أصغر ومن ثم يتم ارسالها الى المرحلة الثانية ليتم معالجتها حسب إمكانيات عقد النظام.

اقترح Junjie خوارزمية موازنة الحمل في السحابة الخاصة [13] مستخدماً الآلات الافتراضية لتعيين الآلات الفيزيائية. تحتوي الخوارزمية على متحكم جدولة مركزي ومراقب موارد، يقوم المتحكم بمعرفة اي الموارد قادرة على تنفيذ مهمة معينة ويقوم بتوجيه تلك المهمة الى المورد المناسب (العقدة). بينما يقوم مراقب الموارد بمعرفة الموارد المتوفرة في السحابة.

ثالثاً: الخوارزميات الديناميكية لموازنة الحمل:

تضع هذه الخوارزميات في الحسبان المواصفات المختلفة لإمكانيات العقد وعرض النطاق الترددي للشبكة. تعتمد معظمها على مزيج المعرفة المبنية على المعلومات المجموعة مسبقاً عن العقد في السحابة من جهة وخصائص هذه العقد عند تنفيذ مهمة معينة وقت التنفيذ. هذه الخوارزميات توجه المهام وقد تعيد توجيهها بشكل ديناميكي بناء على المعطيات المجموعة والمحسوبة، مثل هذه الخوارزميات تتطلب مراقبة مستمرة للعقد وتقدم المهام وهي بالعادة صعبة التنفيذ، لكنها أكثر دقة وتحقق موازنة حمل أكثر فعالية.

في الخوارزمية المقترحة في [14] الهدف منها تخفيض تكرار وتكدس البيانات اسمها (Index Name Server) INS فهي تدمج بين الغاء البيانات المتكررة وتحسين اختيار نقطة العبور هناك مجموعة من المعاملات لحساب نقط العبور والاختيار المثلى منها: شيفرة التجزئة (الهاش كود) لكثافة البيانات التي سيتم تحميلها، موقع المخدم الذي يحوي كثافة البيانات الهدف (المخدم الهدف)، جودة الانتقال أو الاتصال التي تعتمد على أداء العقدة ومخطط تقدير الوزن، عرض النطاق الترددي الأعظمي للتحميل من المخدم الهدف.

يجب أن يؤخذ اعتبار اخر وهو اكتشاف فيما كان الاتصال يمكنه التعامل مع عقد اضافية او لا (المستوى المشغول)،

يمكن تصنيف المستويات المشغولة الى ثلاث فئات $B(a), B(b), B(c)$.

$B(a)$ يعني ان الاتصال مشغول جدا ولا يمكنه التعامل مع اية اتصالات اضافية.

$B(b)$ يعني ان الاتصال غير مشغول ويمكن اضافة اتصالات اضافية.

$B(c)$ يعني ان الاتصال محدود وهو بحاجة لمزيد من الدراسة.

$B(b)$ يمكن ان تقسم الى ثلاثة فئات ايضا:

- $B(b1)$: تعني انه على الخوارزمية تحليل وانشاء نسخة احتياطية.
- $B(b2)$: تعني انه على الخوارزمية ارسال الطلبات الى عقد النسخ الاحتياطية.
- $B(b3)$: تتطلب مستوى عال من الفعالية وتعني أنه على الخوارزمية إعادة تحليل وانشاء نسخ احتياطية جديدة.

اقترح REN في المقالة [15] نموذجاً مطوراً للخوارزمية WLC (Weight Least Connection) [16] التي تعمل على أساس توزيع المهام الى العقد ذات أقل عدد من الاتصالات بهذه العقدة غير مهتمة بإمكانيات كل عقدة كقوة المعالجة والسرعة ومساحة التخزين وعرض النطاق الترددي.

جاءت خوارزمية Ren لتحسين هذه الخوارزمية وصار اسمها Weight Least Connection (ESWLC) Exponential Smooth) لتأخذ بعين الاعتبار إمكانيات العقد ذات الاتصالات الأقل وعلى أساسها يتم اختيار العقد بما يتناسب مع نوع المهمة وحجمها على نحو سلس .

في الخوارزمية Dual Direction File Transfer Protocol (DDFTP) المقترحة في المقالة [17] يتم تقسيم الملف المراد تحميله الى المستخدم الى قسمين وتقوم عقدتين بتحميل كل قسم بشكل منفصل مع الاخذ بعين الاعتبار إمكانيات العقد وحمل الشبكة وسرعتها بما يتناسب مع نوع وحجم الملف لمراد تحميله حيث تعمل كل عقدة بالتحميل باتجاه معاكس لعمل العقدة الثانية فنفرض لدينا ملف حجمه m تبدأ العقدة الأولى بالتحميل من المسار 0 حتى $m/2$ والعقدة الثانية تبدأ بالتحميل من ال المسار m بشكل تنازلي حتى المسار $(m/2)+1$. هذه الخوارزمية تقلل من اتصالات الشبكة المطلوبة بين الزبون والعقد وبالتالي تقلل من عبء الشبكة مع الأخذ بالاعتبارات المذكورة مع عدم الحاجة لمراقبة العقد وقت التنفيذ.

المقالة [18] يتم شرح خوارزمية Load Balancing Min-Min)LBMM) التي تعتمد على الخوارزمية الثابتة Opportunistic Load Balancing) OLB) [19] التي تسعى لجعل كل عقد السحابة مشغولة ولا تعتبر زمن التنفيذ في العقدة وهذا بدوره يؤدي الى بطئ شديد بالنظام بالإضافة الى مزيد من الاختناقات حيث أن الطلبات ستبقى معلقة بانتظار تحرر العقد.

جاءت LBBMM لتحسين OLB بالإضافة ثلاث طبقات أول طبقة هي استلام المهمة من قبل مدير الطلب المسؤول وتوجيهها الى الطبقة الثانية إلى مدير الخدمة الذي يقسم المهمة الى مجموعة من المهمات الفرعية لتسريع معالجة هذه المهمة، ويقوم أيضاً بتوزيع كل مهمة فرعية الى الطبقة الثالثة لتنفيذ المهمة الفرعية عن طريق مدير العقدة، مع ملاحظة أن مدير الخدمة يوزع المهام الفرعية الى مدير العقدة بالاعتماد على مجموعة من المعاملات كتوفر العقدة (مساحة المعالجة) والذاكرة المتبقية ومجال النقل.

رابعاً: مقارنة بين خوارزميات موازنة الحمل في الحوسبة الشبكية:

سنناقش في هذا القسم الخوارزميات المختلفة التي شرحنا عنها مسبقاً بالاعتماد على التحديات التي ناقشناها في القسم الأول. كما ناقشنا سابقاً فبعض الحلول المقترحة من خوارزميات موازنة الحمل قد تناسب بعض الحالات ولا تناسب أخرى. فالخوارزميات الثابتة فعالة جداً من حيث التكاليف والنفقات (In terms of overhead) حيث أنها ليست بحاجة لمراقبة المصادر (العقد والمخدمات) وقت التنفيذ لذلك تكون فعالة جداً في البيئات الثابتة أو المستقرة حيث أن الخصائص التشغيلية لا تتغير مع الزمن والحمولات عادة ما تكون متماثلة وثابتة، على الجانب الآخر فإن الخوارزميات المترامنة أو الديناميكية تقدم حلول أفضل حيث تستطيع التكيف مع الحمل وقت التنفيذ بالاعتماد على خصائص الموارد وقت التنفيذ، ولكن هذه الميزات تضيف تكاليف أكبر على النظام حيث أن المراقبة المستمرة والتحكم بالموارد قد تسبب الاختناقات والتأخيرات، بعض خوارزميات موازنة الحمل الديناميكية المقترحة حديثاً تحاول التقليل من التكاليف عن طريق نماذج توزيع مهمات جديدة.

في الجدول رقم (1) يظهر مقارنة بين مجموعة الخوارزميات التي تمت مناقشتها لتظهر ايجابيات وسلبيات كل خوارزمية. فمثلا خوارزمية ال INS تتعامل مع مشكلة موازنة الحمل بشكل ديناميكي ولكنها بنفس الوقت تتسم بالتعقيد لذلك يجب تغيير الهيكل العام للخوارزمية لتحقيق النتائج المرجوة بأقل درجة من التعقيد. كذلك الأمر فإن خوارزمية ال CLDBM حلت مشكلة الحاجة الى مسؤول بشري كل الوقت للتحكم بالنظام ولكنها بنفس الوقت تقدم نموذجاً للتحكم المركزي الذي يؤدي الى انهيار كامل للنظام في فشل المتحكم المركزي باي وقت ضمن عمل النظام لذلك فإن الحصول على نسخة احتياطية من المتحكم المركزي قد يحل المشكلة عند انهيار النظام. ونلاحظ أيضاً في نمط مستعمرة النمل Ant Colony ظهور نموذج اللامركزية الذي يقدم حل جيد لمشكلة الانهيار من نقطة واحدة ولكنه بنفس الوقت يزيد عبء الشبكة نتيجة العدد الكبير من النمل المتحرك وعدم اعتبار مجموعة من المعاملات التشغيلية مما يؤثر على أداء النظام، يمكن تحسين هذه الخوارزمية عن طريق الأخذ بالاعتبار لحالة العقد والموارد المتوفرة حالياً بالإضافة للحد من عدد النمل المستخدم في عمليات البحث عن طريق تقديم متحكمات تقلل مستويات التشعب أثناء عمليات البحث. في خوارزمية DDFTP يكون التحكم بأقل مستوى له كما أنه لسنا بحاجة المراقبة وقت التنفيذ للتعامل مع التغييرات الحاصلة ونتيجة لذلك فهي تحقق فعالية عالية لموازنة الحمل، ولكن حتى الآن فإنها تحتاج بعض التحسينات لاستغلال الموارد بشكل أفضل، كالحل من التكرارات المطلوبة مع الحفاظ على نفس المستوى من الأداء وذلك قد يكون ممكناً بالأخذ بعين الاعتبار التكرارات الفرعية مع مستوى معين من التداخل والذي يؤدي بدوره الى استغلال الموارد بشكل أفضل وتحقيق الحد الأدنى من التكاليف في موازنة الحمل.

الجدول رقم (1) إيجابيات وسلبيات خوارزميات موازنة الحمل

الخوارزمية	الإيجابيات	السلبيات
INS	-قادرة على التعامل مشاكل موازنة الحمل بشكل ديناميكي	-غير تنبؤية بالتصرف المتوقع للعقد -معددة عند التنفيذ -لا تأخذ بالاعتبار الا بعض المعاملات التشغيلية كالمسافة والزمن
ESWLC	-تحقق نتائج أكثر دقة من WLC	-معددة -تنبؤية تتطلب بيانات موحدة ووقت معالجة طويل
CLDBM	-تحل مشكلة خوارزمية جولة روبن -تحل من المسؤولية البشرية	-عدم اعتبار امكانيات العقد -نقطة انهيار وحيدة(مركزية) -لا تناسب العتبة جميع الحالات
Ant Colony	-أفضل سيناريو هو ايجاد العقدة الخاملة بداية البحث -غير مركزية -يستطيع النمل جمع المعلومات بسرعه	-تكاليف باهظة بسبب العدد الضخم للنمل -نقط البدء وعدد النمل غير واضح -لا تؤخذ حالة العقد بالاعتبار بعد زيارة النمل لها
Enhanced MapReduce	-تكاليف أقل بسبب تخفيض المهام	-وقت معالجة ضخم -الحد من امكانيات المهام لا يؤخذ بالاعتبار
VM Mapping	-طريقة حساب موثوقة	-نقطة فشل وحيدة -لا تأخذ بالحسبان حمولة الشبكة وامكانيات العقد
DDFTP	-سريعة -تحميل موثوق للملفات	-التكرارات الكاملة للملفات تتطلب مساحة تخزين كبيرة في كل العقد
LBMM	موثوقية بتوزيع المهام على العقد	-أبطى من الخوارزميات الاخرى لان عملها يمر خلال ثلاث طبقات لتنتم معالجته

الجدول رقم (2) يظهر المقارنة بين الخوارزمية آفة الذكر من حيث التحديات المطروحة في الفصل الثاني من هذه المقالة فمثلا نلاحظ أن الخوارزمية الوحيدة التي تتجنب تكرار البيانات والاستخدام المتكرر لمساحات التخزين هي ال INS ولكنها خوارزمية مركزية ومعقدة وامكانية الفشل من نقطة واحدة واردة. على الجانب الآخر فان خوارزمية DDFTP تعتمد على الموارد المتكررة ولا تقلل من حجم مساحة التخزين المطلوبة ولكنها نموذج للامركزية الديناميكية في موازنة الحمل وهي أيضا أكثر بساطة لتحميل البيانات المخزنة ويمكن تحسينها بحيث تقلل من حجم مساحة التخزين عن طريق النسخ الجزئي. بشكل عام كل خوارزمية تعالج مجموعة من هذه التحديات لتصبح مناسبة لحالة معينة. مثلا الخوارزميات INS و CLBDM و VM Mapping لديها نقطة الفشل الوحيدة لذلك فهي تعمل بشكل جيد جدا في البيئات المستقرة حيث تكون موثوقية الموارد عالية جدا. كل الخوارزميات باستثناء Ant Colony و VM Mapping التعامل بقوة مع الانظمة الموزعة، لذلك فهي أكثر مناسبة للسحابات العامة من هاتين الخوارزميتين. كذلك فان الكل باستثناء ال DDFTP تنتج تكاليف كبيرة على الشبكة. وكنتيجة لذلك فإنه عندما تسوء أحوال الشبكة فكل الخوارزميات ستعاني بشكل كبير من التأخيرات التي ستؤخر بدورها عملية موازنة الحمل الكلية، بينما في خوارزمية ال DDFTP هناك قدرة أكبر للتعامل مع تلك التأخيرات فهي ليست بحاجة للاعتماد على التحكم ومراقبة الموارد وقت التنفيذ.

الجدول رقم (2) مقارنة بين خوارزميات موازنة الحمل

الخوارزمية	التكرار	السرعة	عدم التجانس	تكاليف الشبكة	موزعه مكانيا	تعقيد التنفيذ	التسامح مع الخطأ
INS	جزئي	معتدل	نعم	نعم	نعم	عالي	لا
ESWLC	كامل	سريع	نعم	نعم	نعم	عالي	نعم
CLBDM	كامل	بطيء	نعم	نعم	نعم	منخفض	لا
Ant Colony	كامل	سريع	لا	نعم	لا	عالي	نعم
MapReduce	كامل	بطيء	نعم	نعم	نعم	عالي	نعم
VM Mapping	كامل	سريع	نعم	نعم	لا	عالي	نعم
DDFTP	كامل	سريع	نعم	لا	نعم	منخفض	نعم

الاستنتاجات والتوصيات:

لقد استعرضنا في هذه المقالة مجموعة من خوارزميات موازنة الحمل في الحوسبة السحابية وناقشنا التحديات التي يجب معالجتها للحصول على أكثر خوارزميات موازنة الحمل فعالية كما ناقشنا فوائد واضرار هذه الخوارزميات، ومن ثم قمنا بمقارنة الخوارزميات المذكورة بالاعتماد على التحديات آفة الذكر. بحثنا عن خوارزمية [20] DDFTP تركز على فعالية موازنة الحمولة وتعرفنا على أساسيات تحسين هذه الخوارزمية وجعلها أكثر فاعلية في موازنة الحمولة واستغلال الموارد. خوارزمية DDFTP الحالية تتسامح مع التأخيرات وتستطيع التعامل مع الموارد المتجانسة وتتمتع توزيع بنمط موزع للمهام المختلفة لكنها تعتمد على النسخ المتعدد للملفات على عدة مواقع وهذا يؤدي بدوره الى استهلاك اكبر للمساحات التخزينية لذلك نطمح للعمل المستقبلي المتجسد بجعل هذه الخوارزمية أكثر مواءمة للبيئات السحابية وأكثر فعالية من ناحية التخزين السحابي.

References:

- [1] Randles, M., D. Lamb and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing," in Proc. IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), Perth, Australia, April 2010.
- [2] Rimal, B. Prasad, E. Choi and I. Lumb, "A taxonomy and survey of cloud computing systems." In proc. 5th International Joint Conference on INC, IMS and IDC, IEEE, 2009.
- [3] Buyya R., R. Ranjan and RN. Calheiros, "InterCloud: Utility-oriented federation of cloud computing environments for scaling of application services," in proc. 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), Busan, South Korea, 2010.
- [4] Foster, I., Y. Zhao, I. Raicu and S. Lu, "Cloud Computing and Grid Computing 360-degree compared," in proc. Grid Computing Environments Workshop, pp: 99-106, 2008.
- [5] Grosu, D., A.T. Chronopoulos and M. Leung, "Cooperative load balancing in distributed systems," in Concurrency and Computation: Practice and Experience, Vol. 20, No. 16, pp: 1953-1976, 2008.
- [6] Ranjan, R., L. Zhao, X. Wu, A. Liu, A. Quiroz and M. Parashar, "Peerto- peer cloud provisioning: Systems and Applications, pp: 195-217, 2010.
- [7] Radojevic, B. and M. Zagar, "Analysis of issues with load balancing algorithms in hosted (cloud) environments." In proc.34th International Convention on MIPRO, IEEE, 2011.
- [8] Sotomayor, B., RS. Montero, IM. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," in IEEE Internet Computing, Vol. 13, No. 5, pp: 14-22, 2009.
- [9] Nishant, K. P. Sharma, V. Krishna, C. Gupta, KP. Singh, N. Nitin and R. Rastogi, "Load Balancing of Nodes in Cloud Using Ant Colony Optimization." In proc. 14th International Conference on Computer Modelling and Simulation (UKSim), IEEE, pp: 3-8, March 2012.
- [10] Zhang, Z. and X. Zhang, "A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation." In proc. 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), IEEE, Vol. 2, pp:240-243, May 2010.
- [11] Kolb, L., A. Thor, and E. Rahm, E, "Load Balancing for MapReducebased Entity Resolution," in proc. 28th International Conference on Data Engineering (ICDE), IEEE, pp: 618-629, 2012.
- [12] Gunarathne, T., T-L. Wu, J. Qiu and G. Fox, "MapReduce in the Clouds for Science," in proc. 2nd International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, pp:565-572, November/December 2010.
- [13] Ni, J., Y. Huang, Z. Luan, J. Zhang and D. Qian, "Virtual machine mapping policy based on load balancing in private cloud environment," in proc. International Conference on Cloud and Service Computing (CSC), IEEE, pp: 292-295, December 2011.
- [14] , T-Y., W-T. Lee, Y-S. Lin, Y-S. Lin, H-L. Chan and J-S. Huang, "Dynamic load balancing mechanism based on cloud storage" in proc. Computing, Communications and Applications Conference (ComComAp), IEEE, pp:102-106, January 2012.
- [15] Ren, X., R. Lin and H. Zou, "A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast" in proc. International Conference on Cloud Computing and Intelligent Systems (CCIS), IEEE, pp: 220-224, September 2011.

- [16] Lee, R. and B. Jeng, "Load-balancing tactics in cloud," in proc. International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), IEEE, pp:447-454, October 2011.
- [17] Al-Jaroodi, J. and N. Mohamed. "DDFTP: Dual-Direction FTP," in proc. 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), IEEE, pp:504-503, May 2011.
- [18] Wang, S-C., K-Q. Yan, W-P. Liao and S-S. Wang, "Towards a load balancing in a three-level cloud computing network," in proc. 3rd International Conference on. Computer Science and Information Technology (ICCSIT), IEEE, Vol. 1, pp:108-113, July 2010.
- [19] Sang, A., X. Wang, M. Madihian and RD. Gitlin, "Coordinated load balancing, handoff/cell-site selection, and scheduling in multi-cell packet data systems," in Wireless Networks, Vol. 14, No. 1, pp: 103- 120, January 2008.
- [20] Mohamed, N. and J. Al-Jaroodi, "Delay-tolerant dynamic load balancing," in proc. (HPCC), pp:237-245, September 2011.
- [21] Nasser Ali Nasser "Counting Classes Defined by Prime Numbers", pp:2663-4252, 2017.
- [22] Nasser Ali Nasser "Polygonal Complexity Counting Classes" pp:2079-3081, 16-9-2018.