تقليل تلوث كاش الويب في خوارزمية الكاش GDFS باستخدام مسافة غوغل المقيّسة NGD للتشابه الدلالي

د. إيهاب الديباجة*

(تاريخ الإيداع 12 / 7 / 2021. قُبِل للنشر في 27/ 8 / 2021)

□ ملخّص □

إنّ الكاش هو أحد التقنيات التي تساهم بشكل فعال في تحسين كفاءة أنظمة استعادة المعلومات Information Retrieval Systems، ويعتبر كاش الويب، وهو تقنية الكاش الخاصة بشبكة الويب، أحد الأدوات لتحسين زمن استجابة أنظمة المعلومات القائمة على الويب Web-Based Information Systems (WIS)، ويتم ذلك من خلال خوارزميات تختلف فيما بينها في عمل تابع استبدال الخوارزمية.

نبحث في هذه الورقة تقليل تلوث الكاش الساخن Hot Pollution والتلوث البارد Cold Pollution والذي قد يصيب عمل خوارزمية GDFS (Greedy Dual Frequency Size) والتي تعتبر خوارزمية أساسية ومرجعية في مجال كاش الويب، وذلك من خلال تحسين تابع الاستبدال بإدخال مفهوم التشابه الدلالي بين الأغراض المخزنة في الكاش، باستخدام مسافة غوغل المقيّسة Normalized Google Distance (NGD) إلى عمل تابع الاستبدال لهذه الخوارزمية.

بينت النتائج أن إدخال مفهوم التشابه الدلالي إلى عمل هذه الخوارزمية أدى إلى تقليل تلوث الكاش، من خلال التحكم الأفضل ببقاء الأغراض في الكاش، والمساهمة مع وظيفة تابع الاستبدال الأساسية في تقييم مدة بقاء الأغراض في الكاش، وبالتالي تحسين نسبة الإصابة Hit Rate للأغراض من ذاكرة الكاش بدلاً من مصدر البيانات الأساسي.

الكلمات المفتاحية: كاش الويب – نظم استعادة المعلومات – تلوث الكاش – GDFS – Greedy – Dual– الويب الدلالي – مسافة غوغل المقيسة

* مدرس – كلية الهندسة – قسم المعلوماتية – جامعة المنارة– اللاذقية – سورية.Email: ihab.aldibaja@manara.edu.sy

Reducing Cache Pollution in GDFS Cache Algorithm Using Normalized Google Distance (NGD) for Semantic Similarity

Dr. Ihab Aldibaja*

(Received 12 / 7 / 2021. Accepted 27 / 8 / 2021)

□ ABSTRACT □

Cache is considered as one of techniques that effectively contributes in enhancing performance of information retrieval systems. Web cache which is the cache technology specified for web is one of the tools for enhancing response time in web-bases information systems (WIS). That's can be done by algorithms which varying in its replacement function. We study in this paper how to reduce hot cache pollution and cold cache pollution which may affects GDFS (Greedy Dual Frequency Size) web cache algorithm. This algorithm is considered as a fundamental web cache algorithm, by improving replacement function using Normalized Google Distance (NGD) between cache objects.
Results shows that employing semantic similarity concept to GDFS replacement function have had reduced cache pollution by improving the control of existence of objects in cache memory and collaboration with original function of GDFS algorithm in evaluation the existence of objects in cache and thus improving hit rate of objects from cache memory instead of original data source

Keywords: Web Cache – information Retrieval Systems, Cache Pollution – GDFS – Greedy Dual, Semantic Web - Normalized Google Distance.

* **Assistant Professor – Faculty of Engineering – Informatics Department – Almanara University – Lattakia – Syria. Email: ihab.aldibaja@manara.edu.sy**

مقدمة:

إن التطور الكبير في حجم المعلومات المخزنة في أنظمة المعلومات يفرض حتماً تطوير خوارزميات استعادة المعلومات من هذه الأنظمة وايصالها إلى المستخدم بأسرع وقت ممكن، وتأتي خوارزميات الكاش كجزء من التقنيات التي تساهم في تحسين استجابة الأنظمة لطلبات الوصول الى المعلومات، ويعتبر كاش الويب أحد هذه التقنيات التي تساهم في تحسين أداء الأنظمة القائمة على الويب.[1]

يوجد العديد من خوارزميات الكاش التي تتبع سياسات مختلفة لاستبدال الأغراض الموجودة في الذاكرة، وبالتالي التعامل بأفضل شكل مع حجم ذاكرة الكاش، إلا أنه لكل من هذه الخوارزميات إيجابيات وسلبيات، ويعتبر تلوث الكاش أحد السلبيات التي تعاني منها هذه الخوارزميات[2]، وإن تطوير توابع الاستبدال لهذه الخوارزمية هو أحد النقاط البحثية الأساسية التي يتم العمل عليها لرفع كفاءة خوارزميات الكاش المختلفة.[3]

مشكلة البحث

تعاني خوارزميات الكاش من عدد من المشاكل، تبرز هذه المشاكل بشكل متزايد مع ازدياد حجم المعلومات المخزنة في أنظمة المعلومات، وانفجار هذا الحجم، مقارنة مع محدودية الموارد الموجودة، ومع وجود نمط طلبات معين للبيانات، فإن مشكلة تلوث الكاش تبرز بشكل واضح وتؤدي بشكل ملحوظ إلى تراجع أداء الخوارزمية المستخدمة لإدارة الكاش[4]. تم تطبيق البحث في الشركة العامة لمرفأ اللاذقية والتي تشغل نظام معلومات مطور محلياً. إن نموذج الطلب الموجود في المرفأ (طلب عالٍ على أغراض لفترة محددة لدى وصول السفينة، أو طلب أغراض لمرة واحدة مثل بيانات دخول السفينة) يوفر بيئة مناسبة لحصول تلوث الكاش، حيث لوحظ انخفاض أداء خوارزمية كاش الويب المدروسة بسبب تلوث الكاش، وبالتالي وجود أغراض غير ضرورية في ذاكرة الكاش تشغل مساحة يمكن استخدامها لتحسين أداء الخوارزمية.

أهمية البحث وأهدافه:

أهمية البحث

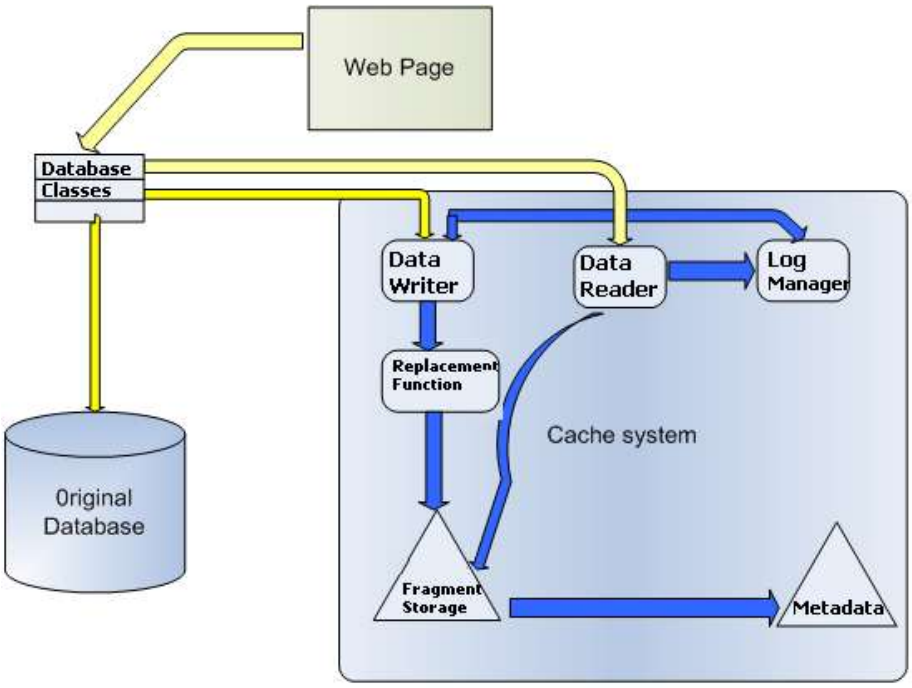
تكمن أهمية البحث في دراسة أثر تطوير تابع استبدال الخوارزميات المدروسة باستخدام مسافة غوغل المقيسة على تخفيض تلوث الكاش البارد والساخن، وبالتالي إخراج الأغراض من الذاكرة في الوقت المناسب، وهو ما سينعكس إيجاباً على أداء نظام المعلومات في المؤسسة، خصوصاً في بيئة عمل تتطلب عمليات مكثفة وسريعة للوصول إلى طيف متنوع وواسع من المعلومات، وبالتالي، تحسين أداء هذه النظم من ناحية سرعة الاستجابة لطلبات الزبائن *Clients*، ومن ناحية تخفيض عرض الحزمة المتبادل بين الزبائن والمخدمات على الشبكة.

أهداف البحث

يهدف البحث إلى إدخال معيار التشابه الدلالي، باستخدام مسافة غوغل المقيسة للأغراض الموجودة في ذاكرة الكاش، إلى آلية عمل تابع الاستبدال لخوارزمية *GDFS* ومن ثم دراسة أثر هذا التعديل على أداء الخوارزمية لدى تعرضها لنموذج طلبات يسبب تلوث الكاش.

طرائق البحث ومواده:

تم بناء مخدم ويب كاش لاختبار التعديل المقترح كما هو مبين في الشكل 1، مكون من الوحدات الأساسية التالية: وحدة الاتصال بقاعدة البيانات الأصلية *Original Datamasse Connection Unit*، وحدة تخزين أجزاء الصفحات *Page Segment Storage Unit*، مخزن البيانات الوصفية *Data Description Storage* : وهو عبارة عن قاعدة بيانات تحتوي على البيانات الوصفية *Metadata* المقترنة بجزء ديناميكي من الصفحة، قارئ الكاش *Cache Reader*: يقوم هذا المكون بقراءة جزء الصفحة من الذاكرة المؤقتة ويقوم بتحديث البيانات الوصفية المرتبطة به مثل عدد مرات الزيارة *Hit*، كاتب الكاش *Cache Writer*: يقوم هذا المكون بتخزين جزء الصفحة في ذاكرة الكاش وتحديث المعلومات الوصفية المرتبطة بها، مخزن سجل المناقلات *Log Manager Storage*: يقوم هذا المكون بتخزين الحركات التي تتم على الكاش.



الشكل(1) المخطط الصندوقي المبسط لنظام الكاش

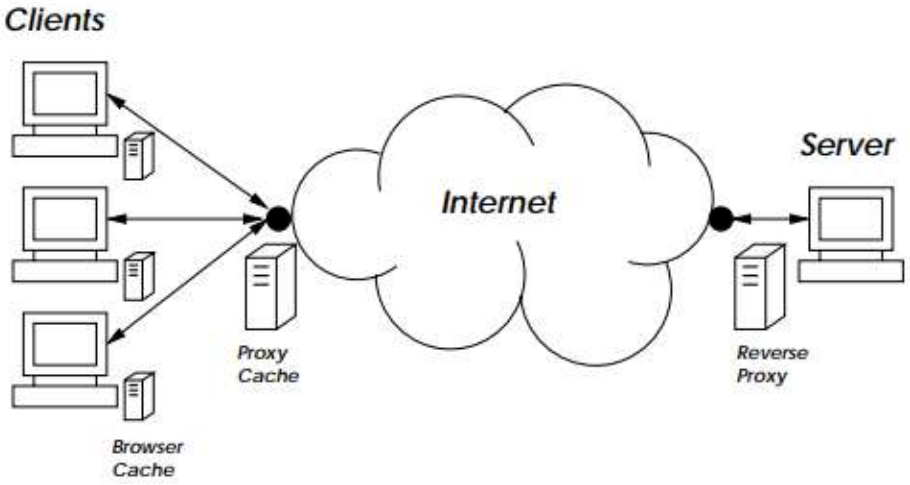
1- مجتمع وعينة البحث

تم تحليل سجلات الوصول الى الأنظمة المعتمدة على الويب في النظام التجريبي وعددها مليون سجل لوصول المستخدمين إلى صفحات الويب الخاصة بأنظمة المرفأ بين عامي 2018 -2020، تم الحصول عليها من سجلات مخدم الويب المشغل لهذه الأنظمة *IIS (Internet Information Service Log)*، وتم عمل محاكاة لهجوم يسبب تلوث الكاش بالشكل البارد (طلب صفحات غير متكررة) وبالشكل الساخن (طلب صفحات معينة بتردد عالٍ غير مألوف لعمل النظام)، ثم تم استخلاص القيم المطلوبة من هذه السجلات، وتحويلها لشكل احصائي، ورسم المخططات البيانية المناسبة، واستنتاج التغييرات الحاصلة نتيجة التطوير المقترح.

2-الإطار النظري للبحث

2-1- كاش الويب:

يمكن تحقيق كاش الويب في 3 مستويات أساسية على الشبكة، كما هو مبين في الشكل 2: المستوى الأول يمكن تحقيقه من جهة متصفح الويب الذي يقوم بتخزين المستندات التي تم الوصول إليها مؤخراً، إلا أن أغلب حلول الكاش يتم عملها في المستوى الثاني، على مخدم وكيل *Proxy Server*، والذي يكون عادة قريباً من بوابات الشبكات *Network Gates*، حيث تكون هذه المخدمات شفافة *Transparent* لمجموعة مستخدمي الشبكة مثل المستخدمين ضمن شركة او المستخدمين التابعين لمزود خدمات انترنت *ISP Internet Service Provider*، والمستوى الثالث – وهو المستوى الذي تم تطبيق البحث عليه – يكون على مخدم الويب نفسه، وهذه الحالة تطابق وجود نظام المعلومات ضمن مخدمات المنظمة المطلوب وضع حلول الكاش لها.[5]



الشكل(2) مستويات الكاش في الويب

2-2- خوارزمية *GDFS*:

إنّ خوارزمية Greedy Dual هي تعميم للخوارزمية المعروفة LRU Least Recently Used مراعيةً احتياجات التخزين المؤقت لأغراض الويب، تقوم الخوارزمية على مبدأ الاحتفاظ بقيمة تقديرية H(p) لكل مستند أو صفحة p تم تخزينها، كما هو مبين في الشكل 3 الذي يضم الترميز الزائف للخوارزمية.

إنّ قيمة المفتاح للمستند P تحسب وفق المعادلة:

$$K(p) = Clock + F(p) * C(p)/S(p) \qquad (1)$$

حيث $F(p)$ هو عدد مرات النفاذ للمستند. عندما يتم استعادة المستند p إلى الذاكرة لأول مرة فإن قيمة التردد الخاص به $F(p)$ تكون مساوية للواحد. C(p) هي التكلفة التي حصلت نتيجة تخزين p. تم اعتبار C(p) هي كلفة المعالجة للاستعلام اللازم لتخزين الغرض CPU Cost، وكلفة الدخل والخرج للحصول على الغرض I/O Cost، وهي كلف يمكن الحصول عليها من محرك قاعدة البيانات.

إذا تمّ طلب النفاذ إلى p مجدداً في الذاكرة، فإن قيمة التردد الخاص به تزداد بمقدار 1:

$$F(p) = F(p) + 1 \qquad (2)$$

حيث أنّ $S(p)$ هو حجم المستند p، وبالتالي، يتم هنا أخذ حجم المستند بعين الاعتبار.[6]

تضيف هذه الخوارزمية عوامل الحجم وتردد طلب الغرض إلى كلفة الغرض، وبذلك تصبح المعادلة التي تصف مفتاح الغرض هي:

$$K(p)= Clock +F(p)*\ ((CPU\ Cost + IO\ Cost))/S(p) \quad (3)$$

يبين الشكل (8−3) الترميز الزائف للخوارزمية.

Algorithm Greedy Dual Frequency Size

Initialize L=0 //L is Queue Clock
Process each requested document in turn:
Let p be the current document:
IF p is already in cache
K(p) = L + C(p) //C(p) Object Cost , K(p) Object Key
ELSE
WHILE there is not enough space for p
Let L = $min_{q\epsilon M}$ k(q), for all q in cache M
Evict q such that K(q) = L
END WHILE
Deposit p into cache and set
K(p) = L + C(p)
END IF

الشكل (3) الترميز الزائف لخوارزمية *GDFS*

2−3− تقييم أداء خوارزمية كاش الويب:

يوجد عدد من المعايير التي يمكن أن تستخدم لتقييم أداء خوارزمية كاش الويب، إلا أن المعيار الأكثر أهمية واستخداماً هو نسبة الإصابة *Hit Rate*، تعكس نسبة الإصابة مقدار التحسن الذي تمّ الحصول عليه، باستخدام خوارزمية "ويب كاش"، في تلبية طلب المستخدم بشكل أسرع، نتيجة الحصول على الغرض من ذاكرة الكاش بدلاً من المخدم الأصل.[7]

وتعرّف نسبة الإصابة على أنها النسبة المئوية من الطلبات التي يمكن تلبيتها من الكاش، مقارنةً مع عدد الطلبات الكلي، وهي تكتب بالمعادلة:

$$Hit\ Ratio = \frac{\sum_{i\in R} h_i}{\sum_{i\in R} f_i} \quad (4)$$

عندما تكون هذه القيمة عاليةً، فإنّ ذلك يدلّ على أنه تمّ استرجاع نسبة عالية من الطلبات من التخزين المؤقّت، وهو عادةً النتيجة المرجوّة لمعظم مديري النظم.

2−4− تلوث الكاش *Cache Pollution*

إن مصطلح تلوث الكاش يعني أن الكاش تضم أغراضاً غير مستخدمة بتواتر عال، وذلك في المدى المنظور. يسبب تلوث الكاش انخفاض فعالية الكاش وإنقاص أداء الخوارزمية، وحتى عند تخصيص حجم كبير لذاكرة الكاش، فإن ذلك لن يكون فعالاً لأن زمن البحث عن غرض ضمن هذا الحجم الكبير سوف يؤدي إلى انخفاض كبير في أداء الخوارزمية، فضلاً عن الزيادة الغير مرغوبة في استهلاك موارد المعالجة.[9][8]

يمكن تقسيم تلوث الكاش إلى التلوث البارد *Cold Pollution* والتلوث الساخن *Hot Pollution*.

إن السبب الأساسي لتلوث الكاش البارد هو الأغراض التي تبقى في الذاكرة لأنه تم الوصول إليها مرة واحدة، ومن ثم لم يتم الرجوع لها مرة أخرى، سوف تبقى هذه الأغراض في الذاكرة لوقت طويل إلى أن يتم إزاحتها من مكدس التخزين الخاص بالكاش مع وصول أغراض جديدة إلى الذاكرة، فبفرض أن طول مساحة التخزين هو D عندها سوف يتم الانتظار D وحدة زمنية ليتم إخراج الغرض من الكاش. يعتبر التلوث البارد للكاش أحد أبرز نقاط الضعف لخوارزمية *LRU* لدى تطبيقها في مجال الويب.[4]

يشير تلوث الكاش الساخن *Hot Cache Pollution* الى الأغراض التي كانت مطلوبة بشكل متكرر في مرحلة معينة، ولم تعد مطلوبة الان، وهذا سوف يسبب بقاء الأغراض في الكاش بدون فائدة لحين قدوم أغراض ذات تواتر أعلى لإخراجها من الكاش، وهذا التأثير يكون واضحاً في خوارزمية *LFU*.

يعتبر الكشف عن سبب تلوث الكاش أمراً معقداً، ولا يمكن ضبطه بسهولة، لأنه من الوارد جداً أن يتم الخلط بين هذه الظاهرة وبين الطلبات الاعتيادية للزبائن، والتي قد تأخذ طابعاً خاصاً في بعض الأحيان (مثلاً الطلب على صفحة نتائج البكالوريا في لحظة معينة ولزمن قصير معين).[10]

2-5- مسافة غوغل المقيسة

تم في البحث تطوير تابع الاستبدال لخوارزمية *GDFS* من خلال دراسة التشابه الدلالي بين الأغراض الموجودة في ذاكرة الكاش باستخدام مسافة غوغل المقيسة، حيث تم إدخال قيمة هذه المسافة إلى تابع الاستبدال للخوارزمية، تعتمد هذه الطريقة على عدد الصفحات التي يعيدها محرك بحث معين كنتيجة للبحث لحساب التشابه الدلالي بين الكلمات، وعند استخدام محرك البحث*Google* تسمى هذه المسافة بمسافة غوغل المقيسة لقياس التشابه بين المصطلحات.[11]

تعرف مسافة غوغل المقيسة بالقانون:[12]

$$NGD(x,y) = \frac{max\{\log f(x), \log f\ (y)\} - \log f(x,y)}{\log N - \min\{\log f(x), \log f(y)\}} \qquad (5)$$

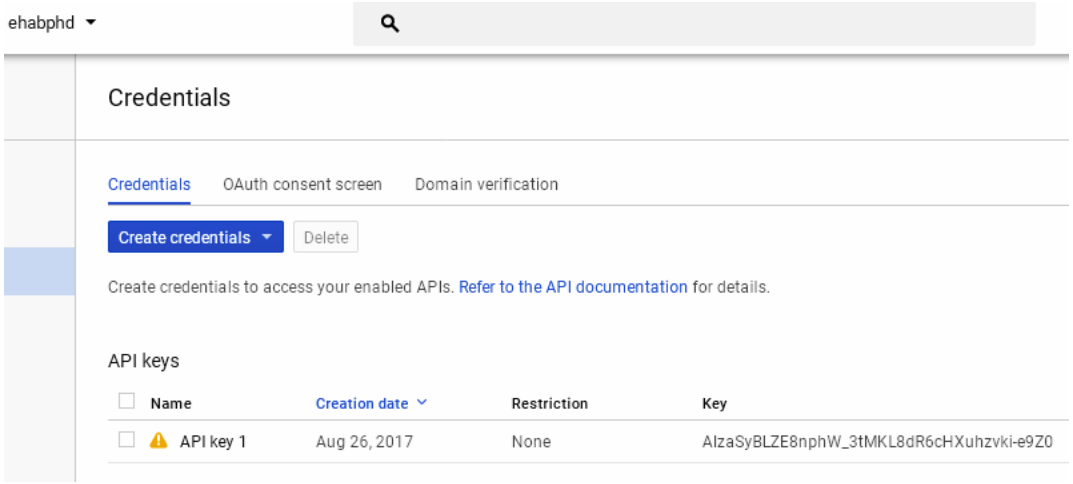
حيث $f(x)$,$f(y)$ هي عدد نتائج بحث غوغل للمصطلحات x,y على التوالي، $f(x,y)$ عدد نتائج بحث غوغل عن كل من x,y معا، N إجمالي عدد الصفحات التي تم فهرستها من قبل محرك البحث.

تمّ اعتماد مسافة غوغل المقيسة لدراسة التشابه الدلالي كونه يعتبر من أحدث مقاييس التشابه ضمن تصنيفات المقاييس، كما أعطى نتائج أفضل لتحديد التشابه الدلالي وفق الدراسات المرجعية التي تم العودة إليها.[13]

النتائج والمناقشة:

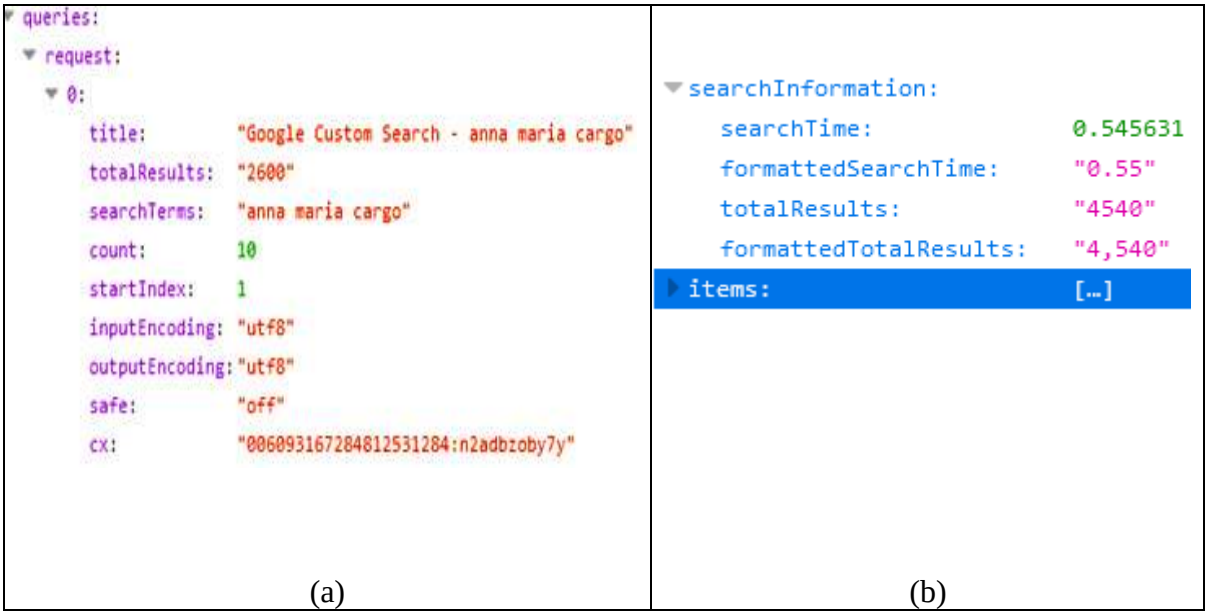
تصميم تابع الاستبدال المطور للخوارزمية:

تمّ التسجيل في موقع *Google* للوصول إلى خدمة *Google API*، التي تسمح بالوصول إلى نتائج البحث عن كلمات معينة بطريقة مؤتمتة يمكن معالجتها، للحصول على تردد ورود كلمة بحث معينة كما هو مبين في الشكل 4 والشكل 5.



الشكل (4) التسجيل في *Google API*

يقوم محرك البحث *Google* بإعادة نتيجة البحث على شكل ملف *XML*، كما هو مبين في الشكل 5. ويتضمن هذا الملف معلومات متنوعة، مثل: عدد نتائج البحث، والزمن اللازم للحصول على النتيجة من محرك البحث، كما هو مبين في الشكل (*5–b*). وتقوم وحدة المعالجة التي تمّ تصميمها في البحث باستخلاص القيم الخاصة بتكرار الكلمة من الحقل *totalresults* من أجل استخدامه في حساب *NGD*.



الشكل (5) مثال عن النتيجة التي يعيدها محرك البحث عن نتائج كلمة معينة

من أجل خوارزمية *GDFS*، تصبح المعادلة التي تقدم المفتاح الخاص بالأغراض الموجودة في الكاش، والتي يستخدمها تابع استبدال الخوارزمية من أجل تحديد الغرض المتوجب إخراجه من الكاش لإضافة غرض جديد هي:

$$K(p) = \left(sim(o_{new}, o_j)\right) + L + F(p) * \ (CPU\ Cost + IO\ Cost)/S(p) \qquad (6)$$

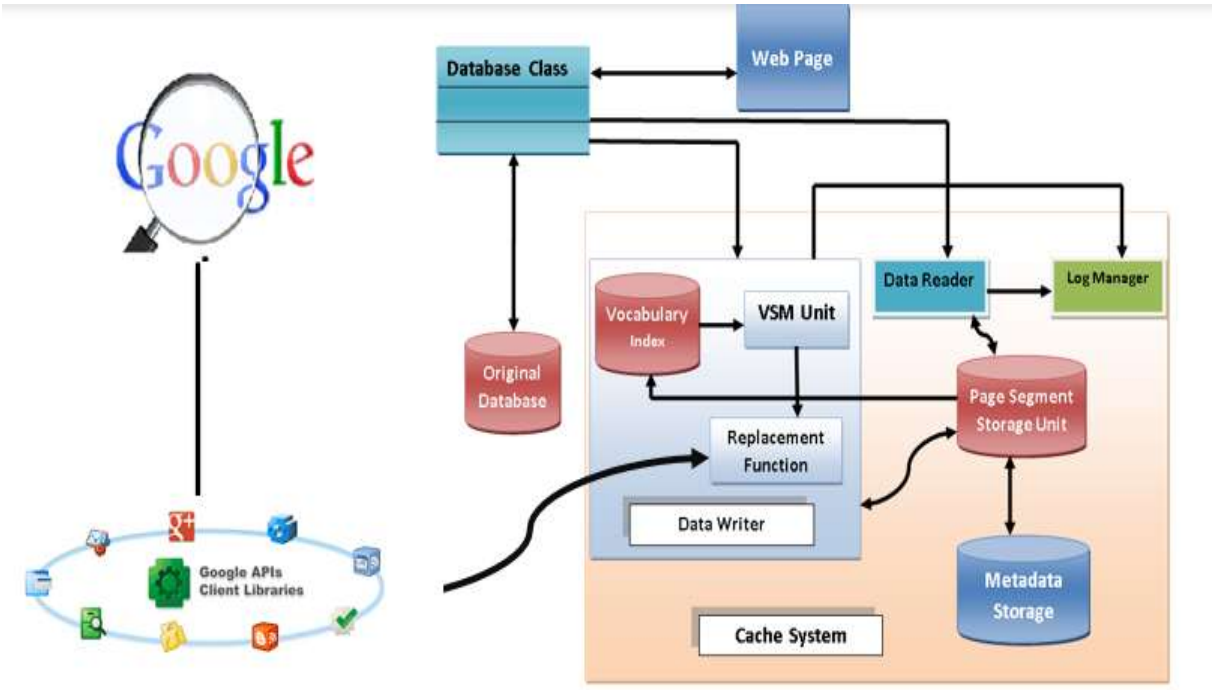
حيث o_{new} هو الغرض الجديد المطلوب إضافته إلى الكاش، و o_j هو غرض موجود سابقاً في الكاش، و $sim(o_{new}, o_j)$ تعطى بالمعادلة:[14]

$$sim(o_{new}, o_j) = \sum_{r=1}^{num} \sum_{q=1}^{num} 1 - NGD(k\_new_r, k\_current_q) \qquad (7)$$

حيث num هو طول الشعاع المعتمد في النظام لتمثيل الأغراض، k_new_r هي الكلمة المفتاحية ذات الترتيب r للغرض المطلوب إضافته إلى الكاش، $k\_current_q$ هي الكلمة ذات الترتيب q في الغرض الموجود حالياً في الكاش والذي تتم مقارنته مع الغرض الجديد، وبما أنّ *NGD* يعبر عن المسافة فإنّ *1-NGD* يعبر عن التشابه.[14]

تكرّر عملية المقارنة من أجل كل الأغراض الموجودة في الكاش لحساب الغرض الأبعد عن الغرض الحالي، حيث يتم إزالة الغرض ذو القيمة الأدنى من ذاكرة الكاش.

يبين الشكل (6) المخطط الصندوقي لمخدم "ويب كاش" المطور بعد إدخال وحدة حساب التشابه الدلالي اعتماداً على مسافة غوغل المقيسة، إلى تابع الاستبدال للخوارزميات المطوّرة.



الشكل (6) المخطط الصندوقي لمخدم "ويب كاش" المطور باستخدام مسافة غوغل المقيسة

يظهر في الشكل (6) ربط مخدم ويب كاش المطور في البحث مع محرك البحث غوغل باستخدام خدمة *Google API*، والتي يمكن من خلالها استعادة نتائج عمليات البحث عن المصطلحات من محرك البحث غوغل على شكل ملف *XML* برمجياً، ومن ثم تمرير المعلومات إلى تابع الاستبدال الذي يقوم بالاستفادة من هذه المعلومات لحساب مسافة غوغل المقيسة، ومن ثم إدخال المسافة إلى قيمة الأغراض الموجودة في الكاش بحيث تقوم الخوارزمية باتخاذ القرار بإخراج الأغراض من الكاش، وفق سياسة الاستبدال المستخدمة.

النتائج التجريبية

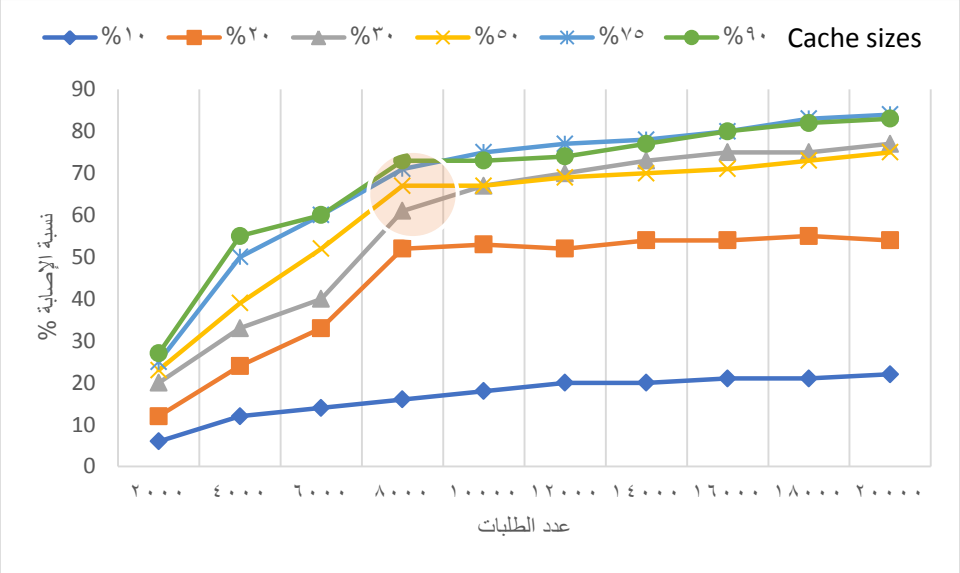
تم تجربة أحجام مختلفة لذاكرة الكاش، مع عدد طلبات مختلف في النظام المدروس للحصول على نقاط استقرار النظام، وهي النقاط التي يصبح فيها تحسن أداء الخوارزمية محدوداً مع زيادة الموارد المخصصة للكاش، حيث تعتبر هذه النقاط هي النقاط المرجعية من أجل إجراء التجارب والقياسات والمقارنات اللاحقة على النظام، والجدول 1 يبين القيم التي تم الحصول عليها لأداء خوارزمية *GDFS* مع أحجام مختلفة للذاكرة وأعداد طلبات مختلفة.

الجدول (1) معدل الإصابة لخوارزمية *GDFS* مع قيمة مختلفة لعدد الطلبات وأحجام مختلفة لذاكرة الكاش

حجم الذاكرة / عدد الطلبات	10%	20%	30%	50%	75%	90%
2000	6%	12%	20%	23%	25%	27%
4000	12%	24%	33%	39%	50%	55%
6000	14%	33%	40%	52%	60%	60%
8000	16%	52%	61%	67%	71%	73%
10000	18%	53%	67%	67%	75%	73%

12000	20%	52%	70%	69%	77%	74%
14000	20	54	73	70	78	77
16000	21	54	75	71	80	80
18000	21	55	75	73	83	82
20000	22	54	77	75	84	83

من الشكل (7) يتبين أن الخوارزمية المدروسة يستقر أداؤها نسبياً عند حجم 50% من الذاكرة الكلية التي تشغلها الأغراض التي تم طلبها، حيث تبلغ نسبة الإصابة 70% تقريباً.



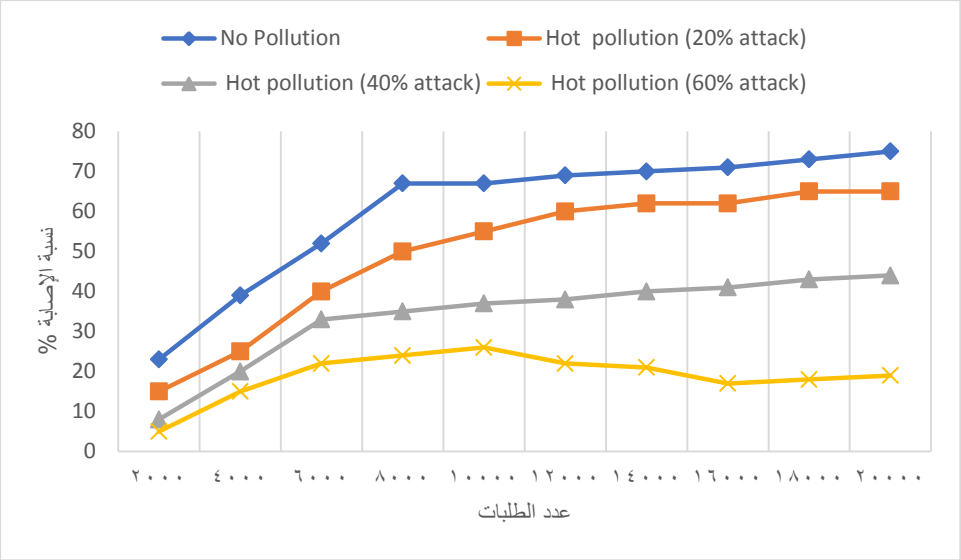
الشكل (7) معدل الإصابة لخوارزمية *GDFS* مع قيمة مختلفة لعدد الطلبات وأحجام مختلفة لذاكرة الكاش

لدى القيام بتجربة أداء الخوارزمية مع تجربة هجوم مسبب للتلوث الساخن (طلب أغراض غير متكررة بتواتر كبير) وجدنا النتائج المبينة في الجدول 2.

الجدول (2) معدل الإصابة لخوارزمية *GDFS* مع نسب مختلفة للتلوث الساخن وعدد طلبات مختلف

نوع التلوث / عدد الطلبات	No Pollution(%)	Hot pollution(%) (20% attack)	Hot pollution(%) (40% attack)	Hot pollution(%) (60% attack)
2000	23	15	8	5
4000	39	25	20	15
6000	52	40	33	22
8000	67	50	35	24
10000	67	55	37	26
12000	69	60	38	22
14000	70	62	40	21
16000	71	62	41	17
18000	73	65	43	18
20000	75	65	44	19

تبين النتائج كما يظهر في الشكل 8 انخفاض أداء الخوارزمية بسبب التلوث الساخن وبنسب مختلفة مع ازدياد مقدار التلوث (زيادة تردد طلب الأغراض بشكل شاذ *anomalies*)، وينخفض الأداء بشكل شديد بعد نسبة تلوث 40% من عدد الطلبات النظامي.

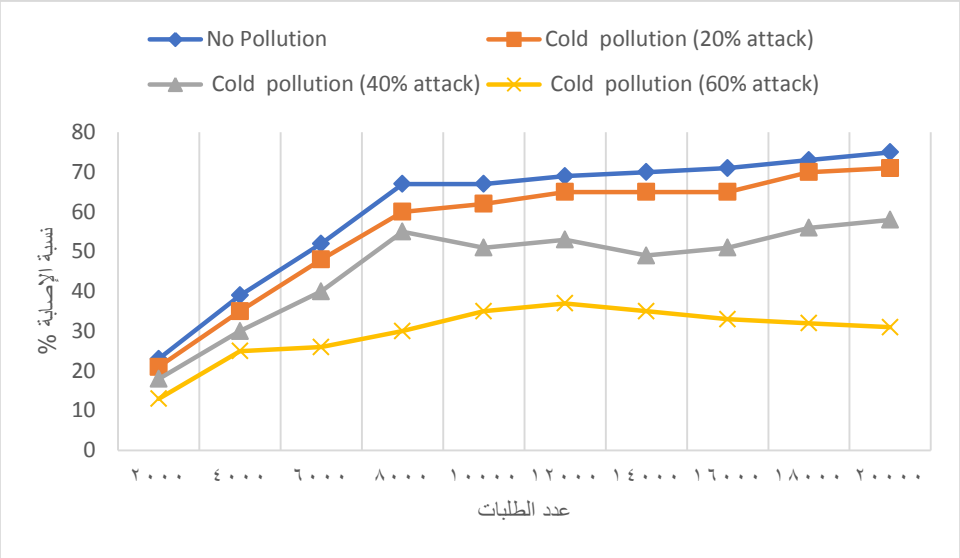


الشكل (8) معدل الإصابة لخوارزمية *GDFS* عند حجم ذاكرة *50%* ونسب مختلفة من الطلبات الزائفة (هجوم التلوث الساخن)

بتجربة التلوث البارد ظهرت النتائج في الجدول 3 والتي تم تمثيلها في الشكل 9، وهي تبين انخفاض أداء الخوارزمية أيضاً بسبب التلوث البارد (طلب أغراض عديدة لمرة واحدة وبقائها في الذاكرة).

الجدول (3) معدل الإصابة لخوارزمية *GDFS* مع نسب مختلفة للتلوث البارد وعدد طلبات مختلف

نوع التلوث / عدد الطلبات	No Pollution(%)	Cold pollution(%) (20% attack)	Cold pollution(%) (40% attack)	Cold pollution(%) (60% attack)
2000	23	21	18	13
4000	39	35	30	25
6000	52	48	40	26
8000	67	60	55	30
10000	67	62	51	35
12000	69	65	53	37
14000	70	65	49	35
16000	71	65	51	33
18000	73	70	56	32
20000	75	71	58	31



الشكل (9) معدل الإصابة لخوارزمية *GDFS* عند حجم ذاكرة *50%* ونسب مختلفة من الطلبات الزائفة (هجوم التلوث البارد)

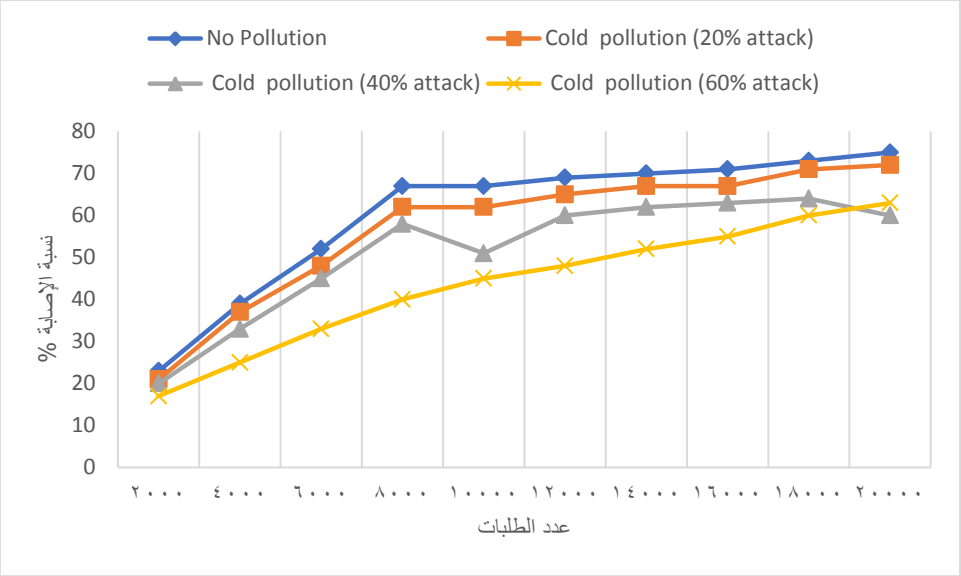
بمقارنة المخطط البياني في الشكل (9) مع المخطط البياني في الشكل (8) نجد أن الخوارزمية تتأثر بشكل أكبر بالتلوث الساخن، ويعود ذلك إلى وجود معيار التردد في مفتاح الخوارزمية، والذي يحد من بقاء الأغراض غير متكررة في الذاكرة، وهي حالة التلوث البارد، وبالتالي يتم إخراجها باكراً نسبياً، وهذا المعيار (معيار التردد) يتأثر بشكل كبير في التلوث الساخن كونه يتم طلب الأغراض الشاذة بشكل كبير، وبالتالي تحصل على قيم كبيرة لمفتاح الخوارزمية، ويبقيها في الذاكرة لفترة أطول.

تم إدخال مسافة غوغل المقيسة وفق المعادلات المذكورة سابقاً، ودراسة أداء الخوارزمية مع التلوث البارد عند عدد طلبات مختلف، والنتائج كانت كما هو مبين في الجدول 4.

الجدول (4) معدل الإصابة لخوارزمية *GDFS* مع نسب مختلفة للتلوث البارد ومع قيم مختلفة لعدد الطلبات

نوع التلوث / عدد الطلبات	No Pollution(%)	Cold pollution(%) (20% attack)	Cold pollution(%) (40% attack)	Cold pollution(%) (60% attack)
2000	23	21	20	17
4000	39	37	33	25
6000	52	48	45	33
8000	67	62	58	40
10000	67	62	51	45
12000	69	65	60	48
14000	70	67	62	52
16000	71	67	63	55
18000	73	71	64	60
20000	75	72	60	63

يبين الشكل (10) تحسن أداء الخوارزمية، حيث ارتفع معدل الإصابة وخصوصاً الحالة الأسوأ وهي 60%.



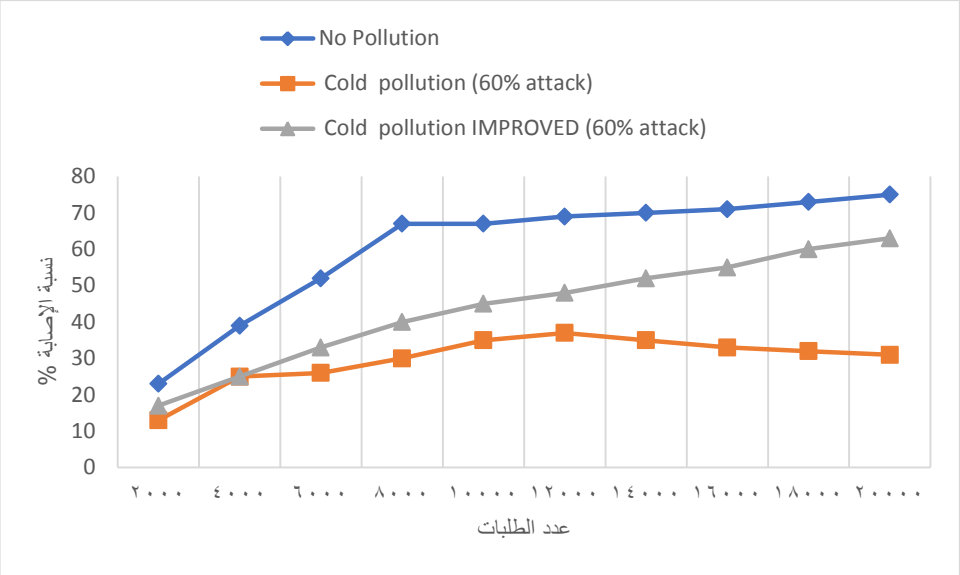
الشكل (10) معدل الإصابة لخوارزمية *GDFS* عند حجم ذاكرة *50%* ونسب مختلفة من الطلبات الزائفة (هجوم التلوث البارد)

يبين الجدول 5 والذي تم تمثيله بالشكل 11، مقارنةً للحالة الأسوأ للتلوث البارد قبل التحسين وبعده، ومقارنة أداء الخوارزمية المحسنة مع الخوارزمية الأصلية.

الجدول (5) مقارنة لمعدل الإصابة للخوارزمية الأساسية والخوارزمية المحسنة عند نسبة تلوث بارد *60%*

نوع التلوث / عدد الطلبات	No Pollution(%)	Cold pollution(%) (60% attack)	Cold pollution IMPROVED(%) (60% attack)
2000	23	13	17
4000	39	25	25
6000	52	26	33
8000	67	30	40
10000	67	35	45
12000	69	37	48
14000	70	35	52
16000	71	33	55
18000	73	32	60
20000	75	31	63

يظهر في الشكل (11) التحسن الملحوظ الذي طرأ على أداء الخوارزمية وخصوصاً الحالة الأسوأ بنسبة تلوث 60% ويزداد التحسن مع ازدياد عدد الطلبات، ومع أن التلوث بقي مؤثراً على أداء الخوارزمية، مقارنة مع أداء الخوارزمية بدون تلوث، إلا أن التحسن كان واضحاً في معدل الإصابة كما هو مبين في الشكل.



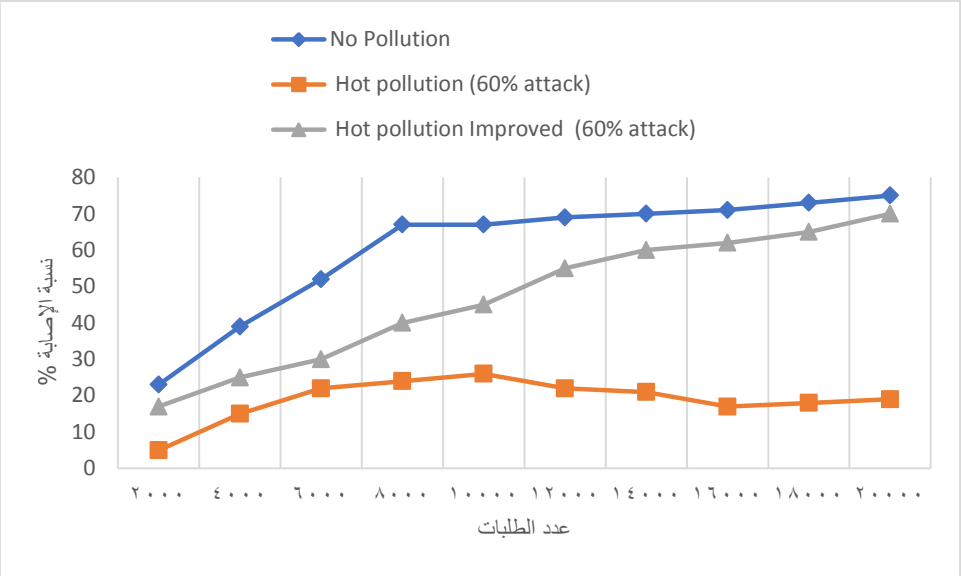
الشكل (11) التحسين الحاصل على الخوارزمية مع التلوث البارد بعد ادخال مفهوم التشابه الدلالي باستخدام *NGD* مع نسبة طلبات زائفة *60%*

يبين الجدول 6 مقارنة أداء الخوارزمية المحسنة مع الخوارزمية الغير محسنة والخوارزمية الأصلية لدى تعرضها للتلوث الساخن.

الجدول (6) مقارنة لمعدل الإصابة للخوارزمية الأساسية والخوارزمية المحسنة عند نسبة تلوث ساخن *60%*

نوع التلوث / عدد الطلبات	No Pollution(%)	Hot pollution(%) (60% attack)	Hot pollution Improved(%) (60% attack)
2000	23	5	17
4000	39	15	25
6000	52	22	30
8000	67	24	40
10000	67	26	45
12000	69	22	55
14000	70	21	60
16000	71	17	62
18000	73	18	65
20000	75	19	70

من الشكل (12) يمكن استنتاج التحسن الواضح في أداء الخوارزمية المحسنة مقارنة مع الخوارزمية بدون تحسين عند نسبة تلوث 60%، وهي تقترب من أداء الخوارزمية بدون تلوث وخصوصاً مع عدد طلبات عالٍ.



الشكل (12) التحسين الحاصل على الخوارزمية مع التلوث الساخن بعد ادخال مفهوم التشابه الدلالي باستخدام *NGD* مع نسبة طلبات زائفة *60%*

الاستنتاجات والتوصيات

الاستنتاجات

1. يكون أفضل أداء نسبي لخوارزمية *GDFS* مقارنة مع حجم التخزين المستخدم لذاكرة الكاش عند حجم 50% من حجم الذاكرة المطلوبة لكافة الأغراض التي تم طلبها من قبل مخدم الويب.
2. تتأثر خوارزمية *GDFS* بكل من التلوث البارد والتلوث الساخن.
3. تتأثر خوارزمية *GDFS* بشكل أكبر بالتلوث الساخن، ويعود ذلك إلى وجود معيار التردد في مفتاح الخوارزمية، والذي يحدّ من بقاء الأغراض غير المتكررة في الذاكرة، وهي حالة التلوث البارد.
4. إدخال مسافة غوغل المقيسة إلى عمل تابع الاستبدال لخوارزمية *GDFS* أدى إلى تحسن أداء الخوارزمية وخصوصاً الحالة الأسوأ عند مقدار تلوث 60% من الطلبات.
5. كان التحسن ملحوظاً بشكل واضح في حالة التلوث الساخن، حيث أمكن التغلب بشكل كبير على المشكلة التي تعاني منها خوارزمية *GDFS* بسبب التلوث، واقتربت الحالة الأسوأ سابقاً للخوارزمية من الحالة المثالية.

التوصيات

1. التطوير المستمر لتوابع الاستبدال لخوارزمية كاش الويب للتغلب على مشكلة تلوث الكاش حيث تم التغلب على مشكلة التلوث باستخدام تطوير تابع الاستبدال باستخدام مسافة غوغل المقيسة.
2. تطوير تابع الاستبدال للخوارزمية لتقليل أثر تلوث الكاش باستخدام تقنيات تشابه دلالي أخرى مثل التشابه الدلالي المعتمد على الحواف *Edge based semantic similarity* وخوارزميات أخرى تعتمد على محتوى المعلومات Information Content-Based Measure.

References:

[1] Mertz, Jhonny & Nunes, Ingrid. (2017). Understanding Application-Level Caching in Web Applications: A Comprehensive Introduction and Survey of State-of-the-Art Approaches. ACM Computing Surveys. 50. 1-34. 10.1145/3145813 .

[2] Liu Y, Zhi T, Xi H, Quan W, Zhang H. 2019b. A novel cache replacement scheme against cache pollution attack in content-centric networks. In: 2019 IEEE/CIC International Conference on Communications in China (ICCC). Piscataway: IEEE, 207–212.

[3] Janyaadisai, A., Sridama, P., Singkhleewon, N., & Wisedsind, N. (2020). Development to speed in Internet Utilizationby Web Cache Replacement Model .IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 22, Issue 2, Ser. I (Mar - Apr 2020), PP 14-20

[4] Ali, Waleed & Shamsuddin, Siti Mariyam. (2009). Intelligent Client-Side Web Caching Scheme Based on Least Recently Used Algorithm and Neuro-Fuzzy System. 70-79. 10.1007/978-3-642-01510-6_9 .

[5] Sulaiman, Sarina & Shamsuddin, Siti Mariyam & Abraham, Ajith & Sulaiman, Shahida. (2008). Web caching and prefetching: What, why, and how?. Proceedings - International Symposium on Information Technology 2008, ITSim. 3. 1 - 8. 10.1109/ITSIM.2008.4631949 .

[6] J. Li; J. Wu; G. Dan; A. Arvidsson and M. Kihl.(2014). Performance Analysis of Local Caching Replacement Policies for Internet Video Streaming Services. IEEE 22nd International Conference on Software, Telecommunications and Computer Networks (SoftCom).

[7] Mohammed,Salah; Abdalaziz,Khaleel; Fattoh,Saif Eldin Osman and Hiba Ali Nasir Sirour. (2016). A Survey on Web Cache Replacement Technology. International Journal of Computer Science and Telecommunications,Volume 7, Issue 7, October2016

[8] Lutz, Roman. (2016). Security and Privacy in Future Internet Architectures - Benefits and Challenges of Content Centric Networks. , arXiv:1601.01278v2 [cs.CR] 13 Jan 2016

[9] Ayani, Rassul & Teo, Yong & Ng, Yean. (2003). Cache Pollution in Web Proxy Servers.. 248. 10.1109/IPDPS.2003.1213450 .

[10] Y. Wang, Y. Yang, C. Han, L. Ye, Y. Ke and Q. Wang, "LR-LRU: A PACS-Oriented Intelligent Cache Replacement Policy," in IEEE Access, vol. 7, pp. 58073-58084, 2019, doi: 10.1109/ACCESS.2019.2913961.

[11] Ayesha, Banu; Syeda, Sameen Fatima; Khaleel and Ur Rahman Khan.(2015). Information Content Based Semantic Similarity Measure for Concepts Subsumed By Multiple Concepts. International Journal Web Applications Volume 7 Number 3 September 2015

[12] Cedric De, Boom; Steven Van,Canneyt; Steven,Bohez; Thomas,Demeester and Bart Dhoedt. (2015). Learning Semantic Similarity for Very Short Texts. 2015 IEEE International Conference on Data Mining Workshop (ICDMW)

[13] ten Kate, Warner ; Aleksovski, Zharko and Gligorov, Risto. (2007). Using Google Distance to Weight Approximate Ontology Matches. Proceeding WWW '07 Proceedings of the 16th international conference on World Wide Web, Pages 767-776

[14] Aldibaja, Ihab; Suleiman, Ali" .(2018). Web Cache Algorithm Development Using Google Measured Distance GDFS ،Tishreen University Journal -Engineering Sciences Series,, 39(6). Retrieved from http://journal.tishreen.edu.sy/index.php/engscnc/article/view/4241